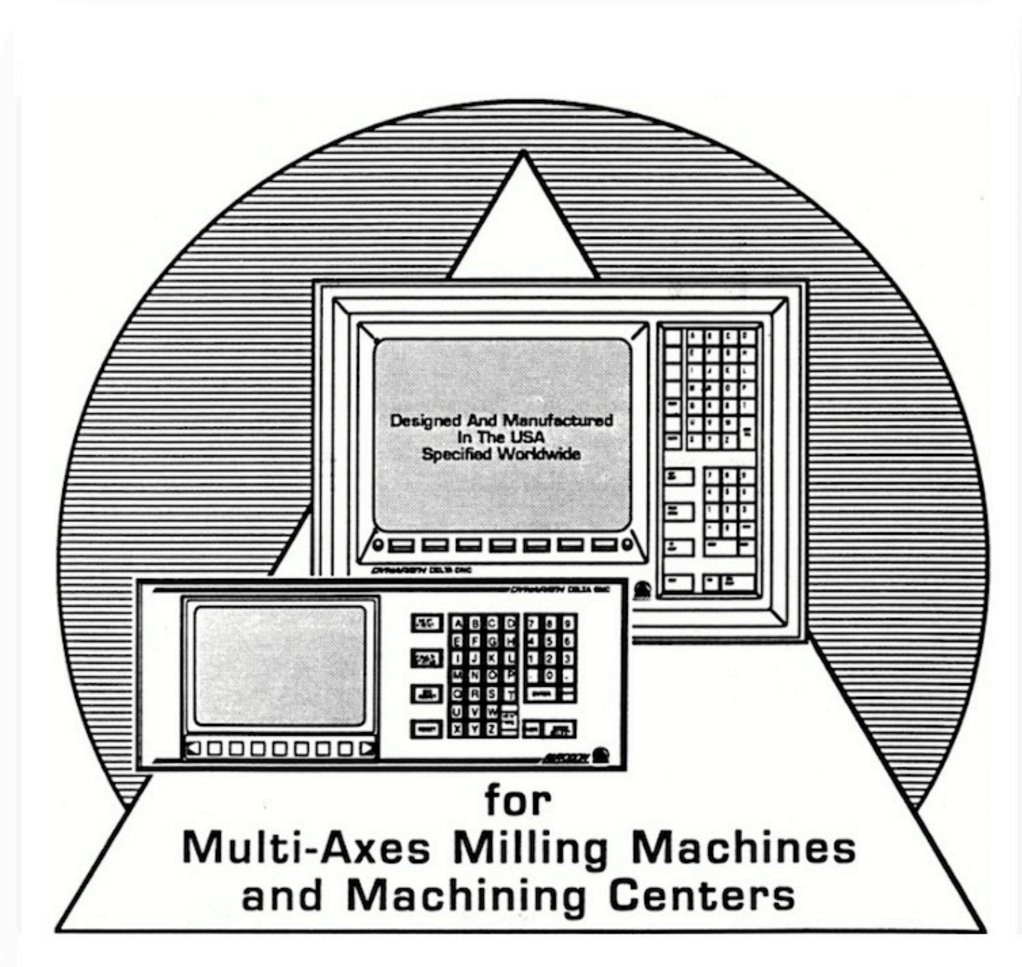
DynaPath® Delta

40M/50M/60M
Computer Numerical Controls

CUSTOMER MANUAL

OPERATIONS
•
PROGRAMMING
•
CONVERSATIONAL PROGRAMMING
•
CALC-ASSIST
•
CUSTOMER PREPARED CYCLES
•
EIA/ISO PROGRAMMING
•
APPENDICES



for
Multi-Axes Milling Machines
and Machining Centers

AUTOCON

38455 Hills Tech Drive Farmington Hills, MI 48331-5751 Tel: (810) 488-0440 FAX: (810) 488-0430

Autocon Delta 40M, 50M, 60M Features

The following is a list of features that may interest customers who have purchased D10M/20M controls in the past. Please note that most of these features are also available in the D30M control.

6th, 7th, 8th axis (contouring or indexing, linear or rotary)

H and D tool offset codes

Second Reference plane

Automatic Fixture Offset Calibration

Ability to control Gantry style machines

200 Tool Offsets

Selectable 2 through 6 Digit T Codes

M or T Codes execute Macro/CPC

500 CPC Variables

Auto mode to Single Event mode soft key

Multiple Position Display Coordinates

Auxiliary Axes named ABCUVW

Enhanced Capability to Handle Tool Changers

Axis Interchange and Switching capability

EIA/ISO Selectable G codes

New G codes: G10, G11, G28, G29, G53

Fixture Offsets for 4th, 5th, and 6th axes

System Settings Display Screen

38,400 Kbaud data transfer rate

Isometric and Blueprint Graphics Views.

Graphics Auto Zoom feature (available later this year)

Hard Disk Drive (available later this year)

Dear Customer:

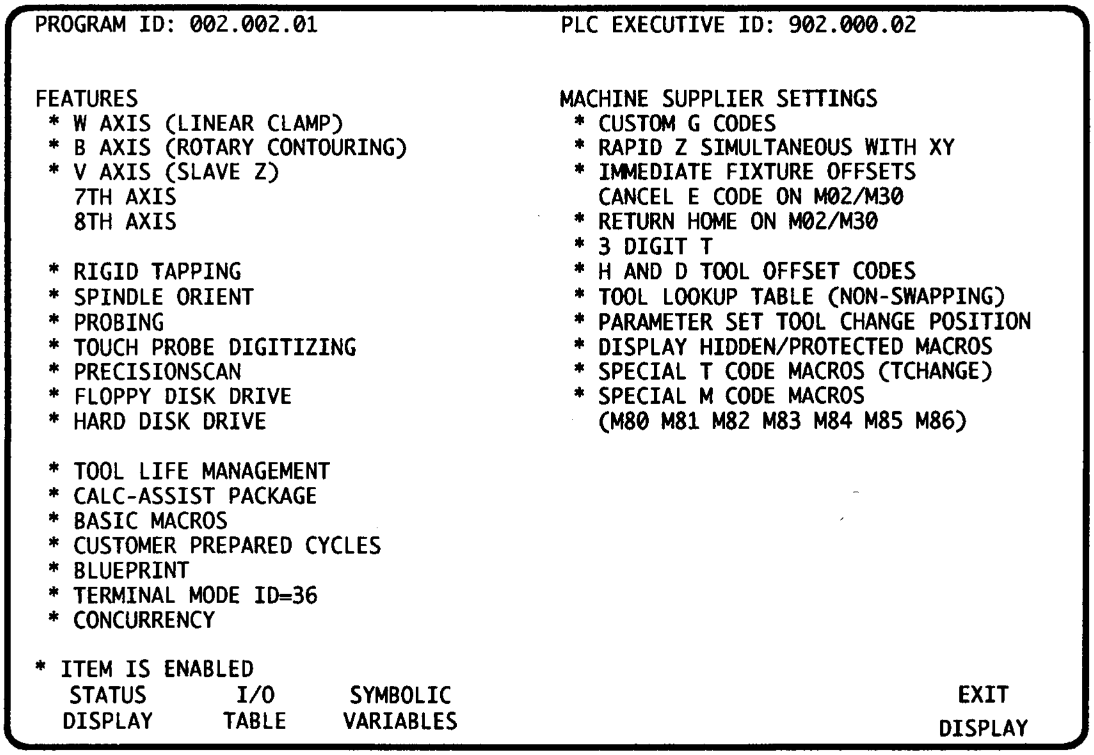
Hopefully, you have just read a partial list of features that are provided in our new line of controls.

The ideas for most features described here originated directly from our customers and end-users. We thank you for providing us with this valuable feedback and we hope you will continue to keep us informed about features you would like to see in our control.

Please feel free to contact our Sales Engineering Department if you require any further information about the items mentioned in this document. Again, thank you for your continuing interest in our line of Computer Numerical Controllers.

System Settings Screen

The System Settings screen may be accessed at any time to provide machine operators with information about the control's operating system, CNC options that are installed in the control, and machine parameters that will affect tool motion.



```
PROGRAM ID: 002.002.01                    PLC EXECUTIVE ID: 902.000.02


FEATURES                                  MACHINE SUPPLIER SETTINGS
 * W AXIS (LINEAR CLAMP)                   * CUSTOM G CODES
 * B AXIS (ROTARY CONTOURING)              * RAPID Z SIMULTANEOUS WITH XY
 * V AXIS (SLAVE Z)                        * IMMEDIATE FIXTURE OFFSETS
   7TH AXIS                                  CANCEL E CODE ON M02/M30
   8TH AXIS                                * RETURN HOME ON M02/M30
                                           * 3 DIGIT T
 * RIGID TAPPING                           * H AND D TOOL OFFSET CODES
 * SPINDLE ORIENT                          * TOOL LOOKUP TABLE (NON-SWAPPING)
 * PROBING                                 * PARAMETER SET TOOL CHANGE POSITION
 * TOUCH PROBE DIGITIZING                  * DISPLAY HIDDEN/PROTECTED MACROS
 * PRECISIONSCAN                           * SPECIAL T CODE MACROS (TCHANGE)
 * FLOPPY DISK DRIVE                       * SPECIAL M CODE MACROS
 * HARD DISK DRIVE                           (M80 M81 M82 M83 M84 M85 M86)

 * TOOL LIFE MANAGEMENT
 * CALC-ASSIST PACKAGE
 * BASIC MACROS
 * CUSTOMER PREPARED CYCLES
 * BLUEPRINT
 * TERMINAL MODE ID=36
 * CONCURRENCY

* ITEM IS ENABLED
   STATUS      I/O      SYMBOLIC                                      EXIT
   DISPLAY     TABLE    VARIABLES                                     DISPLAY
```

Feature List:

- Part Program Storage on floppy disk drive (buffered input is supported).
- Graphics Setup event.
- User selectable Coordinate System Display (user picks part coordinates, machine coordinates or one of six other options).
- Soft key to switch from Auto mode to Single Event Mode.
- 38.4 Kbaud serial communications rate.
- Enhanced PLC receive procedure. CNC displays incoming data and prevents user from accidentally overwriting the PLC Executive program.
- Allow Rigid Tapping in 2 gear ranges. Added spindle reversal compensation parameters.
- 500 CPC variables.
- Metric Jog increments .001, .01, etc. (instead of .002, .02, etc.)
- Ability to restrict operator from changing reversal compensation entries from the Setup mode.
- New PLC signals to simplify the programming of a jog handwheel pendant.
- Separate T, H, and D codes.
- Second reference plane definition in quill cycles.
- 6th axis capability - Rotary Contouring, Rotary Indexing, Linear.
- Allow auxiliary axes to be named A, B, C, U, V, W.
- Allow auxiliary axes to be slaved to any other similar axis.
- Allow user to complete a "lookup" table that maps the programmed T code into a tool pocket number for the control.
- Define M codes to execute a special Macro/CPC.
- Specify T code to execute a special Macro/CPC.
- Allow machine builder to enable hidden and/or protected Macro/CPCs.
- Machine builder selectable G codes.
- Gantry support with cross-error correction.
- Axis Interchange capability (e.g. send 4th axis commands to 5th axis).
- Allow PLC to command axes to a predefined position.
- Fixture Offset support for 4th, 5th, and 6th axes.
- Automatic storage of Fixture Offset coordinates via Fixture Calibration feature.
- Incremental adjustments to Tool Table entries.
- G10, G11 codes to store Tool and Fixture Offset data in part program.
- G27, G28, G29, G53 G code functions.
- Bi-directional Lead Screw Error Compensation.
- Handwheel Interrupt (in Jog and/or Auto mode).
- Spindle orient to position(s) established via PIC program. (This feature may not be activated until the next release of the firmware.)

Support for High-End Machining Centers

The following features are being added to allow the control to be applied to a wider range of high-end machining centers.

Request . . . Support separate T, H, and D codes for compatibility with other CNCs used throughout the industry.

Feature . . . . D30M Supports up to 200 separate T, H, and D codes. Of course, machine builders can set parameters to be compatible with part programs written for older systems (i.e. supports 2 and 4 digit T codes found on D10/20M controls).

Notes . . . . . We believe that separate T, H, and D codes will be preferred by most customers who do not have to worry about backwards compatibility with older DynaPath part programs. It is a simpler and more efficient method of handling tool offsets.

Request . . . Allow customer to program a tool number (up to 6 digits long) that relates to a company's inventory system.

Feature . . . . Implement a special non-swapping Q table in CNC. Program T codes up to 6 digits in length. Q table will contain a list that maps the T code into a 3 digit tool pocket number. The Q table is entered by the machinist who loads the tool carousel.

Notes . . . . . Separate T, H, and D codes must be selected to use this feature.

Request . . . Support servo-controlled tool changer.

Feature . . . . Call a special Macro/CPC when an M06 command executes in an M Function event. The special Macro/CPC will position the tool changer in the correct position, then it will initiate the tool change.

Notes . . . . . none

Request . . . Support a gantry style machine.

Feature . . . . Axis slaving is more robust - any auxiliary axis can be slaved to any other similar axis. A special cross-error skew correction feature provides control for gantry systems.

Notes . . . . . none.

Request . . . Support a dual spindle machine with rotary axes.

Feature . . . . Slave 5th linear axis to the Z axis. Slave 4th rotary axis to the 6th rotary axis. A "simple" XYZ part program will now cut simultaneous parts.

Notes . . . . . none.

Request . . . Allow customer to name auxiliary axes A, B, C, U, V, W in accordance with EIA/ISO standards relating to linear and rotary axes.
Actual. . . . . Allow user to select A, B, C, U, V, W for auxiliary axes.
Notes . . . . . Caution: since the control uses the letters C, V, and W in normal part programming, naming an auxiliary axis with these letters will cause compatibility problems with programs that have been developed for other DynaPath controls. I would strongly recommend that machine builders do not use the letters C, V, and W to name an auxiliary axis if part program compatibility is an issue.

Request . . . Support a machine with a rotary axis on each pallet.
Feature . . . . Implement the Axes Interchange feature so the same part program can be used for each pallet. This feature lets the CNC direct 4th axis commands to the 4th axis when pallet #1 is in position and to the 5th axis when pallet #2 is in position.
Notes . . . . . The PLC User program controls the Axes Interchange feature.

Request . . . Enhance support for machines with a pallet changer.
Feature . . . . •Implement special M code Macro/CPC feature that lets a predefined M code call a special Macro/CPC into execution. Up to seven special Macro/CPCs can be called via M code.
• Implement feature that allows PLC to command the axes to a parameter stored machine coordinate position.
• Refer to Axes Slaving and Axes Interchange features described elsewhere in this memo.
Notes . . . . . none.

Request:. . . Prevent Macro/CPC code from being deleted or edited by the machine operator.
Feature . . . . A special character embedded in the Macro/CPC title identifies the program as:
•protected - program may be viewed, but not edited nor deleted from the Macro catalog.
•hidden and protected - program may not be viewed, edited, nor deleted from the Macro catalog.
Notes . . . . . Machine parameter setting can turn the hidden/protected feature on and off.

International Programming Compatibility Features

The following enhancements are being added to make D30 part programs more easily adaptable to other part programming styles that are common throughout the industry.

- **Separate T, H, and D codes** - this is the most common method in the industry to activate tool length (H code) and tool diameter (D code) offsets.

- **G10, G11 codes** - this code is used to load the tool tables directly from the part program. Our current system requires that the tool table be stored in a file that is separate from the part program. T, H, D, E, and Q tables can be loaded when G10 is active. G11 cancels a G10.

- **Second reference plane definition in quill cycles** - this features provides an optional entry in the Position, Bolt Circle, Frame Mill and Circle Mill, and EIA events that allows the programmer to define a second reference plane when the tool must jump over a clamp.

- **G28, G29 codes** - go to and return from reference point.

- **G53, G27 code** - move to a coordinate programmed in machine coordinates.

- **Allow auxiliary axes to be named A, B, C, U, V, W.**

- **Machine builder selectable G codes.**

- **Metric Jog increments .001, .01, etc.** - instead of .002, .02, etc.

- **User selectable Coordinate System Display** - user picks part coordinates, machine coordinates or one of six other options.

- **Fixture Offset support for auxiliary axes.**

PROPRIETARY INFORMATION

This document contains information considered to be proprietary to Autocon Technologies, Inc. Such information may not be used for the manufacture of equipment of the type described here; nor may this document or any part thereof be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopying, recording or otherwise without written permission from a duly authorized representative of Autocon Technologies, Inc.

PATENTS

The equipment described herein is covered by vendor patents and/or one or more United States patents or other patents pending and licensed under various Foreign Countries' patents. Refer to the patent placard on the unit.

OBJECTIVE

This publication is intended to provide the reader with significant data regarding the essential characteristics of equipment incorporated in a Numerical Control System. Although every attempt has been made to present information which is both complete and current, the large number of variations both in equipment itself and in its application make it impossible to cover all of the contingencies which arise in its installation, operation, or maintenance. In addition, incorporation of design changes to improve performance or reliability may occur at any time and are not necessarily reflected in the information provided. The right to revise or change this publication in whole or in part without prior notice is reserved. Should further information be desired, the matter should be referred to the Machine Supplier or Autocon Technologies, Inc.

First Printing: February 1995

Autocon Technologies, Inc.

Farmington, Michigan

TRADEMARK ACKNOWLEDGEMENTS

- IBM, PERSONAL COMPUTER and PC/AT are trademarks of International Business Machines Corporation.
- DOS is is a registered trademark of Microsoft Corporation.
- HAYES is a trademark of Hayes MicroComputer Products Inc.
- Flash is a trademark of Intel Corporation.
- PrecisionScan is a trademark of Autocon Technologies, Inc.
- All other product names mentioned herein are used for identification purposes only, and may be the trademarks or registered trademarks of their respective companies.

NOTICE

This Computer Numerical Control is designed to command the operation of a wide range of milling machines and machining centers. When operating this type of machinery, consideration **must** be given to meeting applicable safety and guarding standards.

The guarding standards which are most widely used in the drilling, milling, and boring industry are:

1. "Safety Requirements for the Construction, Care and Use of Drilling, Milling and Boring Machines", ANSI Standards B11.8 1974 which may be obtained from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

2. "Point of Operation Guarding" Section 1910.212, U.S. Occupational Safety and Health Administration.

3. "Power Transmission Guarding" Section 1910.212, U.S. Occupational Safety and Health Administration both of which are available from the U.S. Department of Labor.

PART 1 - INTRODUCTION

1-1) GENERAL INFORMATION and SPECIFICATIONS

1-2) CONTROL FEATURES and SYSTEM SETTINGS

1-3) START UP PROCEDURES and BASIC DATA ENTRY

1-3) MEMORY SUPPORT

PART 2 – OPERATIONS

2-1) JOG MODE

2-2) SET UP MODE

2-3) T, H, and D TABLES

2-4) TOOL LOOKUP TABLE

2-5) TOOL LIFE MANAGEMENT TABLE

2-6) FIXTURE OFFSET TABLE

2-7) CATALOG MODE

2-8) AUTO MODE

2-9) SINGLE EVENT MODE

2-10) PROGRAM MODE

2-11) EDIT MODE

2-12) GRAPHICS MODE

PART 3 – PROGRAMMING

PART 4 – CONVERSATIONAL PROGRAMMING

4-1) GRAPHICS EVENT

4-2) TEXT EVENT

4-3) M FUNCTION EVENT

4-4) DWELL EVENT

4-5) POSITION EVENT AND G CYCLES

4-6) LINEAR MILL EVENT

4-7) ARC MILL EVENT

4-8) CUTTER RADIUS COMPENSATION

4-9) BOLT CIRCLE EVENT

4-10) FRAME MILL EVENT

4-11) CIRCLE MILL EVENT

4-15) ROTATE EVENT

4-16) CAVITY MILL EVENT

4-17) PROBE EVENT

4-18) MACRO CALL EVENT

PART 5 - CALC-ASSIST

5-1) BASIC CALC-ASSIST

5-2) ADVANCED CALC-ASSIST

PART 6 - CUSTOMER PREPARED CYCLES (CPCs)

6-1) INTRODUCTION

APPENDIX A - OPERATOR MESSAGES

APPENDIX B - OFF-LINE COMMUNICATIONS

INTRODUCTION

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON

INTRODUCTION

The DynaPath Delta 40M. 50M, and 60M CNC units are microprocessor based numerical control units which can control most varieties of multiaxes machining centers and milling machines.

As the operator/programmer of a machine that has been fitted with a DynaPath Series CNC, and as a reader of this manual, you should be aware that there are many features and control settings that may or may not be available for your use. We strongly recommend that you learn about your control's configuration by studying Part 1 - Section 2 of this manual.

Further, your control may have special functions which have been designed and implemented for your particular machine. These features obviously fall beyond the scope of this manual and require that you consult documentation that is supplied with your machine.

HOW TO USE THIS MANUAL

The purpose of this manual is to provide you with the information required to set up, program, and run a machine equipped with a DynaPath Series control. In describing these operations, this manual uses terminology and conventions which are widely used throughout the machine tool industry. If you lack conversance in the terms and dimensioning standards listed below, you will find that the information in this document will be greatly enhanced if you first obtain and study a Computer Numerical Control tutorial guide. Such a guide is a available from many third party sources. A short list of terms that you should be familiar with include:

Cartesian coordinate system
Polar coordinate system
modal
absolute dimensions
incremental dimensions
datum

Once you are familiar with these terms, we recommend that you read this manual's Table of Contents to gain familiarity with its topical content and organization. If this is your first time using a DynaPath control we recommend that you first read the Start Up Procedures section. When you are familiar with this information, read each section as it becomes pertinent to your next operation. This manual should be kept near the control to be available for easy reference.

MANUAL ORGANIZATION

The structure of this manual is based on the following general categories;

Part 1	General Information and Specifications
Part 2	Operational Modes
Part 3	Programming - General
Part 4	Conversational Program Events
Part 5	Calc-Assist
Part 6	Customer Prepared Cycles
Part 7	EIA/ISO Programming
Appendix A	Operator Messages
Appendix B	Off-line Communications

CONTROL SPECIFICATIONS

GENERAL

Ambient Temperature	40°F to 120°F (4°C to 49°C)
Humidity	5% to 95%, noncondensing
Front Panel Seals	Dust and Oil Tight, Splash Proof
Cooling	Internal fans for System Module
Input Voltage	115/230 VC ± 10%, Single Phase, 47-63 Hz
Max. Power	425 VA with color CRT 300 VA with monochrome CRT

INTERFACES

RS232C serial communications	Standard on D40, D50, D60
Terminal mode for host communications	Optional D40, D50; Standard D60
RS-422 Interface to replace RS232C interface	Optional on D40, D50, D60
1.44 Meg PC Floppy Disk Drive	Optional D40, D50, Standard D60
270 Meg (or larger) PC Hard Disk Drive	n/a D40, Optional D50, Standard D60

MECHANICAL

Modular Configuration (a complete system consists of a System Module, one of the CNC Operator's Panels, and an Auxiliary Operator's or Machine Operator's panel.)

System Module	14.6"W x 16.7"H x 8.9"D[1] (371mm x 424mm x 226mm) weight approx. 19 lb. (8.6 kg)
CNC Operator's Panel[2] with 14" color CRT	20.50"W x 15.10"H x 15"D (520.7mm x 383.5mm x 381mm) weight approx. 36 lb. (16.4 kg)
CNC Operator's Panel[2] with 9" monochrome CRT	20.50"W x 7.88"H x 12"D (520.7mm x 200.2mm x 304.8mm) weight approx. 16 lb. (7.3 kg)
Auxiliary[2] Operator's Panel	20.50"W x 4.88"H x 3.50"D (520.7mm x 124.0mm x 88.9mm) weight approx. 3 lb. (1.4 kg)
Machine Operator's Panel	20.50"W x 7.19"H x 3.50"D (520.7mm x 182.6mm x 88.9mm) weight approx. 4 lb. (1.8 kg)

[1] D includes 2.75" (70mm) connector allowance.

[2] The 9" CNC Operator's Panel and the Auxiliary Operator's Panel can be configured as a single panel. The panel size is 20.50" W x 12.76"H (520.7 x 324.0 mm).

MEMORY

Flash Memory on Memory board	CNC Operating System, Machine Parameters
Flash Memory on PLC board	PLC Exec and User programs
SRAM (battery retained)[3]	Tool Tables, Part Programs, PLC keep relays
DRAM	Part Program, Tool Table data

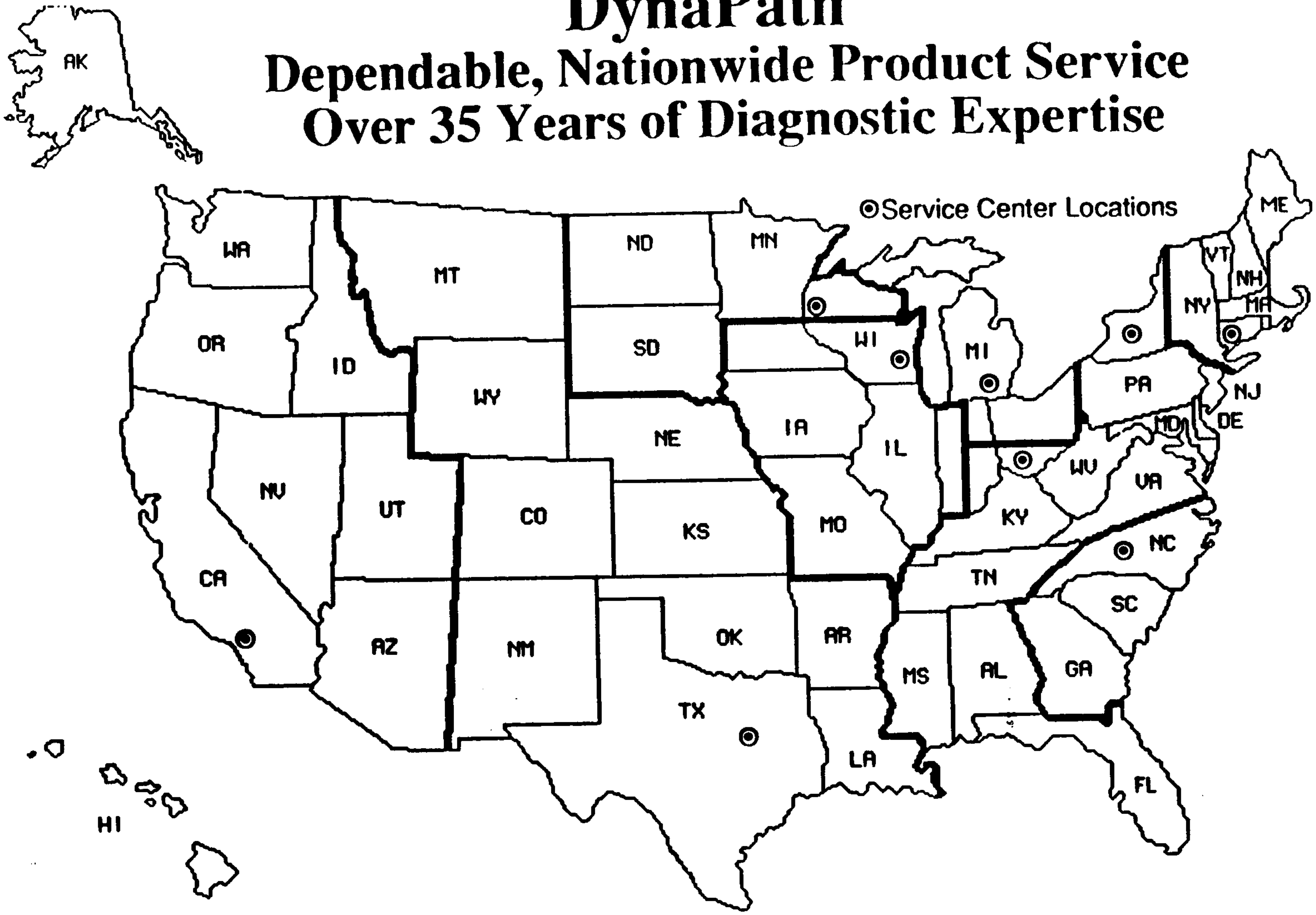
Part Program Storage

D40 Standard	64,000 characters (530 feet or 160 meters)
D40 Optional	192,000 optional (1600 feet or 426 meters)
D50 Standard	192,000 optional (1600 feet or 426 meters)
D50 Optional	448,000 optional (3730 feet or 1120 meters)
D60 Standard	192,000 optional (1600 feet or 426 meters)
D60 Optional	448,000 optional (3730 feet or 1120 meters)

[3] Backup Battery Retention: approximately 10 years cumulative off time. Battery may be replaced without loss of part program and Setup data.

DynaPath®

Dependable, Nationwide Product Service
Over 35 Years of Diagnostic Expertise



Call 1-800-367-3423 for Service

INTRODUCTION

This section discusses optional features on the DynaPath CNCs. It is important that you familiarize yourself with the information in this section before you attempt to operate your machine.

Why there are options on the DynaPath CNCs.

Options are made available on our CNC to provide the following capabilities:

- provide machine builders with a high degree of flexibility in fitting the control to a wide range of machining centers and milling machines. Examples of these options include provisions for handling up to eight axes, handling a wide variety of tool changers, and handling machines with an automatic pallet changer. Further examples include the ability to perform advanced machining functions such as probing, digitizing, and rigid tapping.

- allow machine builders to customize the CNC to conform with programming and machine operation styles that certain users are already familiar with. Examples of these options include the ability to customize G code assignments for EIA/ISO programming, the ability to specify immediate/deferred fixture offsets, the ability to allow simultaneous XYZ rapid moves on your machine, and the ability to select H and D codes to specify tool offsets on the control.

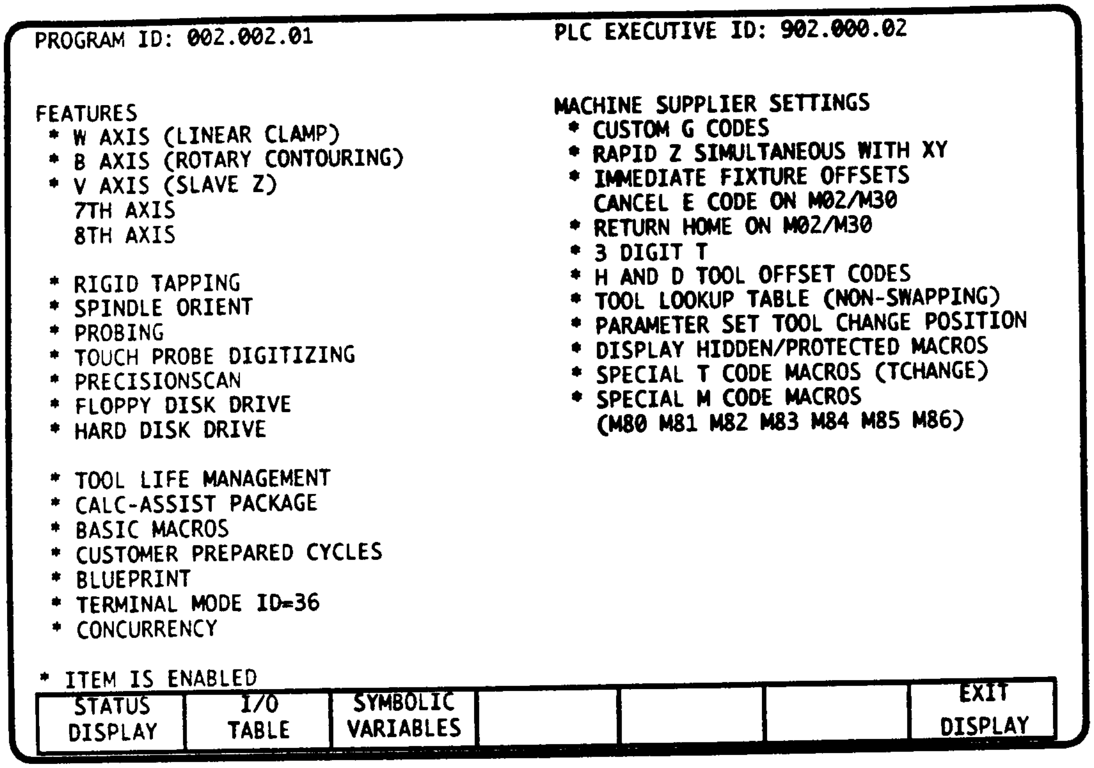
Most options on the CNC can be placed in one of the following categories.

- **Hardware and Software Options** are control items designed to boost the performance and flexibility of the control by: supporting special hardware (auxiliary axes, rigid tapping spindle, or a floppy disk drive) that is installed on your machine, or by providing software features (Calc-Assist, Macros/CPCs, etc.) that simplify the task of creating part programs and operating your machine. Many features in this group may be purchased directly from your machine supplier.

- **Machine Supplier Settings** are software settings that have been selected by your machine builder to configure the control to be consistent with programming and operating styles that you are already familiar with. It is important to understand that there is not a correct or incorrect setting for these options. It is, however, important that everyone who uses the control is aware of how these options have been configured by the machine supplier. These settings are programmed as parameters in a non-volatile memory device known as a Flash memory chip. This memory device acts as a table of values for the control. Once this device is programmed or initialized, the values may not be redetermined unless the Machine Parameter Flash chip is reprogrammed by the supplier. Any desired change by the customer should be communicated to the machine supplier, who will make recommendations based on the capabilities of both the machine and control unit.

How to determine which options have been installed in your CNC.

You can determine how your control has been configured by pressing the control's fault/status key and then the system settings key. The screen, below, lists the options that are available. Any item proceeded with an asterisk (*) character has been enabled by your machine builder or Autocon Technologies.



```
PROGRAM ID: 002.002.01                      PLC EXECUTIVE ID: 902.000.02

                                            MACHINE SUPPLIER SETTINGS
FEATURES                                     * CUSTOM G CODES
 * W AXIS (LINEAR CLAMP)                     * RAPID Z SIMULTANEOUS WITH XY
 * B AXIS (ROTARY CONTOURING)                * IMMEDIATE FIXTURE OFFSETS
 * V AXIS (SLAVE Z)                            CANCEL E CODE ON M02/M30
   7TH AXIS                                  * RETURN HOME ON M02/M30
   8TH AXIS                                  * 3 DIGIT T
                                             * H AND D TOOL OFFSET CODES
 * RIGID TAPPING                             * TOOL LOOKUP TABLE (NON-SWAPPING)
 * SPINDLE ORIENT                            * PARAMETER SET TOOL CHANGE POSITION
 * PROBING                                   * DISPLAY HIDDEN/PROTECTED MACROS
 * TOUCH PROBE DIGITIZING                    * SPECIAL T CODE MACROS (TCHANGE)
 * PRECISIONSCAN                             * SPECIAL M CODE MACROS
 * FLOPPY DISK DRIVE                           (M80 M81 M82 M83 M84 M85 M86)
 * HARD DISK DRIVE

 * TOOL LIFE MANAGEMENT
 * CALC-ASSIST PACKAGE
 * BASIC MACROS
 * CUSTOMER PREPARED CYCLES
 * BLUEPRINT
 * TERMINAL MODE ID=36
 * CONCURRENCY

* ITEM IS ENABLED
```

A short introductory description of each option and setting is provided below. As you read each section, consider how the state (installed or not installed) of each option will affect your machining and programming operations. In general, you should consider the following issues:

> **How does the item affect part programming?** Many of the items relating to tool motion and customization will directly influence the manner in which you design part programs. For example, to design a safe and efficient part program, it is important that every programmer understands how the tool will move during an XYZ rapid command, or how the tool will move following the activation of a Fixture Offset.
>
> **How does the item affect part program compatibility with other DynaPath controls?** This is important to consider if you wish to share the same part program with two different DynaPath controls. Differences between these items (particularly the machine supplier settings) may make the program non-functional or dangerous on a second DynaPath control. Always proof out a part in the control's Graphics mode if you are unsure how it will perform on the second DynaPath control.

Each topic below references a section of the manual where you can obtain more information about an option or machine supplier setting.

HARDWARE and SOFTWARE OPTIONS

Auxiliary Axes

The first five items in the Features list show the number of auxiliary axes that are controlled by the CNC. Each line shows the type of axis that is installed (rotary, indexing, etc.) and the letter identifier that is used to program the axis.

> **Note: an auxiliary axis may be labeled as A, B, C, U, V, or W. Having an axis labeled as C, V, and/or U causes the following changes to the control's Conversational programming language.**
>
> **If an axis is labeled C** - the letter H replaces the letter C in Linear Mill, Arc Mill, Frame Mill, Circle Mill, Repeat, Rotate, and Cavity Mill events.
>
> **If an axis is labeled V** - the letter J replaces the letter V in Position, Linear Mill, and Repeat events.
>
> **If an axis is labeled W** - the letter I replaces the letter W in Position, Frame Mill, Circle Mill and Bolt Circle events.
>
> **These changes will be displayed on the appropriate Program mode screens. Having a C, V, or W axis may cause some compatibility problems when working with programs that were created on a DynaPath Series control without a C, W, or V axis. Some minor edits will be required if you wish to interchange part programs between the two controls.**

Rigid Tapping - A hardware and software feature that allows high-speed tapping using a rigid tap holder. (See Part 4-Section 5).

Spindle Orient - A hardware and software feature that allows the spindle to orient along any angle. If the machine is capable of orienting, the machine builder must implement this feature and design the manner in which spindle orient is programmed. Contact the machine builder for information regarding this feature.

Probing - A hardware and software feature that quickly and accurately locates fixture offsets and/or measures tool lengths. (See Part 4-Section 16).

Touch Probe Digitizing - A hardware and software feature that uses a spindle probe to "trace" an existing part and send its coordinates to a personal computer. You can then use the computer to perform miscellaneous CAD and CAM operations on the coordinate data in order to create a part program which can be run on the CNC. (See Part 2-Section 5).

PrecisionScan - PrecisionScan is a high speed continuous trace scanning system which uses spindle probe to "scan" an existing part and send its coordinates to a personal computer. You can then use the computer to perform miscellaneous CAD and CAM operations on the coordinate data in order to create a part program which can be run on the CNC.

Floppy Disk Drive - A hardware and software feature that lets you store part programs on a PC-compatible floppy disk drive mounted in the control. System updates can also be loaded directly from a floppy disk. (See Part 2-Section 10).

Hard Disk Drive - A hardware and software feature that lets you store part programs on a hard disk drive mounted in the control. If you connect an AT keyboard to the control and you have installed DOS, or any other PC operating system, on the hard drive, the CNC may be booted as a personal computer. (See Part 2-Section 10).

Tool Life Management - A software feature that monitors tool life. Since tools with identical machining purposes (.500" end mills for example) are grouped together, the control can automatically select the tool which has the most remaining life. This feature is primarily intended for a production environment where "redundant" tools are readily available. (See Part 2-Section 3).

Calc-Assist Package - A software feature that performs trigonometric calculations on part programs that mix polar and Cartesian data in the same event. The Calc-Assist Package consists to Basic Calc-Assist (which uses data that is present in a single block) and Advanced Calc-Assist (which uses data from three consecutive blocks) to calculate line and arc data. (See Part 5).

Macro and Customer Prepared Cycles - Macros is a software feature that allows a main part program to call a program that resides in a "protected" Macro Catalog. Customer Prepared Cycles expands the Macro feature to allow the use of variables in any program that resides in the Macro Catalog. This allows users to create custom "User Macros" to handle special functions on their machine, or to use variable data to define a family of parts. (See Part 6).

Blueprint - A software feature which provides an isometric view in the Graphics mode (See Part 2-Section 12).

Terminal Mode - A software feature which allows the CNC to communicate with a host computer to exchange part programs and tool data. (See Part 2-Section 14).

Buffered Input - A software feature which allows you to execute a part program whose size exceeds the control's available storage. The part program may reside on the CNC's floppy or hard disk drive, or it may be reside on a personal computer and be transmitted to the control's serial port (See Part 2-Section 8).

Concurrency - A software feature which allows the machine operator to create or edit a second part program while the CNC is also executing a primary part program. (See Part 2-Section 8).

MACHINE SUPPLIER OPTIONS

Custom G Codes - This feature lets your machine builder select a set of G codes other than the EIA/ISO G Code set which is standard on all of our controls. This provides the ability, for example, to make the G20/G21 codes control the Inch/Metric selection on your particular control (an unaltered DynaPath Series control uses G70/G71 to control this function). This feature does not affect Conversational programming. (See Part 7).

Note: an asterisk before the Custom G Code item indicates that an alternate set of G codes has been selected for your control. If an asterisk appears, use the control's ISO ASSIST key to display the set of G Codes that program your control. This manual describes the G Code set that is Standard on all DynaPath Series controls. If an alternate set has been selected, you must cross-reference the function of each of your control's G Codes with the G Code that appears in the manual.

Having an alternate G Code set may cause some compatibility problems when working with programs that were created on a DynaPath Series control that has a different set of G Codes selected. Some minor edits will be required if you wish to interchange part programs between the two controls.

Rapid Z simultaneous with XY - This parameter setting affects the way that the control executes rapid moves. The majority of DynaPath series controls provide the programmer with maximum safety by moving the Z axis independent of X and Y. Your machine supplier can allow a simultaneous XYZ move by enabling this parameter. (See Part 4-Section 5).

Immediate or Deferred Fixture Offsets - This parameter setting affects the way the control executes a motion command following the activation of a Fixture Offset E code. The majority of DynaPath series controls use Immediate Offset Activation. Your machine supplier may enable Deferred Offset Activation. (See Part 2-Section 3 and Part 4-Section 3).

Cancel E code on M02/M30 - This parameter setting determines whether a Fixture Offset E code is canceled when an M02/M30 executes. The majority of DynaPath series controls keep the Fixture Offset active after an M02/M30. (See Part 4-Section 3).

Return Home on M02/M30 - This parameter setting determines whether the tool stops at its last programmed position or returns to the machine's Home position when an M02/M30 executes. The majority of DynaPath series controls rapid the tool's to the Home position when an M02/M30 executes. (See Part 4-Section 3).

Number of Digits in T code - This parameter setting determines the number of digits that are used to program a tool number on your control. (See Part 4-Section 3).

H and D Tool Offset Codes - This parameter setting determines how tool length and diameter offsets are activated on your control. (See Part 2-Section 3). The options are:

- Use T code to specify the physical tool number, tool length, and tool diameter.
- Use T code to specify the physical tool number, use H to specify tool length, and use D to specify tool diameter.

Tool Lookup Table - This parameter setting determines whether a Q Lookup table is required on your control. Your machine supplier will describe how to utilize this table since it can be enabled as a Swapping or Non-Swapping Lookup table. (See Part 2-Section 3).

Parameter Set Tool Change Position - This parameter setting specifies where the tool will move to when an M06-Tool Change command executes. (See Part 4-Section 3).

Display Hidden/Protected Macros - This parameter setting indicates the state of the parameter that allows a programmer to edit and/or delete part programs that were specially loaded into the control as "hidden" or "hidden and protected". These programs are usually loaded into your control by your machine supplier. (See Part 6).

Macro/CPC T codes - When this parameter setting is enabled, a programmed T code will cause the control to automatically execute a Special User Macro. The title of the Special Macro that will execute appears in parentheses. (See Part 4-Section 3).

Macro/CPC M codes - This parameter setting identifies that Macro/CPC M codes have been designed for your machine. Macro/CPC M codes are macros that execute when its corresponding M Code is programmed in an M Function or EIA event. The M codes that have been specified to call a Macro/CPC M code appear within parentheses. (See Part 4-Section 3).

GETTING STARTED

This section describes the basic information that you need to operate the DynaPath control. The topics described in this section are:

- • Turning the control ON.
- • Basic data entry and paging.
- • Turning the control OFF.
- • Daily Start Up Routine.
- • First Time Start-up Routine.

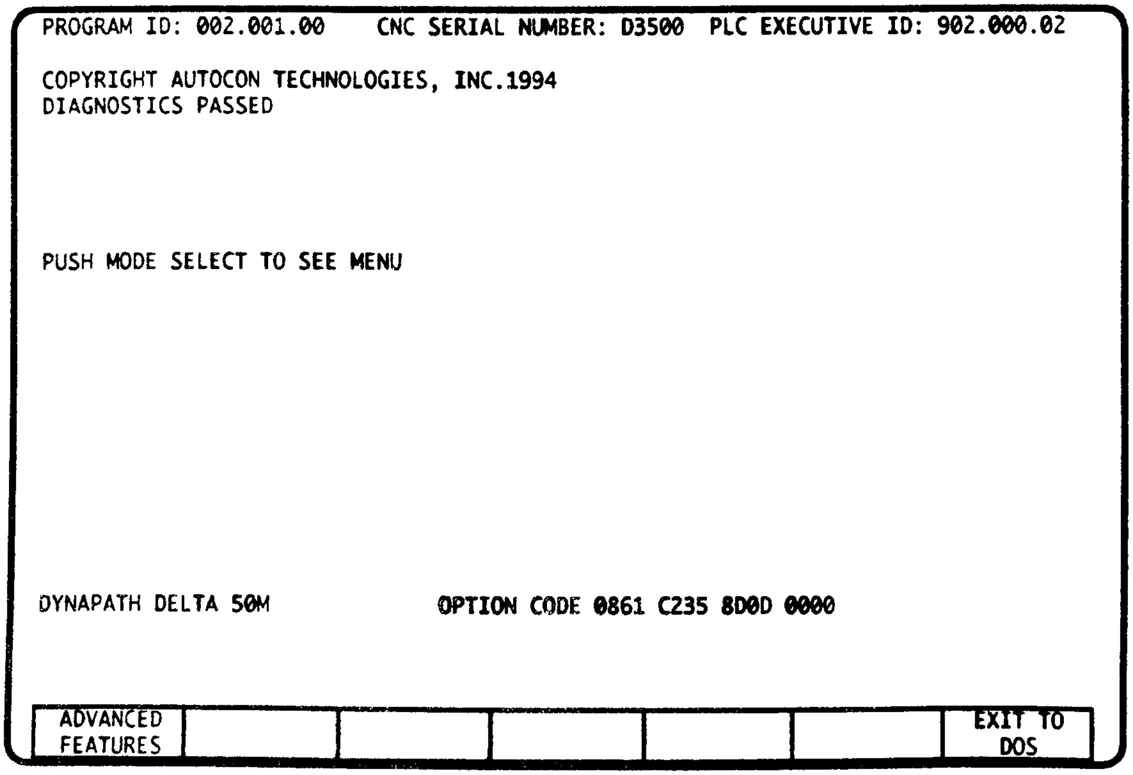
If you are unfamiliar with operating the control, you should familiarize yourself with the first four of the above topics before attempting to operate the machine. If it is your responsibility to bring the control into service for the first time, complete the procedure described in First Time Start Up Routine topic.

TURNING THE CONTROL ON

Follow your machine supplier's power ON directions. In general, you must;

1) Move the control's POWER SWITCH to the ON position. The Start Up screen, **below**, will appear after the control completes its startup diagnostics tests. The first line of this screen identifies the versions of the control's Firmware and PLC Executive program. A Diagnostics screen will appear if the control detects a hardware error during its startup diagnostics. Contact your machine supplier for assistance if this screen appears.

   If a WARNING-BACKUP BATTERY LOW or POWER FAIL message is displayed on the Start Up screen, refer to the following Memory Support section.



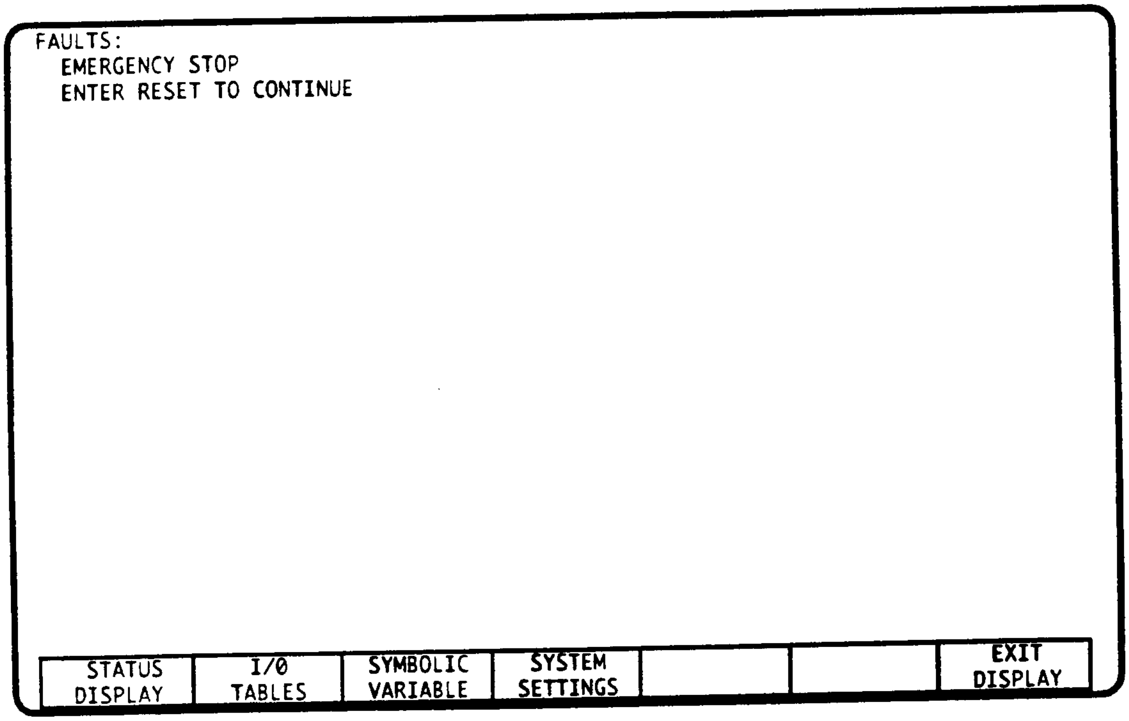
```
PROGRAM ID: 002.001.00    CNC SERIAL NUMBER: D3500  PLC EXECUTIVE ID: 902.000.02

COPYRIGHT AUTOCON TECHNOLOGIES, INC.1994
DIAGNOSTICS PASSED



PUSH MODE SELECT TO SEE MENU






DYNAPATH DELTA 50M               OPTION CODE 0861 C235 8D0D 0000
```

If a PARAMETER DATA DISCREPANCY message or any other error message is displayed, contact your machine supplier for assistance.

Pressing the ADVANCED FEATURES soft key on the Start Up screen will display a list of the optional software features that are installed on your control; all installed features are proceeded with an asterisk "*".

2) Press the MODE SELECT key. This will cause a FAULT OR MESSAGE display to flash on the screen.

3) Press the FAULT/STATUS key. This will display a Fault/Status screen which lists all conditions that must be cleared before you can operate the machine. A typical Start Up Fault/Status[1] screen is shown below.



```
FAULTS:
  EMERGENCY STOP
  ENTER RESET TO CONTINUE
```

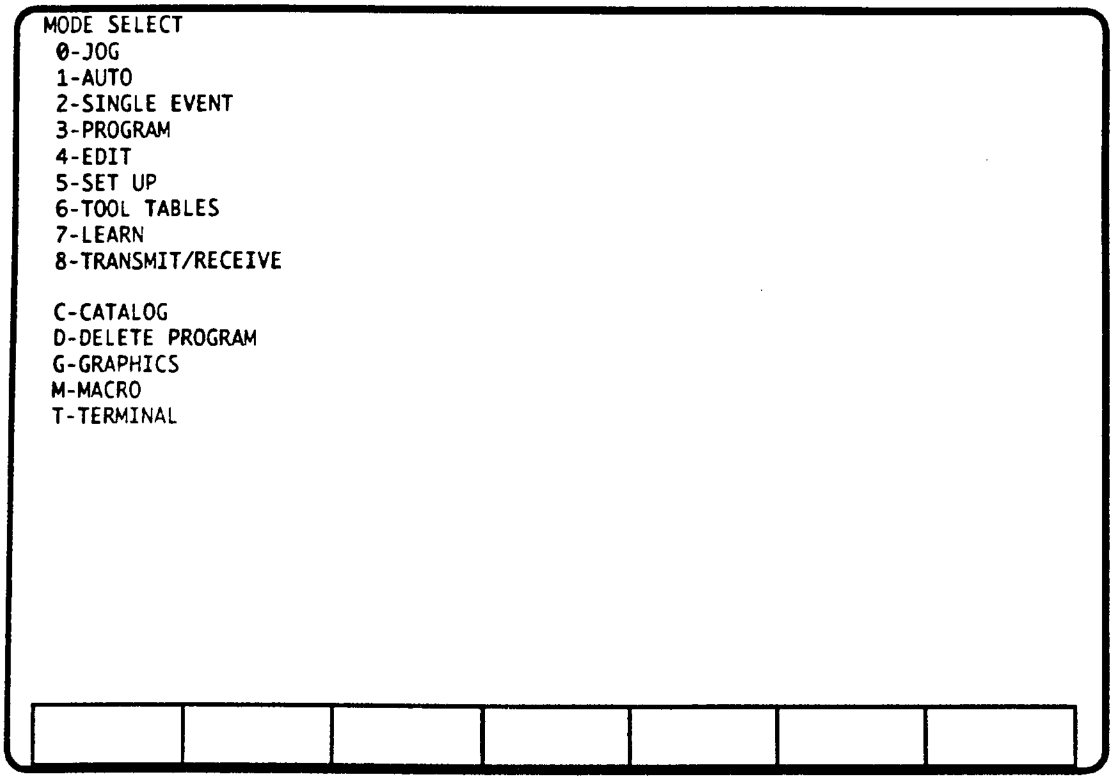
4) The procedure to clear these messages will vary from machine to machine - consult your machine supplier for the proper sequence. One step in the procedure must be to clear the EMERGENCY STOP fault by releasing the EMERGENCY STOP button.

5) Press the RESET key. If the fault conditions have been cleared, the ENTER RESET TO CONTINUE message will disappear to indicate that the control is initialized.

[1] The soft keys on this page are used to display CNC operating tables. These tables can be viewed at any time, even when the control is in cycle. The tables are:

- **Status Display** - shows whether the machine axes have been reference, the status of modal EIA/ISO programming G codes, and shows the status of the Option Stop and Block Delete functions.
- **I/O Tables** - shows the status of the control's inputs and outputs. This screen is used by service personnel to diagnose machine problems.
- **Symbolic Variables** - displays the current status of all variables which are used with the control's Customer Prepared Cycles feature.
- **System Settings** - provides a list of the control's advanced features and parameter settings.

6) Press the EXIT DISPLAY soft key to display the Mode Select screen, below.



```
MODE SELECT
 0-JOG
 1-AUTO
 2-SINGLE EVENT
 3-PROGRAM
 4-EDIT
 5-SET UP
 6-TOOL TABLES
 7-LEARN
 8-TRANSMIT/RECEIVE

 C-CATALOG
 D-DELETE PROGRAM
 G-GRAPHICS
 M-MACRO
 T-TERMINAL
```

BASIC DATA ENTRY and PAGING

Any of the primary[2] operating modes can be selected when the control is not in cycle[3]. To select a mode, press the MODE SELECT key, then type the single letter or number that identifies the mode on the display screen. For example, MODE SELECT 0 selects the Jog mode; MODE SELECT G selects the Graphics mode.

Once you have selected an operational mode, there are three key sequences that you must learn to select options and enter data into the control.

Paging

Some of the operational modes require multiple pages to display all of the mode's available entries. For example, the Catalog mode requires two pages to list up to 150 part program titles that are available. Whenever a display screen has more than one page, you can use the ARROW keys to move from page to page. The ARROW keys are located on either side of the soft keys. Multiple page screen displays are easily identifiable because the active page number will appear on the lower right corner of the screen. If a page number does not appear on the screen, the ARROW keys will not be functional.

[2] The primary modes, shown on the above screen, are modes that can be selected when Concurrent operations are not being used. Secondary modes can be selected when Concurrent operations are in use. For more information on Concurrency, see the Auto Mode section of this manual.

[3] The control is "in cycle" when a part program is executing in Graphics, Program, Auto, Single Event modes, or when you are jogging the axes. The CYCLE START lamp, if installed, will light whenever the control is in cycle. Although the CYCLE START lamp remains off, the control is also considered to be "in cycle" when data is being transmitted through one of its serial ports.

Selecting From a List of Options

To select an item from a list on the display screen, simply type in the 1, 2, or 3 character identifier that precedes the item, then press ENTER. Once you have selected an item, it will be highlighted on the display screen.

Entering Numeric Values

To enter numerical information into the control, type in the 1, 2, or 3 character identifier that proceeds the item, type the number you wish to store, then press ENTER. Leading and trailing zeros are not required. Incremental values are entered by pressing the INCR key. Negative values must include a minus sign character.

If the number you enter is not valid for the selected item, a FORMAT ERROR message will be displayed. Press the CLEAR key to remove the error.

To remove a key sequence before you press the ENTER key, press CLEAR.[4]

To remove a part program entry after you press ENTER, key in the letter identifier, then press CLEAR.

TURNING THE CONTROL OFF

In general, you must:

1) Follow your machine supplier's power off instructions. One step in this procedure will include releasing the EMERGENCY STOP button to remove power from the drives.

2) After pressing the EMERGENCY STOP button, **observe a 30 second delay.** This short delay before step 3 allows power to dissipate from the machine's electronics to avoid axis motion. Otherwise, power dissipation through the axes motors could cause the axes to jump.

3) Move the control's POWER SWITCH to the OFF position.

DAILY START UP ROUTINE

Once you have turned the machine on, follow this general Start Up Routine. The operation specified by each step is followed, in parenthesis, by the section in this part of the manual that details the operation.

If this is the machine's first start up, see the next topic instead.

1) Turn the control and machine ON.

2) Reference your machine using REFERENCE ZERO or MACHINE ZERO as specified by your machine supplier. (Jog mode)

3) Select a part program from the Catalog mode. (Catalog mode)

4) Complete any tool offsets that are required by your part program. (Tool Tables mode)

[4] If you are in Program mode, the sequence that you clear must have both a letter and a number. If only a letter is present, the existing part program data for the letter will be erased when you press the CLEAR key.

5) Set a part zero using one of the following techniques:
   - Use the reference tool and the FIXTURE CALIBRATE feature to locate and store all fixture offset locations that are used in your part program (Tool Tables mode).
   - Use the reference tool to perform a ZERO SET operation. (Jog mode)

6) Choose the Auto or Single Event mode to begin machining. (Auto or Single Event mode)

FIRST TIME START UP ROUTINE

If the control is being brought into service for the first time, or a POWER FAIL message was displayed on the Start Up screen, you must complete this procedure in the order described below. Each item is described later in this manual. The operation specified in each step is followed, in parenthesis, by the section of the manual that details the operation.

One or more of items 2, 3, and 5 may have been completed by the machine supplier prior to the machine's delivery. If an item was completed, the measured values will be stored in the control's parameter PROM and will be displayed in the Set Up mode. Items which have been completed by the supplier will display non-zero values in their respective Set Up mode (Mode 5) entries.

1) Establish Inch or millimeters units of measurement. (Set Up mode)

2) Test and enter Axis Reversal Compensation for all axes. (Set Up mode)

3) Enter Preliminary Software Travel Limits. (Set Up mode)

4) Reference all axes. (Jog mode)

5) Locate and enter final Software Travel Limits. (Set Up mode)

6) Locate and enter Home position. (Set Up mode)

7) Calibrate and store tool length and cutter diameters. (Tool Tables mode)

8) Enter tool numbers into the Q - Random Tool Table[5]. (Tool Tables mode)

9) Set a part zero with using of the following techniques:
   - Use the reference tool and the FIXTURE CALIBRATE feature to locate and store all fixture offset locations that are used in your part program (Tool Tables mode).
   - Use the reference tool to perform a ZERO SET operation. (Jog mode)

[5] This step need not be performed unless your machine has an automatic tool changer that stores the last tool in the same pocked from which the new tool is retrieved.

MEMORY SUPPORT

Your control contains advanced CMOS memory circuitry to reduce "off-time" power consumption and extended memory support time. External batteries and chargers are not required. SRAM memory retains information such as Part Programs, Set Up mode information, and Tool Table information. A lithium battery will retain this data for ten years or more of cumulative off time under normal operating conditions.

BATTERY WARNING

When monitoring circuitry senses the need to change the battery, you will be informed by a WARNING-BACKUP BATTERY LOW message on the Start Up display. When this screen message appears, you must add an second battery to the CNC's memory board. The battery and battery pack may be obtained from your local electronics supply retail store. The battery addition will permit battery change without loss of stored data.

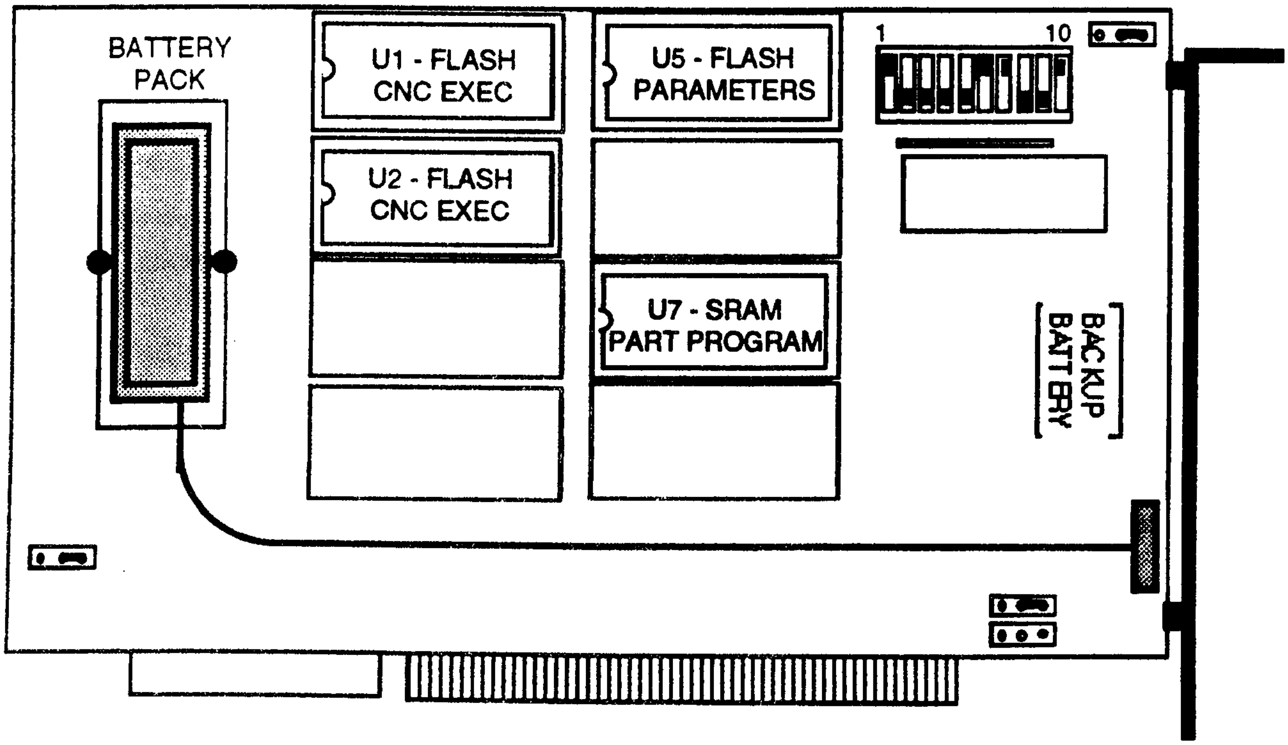
BATTERY CHANGE INSTRUCTIONS

The following procedure describes how to replace the Memory board's backup battery.

1) Turn system power off and gain access to the CNC System Module in the machine's electronic cabinet.

2) Note the Memory board's location from the card rack cover, remove the card rack cover, then remove the Memory board. Place the Memory board on an electrostatic mat.

3) The backup battery will consist of a battery holder, securing tie wraps, cable, and a 3.6 volt battery. The replacement pack is available from Autocon Technologies, or it may be purchased from most stores which carry electronic supplies. Attach the battery pack as shown below.



4) After securing the battery pack to the Memory board, insert the battery (ensuring that you align the plus and minus terminals with the guide on the battery pack), and connect the cable to the Auxiliary Battery connector. The Auxiliary Battery connector is keyed.

5) Insert the Memory board into its card slot and secure it with the appropriate fasteners. Replace the card rack cover and turn system power ON.

BATTERY FAIL

If you fail to replace the battery when the caution appears, the battery will continue to discharge. When the battery can no longer retain stored data, you will be advised on the next Start Up screen by a POWER FAIL message. At this point, you must replace the battery and reload the lost information.

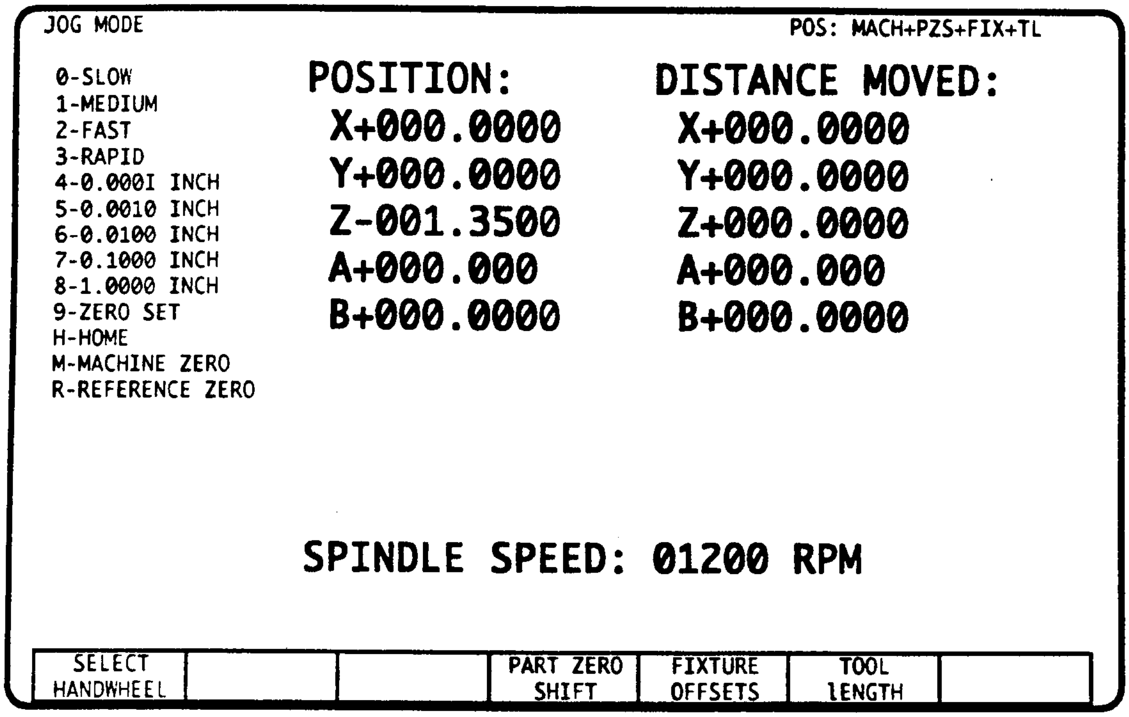
CRT BLANKING

In the process of setting up your control, your machine supplier may add a screen blanking feature that darkens the screen when there is no activity for a predetermined length of time. To recover press any key on the keyboard. Screen blanking is disabled while you are cycling a part in Auto or Single Event modes.

OPERATIONS

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON



```
POS: MACH+PZS+FIX+TL

POSITION:        DISTANCE MOVED:
 X+000.0000       X+000.0000
 Y+000.0000       Y+000.0000
 Z-001.3500       Z+000.0000
 A+000.000        A+000.000
 B+000.0000       B+000.0000

SPINDLE SPEED: 01200 RPM
```

JOG MODE – INTRODUCTION

The Jog mode is a primary operational mode. Select this mode to position one or more axes by choosing a rate, increment distance, or specifying a return to an assignable Home position. This mode is also selected to reference the axes and set a part program datum.

The jog rate or incremental distance you choose here will be active for jogging the axes in this mode, the Tool Tables mode, or the Learn mode. The selected jog rate also specifies how fast you can move the cursor in the Graphics mode.

The PART ZERO SHIFT, FIXTURE OFFSETS, AND TOOL LENGTH soft keys let you select the coordinate system that is displayed in the control's Position Display registers. This feature lets each user select the coordinate system that is most useful for the operation being performed. Eight different coordinate systems are available.

The letters A, B, C, U, V, W identify auxiliary axes. Some of these letters will appear when one or more auxiliary axes are installed on your machine.

DEFINITIONS

Machine Coordinate Zero - this point is established every time you reference the axes with the MACHINE ZERO or REFERENCE ZERO operation. Machine Coordinate Zero serves as an anchor for all machine dimensions such as Software Travel Limits and Home position. To determine the location of Machine Coordinate Zero, reference the axes then jog each axis to the point where its Position Display reads zero[1].

Datum - is the origin of each part, established with a ZERO SET operation. After you complete a ZERO SET, all part program dimensions are measured from this point. If your setup procedure uses a ZERO SET operation, the ZERO SET must be performed each time you reference the axes or turn the control on.

If your setup does not include a ZERO SET operation, the control assumes that Machine Coordinate Zero, by default, is the datum. Using this default datum (i.e. not performing a ZERO SET) is often preferred when Fixture Offsets are used to locate part zero.

Part Zero - is the current location of the coordinate X0, Y0, Z0 on the machine. In most situations, thc datum and part zero will be the same point. However, you can move the part zero location away from the datum by activating a Fixture Offset, a Part Offset, or a G6 (EIA/ISO G92) Load Position. Each of these operations is described later in this manual. Using Fixture Offsets to establish part zero is preferred by most machinists.

POSITION DISPLAY REGISTERS

The upper right column of the Jog mode screen displays the position of each axis.

You can specify which coordinate system is displayed in these registers through the use of the PART ZERO SHIFT, FIXTURE OFFSETS, and TOOL LENGTH soft keys.

Each soft key acts as a toggle that determines whether a specific type of offset will be included in the Position Display. The highlighted text in the MACH+PZS+FIX+TL field at the top of the screen identifies which offsets are currently included in the Position Display.

Here are the settings that you will find most useful as you operate your machine:

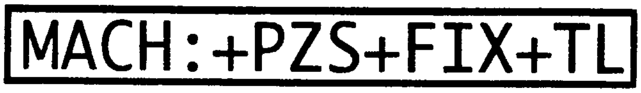
Display machine coordinates: Press the three soft keys until only the MACH text field is highlighted on the display screen as shown below. When these options are selected, the Position Display registers will show the position of the spindle nose relative to Machine Coordinate Zero[2].

MACH:+PZS+FIX+TL

[1] Use caution, some machines may be set up in a manner that will not allow you to position the spindle nose at machine zero.

[2] If you have not referenced the machine, this setting will display the position of the spindle nose relative to its position at the time that the machine was turned on.

Display part coordinates: Press the three soft keys until all of the text fields are highlighted on the display screen as shown **below**. When these options are selected, the Position Display registers will show the position of the tool tip relative to the current location of part zero.



```
MACH:+PZS+FIX+TL
```

In technical terms, these three soft keys allow you to select which types of offsets are reflected in the control's Position Display registers. Notice that the MACH field on the screen is always highlighted and cannot be turned off. Highlighting one or more of the remaining three fields will cause the control to include the following offsets in the Position Display.

PZS - Part Zero Shift Offsets
- Jog mode Zero Set offset
- Position event G6 Load Position command
- EIA/ISO G92 Axis Preset command
- Handwheel Interrupt Offsets

FIX - Fixture and Setup Offsets
- E Code Fixture Offset
- Setup mode PX and PY Part Offset
- Setup event A and B Part Offset
- Offset for a slaved axis.

TL - Tool Length Offset
- Tool Length offset for the active tool

DISTANCE MOVED OR AXIS LAG DISPLAY

Each time you enter Jog mode, the Distance Moved display is set to zero. This display retains the distance each axis has moved relative to its position on entry to this mode. Use this display to track how far you have jogged.

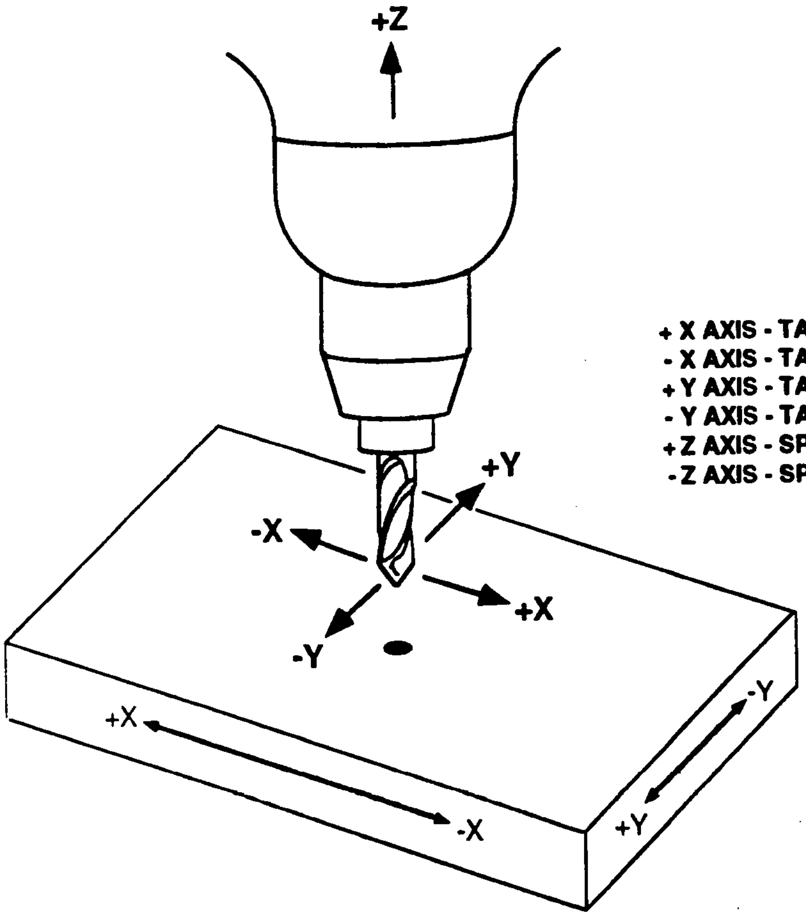
An Axis Lag display replaces the Distance Moved display if TM4 (Test Mode 4) is selected in the Set Up mode. This display assists the machine installer while adjusting your machine's servo drives.

SPINDLE SPEED DISPLAY

If the spindle is controlled by part program commands, this display will identify the commanded spindle speed. Any speed adjustment due to the SPINDLE OVERRIDE potentiometer (if installed) will be reflected in this display.

AXES ORIENTATION

The illustration **below** shows the relationship between tool motion commands (issued by jogging an axes or by executing a part program) and actual table motion. Even though the tool is stationary in the XY plane, most part programmers create part programs by visualizing the tool moving around a fixed part.



+ X AXIS - TABLE MOVES LEFT
- X AXIS - TABLE MOVES RIGHT
+ Y AXIS - TABLE MOVES OUT
- Y AXIS - TABLE MOVES IN
+ Z AXIS - SPINDLE MOVES UP
- Z AXIS - SPINDLE MOVES DOWN

JOG MODE OPTIONS

0, 1, 2, 3 - CONTINUOUS JOG RATES

Press 0, 1, 2, or 3 to enable continuous jog at Slow, Medium, Fast or Rapid rate. Your Jog mode selection will be highlighted on the video screen.

The jog rates for each selection are:

Identifier	Selection	X,Y,Z or Aux. linear axis Rate (ipm/mmpm)	Auxiliary rotary axis Rate (degrees/min)
0	Slow	1/25	10
1	Medium	10/254	100
2	Fast	100/2540	1000
3	Rapid	determined by machine supplier	

Once the rate is selected, jog an axis by toggling its respective JOG switch. The jog will continue at the selected rate until the switch is released. The X and Y axes may be jogged simultaneously, but neither may be jogged simultaneously with Z.

The FEEDRATE OVERRIDE potentiometer can modify the Slow, Medium and Fast jog rates. The FEEDRATE OVERRIDE potentiometer will also control the Rapid jog rate when TM1 (Test Mode 1) is selected in the Set Up mode.

4, 5, 6, 7, 8 - INCREMENTAL JOG DISTANCES

Press 4, 5, 6, 7, or 8 to enable the control to move an axis a precise incremental distance, then halt. The increment distances for each selection are:

Identifier	X,Y,Z, Aux. linear axis increment (inch/mm)	Rotary Index axis, either 1°/5° minimum	Rotary Contouring axis
4	0.0001/00.001	1°/5°	.001°
5	0.0010/00.010	10°/10°	.01°
6	0.0100/00.100	100°/100°	0.1°
7	0.1000/01.000	100°/100°	1.0°
8	1.0000/10.000	100°/100°	10.0°

Once the increment is selected, jog an axis by toggling its respective JOG switch. The jog continues until the axis has moved the selected distance. One axis may be incrementally jogged at a time. Incremental jogs occur at 10 ipm (254 mmpm), adjustable with the FEEDRATE OVERRIDE potentiometer. To temporarily halt an incremental jog, switch the FEEDRATE OVERRIDE potentiometer OFF. To abort an incremental jog, press MOTION STOP.

WARNING - INITIALIZE THE MACHINE FIRST

Before using any of the following operations, you must conform to your machine supplier's prescribed initialization sequence to avoid damage to the machine or injury to personnel. The Start Up Procedures section in Part 1 of this manual describes initialization activities and refers to the following operations.

H - RETURN TO HOME POSITION

Do not use this option until the axes have been referenced. Use this selection to return an axis at rapid rate to its Home coordinate. Home is a position that allows the spindle to clear the part for tool change operations or fixture adjustment. You can assign the coordinates of the Home position in the Set Up mode.

To return all axes to the Home position:

1) Press H to select the Home option.
2) Toggle the Z axis JOG switch to Home the Z axis.
3) Toggle the remaining JOG switches to Home each axis in the same manner.

REFERENCE THE MACHINE

You must reference the machine's axes each time the control is turned on. Referencing establishes a reference position for Home coordinates, Software Travel Limits, and the tool change position. Referencing also activates Software Travel Limit checking. In some applications you may not be able to jog at the Rapid rate, jog Home, or execute a part program until you reference all axes. The DynaPath control provides two referencing options: REFERENCE ZERO and MACHINE ZERO. Your machine supplier will tell you which one to use. The Status[3] screen identifies the axes that have not been referenced.

Your machine supplier may design a custom referencing cycle that simplifies the following procedure. Always follow your machine supplier's referencing instructions.

R - REFERENCE ZERO OPERATION

If your machine supplier has installed reference limit switches, you must perform this operation to reference the axes each time the control is turned on. To reference the axes with REFERENCE ZERO:

1) Select a continuous jog rate.
2) Jog each axis near its reference limit switch.
3) Press R to select the Reference Zero operation.
4) Toggle and release a JOG switch to move an axis onto its Reference Limit switch.

Once an axis contacts its reference limit switch, the control halts when it detects the encoder's next marker pulse. Each axis display will reset to zero, or a coordinate which has been assigned by your machine supplier.

M - MACHINE ZERO OPERATION

If reference limit switches are not installed, perform this operation to reference the axes each time the control is turned on. To reference the axes with MACHINE ZERO:

1) Jog each axis to its reference position. The reference position is usually located by aligning a pointer to an index mark along each axis.
2) Press M to select the Machine Zero operation.
3) Toggle and release a JOG switch to identify that the axis is at the reference position. Each axis display will reset to zero.

9 - ZERO SET

The ZERO SET operation should **not be attempted** until initialization activities have been completed, tool offsets have been entered, a reference tool has been loaded into the spindle, and the fixture and work are in place.

This option sets the part program zero point (datum). After this operation, the control will interpret part program commands and display axes position with respect to this datum.

Zero Set offsets are cleared each time you reference the axis or turn the control off. For this reason, most users who require an identical setup for a multiple shift production run prefer to use Fixture Offsets as an alternative to performing a ZERO SET at the start of every shift.

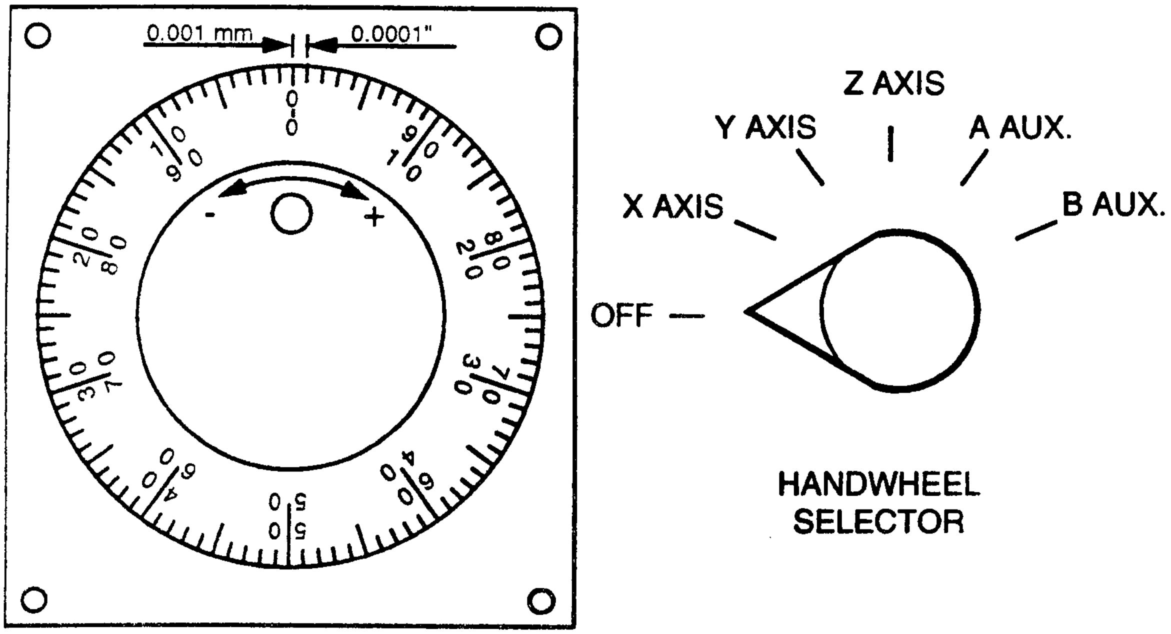
[3] To view the Status screen, press the FAULT/STATUS key, then press the STATUS DISPLAY soft key.

To perform a ZERO SET:

1) Put the reference tool[4] in the spindle. Cancel any active tool length compensation and fixture offset by executing T00[5] and E00 in an M Function event or EIA/ISO block.
2) Jog the tool tip to the desired part zero.
3) Choose the ZERO SET option.
4) Toggle each JOG switch to identify that the axis is at the Zero Set position. Each corresponding Position Display register will reset to zero.

HANDWHEEL JOG

The JOG HANDWHEEL is a machine supplier installed option that allows quick and accurate incremental jogs. To operate the JOG HANDWHEEL, select an incremental distance 4, 5, or 6[6]. Next choose the axis to jog by pressing the SELECT HANDWHEEL soft key[7] which appears when one of these incremental distances is selected. Repeatedly press the soft key until the desired axis is displayed on the video screen. You may now jog the axis by turning the handwheel. Each division, or click, on the handwheel represents 1 incremental jog at the selected distance. A typical JOG HANDWHEEL is shown below.



[4] Implementation and use of the reference tool is described in the Tool Tables mode section of this manual.

[5] If your control uses H and D codes to activate tool length and diameter offsets (see Part 1-Section 2) H00 and D00 must be the active codes when you perform the ZERO SET operation with the reference tool.

[6] Incremental distances 7 and 8 are not available for handwheel jogs since a handwheel turn can exceed the rapid rate for the selected axis. Your machine supplier may also lock out handwheel operation with Jog mode selections 6 and/or 5.

[7] Instead of using a soft key, your machine supplier may install a rotary HANDWHEEL SELECT switch to select the axis.

HANDWHEEL INTERRUPT

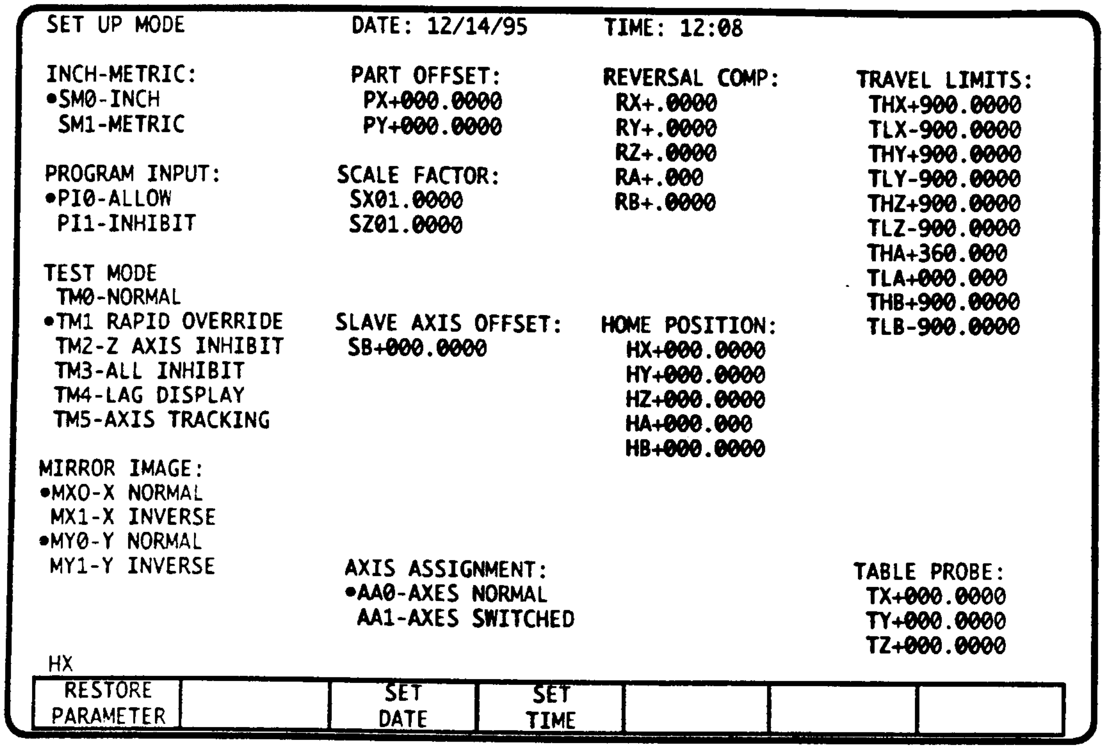
Handwheel Interrupt will only be installed on a small number of machines that have been specially designed to utilize this feature. In most cases, your machine supplier will provide you with instructions for using this feature. You can determine whether handwheel interrupt is available by searching for a HANDWHEEL INTERRUPT, or similarly titled, switch or pushbutton mounted on your machine's operator panel.

Regardless of the position of the HANDWHEEL INTERRUPT control, each click of the jog handwheel will **always move** the selected axis a distance equal to the selected jog increment (selection 4, 5, or 6 in the Jog mode).

When the HANDWHEEL INTERRUPT control is active, your part datum will automatically shift with each motion generated by the handwheel. Since the part datum is being shifted, any axis motion from the jog handwheel will not be displayed on the screen when PZS[8] is selected as part of your position display coordinates. However, since the control continues to keep track of machine coordinates, any axis motion will be displayed when the PZS[8] component is not selected as part of the position display coordinates.

The handwheel interrupt offset will be canceled when you ZERO SET or REFERENCE an axes, select a new part program from the control's Catalog, or when you turn the control off.

[8] The PZS offset component can be added and removed from the position display by pressing the PART ZERO SHIFT soft key in the Jog mode.



```
SET UP MODE                DATE: 12/14/95       TIME: 12:08

INCH-METRIC:               PART OFFSET:         REVERSAL COMP:       TRAVEL LIMITS:
•SM0-INCH                   PX+000.0000          RX+.0000              THX+900.0000
 SM1-METRIC                 PY+000.0000          RY+.0000              TLX-900.0000
                                                 RZ+.0000              THY+900.0000
PROGRAM INPUT:             SCALE FACTOR:         RA+.000               TLY-900.0000
•PI0-ALLOW                  SX01.0000            RB+.0000              THZ+900.0000
 PI1-INHIBIT                SZ01.0000                                  TLZ-900.0000
                                                                       THA+360.000
TEST MODE                                                              TLA+000.000
 TM0-NORMAL                                                            THB+900.0000
•TM1 RAPID OVERRIDE        SLAVE AXIS OFFSET:   HOME POSITION:         TLB-900.0000
 TM2-Z AXIS INHIBIT         SB+000.0000           HX+000.0000
 TM3-ALL INHIBIT                                  HY+000.0000
 TM4-LAG DISPLAY                                  HZ+000.0000
 TM5-AXIS TRACKING                                HA+000.000
                                                  HB+000.0000
MIRROR IMAGE:
•MXO-X NORMAL
 MX1-X INVERSE
•MY0-Y NORMAL
 MY1-Y INVERSE             AXIS ASSIGNMENT:                            TABLE PROBE:
                           •AA0-AXES NORMAL                             TX+000.0000
                             AA1-AXES SWITCHED                          TY+000.0000
                                                                        TZ+000.0000
HX
| RESTORE  |        |  SET   |  SET   |        |        |        |
| PARAMETER|        |  DATE  |  TIME  |        |        |        |
```

SET UP - INTRODUCTION

Select this mode to set or change initialization parameters related to machine envelope dimensions. You may also select Test Modes which extend feedrate controls, inhibit axes motion, or assist with installing and calibrating the machine. This mode may also be selected to alter part program execution by establishing a part offset, scale factor, or invoking axis symmetry. Several convenient utility functions are also enabled in this mode.

The Set Up mode screen is shown **above**. AXIS ASSIGNMENT, SLAVE AXIS OFFSET, and TABLE PROBE fields only appear when the appropriate hardware is installed on the machine.

The RESTORE PARAMETER soft key allows you revert to your machine supplier's parameter stored Reversal Compensation, Travel Limits, and Home Position values.

MACHINE ENVELOPE DIMENSIONS

The items in this topic must be entered and tested before you run your machine. Refer to the Start Up Procedures section for the order in which to prepare the machine for operation. Once completed, these entries are retained by the control's backup battery. The dimensional entries in this topic may have been measured by the machine supplier and stored in the control's Parameter PROM prior to the machine's delivery. In this case, record the entries in a written log, and skip to the next step.

These dimensions should be recorded in a written log for easy retrieval in case of a backup battery failure.

SM0, SM1 - INCH/MILLIMETERS - MEASUREMENT UNITS

Choose SM0 for inch, or SM1 for millimeters in the Set Up mode. You must measure all following dimensions in the units you select here.

After you have completed the initialization sequence, you may use this entry to switch the control to INCH or METRIC mode. When you switch modes, the control will convert all physical dimensions (position displays, travel limits, tool sizes, reversal error, etc.) into the new units. However, the control will interpret all part program dimensions in the units (inches or millimeters) that are selected here.

R - AXIS REVERSAL (BACKLASH) COMPENSATION

These entries compensate for mechanical lash whenever an axis reverses direction. If the Reversal Comp entries are nonzero on initial start up, your supplier has already measured the reversal error and entered the correct compensation in the parameter PROM. If your machine uses linear scale feedback, consult your machine supplier for the correct procedure to measure reversal compensation. If all entries are zero, and your machine uses rotary encoders, measure the axis reversal error and enter compensations as follows:

1) Position a dial indicator on the test axis to indicate the slide position.
2) Move to the Jog mode. Choose option 0, the Slow jog rate.
3) Jog the axis in the + direction until the indicator moves for the first time.
4) Press MODE SELECT 0 to reinitialize the Jog mode. This will return the Distance Moved display to 0,0,0.
5) Choose option 4 (.0001" or .001 mm).
6) Repeatedly jog the axis in the - direction until the indicator just begins to move.
7) Log the axis tested and the amount of reversal error. The reversal error is determined by subtracting .0001 inch (.001 mm) from the test axis Distance Moved display.
8) Repeat the procedure for each axis.
9) Move to the Set Up mode and enter the Reversal Compensation for each axis.

SOFTWARE TRAVEL LIMITS

These entries establish a software protected machine envelope which is typically located inside the boundaries established by travel limit switches. A high (TH) and low (TL) limit must by entered for each axis. After the machine is referenced, an axis move or jog that exceeds the Software Travel Limits will cause the control to halt axes motion and display a TRAVEL LIMIT fault message. Power will not, however, be removed from the drives. To recover, move to the Jog mode, select a jog rate or increment, then jog the fault axis back within the defined boundary.

If, at initial startup, the Software Travel Limits are non-zero and are not excessively large (±999.9999" or ±9999.999 mm), then your machine supplier has predetermined the Travel Limits and entered them in the parameter PROM. If hardwired Travel Limit switches are installed on the machine and are functioning properly, you should test the existing Software Travel Limits by referencing the machine, then slowly jogging to each of the defined limits. If the limits are working properly, log these values.

If, at initial startup, the Software Travel Limits are all zero or they define a range much smaller than the true machine envelope, then you must enter preliminary Travel Limits which will exceed the physical limits of your machine. These preliminary values allow you to freely jog the axes while finding the actual limits of your machine. We recommend that you enter +999.9999" (9999.999 mm) for the high limits and -999.9999" (-9999.999 mm) for the low limits. Enter ±360° for a rotary auxiliary axis which requires Software Travel Limits.

Travel limits for the auxiliary axes are required when THx and TLx axis identifiers appear after the travel limits for the X, Y, and Z axes. The "x" character depends on which letter (ABCUVW) is used to program the auxiliary axis.

LOCATE SOFTWARE TRAVEL LIMITS

After entering preliminary software travel limits, you must now determine final high and low settings for each axis. You may determine the necessary values from the Jog mode's Position Display registers. Perform this procedure after referencing the axes, and before setting a part datum with a ZERO SET operation.

Jog to approximately 1/4 inch (5 mm) (2° for a rotary auxiliary axis) inboard of each overtravel switch and log, on paper, the axis, sign, and dimension of this position. You may wish to set the -Z travel limit just above the table surface.

Repeat this operation until a high (+) and low (-) travel limit is logged for each axis.

Move to the Set Up mode and enter these values in the same manner as you entered the preliminary values. Retain these values in a written log. Once the values have been entered, reference all axis and test the values by slowly jogging to each travel limit. Use caution while jogging near the machine's end of travel.

H - HOME POSITION

Home is a position that allows convenient part loading and manual tool changes. The M30- End of Program command and the H option from Jog mode position the Z axis and XY table to this point. The M06-Tool Change command will also position the machine here unless the supplier designates a different tool change position. (Refer to Part 1 - Section 2).

If, at initial start up, the Home dimensions are nonzero, your supplier has located this position and entered its coordinates in the Set Up mode. In this case log the Home position coordinates. You can change this location at a later date, if desirable.

If each entry is zero, Home will be positioned at machine coordinate zero. Perform this step after referencing the axes, and before executing a Jog mode ZERO SET operation. To move Home position, jog to the most convenient position. Log the signed Position Display coordinates for each axis, return to the Set Up mode, then entered the coordinates.

TM - TEST MODES

Test Modes can extend feedrate override control, inhibit axis motion, assist servo adjustment, or facilitate operations with external part tracers.

TM0 - NORMAL OPERATIONS

This entry cancels a TM1 - TM5 selection. This is the recommended setting for production runs of tested part programs since rapid moves will not be unnecessarily slowed by the feedrate override potentiometer.

TM1 - RAPID FEEDRATE OVERRIDE

TM1 adds feedrate override control to programmed rapid motions and rapid jogs. This is the recommended setting for short run parts and prototype work since it provides the operator with feedrate override control of all rapid moves.

TM2 - Z AXIS INHIBIT

TM2 allows the part program to run with X, Y and, if installed, auxiliary axis motion, but not Z. The Position Display registers and Graphics will simulate programmed Z axis motion. This mode also enables the TM1 override.

TM3 - ALL AXIS INHIBIT

TM3 allows the part program to run while inhibiting all axis motion. The Position Display registers and Graphics will simulate all axis motions in the proper program sequence. This mode also enables the TM1 override.

TM4 - LAG DISPLAY

TM4 alters the Jog, Auto, and Program mode's secondary position displays to show axis lag. Maintenance personnel may use these displays to monitor axis following error when making servo adjustments according to the drive manufacturer's specifications.

TM5 - AXIS TRACKING

In some applications, TM5 allows the control to provide readout for external parts tracers. When this mode is selected, the transducers track axis motion and each axis position is displayed, but the axes are driven by means external to the control.

UTILITIES

These functions enable or disable the utilities described below.

PI - PROGRAM INPUT

This option may be used to inhibit program entries or edits from the keyboard. When PI1 PROGRAM INHIBIT is entered, any attempt to enter data from the Program mode will result in a PROGRAM INHIBIT error. The PI1 entry also inhibits operations in Delete Program mode. Access to Transmit and Receive modes is not restricted. Your machine supplier can also add provisions to inhibit entries to the Set Up mode with a key operated switch.

AA - AXIS ASSIGNMENT

Some machine applications provide an auxiliary machining head. Once the auxiliary head is mounted, this option is selected to inform the control that jog and programmed moves should be reassigned to the switched axis configuration. It is necessary to press RESET and reference the axes after changing this setting.

Sx - SLAVED AXIS OFFSET

A slaved axis always moves in tandem with the axis to which it is slaved. The most common use of slaving is to control a machine with two spindles. In this case, your machine's linear 4th, 5th, or 6th axis will be slaved to the Z axis. The slaved axis may not be jogged or programmed independently.

A slave offset entry is available in the Set Up mode for each slaved axis. This offset is used when the slave and master axes do not physically align; the distance that the slaved axis must move to align with the master axis should be entered as the slave axis offset. This offset is typically used when the tool in the slaved axis is longer or shorter than the tool in the Z axis. The difference between the tool lengths is entered as the slave offset.

The CNC will initially add the slave offset into the command position during execution of the first part program motion command following a Reference Zero Operation. The offset will remain active unless the control temporarily removes it when the master axis is programmed to a position that is defined in machine coordinates. Events that program an endpoint in machine coordinates include: M06, M30, G27, G28, G29, and G53. The control will reinstate the slave offset during execution of the next command that is programmed in part coordinates. Since the slave offset is temporarily canceled, it is very important that you consider the motion of both the master and slave axes when you program one of these machine coordinate events.

PART PROGRAM - RUN OPTIONS

The following "run options" let you activate several special effects which can greatly reduce your part programming effort. These entries let you activate a part offset, a part scale factor, or a mirror image along the X and/or Y axes. These features can alternately be controlled by execution of a part program Set Up event.

Introductory descriptions are given here. The Set Up event section of this manual describes each of these effects in further detail.

PX, PY - PART OFFSET

This option allows relocation of the X and Y part zero without modifying the original ZERO SET position. The signed absolute X and Y entries assign the distance each zero coordinate will be offset from the datum[1]. However, if a fixture offset assigns a part zero[1] that is away from the datum, these entries will assign the distance that each zero coordinate will be offset from the fixture offset.

Operationally, the offset is recognized on the first programmed tool motion. Executing M02, M30, or pressing RESET does not cancel the XY offset. The offset will be active for the next part program cycle.

MX, MY - MIRROR IMAGE

These options allow the mirror image of a part to be cut from an existing part program. When these entries command a change in X or Y axis symmetry, the control will interpret the X and/or Y dimensional entries to be sign reversed. Jog directions are not affected by these entries. Executing M02, M30, or pressing RESET does not cancel the mirror image setting.

[1] This term is defined in the Jog mode section.

SX, SZ - SCALE FACTOR

These options allows a .0001 to 99.9999 scale factor to be applied to the program. SX applies the scale factor to programmed X and Y axes commands and SZ applies the scale factor to Z axis commands. Both are unsigned absolute entries. Incremental jogs are not scaled by either entry. Executing M02, M30, or pressing RESET will set both scale factors back to 1.

RESTORE PARAMETER

The RESTORE PARAMETER soft key is intended to be used when you wish to discard one of your Reversal Compensation, Home Position, or Travel Limit entries and restore the value that your machine supplier has stored in the control's parameter PROM.

To use this feature, key in the two or three letter identifier for the value that you wish to restore. If this value is stored in the parameter PROM, a RESTORE PARAMETER soft key will appear. Pressing this key will cause the control to insert the PROM stored value into the correct Set Up mode field.

The following entries are stored in the parameter PROM and may be restored with this feature.

- Reversal Compensation entries
- Home Position entries
- Travel Limit entries

SET DATE and SET TIME

The SET DATE and SET TIME soft keys allow you to enter the date and time into the control. Instructions on how to set the date and time are:

To set the date: press the SET DATE soft key. Notice that the correct format (xx/xx/xx) is displayed on the bottom of the screen. Now type in the month, day, and year of the correct date. You do not have to type in the "/" characters. Example: the keystrokes to enter the date July 12, 1993 are: SET DATE 071293 ENTER.

To set the time: press the SET TIME soft key. Notice that the correct format (xx:xx) is displayed on the bottom of the screen. Now type in the hour and minute of the correct time in military format (i.e. the hours range from 0-24). You do not have to type in the ":" character. Example: the keystrokes to enter the time 9:30 AM are: SET TIME 0930 ENTER. The keystrokes to enter the time 8:53 PM are SET TIME 2053 ENTER.

TOOL OFFSETS - INTRODUCTION

The control has the capacity to store length and diameter offsets for up to 200 tools. Offsets stored in the T, H, or D tables become activate when a T, H, or D code executes in an M Function or EIA/ISO event. When a tool offset is active, the control will adjust all tool motion to compensate for the tool's length, and diameter[1]. Information about activating tool offsets with an M Function event are provided in Part 4 of this manual. Details about activating tool offsets with an EIA/ISO event are provided in Part 7 of this manual.

Your control, depending on how your machine supplier has configured your CNC (refer to Part 1-Section 2), will use one of the following methods to store tool length and diameter offsets.

1) **Use a single T Table to store all tool length and diameter offsets.** A tool's offsets will become active when the control executes an event similar to the M Function event or EIA/ISO event shown below. An example T table is shown on the facing page, upper illustration.

   N10 (9) T04 or N10 (E) T04

   With this configuration, the single T code specifies the tool that must be put in the spindle, the tool length offset, and the tool diameter offset.

2) **Use separate H and D tables to store tool length and diameter offsets.** A tool's offsets will become active when the control executes an event similar to the M Function event or EIA/ISO event shown below. An example H table is shown on the facing page, lower illustration.

   N10 (9) T04 H04 D04 or N10 (E) T04 H04 D04

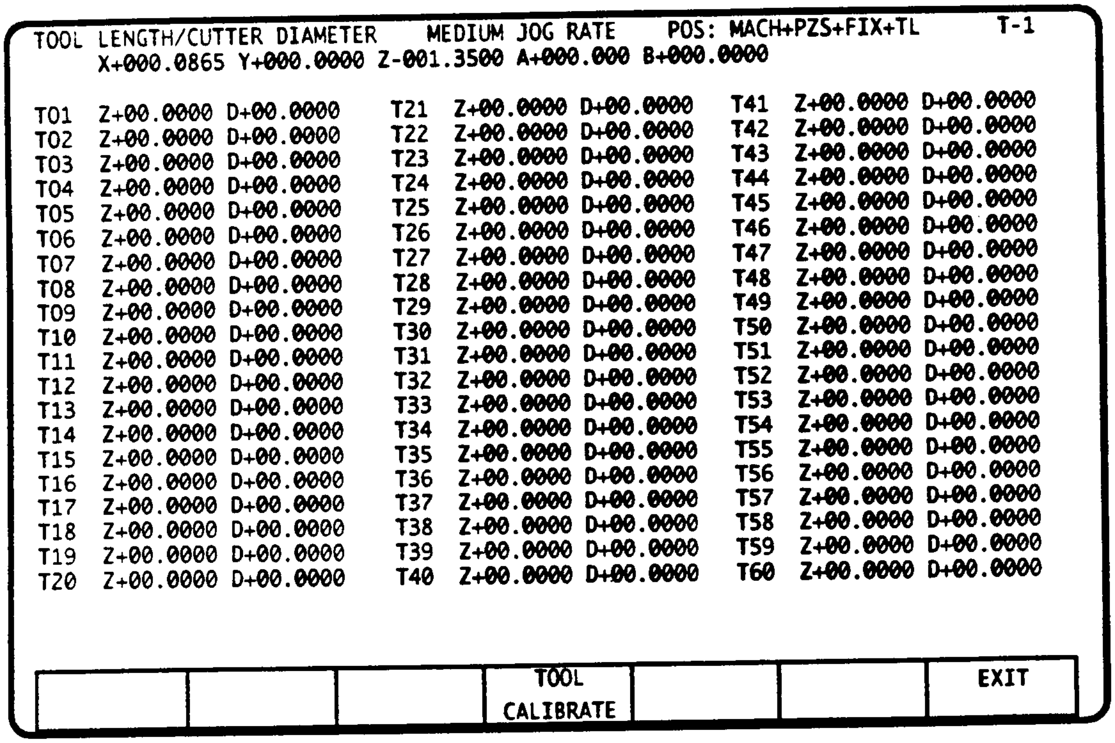
   With this configuration, the T code specifies the tool that must be put in the spindle, the H code specifies the tool length offset, and the D code specifies the tool diameter offset.

Note 1: A procedure to determine how your control has been configured by your machine supplier is provided in Part 1 - Section 2 of this manual. It is very important that you determine how your control has been set to store and activate H and D codes.

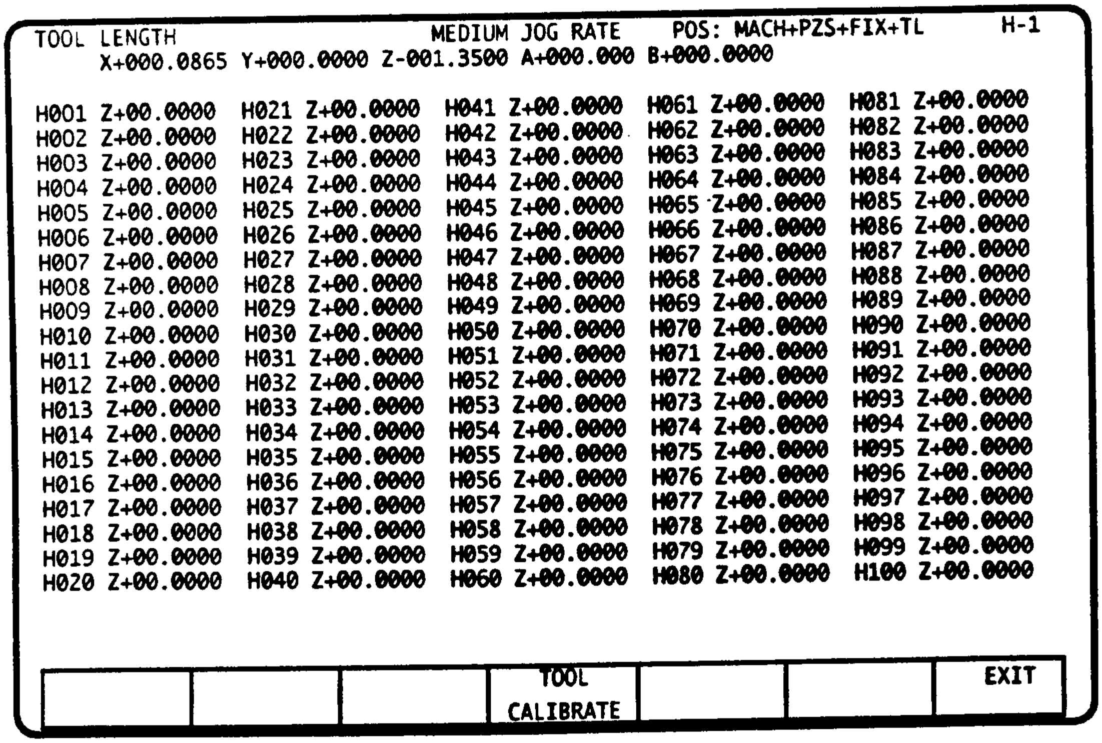
Note 2: The manner in which your control handles tool offsets will affect part program compatibility between two DynaPath controls. In order to cycle a part program that activates tool offsets with H and D codes on a control that uses a single T table, or vice versa, will require you to edit all events that activate tool offsets.

Note 3: Most of the examples in this manual have been programmed for controls that use H and D codes. In order to run these programs on a control that uses T codes, you must first edit all events that activate tool offsets.

[1] The control will offset for the tool's diameter when cutter compensation is active or when the control executes a Frame Mill, Circle Mill, or Cavity Mill event.



```
TOOL LENGTH/CUTTER DIAMETER     MEDIUM JOG RATE     POS: MACH+PZS+FIX+TL      T-1
     X+000.0865 Y+000.0000 Z-001.3500 A+000.000 B+000.0000

T01  Z+00.0000 D+00.0000    T21  Z+00.0000 D+00.0000    T41  Z+00.0000 D+00.0000
T02  Z+00.0000 D+00.0000    T22  Z+00.0000 D+00.0000    T42  Z+00.0000 D+00.0000
T03  Z+00.0000 D+00.0000    T23  Z+00.0000 D+00.0000    T43  Z+00.0000 D+00.0000
T04  Z+00.0000 D+00.0000    T24  Z+00.0000 D+00.0000    T44  Z+00.0000 D+00.0000
T05  Z+00.0000 D+00.0000    T25  Z+00.0000 D+00.0000    T45  Z+00.0000 D+00.0000
T06  Z+00.0000 D+00.0000    T26  Z+00.0000 D+00.0000    T46  Z+00.0000 D+00.0000
T07  Z+00.0000 D+00.0000    T27  Z+00.0000 D+00.0000    T47  Z+00.0000 D+00.0000
T08  Z+00.0000 D+00.0000    T28  Z+00.0000 D+00.0000    T48  Z+00.0000 D+00.0000
T09  Z+00.0000 D+00.0000    T29  Z+00.0000 D+00.0000    T49  Z+00.0000 D+00.0000
T10  Z+00.0000 D+00.0000    T30  Z+00.0000 D+00.0000    T50  Z+00.0000 D+00.0000
T11  Z+00.0000 D+00.0000    T31  Z+00.0000 D+00.0000    T51  Z+00.0000 D+00.0000
T12  Z+00.0000 D+00.0000    T32  Z+00.0000 D+00.0000    T52  Z+00.0000 D+00.0000
T13  Z+00.0000 D+00.0000    T33  Z+00.0000 D+00.0000    T53  Z+00.0000 D+00.0000
T14  Z+00.0000 D+00.0000    T34  Z+00.0000 D+00.0000    T54  Z+00.0000 D+00.0000
T15  Z+00.0000 D+00.0000    T35  Z+00.0000 D+00.0000    T55  Z+00.0000 D+00.0000
T16  Z+00.0000 D+00.0000    T36  Z+00.0000 D+00.0000    T56  Z+00.0000 D+00.0000
T17  Z+00.0000 D+00.0000    T37  Z+00.0000 D+00.0000    T57  Z+00.0000 D+00.0000
T18  Z+00.0000 D+00.0000    T38  Z+00.0000 D+00.0000    T58  Z+00.0000 D+00.0000
T19  Z+00.0000 D+00.0000    T39  Z+00.0000 D+00.0000    T59  Z+00.0000 D+00.0000
T20  Z+00.0000 D+00.0000    T40  Z+00.0000 D+00.0000    T60  Z+00.0000 D+00.0000
```



```
TOOL LENGTH                        MEDIUM JOG RATE     POS: MACH+PZS+FIX+TL      H-1
     X+000.0865 Y+000.0000 Z-001.3500 A+000.000 B+000.0000

H001 Z+00.0000   H021 Z+00.0000   H041 Z+00.0000   H061 Z+00.0000   H081 Z+00.0000
H002 Z+00.0000   H022 Z+00.0000   H042 Z+00.0000   H062 Z+00.0000   H082 Z+00.0000
H003 Z+00.0000   H023 Z+00.0000   H043 Z+00.0000   H063 Z+00.0000   H083 Z+00.0000
H004 Z+00.0000   H024 Z+00.0000   H044 Z+00.0000   H064 Z+00.0000   H084 Z+00.0000
H005 Z+00.0000   H025 Z+00.0000   H045 Z+00.0000   H065 Z+00.0000   H085 Z+00.0000
H006 Z+00.0000   H026 Z+00.0000   H046 Z+00.0000   H066 Z+00.0000   H086 Z+00.0000
H007 Z+00.0000   H027 Z+00.0000   H047 Z+00.0000   H067 Z+00.0000   H087 Z+00.0000
H008 Z+00.0000   H028 Z+00.0000   H048 Z+00.0000   H068 Z+00.0000   H088 Z+00.0000
H009 Z+00.0000   H029 Z+00.0000   H049 Z+00.0000   H069 Z+00.0000   H089 Z+00.0000
H010 Z+00.0000   H030 Z+00.0000   H050 Z+00.0000   H070 Z+00.0000   H090 Z+00.0000
H011 Z+00.0000   H031 Z+00.0000   H051 Z+00.0000   H071 Z+00.0000   H091 Z+00.0000
H012 Z+00.0000   H032 Z+00.0000   H052 Z+00.0000   H072 Z+00.0000   H092 Z+00.0000
H013 Z+00.0000   H033 Z+00.0000   H053 Z+00.0000   H073 Z+00.0000   H093 Z+00.0000
H014 Z+00.0000   H034 Z+00.0000   H054 Z+00.0000   H074 Z+00.0000   H094 Z+00.0000
H015 Z+00.0000   H035 Z+00.0000   H055 Z+00.0000   H075 Z+00.0000   H095 Z+00.0000
H016 Z+00.0000   H036 Z+00.0000   H056 Z+00.0000   H076 Z+00.0000   H096 Z+00.0000
H017 Z+00.0000   H037 Z+00.0000   H057 Z+00.0000   H077 Z+00.0000   H097 Z+00.0000
H018 Z+00.0000   H038 Z+00.0000   H058 Z+00.0000   H078 Z+00.0000   H098 Z+00.0000
H019 Z+00.0000   H039 Z+00.0000   H059 Z+00.0000   H079 Z+00.0000   H099 Z+00.0000
H020 Z+00.0000   H040 Z+00.0000   H060 Z+00.0000   H080 Z+00.0000   H100 Z+00.0000
```

BASIC DATA ENTRY

The key sequences below describe data entry operations that are commonly performed in the tool tables.

Selecting an individual tool table - Press MODE SELECT 6 to select the Tool Tables mode's main menu which displays a list of the available tables. Next press the letter (T, H, or D) that identifies the table that you wish to use.

Highlighting an entry - Prior to making a change in a tool table, you must first highlight the individual line or field that you wish to change. There are three ways to highlight an entry:

- type the letter of the table that you are using followed by the number of the line that you wish to edit. Example: pressing T13 ENTER will highlight the 13th entry in the T table; H54 ENTER will highlight the 54th entry in the H Table.
- press the NEXT EVENT key to select the next field number; - NEXT EVENT to select the previous field number.
- use the LEFT and RIGHT ARROW keys to scroll from page to page.

Entering an offset - After highlighting the field that you wish to edit, you can directly enter a new value by typing the letter that identifies the item that you wish to change followed by the value that you wish to store. Example: D.3875 ENTER will store a tool diameter of .3875" in the T or D table.

Incrementally adjusting an offset - Enter a value with the incremental slash to add your entry to an existing offset. Example: if D.5000 is the stored value in the T or D table, the sequence D-.001/ ENTER will change the stored value to D.499. This is useful when you wish to make a small change to adjust for tool wear.

Clearing a single offset - You can clear (set to 0.000) any highlighted entry by typing its letter identifier then pressing the CLEAR key. Example: typing D CLEAR will set the highlighted tool diameter to 0.000 in the D or T table.

Clearing an entire tool table - You can clear all existing values in any tool table by completing the following sequence. The letter X in this sequence identifies the table that you wish to clear. Example: the sequence T-1 ENTER T CLEAR will erase the entire T table

- Press X-1 ENTER. The X-1 display will be highlighted to identify that all compensations will be removed on the next entry.
- Press X CLEAR. All values will be set to zero. If you wish to abort this sequence, select another tool number prior to pressing the CLEAR key.

Loading Tool Table data with G10 part program commands - Tool offset data can be stored in a part program with a series of EIA/ISO G10 blocks. Execution of a G10 block will cause the programmed value to be stored in the specified tool table. Refer to Part 7 of this manual for further information about the G10 command.

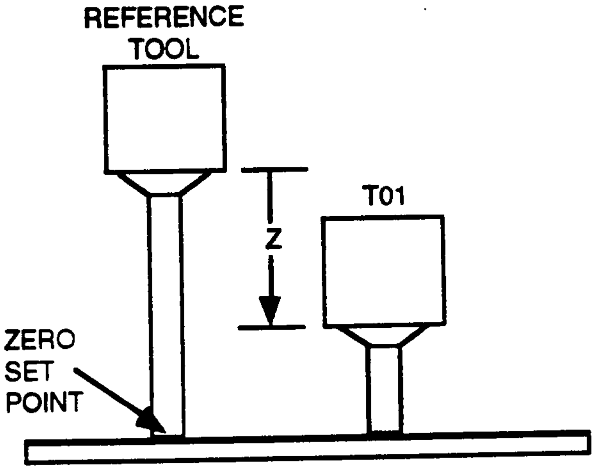
Storing Tool Table data on an off-line device - All dimensions and values stored in the Tool Tables mode may be transmitted to and from an external storage device through the use of the control's the Transmit/Receive mode. You may also store all Tool Table values on a 3.5" DOS diskette if your control has been supplied with a floppy disk drive.

Z - TOOL LENGTH OFFSET

Tool length offsets may be entered directly from the keyboard, or they may be automatically calibrated by touching each tool to a known surface. Using the Tool Calibration feature requires that you implement a reference tool as described below.

REFERENCE TOOL

A Reference tool is a special tool that provides a standard length for measuring all tool lengths. Since accuracy is very important in setting tool lengths, the reference tool is usually an edge-finding tool that is not used for machining. The Reference tool will also be used whenever you Zero Set the Z axis, or when you locate fixture offset coordinates.



Since the Reference tool serves as the standard for all other tool lengths, its length offset must be set to Z0.000 in the Tool Tables. For optimal safety and accuracy, the reference tool should not be used for machining, and should be the longest tool in the tool magazine.

AUTOMATIC TOOL LENGTH CALIBRATION

To calibrate your tools with the Tool Calibration utility, you must first set up a uniform surface or dial indicator to accurately gauge the tool tip position in Z.

1) Load the reference tool into the spindle. Make sure that no tool length offset is active; reference the machine if necessary.

2) Jog in Z to touch off on the surface or establish a repeatable setting on the indicator. ZERO SET the Z axis as described in the Jog mode section.

3) Jog away from Z0 and load the first tool to be calibrated.

4) Move to the T or H Table. Enter a number to identify the tool that is in the spindle. Its number will be highlighted on the monitor.

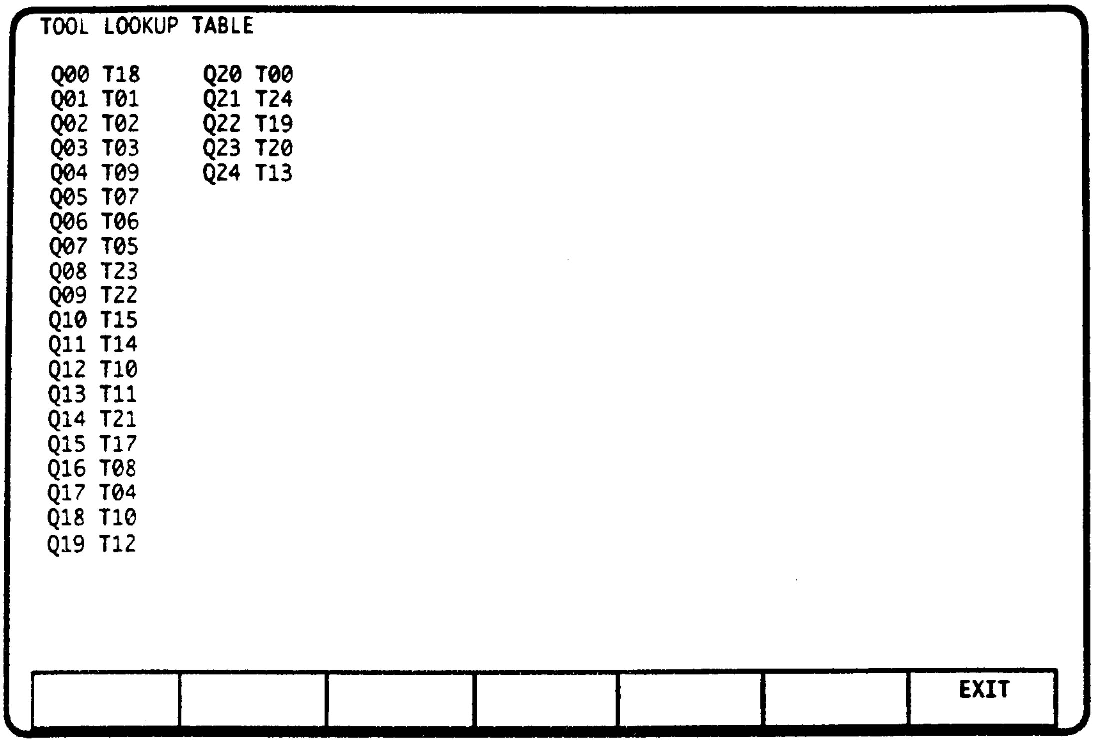
5) Jog the tool to the same surface or indicator setting that was used for the reference tool. You may jog the axes while the Tool Table is selected. The JOG switches are enabled at the rate or increment that was last selected in the Jog mode.

6) Press the TOOL CALIBRATE soft key. This action will store the tool length offset into the selected tool's Z dimension. The value that will be stored is the difference between the length of the Reference tool and the tool that is in the spindle. Tools that are shorter than the Reference tool will have a negative Z offset.

7) Jog away and load the next tool to be calibrated. Repeat this process until all tool length offsets have been stored.

Q - SWAPPING TOOL LOOKUP (RANDOM) TABLE

A swapping tool lookup table is required when an automatic, random tool changer is installed on your machine. A random tool changer is one that swaps the next tool that is called by your part program with the tool that is currently in the spindle. This special tool table is required for this type of a tool changer in order to keep track of how the tools are arranged in the tool carousel. This table is referred to as a "swapping tool lookup table" because it automatically updates itself after each successful tool change. The number of positions in this table is limited by the number of pockets in your tool carousel. A Q table for a 24 position tool changer is shown on the screen below.



```
TOOL LOOKUP TABLE

 Q00 T18      Q20 T00
 Q01 T01      Q21 T24
 Q02 T02      Q22 T19
 Q03 T03      Q23 T20
 Q04 T09      Q24 T13
 Q05 T07
 Q06 T06
 Q07 T05
 Q08 T23
 Q09 T22
 Q10 T15
 Q11 T14
 Q12 T10
 Q13 T11
 Q14 T21
 Q15 T17
 Q16 T08
 Q17 T04
 Q18 T10
 Q19 T12
```

The table below shows how a Swapping Q Table will function during several tool changes. This example assumes that your machine has a 6-pocket tool carousel and that your machine builder has enabled H and D Table to store tool lengths and diameters. The carousel pocket Q00 is used to identify the tool that is currently in the spindle. Any pocket that is noted as containing T00 is considered empty by the control.

	Q00	Q01	Q02	Q03	Q04	Q05	Q06
Initial Setup	T00	T01	T02	T03	T04	T05	T06
(9) T04 H04 D04	**T04**	T01	T02	T03	**T00**	T05	T06
(9) T06 H06 D06	**T06**	T01	T02	T03	T00	T05	**T04**
(9) T03 H03 D03	**T03**	T01	T02	**T06**	T00	T05	T04
(9) T04 H04 D04	**T04**	T01	T02	T06	T00	T05	**T03**

ENTERING TOOL NUMBERS

To assign a particular tool to a pocket, you must first select the pocket's Q number. The selected Q number is highlighted on the monitor. To select a Q number, key in Qxx ENTER, where xx is the Q number whose tool number you need to change. Press the NEXT EVENT key to select the next Q number; - NEXT EVENT to select the previous Q number.

Q - ENTER TOOL ASSIGNMENTS

A Q field represents a pocket on the conveyor. Once you select a Q number, enter the number of the tool that resides in that pocket by keying in Txx ENTER, where xx represents the tool number. A T00 entry indicates that the pocket is empty. Q00 is the spindle. The T number assigned to Q00 refers to the tool that is currently in the spindle.

HOW TO ERASE THE Q TABLE

You can set all tool numbers in the Random Tool table to T00 by completing the following sequence.

1) Press Q-1 ENTER The Q-1 display will be highlighted.
2) Press Q CLEAR. All values will be sequentially numbered. If you wish to abort this sequence, select another Q number prior to pressing the CLEAR key.

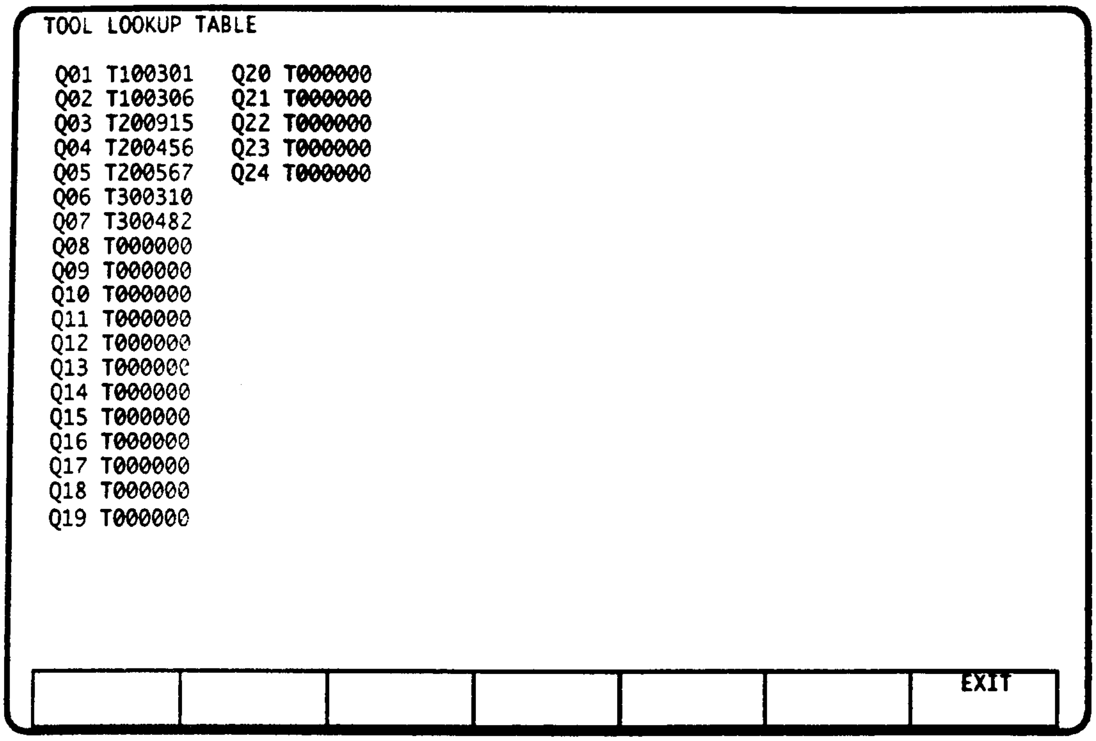
HOW TO SEQUENTIALLY NUMBER THE RANDOM TOOL TABLE

When initially setting up the Tool Lookup Table, you can sequentially number the Random Tool table by completing the following sequence. Sequentially numbering the table will store T01 in Q01, T02 in Q02 etc.

1) Press Q-2 ENTER The Q-2 display will be highlighted.
2) Press Q CLEAR. All values will be sequentially numbered. If you wish to abort this sequence prior to this step, select another Q number.

Q - NON-SWAPPING TOOL LOOKUP TABLE

A non-swapping tool lookup table, **below**, is used only when the tool number that you program must conform to a tool inventory system that has been implemented in your shop. The basic function of this table is to convert your programmed 2 through 6 digit T code into a number that relates to a pocket number on the machine's tool changer.



```
TOOL LOOKUP TABLE

 Q01 T100301   Q20 T000000
 Q02 T100306   Q21 T000000
 Q03 T200915   Q22 T000000
 Q04 T200456   Q23 T000000
 Q05 T200567   Q24 T000000
 Q06 T300310
 Q07 T300482
 Q08 T000000
 Q09 T000000
 Q10 T000000
 Q11 T000000
 Q12 T000000
 Q13 T000000
 Q14 T000000
 Q15 T000000
 Q16 T000000
 Q17 T000000
 Q18 T000000
 Q19 T000000
```

Example

The following example demonstrates how a non-swapping tool lookup table will function. This example uses the sample Q table shown **above**.

N10 (9) T100301 H01 D01	Put tool in pocket #1 in spindle, activate H01 and D01.
N20 (9) T200456 H04 D04	Put tool in pocket #4 in spindle, activate H04 and D04.
N30 (9) T200915 H03 D03	Put tool in pocket #3 in spindle, activate H03 and D03.
N40 (9) T300482 H07 D07	Put tool in pocket #7 in spindle, activate H07 and D07.

HOW TO ERASE THE Q TABLE

You can set all tool numbers in the Random Tool table to T00 by completing the following sequence.

1) Press Q-1 ENTER The Q-1 display will be highlighted.
2) Press Q CLEAR. All values will be sequentially numbered. If you wish to abort this sequence, select another Q number prior to pressing the CLEAR key.

HOW TO SEQUENTIALLY NUMBER THE RANDOM TOOL TABLE

When initially setting up the Tool Lookup Table, you can sequentially number the Random Tool table by completing the following sequence. Sequentially numbering the table will store T01 in Q01, T02 in Q02 etc.

1) Press Q-2 ENTER The Q-2 display will be highlighted.
2) Press Q CLEAR. All values will be sequentially numbered. If you wish to abort this sequence prior to this step, select another Q number.

TOOL LIFE MANAGEMENT - INTRODUCTION

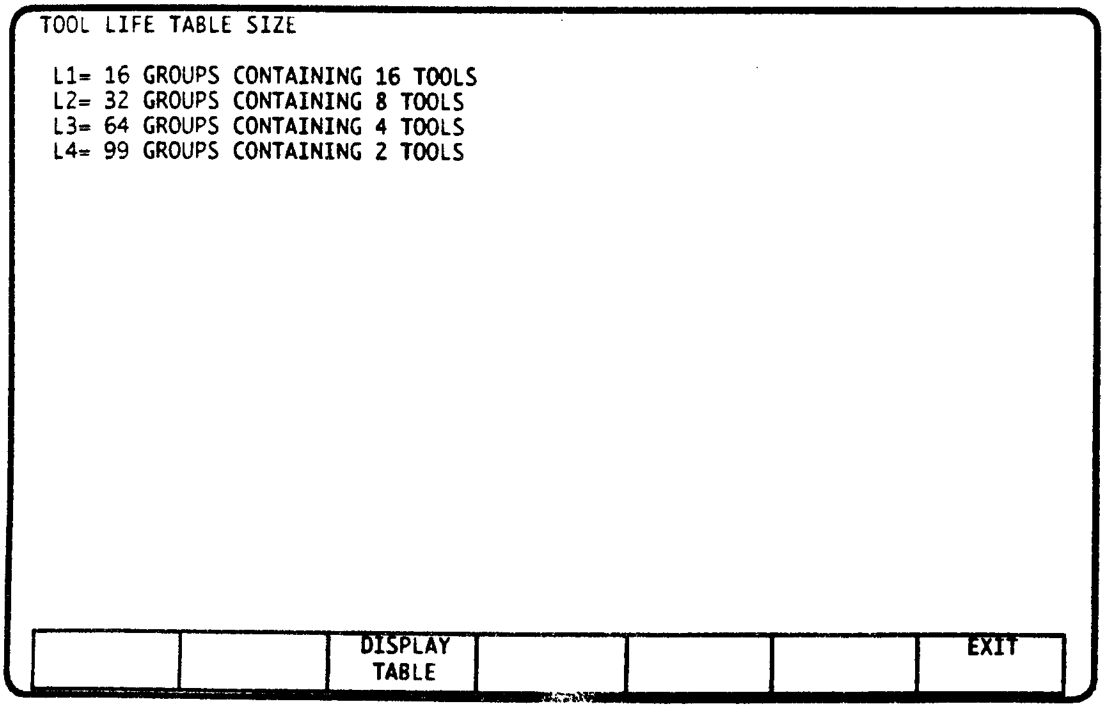
DynaPath Delta series controls offer an optional Tool Life Management feature which tracks tool usage in minutes of feed or number of holes drilled. Initially setting up the Tool Life Management features requires you to define groups of tools. All tools within a given group must be able to serve an identical machining function (i.e., all .500" end mills will be placed in the same group; all .3875" drills will be placed in the same group). Execution of a part program T code, when Tool Life management is active, will command the control to search through the programmed group and select the first tool that has a remaining life.

Other capabilities of the Tool Life Management feature include the ability to select the tool with the most remaining life, and the ability to inform the operator when the active tool's estimated life has expired. The expected cutting life of each tool in a group may be individually specified.

SETTING UP THE TOOL LIFE MANAGEMENT TABLES

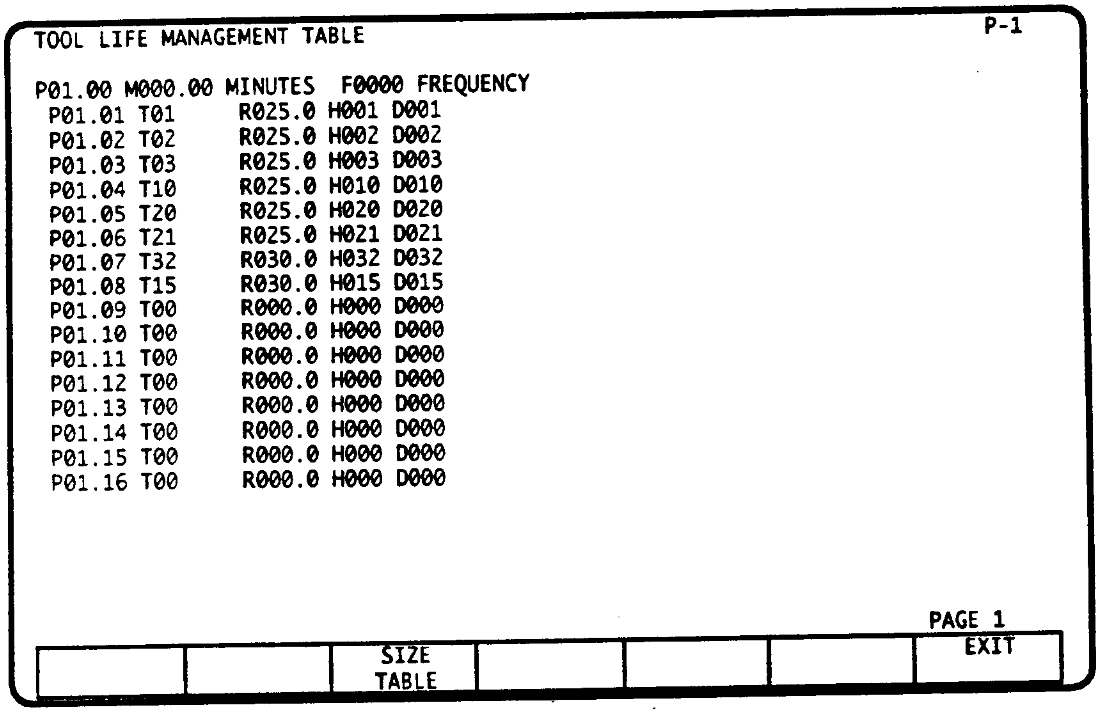
The first step in setting up the Tool Life Management tables is deciding how many tool groups you require. You may choose one of the four groupings shown on the screen below. Each tool group should contain tools which have an identical machining task. For example, .500" end mills should all be in the same group; .375" drills should all be in the same group.

Carefully consider which L option best suits your machining application. Any future changes to this entry will completely erase the Tool Life Management table. In fact, to prevent accidental table erasure, your machine supplier may supply a switch or special M Code that you must use before changing the tool group size. Consult your machine supplier for further instruction.



```
TOOL LIFE TABLE SIZE

 L1= 16 GROUPS CONTAINING 16 TOOLS
 L2= 32 GROUPS CONTAINING 8 TOOLS
 L3= 64 GROUPS CONTAINING 4 TOOLS
 L4= 99 GROUPS CONTAINING 2 TOOLS
```

The screen below shows the Tool Life Management table when the L1 grouping is selected. The H and D codes will not appear on this screen if your control does not use H and D codes to activate tool offsets (refer to Part 1 - Section 1).



```
TOOL LIFE MANAGEMENT TABLE                                              P-1

P01.00 M000.00 MINUTES  F0000 FREQUENCY
 P01.01 T01      R025.0 H001 D001
 P01.02 T02      R025.0 H002 D002
 P01.03 T03      R025.0 H003 D003
 P01.04 T10      R025.0 H010 D010
 P01.05 T20      R025.0 H020 D020
 P01.06 T21      R025.0 H021 D021
 P01.07 T32      R030.0 H032 D032
 P01.08 T15      R030.0 H015 D015
 P01.09 T00      R000.0 H000 D000
 P01.10 T00      R000.0 H000 D000
 P01.11 T00      R000.0 H000 D000
 P01.12 T00      R000.0 H000 D000
 P01.13 T00      R000.0 H000 D000
 P01.14 T00      R000.0 H000 D000
 P01.15 T00      R000.0 H000 D000
 P01.16 T00      R000.0 H000 D000

                                                                    PAGE 1
```

INITIAL SETUP

The following procedure describes how to plan and enter all values that are required to implement Tool Life Management.

1) IDENTIFY YOUR TOOL GROUPS ON PAPER

Write down a list of your required tool groups, the tools that belong in the group, and the expected life of each tool in the group. If your control uses H and D codes, also write down the H and D offset codes that are associated with each tool. The list for this example is shown below. If H and D codes are not used, the T number identifies the Tool Table entry where each tool's Z and D offsets are stored.

GROUP 1 - .5" end mills

T01 - 25 minutes - H01 D01
T02 - 25 minutes - H02 D02
T03 - 30 minutes - H03 D03
T04 - 25 minutes - H04 D04

GROUP 2 - .375" drills

T15 - 32 holes - H15 D15
T16 - 32 holes - H16 D16
T17 - 30 holes - H17 D17
T18 - 30 holes - H18 D18

2) ENTER GROUP INFORMATION

Press the DISPLAY TABLE soft key to select the Tool Life Management table. You must first select each tool group and enter the parameters which apply to all tools in the group. To select a tool group, key in Pxx ENTER, where P is the group identifier and xx represents the group number. To inform you that the entire group is selected, the Pxx.00 row of the group will be highlighted. Any changes made to this row will affect all tools in the group. You must now decide if each group will be monitored in minutes of feed or number of holes.

If all tools in a group will be monitored in **minutes of feed**, key in the Mxxx.x, where M chooses minutes and xxx.x is the expected life in minutes for the average tool in the group.

If all tools in the selected group will be monitored in **number of holes**, key in the Fxxxx, where F chooses hole feeds and xxxx is the expected number of holes for the average tool in the group.

To specify that tools in group 1 will have an expected life of 25 minutes, key in P01 ENTER M25 ENTER. To specify that tools in group 2 will have an expected life of 32 holes, key in P02 ENTER F32 ENTER. The screen display for each group is shown below.

```
GROUP 1 - .500" end mills           GROUP 2 - .375" drills
P01.00 M025.0 MIN.  F0000 FREQ.     P02.00 M000.0 MIN.  F0032 FREQ.
P01.01 T00 R025.0 H00 D00           P02.01 T00 R0032 H00 D00
P01.02 T00 R025.0 H00 D00           P02.02 T00 R0032 H00 D00
P01.03 T00 R025.0 H00 D00           P02.03 T00 R0032 H00 D00
P01.04 T00 R025.0 H00 D00           P02.04 T00 R0032 H00 D00
```

3) ENTER TOOL NUMBERS and SPECIFIC TOOL LIFE

Next, assign a tool number to each group member. First, select the group member by typing Pxx.yy, where xx is the group number and yy is the member number. Alternately, you can press the NEXT EVENT key to select the following member, and - NEXT EVENT to select the previous member. Key in the tool number that corresponds to the selected group member by keying in Txx ENTER. This number always identifies the physical tool that will be put in the spindle. If H and D codes are not used, the T code also identifies the Tool Table entry where each tool's Z and D offsets are stored. If H and D codes are used on your control, you must also enter H and D codes to specify where the tool's length and diameter offsets are stored.

If the tool has an expected life which is different than the rest of the group, enter its expected life by pressing R then the expected life. The units of R are in the same units (minutes or number of holes) that you selected for the entire group. After completing this step, the Tool Life Management table will contain the information shown below.

```
GROUP 1 - .5" end mills             GROUP 2 - .375" drills
P01.00 M025.0 MIN.  F0000 FREQ.     P02.00 M000.0 MIN.  F0032 FREQ.
P01.01 T00 R025.0 H01 D01           P02.01 T00 R0032 H15 D15
P01.02 T00 R025.0 H02 D02           P02.02 T00 R0032 H16 D16
P01.03 T00 R030.0 H03 D03           P02.03 T00 R0030 H17 D17
P01.04 T00 R025.0 H04 D04           P02.04 T00 R0030 H18 D18
```

USING TOOL LIFE MANAGEMENT

Your machine supplier will specify whether you turn Tool Life Management on and off with an M Code, mechanical switch, or pushbutton. This topic describes how the control will operate when Tool Life Management is active. In designing the machine tool interface for your system, your machine supplier may design the operation of this feature different than described here. You should consult your machine supplier for specific details on using this feature. General operations are described here.

EXECUTING T CODES

When a T Code executes in an M Function event or an EIA/ISO block, the T Code specifies the group where the tool will be found. T04, for example, indicates the tool is in group 4. The first tool in the specified group (P04.xx) that has a life greater than zero will be selected by the control. Your machine supplier, however, may provide an M Code that forces Tool Life Management to select the tool that has the most remaining life.

Since H and D codes for each tool are specified in the L table, any H and D code that executes while Tool Life Management is active will be ignored by the control. H and D codes are shown in the Program mode to allow you to develop programs that will not use Tool Life Management.

Once the tool is active, the control reduces its remaining life (R value) during all machining operations. If the group is timed in minutes, the tool's life will be reduced as the control executes a feed operation. Feed operations include Linear and Arc events and the feed portions of Circle Mill, Frame Mill and Cavity Mill Events. If the group life is specified in frequency of use, the R value will decrement each time a fixed or "quill" cycle executes.

If the life for all tools in a given group has expired, the control will issue a fault when a following tool change requests a tool from that group.

ALL AXIS INHIBIT

The remaining tool life (R value) will not be reduced if the control is operating in Test Mode 3 (TM3 from the Set Up mode).

SWAPPING TOOL LOOKUP TABLES

This topic is for operators that use Tool Life Management on machines that are equipped with an automatic random tool changer. A random tool changer always deposits the last tool in the pocket that is vacated by a new tool. A Swapping Lookup table is present whenever your machine has a random tool changer.

TLM, SWAPPING TOOL LOOKUP TABLE, WITHOUT H and D CODES

Tool changes on controls with this combination of features (refer to Part 1 - Section 2) are programmed with a 4 digit T Code which is expressed in the format Txxyy. When Tool Life Management is not active yy identifies the Tool Table offsets for the tool that is in the spindle and xx identifies the tool number for the next tool that will be used.

When Tool Life Management is active, the format is still Txxyy, but now **each pair represents a tool group** from which the control will select the tool to use. If the yy pair is any number other than 00, the control will activate the Z and D offsets for the tool which is currently in the spindle (no other offsets can be made active). If the yy pair is 00, the control will cancel the active Z and D offsets. If the xx pair is 00, and an M06 is programmed in the event (or in a following event) the tool in the spindle will be put in the empty pocket on the tool conveyor. This will leave the spindle empty.

FIXTURE OFFSET TABLE - INTRODUCTION

DynaPath series controls have several different methods for establishing part zero locations on a machine. This introduction discusses the two methods that are commonly used throughout the industry to establish a part zero: Zero Setting and using Fixture Offsets. The remainder of this section describes how to measure, store, and activate fixture offsets.

Both methods described below refer to using a reference tool to locate the position that you wish to become part zero. The reference tool is described in the Tool Tables Mode section.

1) **Jog Mode Zero Set** - Select the Jog mode then move the center of the reference tool to the point that you wish to become part zero. Next select the 9-Zero Set option and toggle the Jog switch for each axis that you wish to set.

   This method is often used in short-run jog shops because it provides a quick method of setting a zero point. However, this method is often discouraged since the zero point is lost when you turn the control off or when you reference an axes. The only way to reestablish the zero set point is to manually set the point with the reference tool. The possible error in multiple zero setting operations during a long part run usually persuades machinists to use Fixture Offsets instead of Zero Set.

2) **Fixture Offsets (E01-E32)** - Use the reference tool to locate coordinates for all of the part zero points that you require. Up to 32 different points can be stored in the Fixture Offset table. Once defined, any fixture offset can be made active via execution of a part program E code. This method is usually preferred by experienced machinists since no manual operation is needed to reestablish the part zero after the control is turned off. The Fixture Calibration feature provides a quick and simple method of locating and storing fixture offset coordinates.

Note 1: Using the Fixture Offset and a Zero Set operation in the same part setup is discouraged since their effects are additive and force the machinist to establish the zero set point each time the control is turned on. Instead, we recommend that you define the initial point as the first fixture offset (E01) and begin the program with an M Function event that activates E01.

Note 2: Set Up mode Part Offsets (PX and PY) will affect both Fixture Offsets and Zero Sets. Since Part Offsets are generally considered to be less useful than Fixture Offsets, we strongly recommend that the Part Offsets always remain set at PX0.0 and PY0.0. As an exception to this rule, many programmers shift the coordinate system within a part program by programming part offsets in a Set Up event. If this method is used, you should always program the part offsets back to X0.0, Y0.0 before ending the program. This canceling ensures that the Part Offsets are not active the next time you locate Fixture Offsets or perform a Zero Set operation.

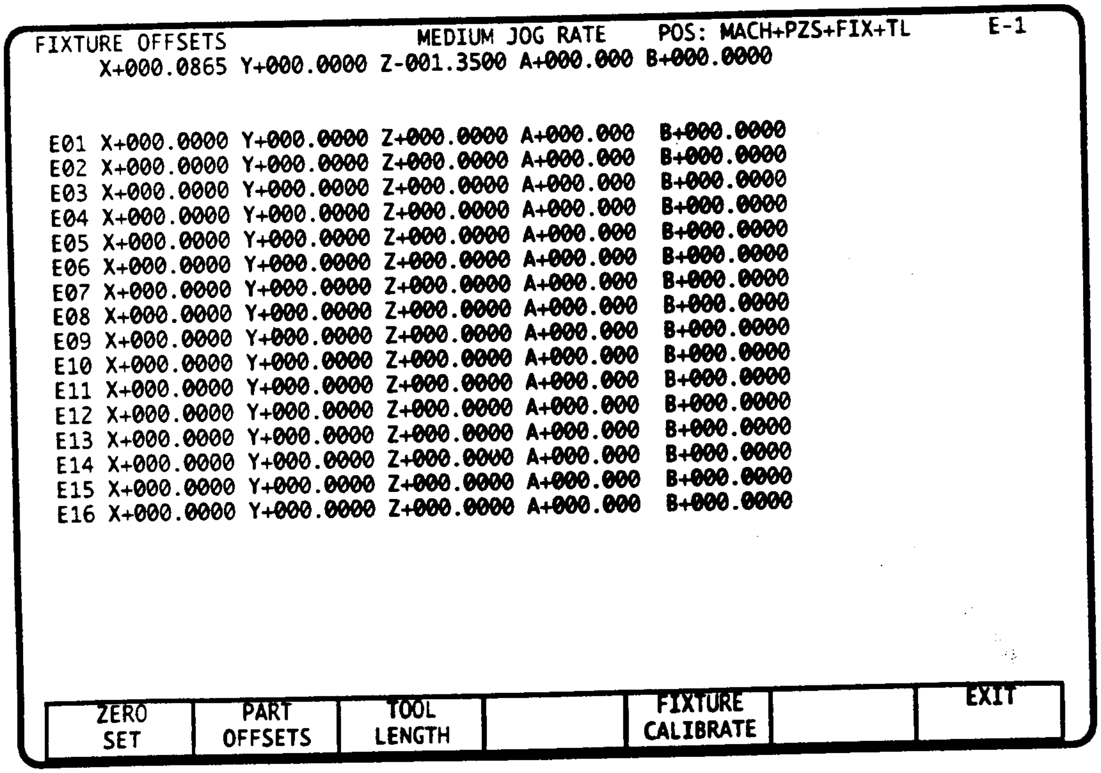
Note 3: Your machine supplier can make two settings which affect how Fixture Offsets function. The first setting determines whether the control cancels the active Fixture Offset when an M02 or M30 - End of Program command executes. The second setting determines whether Fixture Offsets will be Immediate or Deferred. Immediate and Deferred offsets are described at the end of this section.

USING FIXTURE OFFSETS

When a Fixture Offset is active, the control shifts the part datum by the distance that is stored for each axis in the Fixture Offset table. If no ZERO SET operation has been performed since the machine axes were referenced, the fixture offset is measured from Machine Coordinate Zero.

Several features, such as Automatic Fixture Calibration, and incremental adjustment to an existing entry, will simplify the process of locating and adjusting your fixture offset coordinates. The ZERO SET, PART OFFSETS and TOOL LENGTH soft keys allow you to select the appropriate coordinate system to use with the Fixture Calibration feature. We recommend that you always store fixture offsets in machine coordinates by turning off the three soft key settings. This is done by pressing each soft key until its corresponding field in the upper right corner of the screen is unhighlighted. Using the Fixture Calibrate feature when one or more of the soft key settings are highlighted is reserved for special applications.

The M Function event or EIA/ISO block that activates a Fixture Offset (by executing an E Code) does not cause tool motion. However, the new position will be adjusted by the fixture offset dimensions upon execution of the motion command that follows the activation of the E code. The tool motion that occurs depends on your machine supplier's setting of Immediate or Deferred Offset activation.

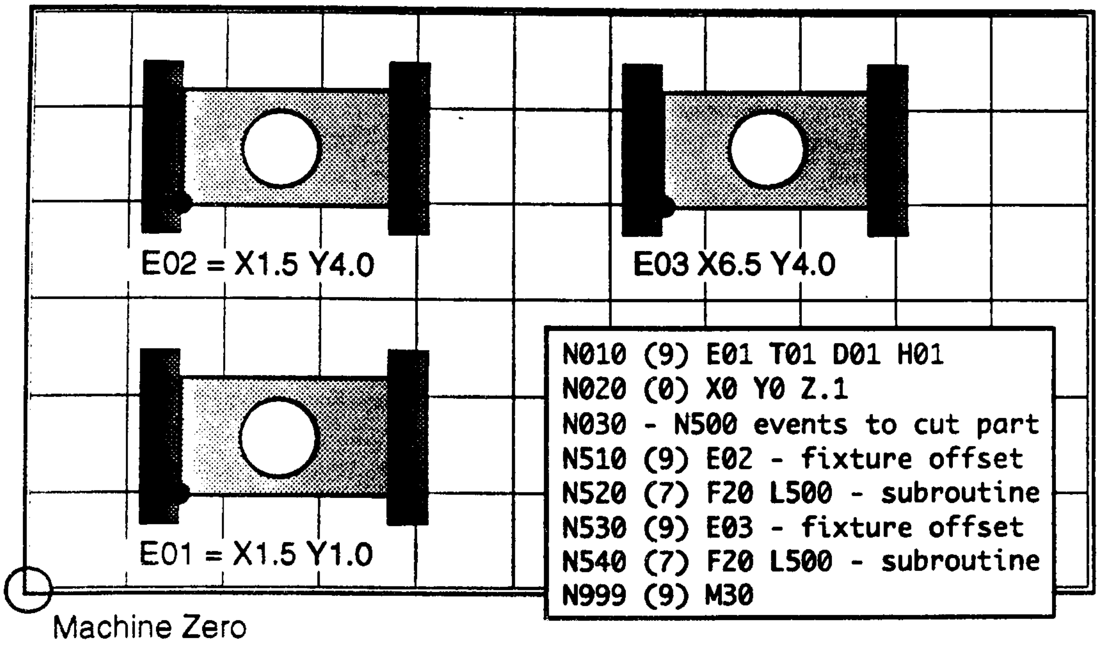


```
FIXTURE OFFSETS                  MEDIUM JOG RATE    POS: MACH+PZS+FIX+TL        E-1
     X+000.0865 Y+000.0000 Z-001.3500 A+000.000 B+000.0000


 E01 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E02 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E03 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E04 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E05 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E06 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E07 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E08 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E09 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E10 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E11 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E12 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E13 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E14 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E15 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
 E16 X+000.0000 Y+000.0000 Z+000.0000 A+000.000  B+000.0000
```

EXAMPLE USING FIXTURE OFFSETS

This example shows how fixture offsets can be used to simplify programming effort and reduce setup time. In order to simplify this example, only X and Y dimensions are shown here - a more realistic setup would include a fixture offset dimension for the Z axis.

This example shows how to create a part program that machines the same part at three different fixtures on a machine. This is accomplished by storing the machine coordinates of the lower left corner of each vice in the first three fixture offset entries (E01, E02, E03). Once you have programmed the first part at fixture offset E01 (at N numbers N30-N500), the remainder of the program simply consists of commands to activate a new fixture (E02, E03), then recall events N30 through N500 with a Subroutine event (event type 7). As many as 32 parts can be programmed in this manner by alternating M Function and Subroutine events.



```
N010 (9) E01 T01 D01 H01
N020 (0) X0 Y0 Z.1
N030 - N500 events to cut part
N510 (9) E02 - fixture offset
N520 (7) F20 L500 - subroutine
N530 (9) E03 - fixture offset
N540 (7) F20 L500 - subroutine
N999 (9) M30
```

LOCATING and STORING FIXTURE OFFSETS

In general, there are two techniques for storing and modifying fixture offset coordinates that you should be familiar with.

1) Using the ENTER key - follow your entry with the ENTER key when:

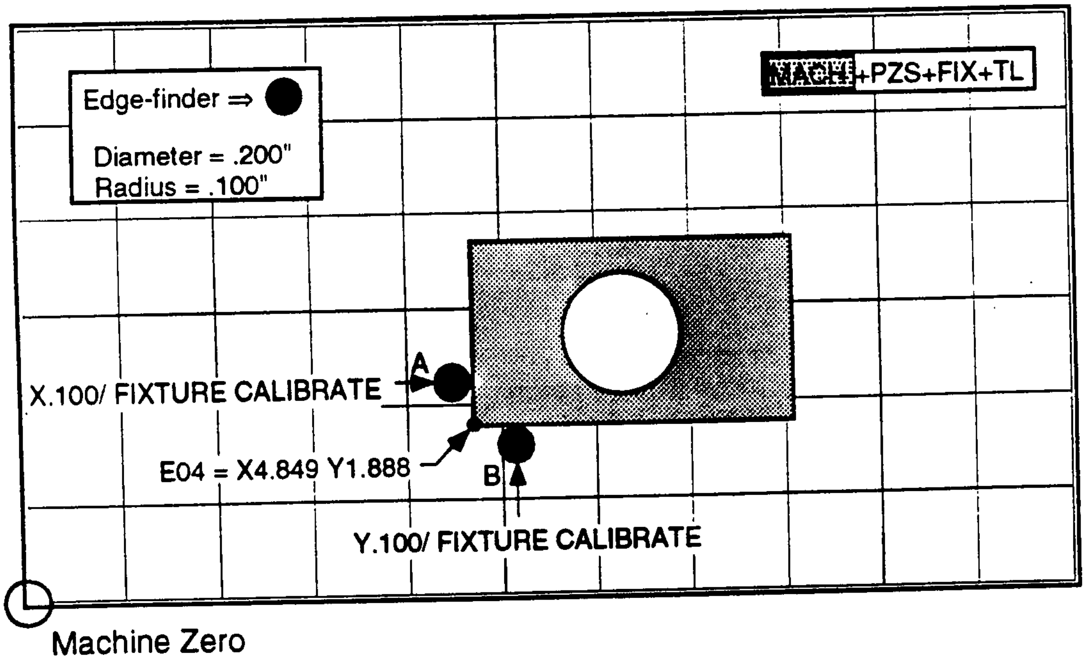
- The coordinates for your required fixture offsets are provided with a part's setup documentation. In this case, select the fixture number then enter the values that you wish to store. Example: to store dimensions for E03 in the above illustration, you would type: E03 ENTER X6.5 ENTER Y4.0 ENTER.

- You wish to adjust the existing fixture offset coordinates by a small number. In this case you must type an incremental dimension before you press the ENTER key. For example, if you wish to change the X6.5 value above to X6.505, you would type X.005/ ENTER. The .005 value will be added to the existing X value in E03.

2) Using the FIXTURE CALIBRATE key - follow your entry with this soft key when:

- **You wish to store the current reference tool position as the zero point for the selected fixture offset.** This is accomplished by specifying the axis letter that you wish to store then pressing the FIXTURE CALIBRATE soft key. For example, the key sequence to store values for all axes is: X FIXTURE CALIBRATE Y FIXTURE CALIBRATE Z FIXTURE CALIBRATE.

- **You wish to use an edge-finder to locate and store your fixture offset values.** This is typically considered the most useful calibration tool because the machinist is not required to move the tool center to the point that is to be stored in the fixture offset table. Instead, the machinists can jog an edge-finder (or a tool with a known diameter) to the surface of a fixture (as shown below) and then perform the calibration by entering the actual position of the fixture relative to the tool center. In the example below, the machinist wants the lower left corner of the part to be part zero. After jogging a .200" diameter edge-finder to point A, the machinist notes that the disired part zero is one-half of a diameter to the positive side of the tool. The key sequence X.100/ FIXTURE CALIBRATE will now store the X coordinate of the fixture's lower-left corner. After positioning the tool to position B, the sequence Y.100/ FIXTURE CALIBRATE will store the Y coordinate of the fixture's lower-left corner.



ZERO SET, PART OFFSETS and TOOL LENGTH SOFT KEYS

Most users of Fixture Calibration should always leave these three soft key in the "off" position. You can determine that they are off, because their respective PZS, FIX, and TL fields will be unhighlighted in the upper-right corner of the screen. With all of the soft keys unhighlighted, the values stored as the result of a Fixture Calibration will always be in machine coordinates.

The only time that you would use Fixture Calibration with one of the PZS, FIX, and TL fields highlighted is when you are performing a calibration while a Jog Mode Zero Set, or a Set Up mode Part Offset is active.

IMMEDIATE or DEFERRED FIXTURE OFFSETS

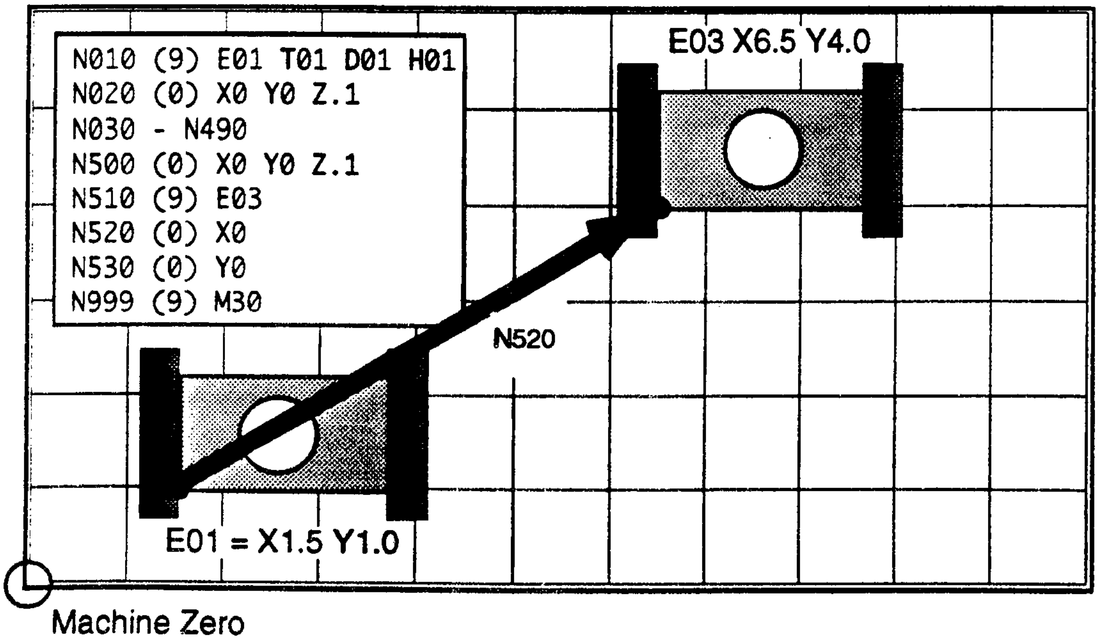
Your machine supplier has selected your control to handle Fixture Offsets as immediate or deferred. The two options are described below. Since this setting affects tool motion on your machine, it is very important that you are aware of which setting your control uses (refer to Part 1 - Section 2). This setting also affects the interchangeability of a part program between two DynaPath controls (i.e., the tool motion generated by a part program will be different on two controls with different Immediate/Deferred offset settings).

The best way to eliminate any possible problems is to make a part programming rule that requires you to always program an absolute XYZ[1] position after an E code activates a new fixture offset. This method of programming ensures that the control will command identical tool motion regardless of the Immediate/Deferred setting.

IMMEDIATE OFFSET ACTIVATION

With Immediate Offset activation, the motion command following an E Code will cause all axes to move to a point relative to the new Fixture Offset. The example below illustrates Immediate Offset activation. The tool starts at X0, Y0 relative to the datum. Next N510 activates E03 whose coordinates are stored in the Fixture Offset table as X6.5 Y4.0. No tool motion occurs on N510. Since Immediate Offset activation is being used, the following motion command (N520) causes both axes to move to X0, Y0 relative to E03. Notice that the nonprogrammed axis (the Y axis) moves to its last commanded position, but relative to the new offset.

IMMEDIATE FIXTURE OFFSET ACTIVATION



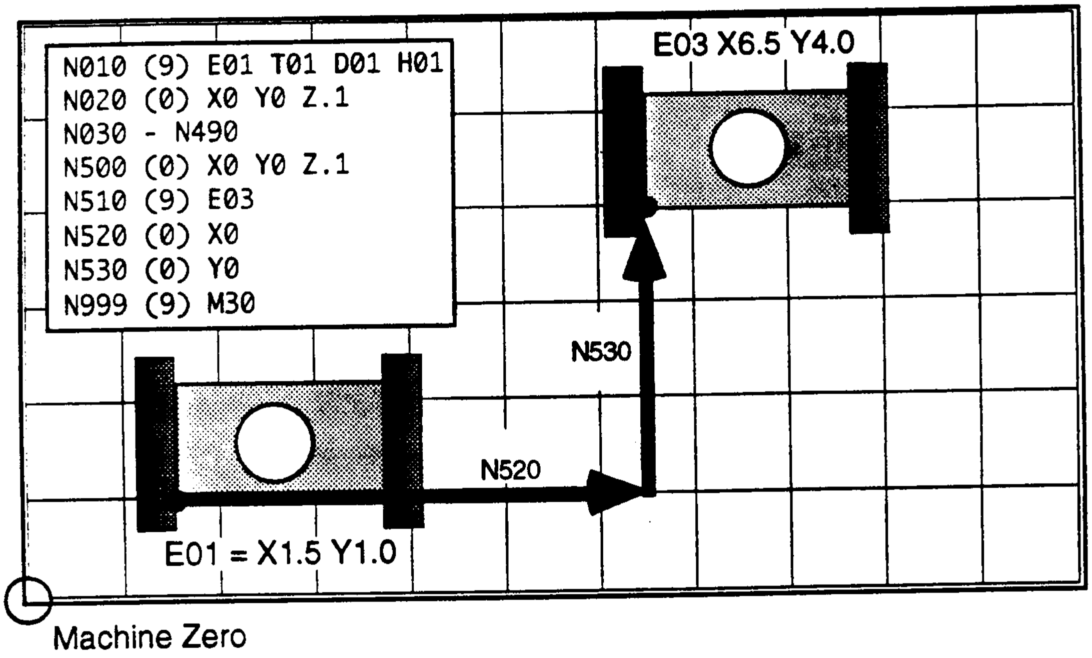
```
N010 (9) E01 T01 D01 H01
N020 (0) X0 Y0 Z.1
N030 - N490
N500 (0) X0 Y0 Z.1
N510 (9) E03
N520 (0) X0
N530 (0) Y0
N999 (9) M30
```

[1] Also program an absolute position for all auxiliary axes that are installed on your machine.

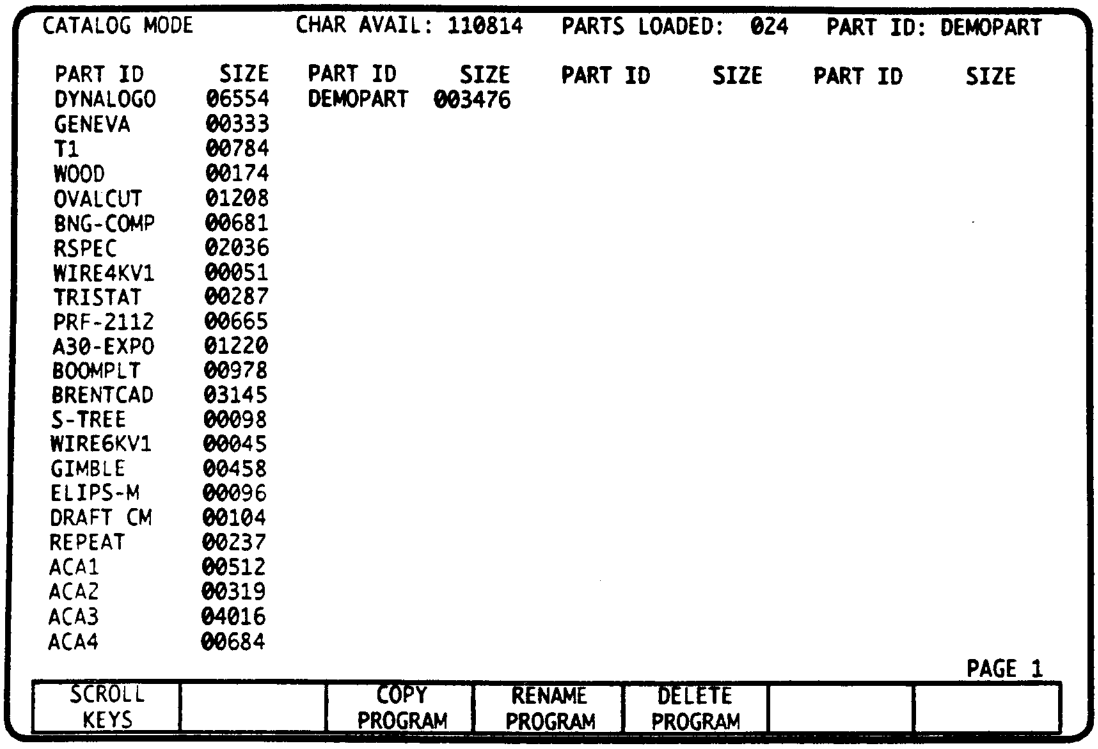
DEFERRED FIXTURE OFFSETS

With Deferred Offset activation, the motion command following an E Code will cause only programmed axes to move to a point relative to the new Fixture Offset. Any nonprogrammed axis will remain at its current position until an events that programs the axis executes. The example below illustrates Deferred Offset activation. The tool starts at X0, Y0 relative to part zero. Next N510 activates E03 whose coordinates are stored in the Fixture Offset table as X6.5 Y4.0. No tool motion occurs. Since Deferred Offset activation is being used, the following motion command (N520) causes only the X axis to move to X0 relative to E03. The Y axis does not move until N530 commands the tool to move to a point relative to E03.

DEFERRED FIXTURE OFFSET ACTIVATION



```
N010 (9) E01 T01 D01 H01
N020 (0) X0 Y0 Z.1
N030 - N490
N500 (0) X0 Y0 Z.1
N510 (9) E03
N520 (0) X0
N530 (0) Y0
N999 (9) M30
```



```
CATALOG MODE        CHAR AVAIL: 110814   PARTS LOADED:  024   PART ID: DEMOPART

 PART ID      SIZE  PART ID     SIZE    PART ID      SIZE    PART ID      SIZE
 DYNALOGO    06554  DEMOPART  003476
 GENEVA      00333
 T1          00784
 WOOD        00174
 OVALCUT     01208
 BNG-COMP    00681
 RSPEC       02036
 WIRE4KV1    00051
 TRISTAT     00287
 PRF-2112    00665
 A30-EXPO    01220
 BOOMPLT     00978
 BRENTCAD    03145
 S-TREE      00098
 WIRE6KV1    00045
 GIMBLE      00458
 ELIPS-M     00096
 DRAFT CM    00104
 REPEAT      00237
 ACA1        00512
 ACA2        00319
 ACA3        04016
 ACA4        00684
                                                                        PAGE 1
```

CATALOG MODE - INTRODUCTION

The Catalog mode may be accessed as a primary or secondary mode. A Part ID (title) must be initially entered in this mode to select a part program for operations in the Auto, Single Event, Graphics, Program, Edit, Transmit, Learn, and Delete Program modes. A Part ID need not be selected for operations in Jog, Set Up, or Tool Tables modes.

Part IDs and programs are retained in SRAM memory. This memory is powered by a lithium battery when the control is turned off. Catalog mode utilities allow you to duplicate, rename or delete existing part programs.

Also use this mode to activate the Buffered Input utility that lets you simultaneously transmit and execute a large part program that resides on the control's floppy disk drive or any storage device (such as a personal computer) that can communicate with the control's serial port.

The optional Digitize feature is also activated from this mode.

DISPLAY SCREEN INFORMATION

The following table shows the amount of part program storage and Catalog titles which are available on each CNC model.

CNC Model	Standard Part Program/ Macro/CPC Storage	Optional Part Program/ Macro/CPC Storage	Number of Part Programs
D40	64,000 characters	192,000 characters	250
D50	192,000 characters	482,000 characters	250
D60	192,000 characters	482,000 characters	250

Your machine supplier has divided the control's available memory and part program titles between the Main and Macro[1] Catalogs. In each Catalog the amount of unused part program storage is shown in the CHAR AVAIL field.

The SIZE field indicates the size (in number of characters) of each part program in the Catalog. The PARTS LOADED field indicates the number of part programs that are stored in the Catalog.

PART ID

Each part program in the Catalog is identified with part ID (program title). A part ID may contain up to 8 characters. The character set for naming a part program includes the 26 alpha characters, the 10 digits, the decimal point, minus sign, and space. A space is generated by pressing the INCR key.

If multiple pages are required to list the part titles, use the LEFT and RIGHT ARROW keys to move from page to page.

PART ID SELECTION - KEYBOARD ENTRY

You must select a part ID to create, edit, run, view, plot, output, or delete in another mode. To do so, key in the part ID and press ENTER. If the part ID already exists in the Catalog, it will be highlighted on the video display. If the part ID does not exist in the Catalog, the control will create the part ID and partition storage for a new program. The part ID you enter remains selected until you change it. The selected part program ID is displayed at the top right corner on most of the control's display screens.

PART ID SELECTION - SCROLL KEYS

As an alternate method of selecting a part program ID, press the SCROLL KEYS soft key, then use the four SCROLL keys, below, to position the cursor bar over the desired part program. To complete the selection, press the ENTER key. The selected part program ID is displayed at the top right corner of the display screen.

SCROLL LEFT	SCROLL DOWN	SCROLL UP	SCROLL RIGHT			EXIT SCROLL

[1] All Macro and CPC programs are stored in a separate Macro Catalog. To access this catalog, select the Macro mode (MODE SELECT M) then select the Macro Catalog mode. The soft keys in the two Catalogs are functionally identical.

RENAME PROGRAM

To change the name of an existing part program, first select its ID as described above. Then type in the new name (do not press ENTER) and press the RENAME PROGRAM soft key.

COPY PROGRAM

To make a duplicate copy of an existing part program, first select its ID as described above. Then type in the desired ID for the duplicate program (do not press ENTER) and press the COPY PROGRAM soft key.

DELETE PROGRAM

To erase a part program, type in its ID (do not press ENTER) then press the DELETE PROGRAM soft key.

ERASE ALL PART PROGRAMS

To erase all part programs in the Catalog, press P-1 then the DELETE PROGRAM soft key. There is no way to recover part programs that have been erased in this manner.

SELECT BUFFERED INPUT

Buffered Input is available when Catalog mode is accessed as a primary mode. The Buffered Input feature allows the control to directly execute a part program that is stored on the control's floppy disk drive or on a personal computer that is connected to the control's COM1 serial port.

To activate Buffered Input enter RS232C as the part program ID. Once you select this ID, the following modes will not be available until a new part ID is entered:

Program mode
Edit mode
Delete Program mode
Learn mode
all concurrent modes

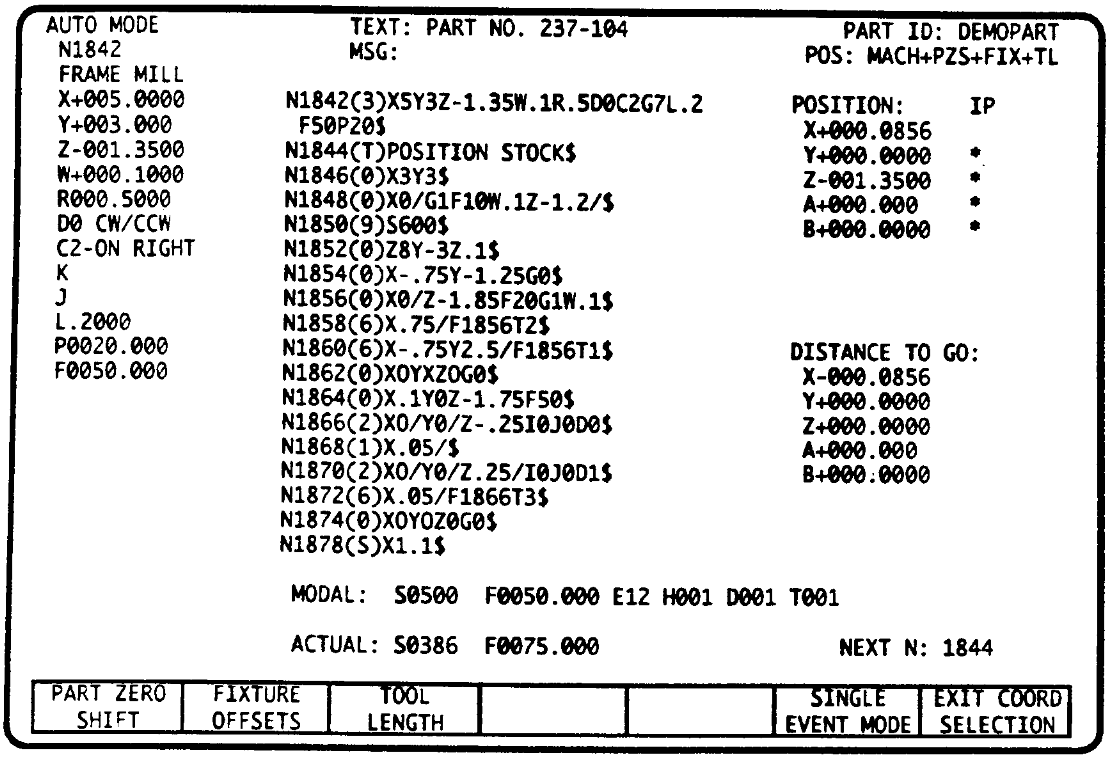
Before using Buffered Input, select the Transmit/Receive mode and choose the required communication parameters as described in the Transmit/Receive mode section.

Next select the Auto or Single Event mode to begin program execution. The Auto mode section further describes Buffered Input execution.

DIGITIZE

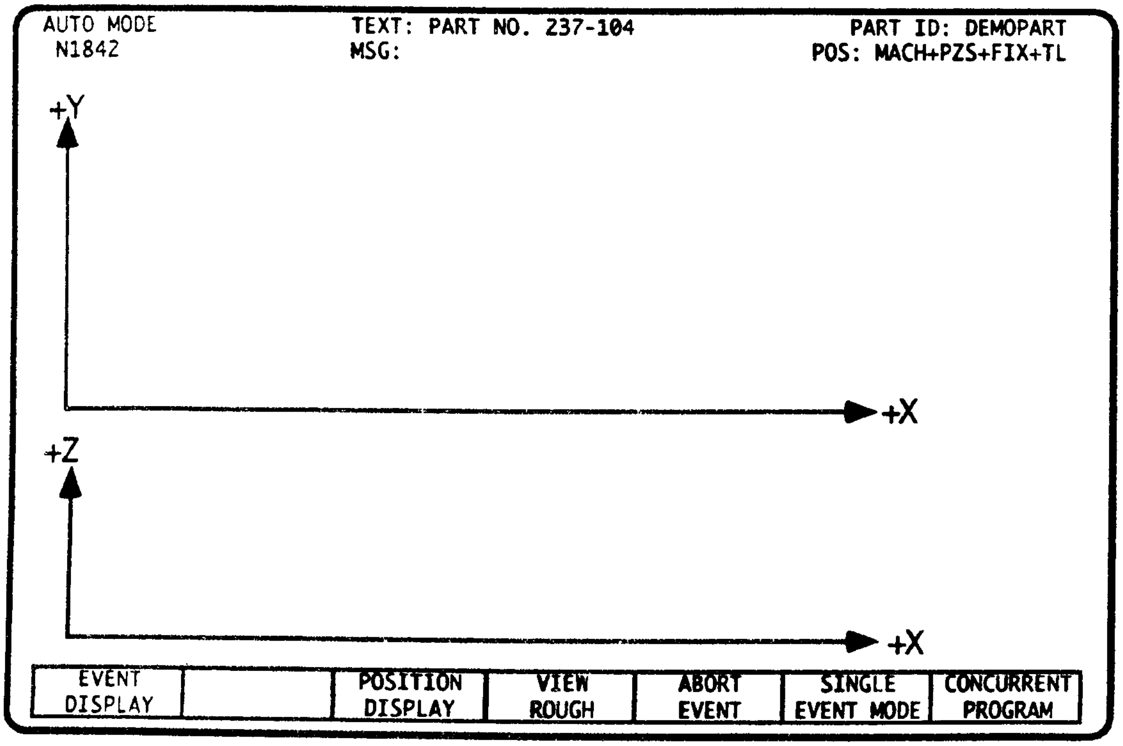
The DIGITIZE part ID may only be selected on units for which the Digitizing option has been purchased. Digitizing uses a touch spindle probe, and special Digitizing software that runs on a personal computer to locate points on an existing part. If installed, Digitizing is activated by selecting the DIGITIZE part ID. This option is further described in the Auto mode section.

EVENT DISPLAY SCREEN



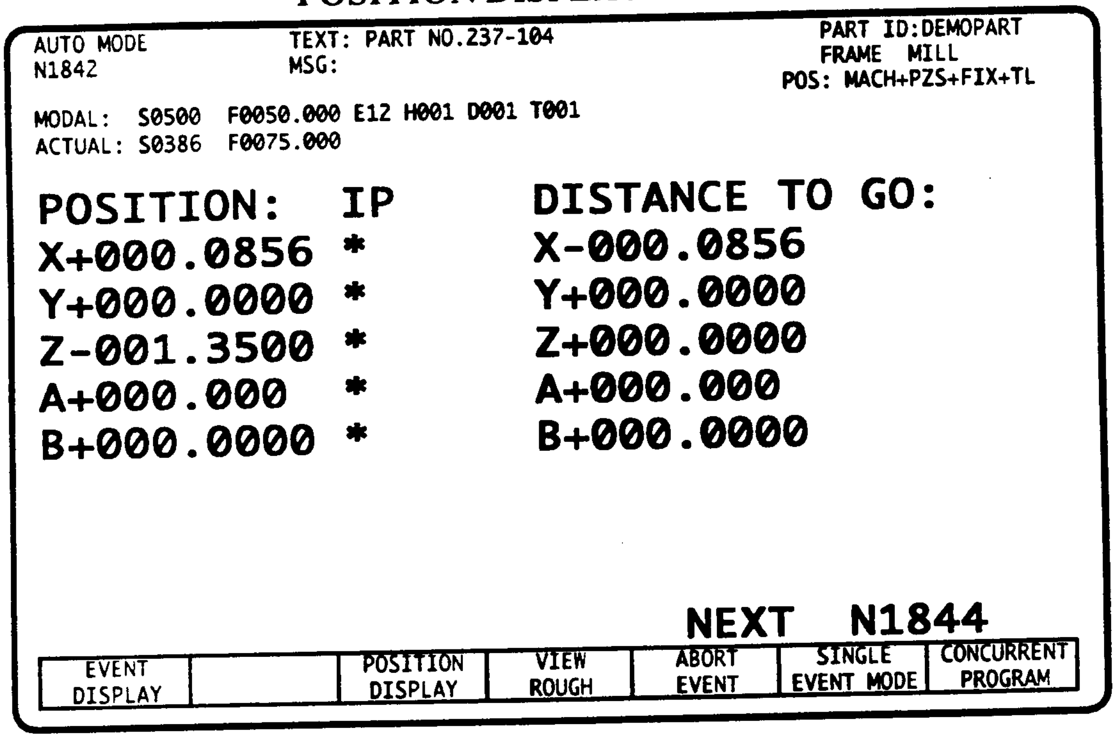
```
AUTO MODE                 TEXT: PART NO. 237-104                 PART ID: DEMOPART
 N1842                    MSG:                                 POS: MACH+PZS+FIX+TL
 FRAME MILL
 X+005.0000          N1842(3)X5Y3Z-1.35W.1R.5D0C2G7L.2        POSITION:     IP
 Y+003.000            F50P20$                                   X+000.0856
 Z-001.3500          N1844(T)POSITION STOCK$                    Y+000.0000    *
 W+000.1000          N1846(0)X3Y3$                              Z-001.3500    *
 R000.5000           N1848(0)X0/G1F10W.1Z-1.2/$                 A+000.000     *
 D0 CW/CCW           N1850(9)S600$                              B+000.0000    *
 C2-ON RIGHT         N1852(0)Z8Y-3Z.1$
 K                   N1854(0)X-.75Y-1.25G0$
 J                   N1856(0)X0/Z-1.85F20G1W.1$
 L.2000              N1858(6)X.75/F1856T2$
 P0020.000           N1860(6)X-.75Y2.5/F1856T1$               DISTANCE TO GO:
 F0050.000           N1862(0)XOYXZOG0$                          X-000.0856
                     N1864(0)X.1Y0Z-1.75F50$                    Y+000.0000
                     N1866(2)XO/Y0/Z-.25I0J0D0$                 Z+000.0000
                     N1868(1)X.05/$                             A+000.000
                     N1870(2)XO/Y0/Z.25/I0J0D1$                 B+000:0000
                     N1872(6)X.05/F1866T3$
                     N1874(0)XOYOZ0G0$
                     N1878(S)X1.1$

                      MODAL:  S0500  F0050.000 E12 H001 D001 T001

                      ACTUAL: S0386  F0075.000                  NEXT N: 1844
```

GRAPHICS DISPLAY SCREEN



```
AUTO MODE                 TEXT: PART NO. 237-104                 PART ID: DEMOPART
 N1842                    MSG:                                 POS: MACH+PZS+FIX+TL
```

POSITION DISPLAY SCREEN



```
AUTO MODE              TEXT: PART NO.237-104              PART ID:DEMOPART
N1842                  MSG:                               FRAME  MILL
                                                         POS: MACH+PZS+FIX+TL

MODAL:  S0500  F0050.000 E12 H001 D001 T001
ACTUAL: S0386  F0075.000

POSITION:   IP          DISTANCE TO GO:
X+000.0856  *           X-000.0856
Y+000.0000  *           Y+000.0000
Z-001.3500  *           Z+000.0000
A+000.000   *           A+000.000
B+000.0000  *           B+000.0000

                                      NEXT  N1844
```

AUTO MODE - INTRODUCTION

Choose the Auto mode to cycle (or run) a part program. While a part is cycling, you may view any of the screens shown on these pages. You may switch between these displays at any time.

- **Event Display** - view axes position, modal machining data, and part program events.
- **Graphics Display** - view tool center plot and axes position.
- **Position Display** - view high visibility axes position, and modal machining data.

The COORDINATE SELECTION soft key allows you to specify the coordinate system that is displayed in the Position Display registers. You have the option of selecting between eight different coordinate systems - including machine coordinates and part coordinates.

The SINGLE EVENT MODE soft key allows you to switch the control into Single Event mode while the control is cycling in Auto mode.

The Auto mode Concurrency feature allows you to perform editing and programming operations on a secondary part program at the same time that the control is executing a primary part program,

Part programs that exceed the control's part program memory can be cycled through the use of the Auto mode's Buffered Input feature.

The control's optional Digitizing feature is also accessed through the Auto mode.

PROGRAM EXECUTION

Prior to running a part program you must select its part program ID in the Catalog mode. You must also enter the required tool and fixture offsets in the Tool Tables mode. When you press CYCLE START, the part program executes without interruption until a halt or end of program M Code is detected.

RESET

When the control is not in cycle, pressing RESET reinitializes the control to its start up conditions. Specifically, pressing RESET causes the actions listed below.

- Resets the program sequence number (as viewed from the NEXT N display of the Event Display and Position Display screens) to the first program event.
- Sets the scale factor to 1.
- Resets any axis rotation to zero degrees.
- Cancels a G6 (Conversational) or G92 (EIA/ISO) Load Position offset without generating tool motion.
- Cancels any active cutter radius compensation without generating axis motion.
- Cancels any quill cycles such as drill, boring or tapping parameters.
- Sets EIA/ISO G codes to their default values[1].
- Your machine may also use the RESET to turn the spindle off and shut off coolant flow.

To ensure that execution begins at the part programs's first N Number, press RESET before you press CYCLE START.

CYCLE START

To begin execution, press CYCLE START. The part program will begin executing with the N Number that appears in the NEXT N field on the Event Display and Position Display screens. Pressing RESET before CYCLE START ensures that the program will begin at its first N Number.

If you wish to start execution in the middle of a part program, move to the Program or Edit mode and enter the N Number you wish to start from. When you return to Auto mode, the start N Number will be in NEXT N field of the Event Display and Position Display screens. To begin execution from this N Number, press CYCLE START, but do not press RESET.

MOTION STOP

To stop execution under normal circumstances, press MOTION STOP. This will cause all axes motion to halt. In most applications, pressing MOTION STOP will not cause the spindle to turn off. If you immediately press CYCLE START, execution will resume from the halt point. However, if you jog any axis or press RESET before you wish to resume, you must ensure that you move to a "safe re-entry" point in the program before you resume execution (see topic on the next page). If you wish to restart from the beginning of the program press RESET, then press CYCLE START

EMERGENCY STOP

When it is necessary to stop all motion and to remove power from the drives, press EMERGENCY STOP. In most applications pressing EMERGENCY STOP also stops the spindle. During Emergency Stop the servo motors are disabled, but the system continues to monitor machine position. You need not re-reference after clearing an Emergency Stop.

[1] To determine the EIA/ISO default values, move the Status screen (press the FAULT/STATUS key twice) and press the RESET key. The G Codes that appear on this screen are the default codes. The exception is the G70/G71 code which remains in its last state until you change it.

FEEDRATE and SPINDLE OVERRIDE POTENTIOMETERS

The FEEDRATE OVERRIDE potentiometer allows adjustment of the programmed feedrate in a range (typically 0 - 150%) selected by your machine supplier. When Test Mode 1 (TM1) is selected in the Set Up mode, the FEEDRATE OVERRIDE potentiometer will also control moves which occur at the rapid rate.

When installed, a SPINDLE OVERRIDE potentiometer allows a similar adjustment to the programmed spindle speed. The spindle control range is determined by your machine supplier.

RETRACT

While running the program, the RETRACT pushbutton will withdraw the tool from a quill cycle (drill, tap, bore, etc.) feed. If you RETRACT while tapping, the spindle direction will reverse before the tap is retracted. The tool withdrawal motion continues as long as the button is pressed. When released, the control will revert to a Motion Stop condition.

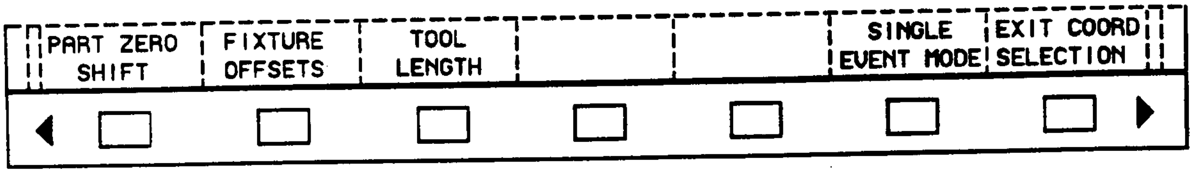
Pressing RETRACT while the control is executing any motion other than a quill cycle has the same effect as pressing MOTION STOP.

ABORT EVENT SOFT KEY

The ABORT EVENT soft key is used to cancel the remainder of an event after MOTION STOP or RETRACT halts execution. Most events except Set Up, M Function, Subroutine and Text events can be aborted. You must jog to the next event's start point before pressing CYCLE START to continue operations. ABORT EVENT disrupts a compensated milling sequence.

COORDINATE SELECTION SOFT KEY

The COORDINATE SELECTION soft key on the Event Display and Position Display screens enables a set of soft keys, **below**, that allow you to select the coordinate system for the Position Display registers.



Each of the PART ZERO SHIFT, FIXTURE OFFSETS, and TOOL LENGTH soft keys acts as a toggle that determines whether a specific type of offset will be included in the Position Display. The highlighted text in the MACH+PZS+FIX+TL field at the top of the screen identifies which offsets are currently included in the Position Display.

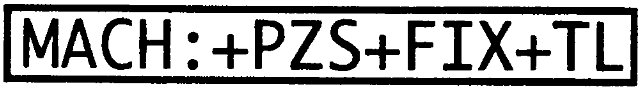
Here are the settings that you will find most useful as you operate your machine:

Display machine coordinates: Press the three soft keys until only the MACH text field is highlighted on the display screen as shown **below**. When these options are selected, the Position Display registers will show the position of the spindle nose relative to Machine Coordinate Zero[2].

MACH:+PZS+FIX+TL

[2] If you have not referenced the machine, this setting will display the position of the spindle nose relative to its position at the time that the machine was turned on.

Display part coordinates: Press the three soft keys until all of the text fields are highlighted on the display screen as shown **below**. When these options are selected, the Position Display registers will show the position of the tool tip relative to the current location of part zero.



MACH:+PZS+FIX+TL

In technical terms, these three soft keys allow you to select which types of offsets are reflected in the control's Position Display registers. Notice that the MACH field on the screen is always highlighted and cannot be turned off. Highlighting one or more of the remaining three fields will cause the control to include the following groups of offsets in the Position Display.

PZS - Part Zero Shift Offsets
- Jog mode Zero Set offset
- Position event G6 Load Position command
- EIA/ISO G92 Axis Preset command
- Handwheel Interrupt Offsets

FIX - Fixture and Setup Offsets
- M Function event E Code offset
- Setup mode PX and PY Part Offset
- Setup event A and B Part Offset
- Offsets for a slaved axis

TL - Tool Length Offset
- Tool Length offset of the active tool

SINGLE EVENT MODE SOFT KEY

The SINGLE EVENT MODE soft key allows you to switch the control into Single Event mode while the control is cycling in the Auto mode.

Pressing this soft key while the control is in cycle will instantaneously put the control in the Single Event mode. The control will halt program execution as soon as it completes the event that is currently executing.

SAFE RE-ENTRY POINT

If you wish to begin execution in the middle of a part program, you must locate a safe re-entry point in the program to begin from. You must also find a safe re-entry point if you halt a program with RETRACT or MOTION STOP, and then press ABORT EVENT, press RESET, or jog any axis.

The "safe re-entry" point in your program must begin a sequence that:

- activates the length and diameter offsets for the tool in use.
- activates any modal EIA/ISO G Codes, M Codes, E Codes, etc.
- positions the tool to an absolute XYZ coordinate. If the part is programmed in incremental coordinates, you must manually jog the axes to the start point of the sequence.
- executes the cutter radius compensation turn on sequence

Once you have identified the safe re-entry point in your part program, move to the Program or Edit mode and enter the event's N number. When you return to Auto mode, the N Number will appear in the NEXT N field on the Event Display and Position Display screens. Do not press RESET before you press CYCLE START.

FAULT/STATUS - FAULT SCREEN

If the control detects a logical error during program execution, the control will halt before the faulty event begins execution. You will be advised of the error condition by a flashing FAULT OR MESSAGE screen display. The Event Display and Position Display's NEXT N field will contain the faulty event's N Number. Note this number, then cancel the flashing screen by pressing the FAULT/STATUS key. The Fault screen will display a three digit error code and a short error message. Note the error code. Press RESET to clear the message. Then press the EXIT DISPLAY soft key to exit the Fault screen. Refer to the Error Codes Appendix to diagnose the problem, then edit the faulty event in the Program mode.

FAULT/STATUS - STATUS DISPLAY SCREEN

You may access the STATUS DISPLAY screen at any time. To view this screen, press the FAULT/STATUS key twice. The STATUS DISPLAY screen contains messages related to initialization parameters and modal G Codes. Also, this screen may contain soft keys which toggle Block Delete and Optional Stop functions. Press EXIT DISPLAY to return to the Mode Select menu.

CONCURRENT MODES

Concurrency allows you to perform operational tasks on a secondary part program while the control executes a primary part program in Auto or Single Event modes. Once you press the CONCURRENT PROGRAM soft key, a secondary Mode Select menu, **following page**, appears.

On initial entry to the Concurrent modes, a message will prompt you to select a secondary program from the Catalog.

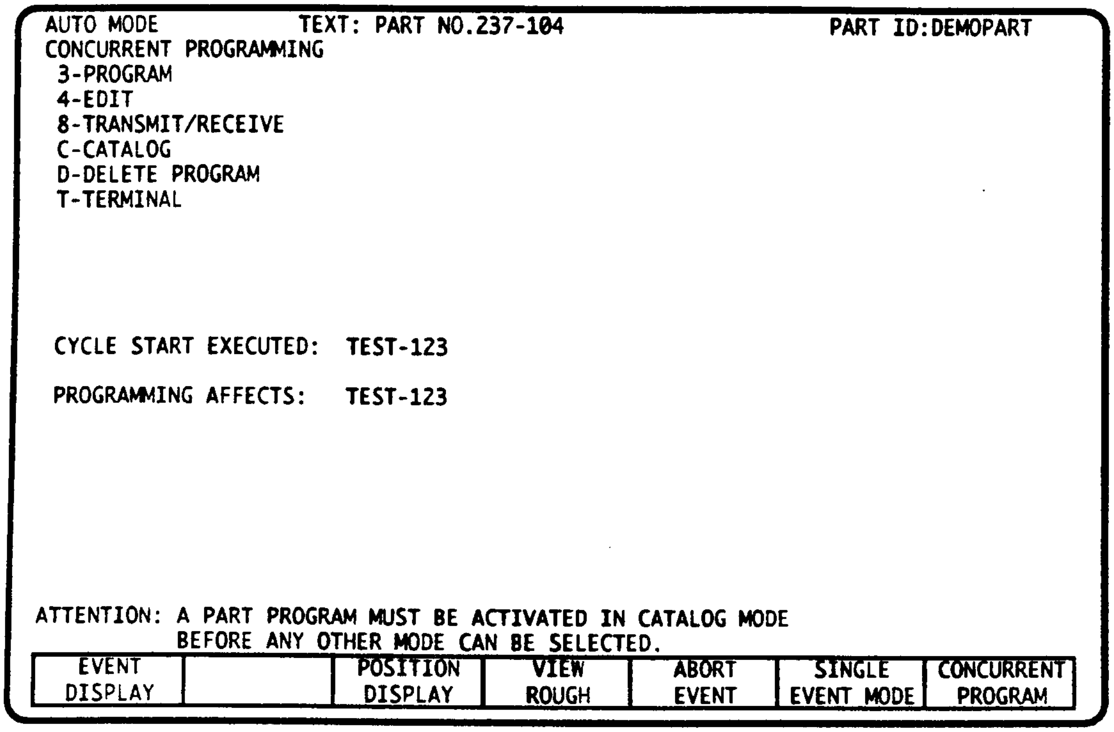
Use a dummy title such as NEW if you intend to receive a program from an off-line device. NEW will be replaced by the ID of the incoming part program once the transmissions is complete.

Once you have selected a program and returned to the Mode Select menu, the primary program will be identified by the message CYCLE START EXECUTES: (part ID). Your secondary program is designated by PROGRAMMING AFFECTS: (part ID).

The following modes are now available to operate on the secondary program:

Program mode:	Enter or edit the secondary program.
Edit mode:	View or edit the secondary program.
Transmit/Receive mode:	Move the secondary program between CNC and RS-232C device.
Delete Program mode:	Delete part or all of the secondary program from memory.
Catalog mode:	Select a new secondary program.
Terminal mode:	

You may now move to any of the secondary modes and begin operations. It is not necessary to move back to Auto mode to maintain operational control of the primary program as CYCLE START and MOTION STOP permit repetitive cycling while you work in the Concurrent modes. If you prefer to move between Auto mode and the Concurrent modes, any activity in progress will not be interrupted. The secondary Program and Edit modes will retain the last selected event. If you are receiving or transmitting a program, the EXIT CONCURRENT soft key will not work until the transmission is completed.



```
AUTO MODE          TEXT: PART NO.237-104                        PART ID:DEMOPART
CONCURRENT PROGRAMMING
 3-PROGRAM
 4-EDIT
 8-TRANSMIT/RECEIVE
 C-CATALOG
 D-DELETE PROGRAM
 T-TERMINAL



 CYCLE START EXECUTED:  TEST-123

 PROGRAMMING AFFECTS:   TEST-123



ATTENTION: A PART PROGRAM MUST BE ACTIVATED IN CATALOG MODE
           BEFORE ANY OTHER MODE CAN BE SELECTED.
```

BUFFERED INPUT EXECUTION

Buffered Input lets you cycle part programs that are too large for the control's part program storage. Your part program can reside on the control's floppy disk drive, hard disk drive, or on a personal computer[3] connected to the control's serial port.

Buffered Input is accomplished by loading several small segments of your part program into the control's memory. Once the control executes a segment of part program code, it will immediately load another segment from the off-line storage device. This process will continue in an uninterrupted fashion until the control executes the part program's M30-end of program command. No Repeat or Subroutines may be present in the buffered input part program.

Selecting the part program title **RS232C**[4] from the main Catalog puts the control into Buffered Input mode.

Move to the Transmit/Receive mode and configure the control to communicate with one of the devices described below.

[3] Actually you can load a part program from any electronic device that can communicate with the DynaPath control through its RS232 serial port. Also the device must support one of the halt/resume protocols (Xon/Xoff or RTS/CTS) that is supported by the control. Refer to the Off-Line Communications Appendix in this manual for further information. A personal computer is the most common device that is used to communicate with the control.

[4] You can confirm that the control is in Buffered Input mode by pressing the MODE SELECT key and observing that the normal Mode Select menu is shortened to only provide access to the modes that are used with this feature. Selecting another part program ID in the Catalog mode will reinstate all of the Mode Select menu options.

Buffered Input from control's floppy disk drive or hard disk drive.

The part program on the floppy or hard disk must be a text file that conforms to the DynaPath format as described in the Off-Line Communications Appendix[5]. The part program may not contain any Repeat or Subroutine events.

Now move to the Transmit/Receive mode and press S2 to select the floppy disk drive, or press S3 to select the hard disk drive. Next press the EDIT SOURCE PARAMETERS softkey, then use the SCROLL soft keys to select the file you wish to run. Remember to press the ENTER key after you move the scroll bar to the correct part program. The title of the selected part program file will appear at the top of the screen.

After you select the file, move to the Auto mode and press CYCLE START. Your program will begin execution.

Buffered Input from a device connected to the control's serial port.

Your off-line device must be able to transmit data that conforms to the DynaPath format as described in the Off-Line Communications Appendix[5]. The transmitted part program may not contain any Repeat or Subroutine events.

Now move to the Transmit/Receive mode and select the S1 - Serial Port Com1 option to tell the control that you wish to load a program through the control's primary serial port. Next press the EDIT SOURCE PARAMETERS soft key then select communication parameters (baud rate, Xon/Xoff control, etc.) which match the settings on your external device.

To begin execution, move to the Auto mode and press CYCLE START. Now move to your external device and begin the transmission. After the control receives several events, it will start cutting the part. The control will continue to execute the part program until it processes an end of program M Code and halts operations.

BUFFERED INPUT ERROR DETECTION

If the control detects an error in the incoming data, a flashing FAULT OR MESSAGE display will report the error code. If communication is interrupted between the transmitting device and control, press MOTION STOP to terminate the sequence, then RESET to reinitialize the control. To recover, you must start program transfer from the first program event.

[5] As an exception to the format shown in the Communications Appendix, N Numbers are not required prior to every event or block. Events are executed in the order that they appear in the file.

TOUCH PROBE DIGITIZING

Touch Probe Digitizing[6] is an optional feature that uses a spindle probe, a personal computer, and special digitizing software to generate CNC programs for existing complex 3D parts. With the part prototype in place and the probe in the spindle, the external computer sends part program instructions through the control's RS-232C port, directing it to move the probe around the prototype in a consistent, matrix-like pattern. Points are read from the part by the probe and reported back to the host; these points are then used by the digitizing software to generate a CNC program to reproduce the part shape.

REQUIREMENTS

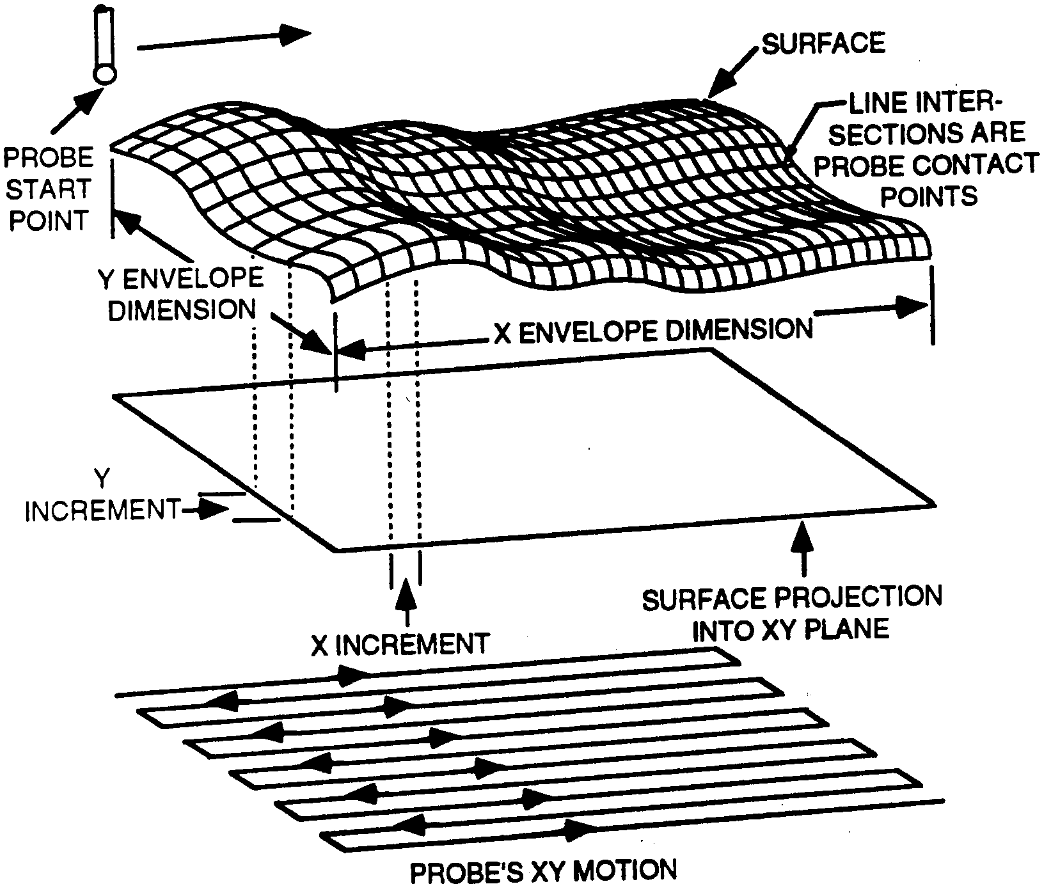
In order to digitize, you must have:

- A Delta 40M, D50M, D60M control equipped with the digitizing option
- The Probing option
- A Spindle Probe
- A host computer and appropriate digitizing software.
- The Part ID: DIGITIZE must be selected in the Catalog.

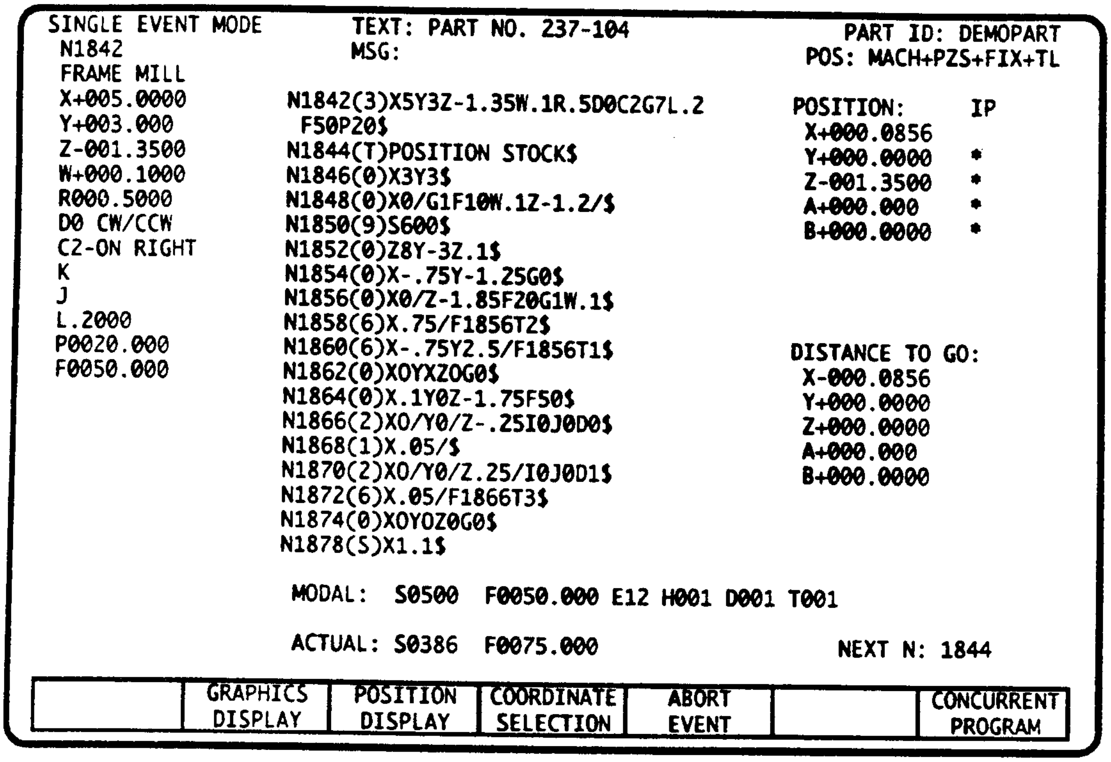
For further information on the Digitizing feature, refer to the Digitizing Operator's Manual that is provided with the Digitizing feature. The illustration, **below**, shows how Digitizing software can sample a 3D surface to approximate its profile. The increments and dimensions are chosen by you to obtain the desired resolution.

[6] Autocon Technologies has two types of systems for reverse engineering parts. The first system, Touch Probe Digitizing, is described above. The second system, PrecisionScan, is a high speed, continuous trace scanning system which traces a part by "dragging" a special three dimensional probe over the part. Contact Autocon Technologies for further information about these systems.

ASSUMING X AS MAJOR AXIS, PROBE PROBES UP AND DOWN IN Z MOVING ALONG X BY X INCREMENT UNTIL IT REACHES END OF X ENVELOPE, THEN INCREMENTS Y AND PROBES BACK. PROCESS IS REPEATED UNTIL END OF ENVELOPE IS REACHED.

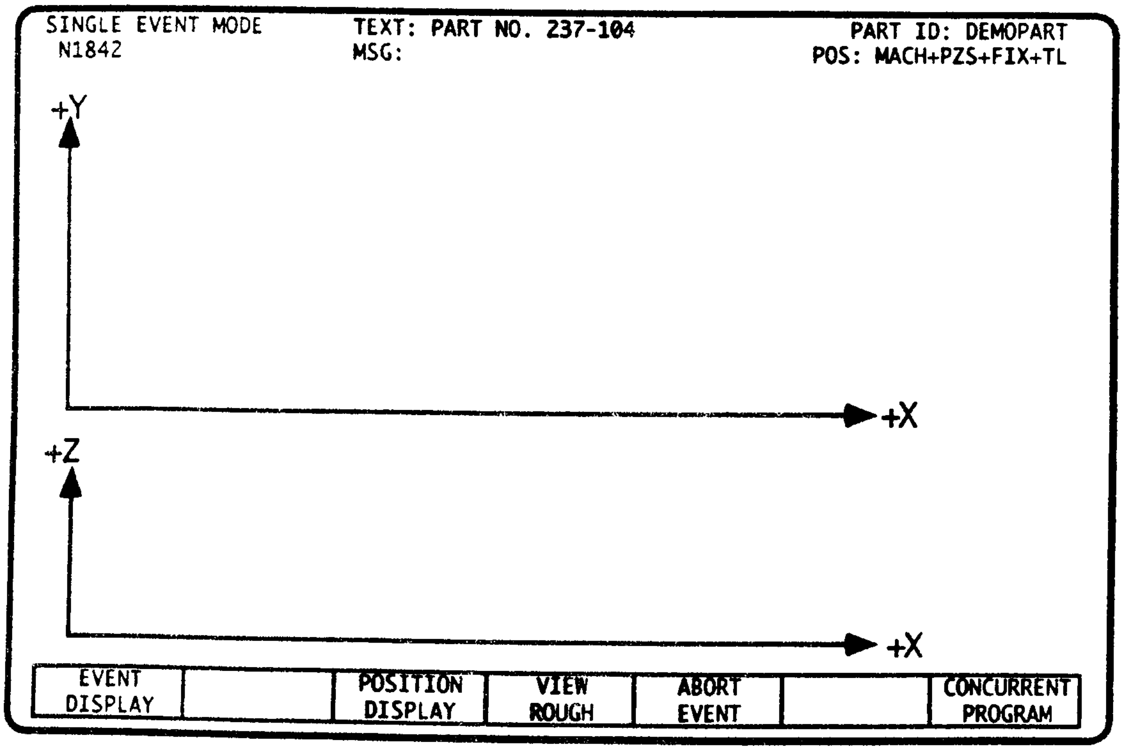


EVENT DISPLAY SCREEN



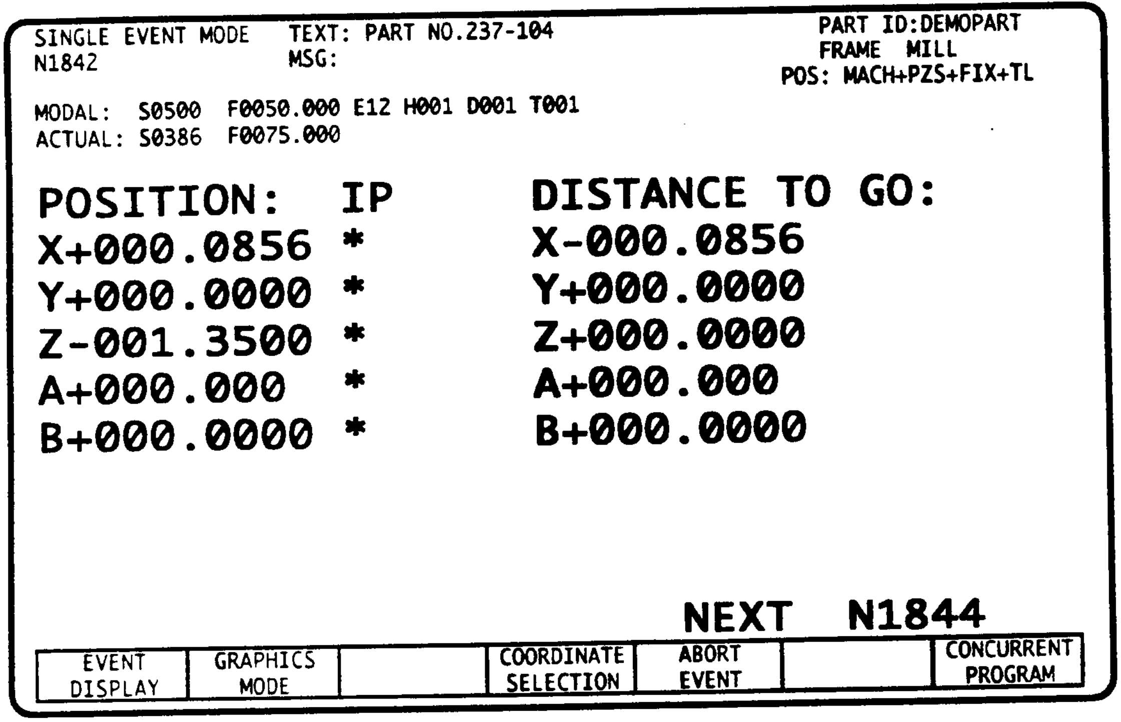
```
SINGLE EVENT MODE        TEXT: PART NO. 237-104                    PART ID: DEMOPART
 N1842                   MSG:                                   POS: MACH+PZS+FIX+TL
 FRAME MILL
 X+005.0000           N1842(3)X5Y3Z-1.35W.1R.5D0C2G7L.2         POSITION:      IP
 Y+003.000             F50P20$                                    X+000.0856
 Z-001.3500           N1844(T)POSITION STOCK$                     Y+000.0000    *
 W+000.1000           N1846(0)X3Y3$                               Z-001.3500    *
 R000.5000            N1848(0)X0/G1F10W.1Z-1.2/$                  A+000.000     *
 D0 CW/CCW            N1850(9)S600$                               B+000.0000    *
 C2-ON RIGHT          N1852(0)Z8Y-3Z.1$
 K                    N1854(0)X-.75Y-1.25G0$
 J                    N1856(0)X0/Z-1.85F20G1W.1$
 L.2000               N1858(6)X.75/F1856T2$
 P0020.000            N1860(6)X-.75Y2.5/F1856T1$                DISTANCE TO GO:
 F0050.000            N1862(0)XOYXZOG0$                           X-000.0856
                      N1864(0)X.1Y0Z-1.75F50$                     Y+000.0000
                      N1866(2)XO/Y0/Z-.25I0J0D0$                  Z+000.0000
                      N1868(1)X.05/$                              A+000.000
                      N1870(2)XO/Y0/Z.25/I0J0D1$                  B+000.0000
                      N1872(6)X.05/F1866T3$
                      N1874(0)XOYOZ0G0$
                      N1878(S)X1.1$

                       MODAL:  S0500   F0050.000 E12 H001 D001 T001

                       ACTUAL: S0386   F0075.000                  NEXT N: 1844
```

GRAPHICS DISPLAY SCREEN



```
SINGLE EVENT MODE        TEXT: PART NO. 237-104                    PART ID: DEMOPART
 N1842                   MSG:                                   POS: MACH+PZS+FIX+TL
```

POSITION DISPLAY SCREEN



```
SINGLE EVENT MODE   TEXT: PART NO.237-104                    PART ID:DEMOPART
N1842               MSG:                                     FRAME  MILL
                                                        POS: MACH+PZS+FIX+TL

MODAL:  S0500  F0050.000 E12 H001 D001 T001
ACTUAL: S0386  F0075.000

POSITION:   IP        DISTANCE TO GO:
X+000.0856  *         X-000.0856
Y+000.0000  *         Y+000.0000
Z-001.3500  *         Z+000.0000
A+000.000   *         A+000.000
B+000.0000  *         B+000.0000

                                     NEXT   N1844
```

INTRODUCTION

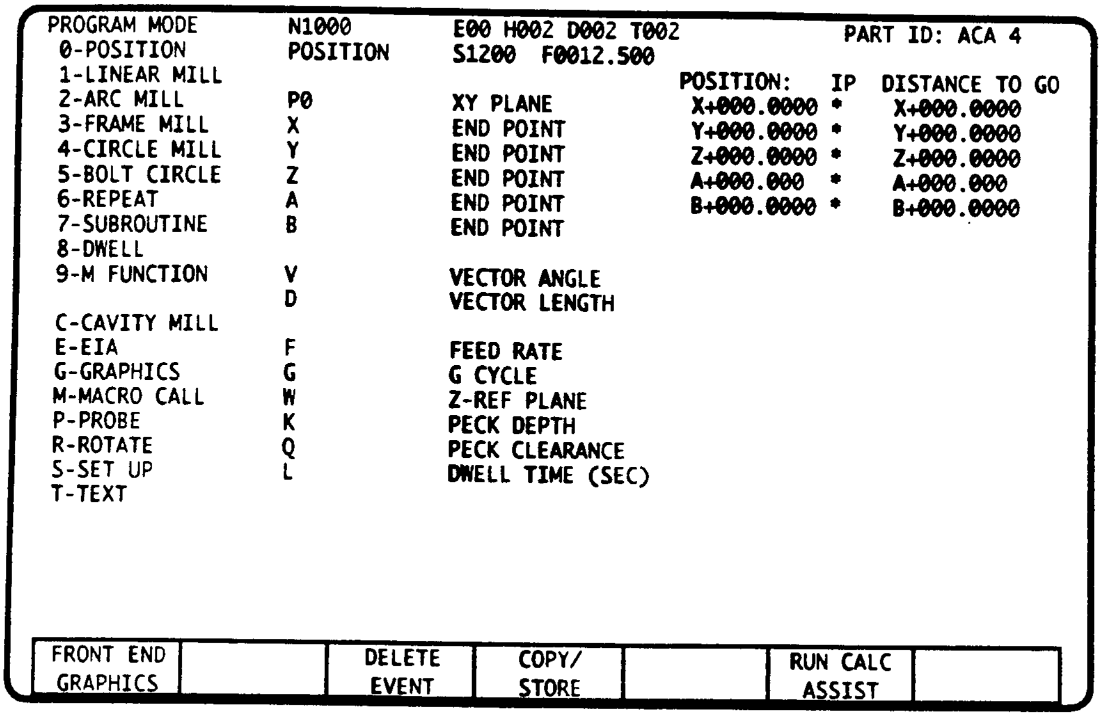
Choose the Single Event mode to run a part program one event at a time. While a part is cycling, you may view any of the following three screens. You can switch between these displays at any time.

- **Event Display** - view axes position, modal machining parameters, and part program data.
- **Graphics Display** - view tool center plot and axes position.
- **Position Display** - view high visibility axes position, and modal machining parameters.

EXECUTION

Execution is similar to that of Auto mode except this mode runs the part program one event at a time. Pressing CYCLE START causes the event shown in the NEXT N field to execute. The control halts part program execution at the end of each event. The following event (now displayed in the NEXT N field) will execute on the next CYCLE START.

All displays and operations described in the Auto mode section are also applicable to this mode. Refer to the Auto mode section for further operational instructions.



```
PROGRAM MODE        N1000        E00 H002 D002 T002              PART ID: ACA 4
 0-POSITION         POSITION     S1200  F0012.500
 1-LINEAR MILL                                     POSITION:    IP  DISTANCE TO GO
 2-ARC MILL         P0           XY PLANE           X+000.0000 *     X+000.0000
 3-FRAME MILL       X            END POINT          Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL      Y            END POINT          Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE      Z            END POINT          A+000.000  *     A+000.000
 6-REPEAT           A            END POINT          B+000.0000 *     B+000.0000
 7-SUBROUTINE       B            END POINT
 8-DWELL
 9-M FUNCTION       V            VECTOR ANGLE
                    D            VECTOR LENGTH
 C-CAVITY MILL
 E-EIA              F            FEED RATE
 G-GRAPHICS         G            G CYCLE
 M-MACRO CALL       W            Z-REF PLANE
 P-PROBE            K            PECK DEPTH
 R-ROTATE           Q            PECK CLEARANCE
 S-SET UP           L            DWELL TIME (SEC)
 T-TEXT
```

INTRODUCTION

This mode may be accessed as a primary or secondary mode. Use this mode to enter or edit the part program which is selected in the Catalog mode. You can also execute single part program events directly from this mode by pressing CYCLE START, or using the "N." Manual Data Input utility.

DISPLAY and DATA ENTRY

You must initially enter a Part ID (program title) in the Catalog mode to identify the program you will enter or edit. A menu of the available Conversational programming events, above, appears when you press the EVENT TYPE key. The Position event is shown on this screen. Each event is described in the Conversational or EIA/ISO Programming sections of this manual.

Creating a part program is simply the repetitive process of entering an N Number (sequence number), choosing an event type, then entering the information required for the selected event type. Each step is described below.

ENTERING N NUMBERS

An N Number, which may range from N0 to N9999.999, must be entered for each Conversational event or EIA/ISO block. This N Number number sets the sequential order of the event relative to other events in the part program.

To initially select an N Number, key in N, the desired number, then press ENTER. Leading and trailing zeros are not required and need not be entered. The decimal point format of 0-9999.999 allows up to 999 events to be later inserted between consecutive integer N Numbers.
After entering the first N Number, all following N Numbers can be entered using one of the following three methods.

Direct entry - To directly enter an N number, key in Nxxxx.xxx[1] ENTER, where xxxx.xxx represents the sequence number for the event. If the N Number already exists in the part program, its data will be displayed; you may now edit this event. If the sequence number does not exist, the control will create the new N Number.

Step by integers - Press the NEXT EVENT key to step to the next integer N Number. To step to the previous integer N Number, press - NEXT EVENT. Stepping by integer N Numbers is often preferred since it allows the fewest key strokes and permits up to 999 events to be inserted between adjacent events. **Example**: you have just completed entries to event N50, if you press NEXT EVENT, N51 will be displayed. To step from N50 back to N49, press - NEXT EVENT.

Program the N Number step - To allow space for integer N Numbers to be inserted at a later time, you may program a step of 5, 10, 20 (any reasonable number) N Numbers by using the N NEXT EVENT sequence. To program an N Number step key in NXXXX.XXX[1] NEXT EVENT, where xxxx.xxx represents the desired step. The control remembers the programmed step until you change it, or you turn the control off. After this, you can step forward, by pressing N NEXT EVENT. To step backwards, press -N NEXT EVENT. **Example**: To program a step of 10 numbers, press N10 NEXT EVENT. If you have just completed event N20, the control will step to N30. After N30 has been programmed you may now step to N40 by pressing N NEXT EVENT, or step back to N20 by pressing - N NEXT EVENT.

When using NEXT EVENT or N NEXT EVENT, intermediate N Numbers are displayed. For example, when going from N50 to N51 using the NEXT EVENT key, an event stored at N50.5 will be displayed. This insures that no events are overlooked.

SELECT EVENT TYPE

After you enter an N Number, press the EVENT TYPE key. The screen will now display a list of the event types that you may select to program. To select an event type, press the number or letter that precedes the event's name. For example, a Linear Mill event is EVENT TYPE 1. When you select the event type, the control will display a list of all entries for the event.

After entering all data for the event, step to the next N Number. To minimize keystrokes, the control assumes that you want to program the same type of event as the last block (a series of Linear Mill events, for example). Press EVENT TYPE only if you wish to change the displayed event type.

ENTERING and EDITING EVENT DATA

After you select an event type, a list of the possible entries for that event will appear on the screen. A short description appears to the right of each field. To enter event data, key in the letter identifier then the desired number. Your entry will be displayed on the lower right corner of the monitor. To complete the entry, press ENTER.

To erase an improper entry that has not been entered, press the CLEAR key. If you have already entered the data, key in the new entry and press ENTER. The new data will replace the last entry. To remove an entry, key in the letter identifier, then press the CLEAR key.

When you have completed all entries to an event, you may move to the next event by entering a new sequence number, pressing the NEXT EVENT key, or initiating a N NEXT EVENT sequence.

[1] Leading and trailing zeros are not required. The keystrokes to enter N10, for example, are N10 ENTER.

BLOCK DELETE

Block Delete allows the control to conditionally skip designated events when a program executes. The Block Delete feature must,be enabled by your machine supplier. If enabled, a BLOCK DELETE soft key[2] will appear on the Status screen[3]. This soft key is used to turn Block Delete on and off. The Block Delete status (on/off) also appears on the this screen.

To program Block Delete, press the INCR key and ENTER after the N Number has been entered. This will put a slash (/) before the N number (e.g. "/N1000"). When you execute a part program with Block Delete On, all Block Delete events will be skipped (only N numbers without slashes will execute). When Block Delete is off all part program events will execute, including Block Delete events. Block Delete can be added to any event but the Text event.

Block Delete slashes may be added to an event at any time. They may be removed by pressing INCR and CLEAR at the desired N number.

DELETE EVENT SOFT KEY

Use the DELETE EVENT soft key to erase a single event from a part program. First, select the N Number you wish to erase. Once the event is displayed on the screen, you can erase it by pressing the DELETE EVENT soft key.

COPY/STORE SOFT KEY

The COPY/STORE soft key allows you to store a copy of the selected event at a new N number. First, select the N Number that you want to copy. Then, key in the N number where you want the duplicate event to appear, but **DO NOT press** ENTER. Instead, press the COPY/STORE soft key. The duplicate event will be displayed at the new N number. The original event is not erased by this operation.

RUN CALC-ASSIST and FRONT END GRAPHICS SOFT KEYS

These two soft keys are used to begin a Calc-Assist run on your part program. Instructions for running the Calc-Assist feature are provided in the Calc-Assist part in this manual.

EXECUTING EVENTS IN PROGRAM MODE

There are two methods you can use to execute program events directly from the Program mode. While the event executes, you can monitor axes position by pressing the POSITION DISPLAY soft key. Each method is described below.

CYCLE START

After entering or viewing a part program event, you may press CYCLE START to execute the event directly from this mode. This feature is useful in testing your program data, activating tool codes and fixture offsets. You may also wish to select one of the TM1, TM2 or TM3 test modes from the Set Up mode to extend feedrate override control or inhibit axis motion prior to executing the event.

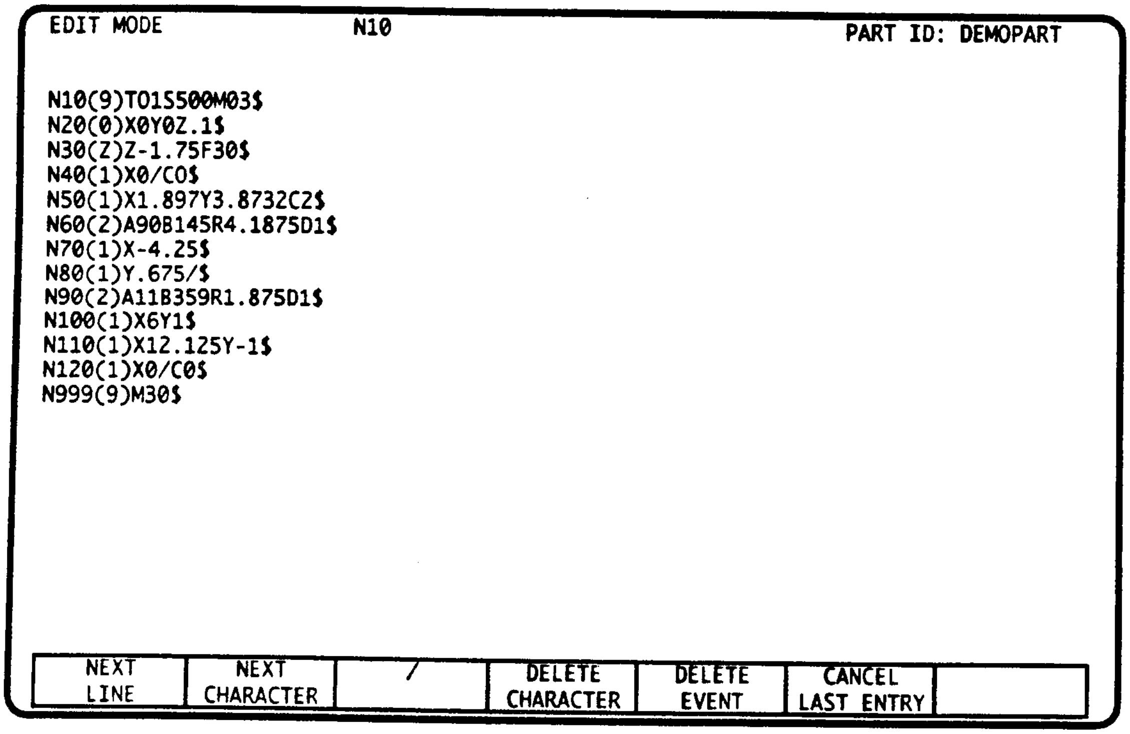
[2] Alternately, your machine supplier may provide an alternating pushbutton or similar type control that allows the operator to activate the Block Delete function. This control will be used to turn Block Delete On and Off.

[3] To view the Status screen, press the FAULT/STATUS key twice.

MDI - N. SEQUENCE NUMBER

The N. utility allows an event to be programmed and executed without being stored in a part program. This allows you to execute M Codes, T Codes and position or feed to any programmed endpoint from this mode without storing any new data. Press N. ENTER[4] to identify the event's sequence number, then enter the data and press CYCLE START to execute the event. When you select a new sequence number, event type or exit this mode, the N. event is erased.

[4] "N." means that the decimal point key is pressed directly after the N key.



```
EDIT MODE                N10                                  PART ID: DEMOPART

N10(9)T01S500M03$
N20(0)X0Y0Z.1$
N30(Z)Z-1.75F30$
N40(1)X0/CO$
N50(1)X1.897Y3.8732C2$
N60(2)A90B145R4.1875D1$
N70(1)X-4.25$
N80(1)Y.675/$
N90(2)A11B359R1.875D1$
N100(1)X6Y1$
N110(1)X12.125Y-1$
N120(1)X0/C0$
N999(9)M30$
```

EDIT MODE - INTRODUCTION

The Edit mode lets you view consecutive part program events on the screen. You may also use this mode as a text editor to modify part programs[1].

EDIT MODE SCREEN and the EDIT LINE

The first line of the Edit mode screen displays the active part program ID and the active N Number. The remaining text shows consecutive part program events. The event at the top of the listing is referred to as the edit line because the control's edit and delete commands operate on this event. The three ways to move an event to the edit line are:

1) Use the LEFT and RIGHT ARROW keys to page through the part program.
2) Enter the N Number of the event that you wish to edit.
3) Press the NEXT LINE to move the next event (the second line of program text) to the edit line; or press -NEXT EVENT to move the previous event to the edit line.

To return to the beginning of the program, enter N0 (N zero), or press the RESET[2] key.

[1] However, the majority of your program editing activities should be performed in the Program mode which provides interactive assist screens and value entry certification to simplify your programming effort.

[2] The RESET key will only cause the first event in the program to appear on the edit line when you are using the primary Edit mode. The RESET key will have no affect when you are using the the Concurrent Edit, or Macro Edit modes.

EDITING THE EDIT LINE

The control is ready to accept changes to the edit line when the N Number at the top of the screen is the same as the N Number on the edit line. When this occurs, you can perform the following operations on the edit line.

INSERT CHARACTERS

Characters may be inserted anywhere within the edit line. To insert a character (or string of characters), use the NEXT CHARACTER (-NEXT CHARACTER) soft keys to position the blinking cursor over the character immediately to the right of the desired insertion point[3]. Now key in the character(s) to be inserted. As you enter them, they will appear on the bottom line of the display. Pressing ENTER will insert the character or string into the event.

Special characters

- A blank space in a Text event is entered by pressing the INCR key.
- An incremental slash following a number is entered by pressing the "/" soft key.
- A Block Delete indicator (a slash preceding the N number) is entered by pressing the "/" soft key twice, then pressing the ENTER key.

DELETE CHARACTERS

To delete one or more consecutive characters, position the blinking cursor over the character (or left most character in a string) you want to delete. Now press the DELETE CHARACTER soft key. The character will be highlighted and the blinking cursor will position one character to the right[4]. By repeatedly pressing the DELETE CHARACTER soft key you can select more characters to be deleted. When all the characters you want to delete are highlighted (the blinking character will not be deleted), press the ENTER key to delete them.

DELETE THE EDIT LINE

To delete the event at the edit line, move the cursor over the event's leftmost character. The easiest way to move the cursor to this point is to enter the event's N Number. Now press the DELETE EVENT soft key to delete the event.

INSERT AN EVENT

The control is ready to insert an event immediately before the edit line when the N Number at the top of the screen differs from the N Number of the edit line. This occurs when you have entered an N Number that does not exist in the part program[5]. To complete the insertion, enter an event type by first pressing the event type key and then keying in the event type character identifier (e.g. "0" for Position). The new N Number and event type will appear at the edit line with the cursor positioned over the terminating dollar sign ("$"), ready for the insertion of data into the event.

[3] If the event is empty or new, the cursor will be properly aligned to the "$" automatically.

[4] Only characters that are highlighted are selected for delete. The character under the blinking cursor is not selected.

[5] As an indication the the control is ready to accept a new N Number, the blinking cursor will not appear.

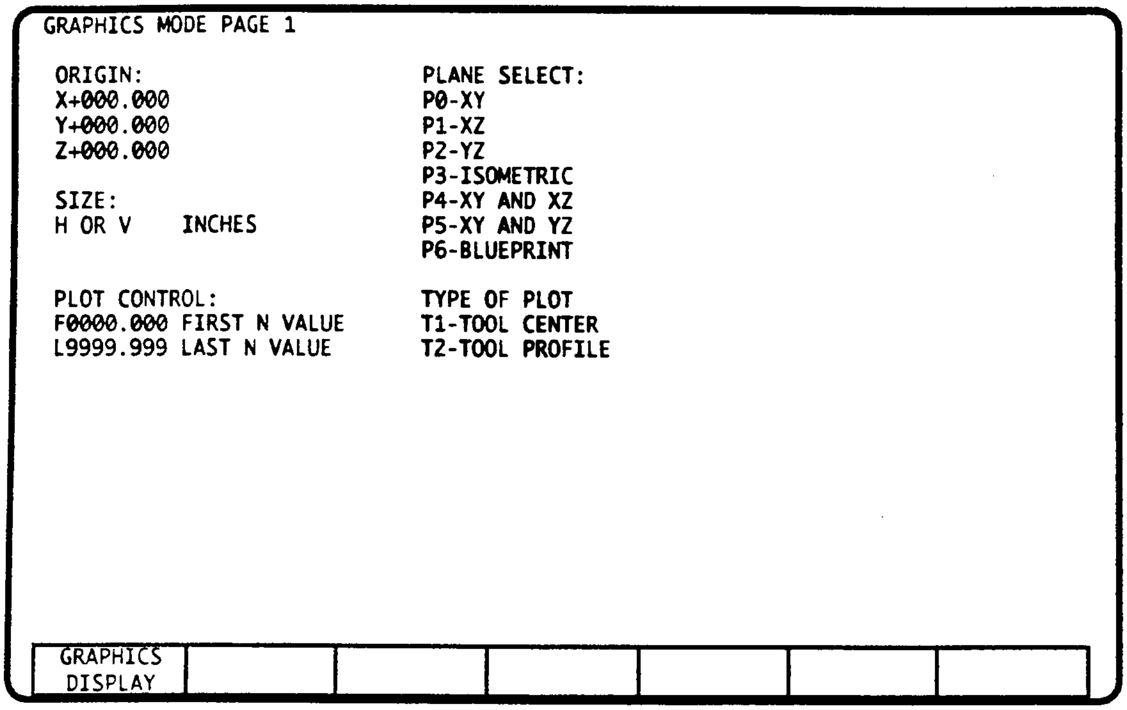
CANCEL LAST ENTRY

The CANCEL LAST ENTRY soft key allows you to cancel the last entry that appears on the entry line at the bottom of the display[6]. It also allows you to deselect characters that have been selected with the DELETE CHARACTER soft key. Repeate dly pressing CANCEL LAST ENTRY will deselect or cancel the remaining characters one at a time. To deselect all the characters that you have selected for deletion or to clear the entire edit line, press the CLEAR key.

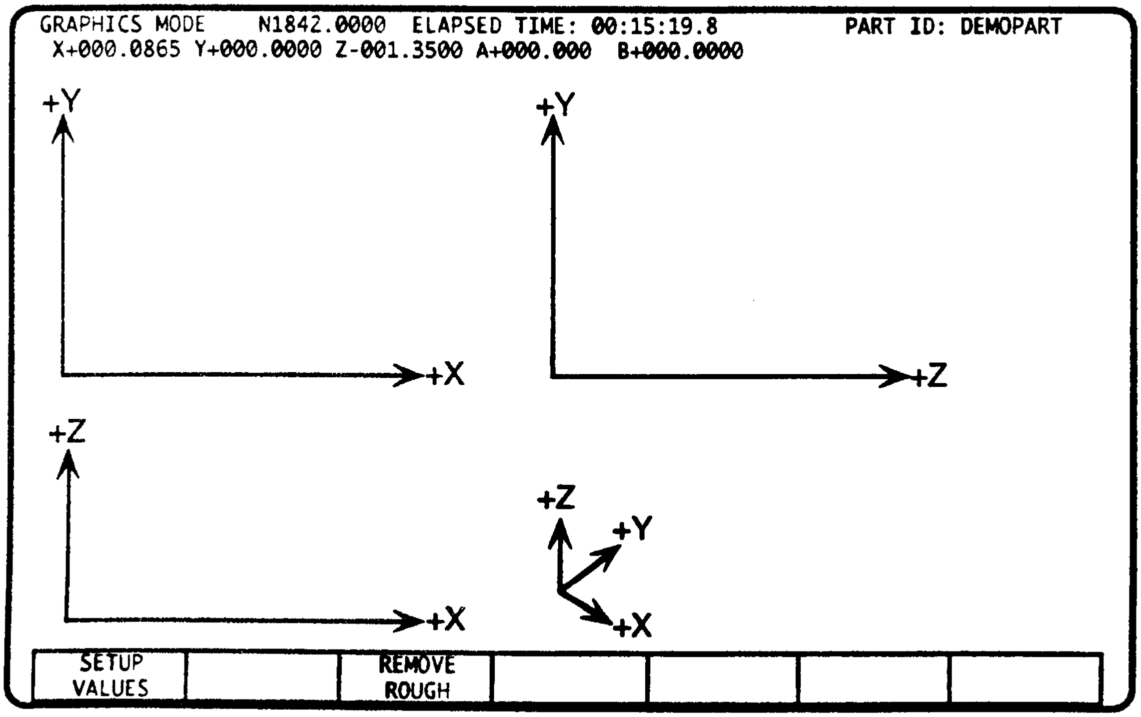
INVALID DATA ENTRY

Since the Edit mode is a simple text editor, no error checking for out of range or invalid entries occurs. However, if you attempt to display an event that contains an error in the Program mode, a PPS FORMAT FAULT will be displayed. When this occurs, you must move to the Edit mode and correct the invalid entry. If you cannot isolate the invalid entry, you should delete the event and reenter it correctly in the Program mode. For this reason, you will find it easier to create and edit all events in the Program mode, and use the Edit mode to inspect your code in a sequential manner.

[6] This must be done **before you press ENTER.**



```
GRAPHICS MODE PAGE 1

 ORIGIN:                     PLANE SELECT:
 X+000.000                   P0-XY
 Y+000.000                   P1-XZ
 Z+000.000                   P2-YZ
                             P3-ISOMETRIC
 SIZE:                       P4-XY AND XZ
 H OR V    INCHES            P5-XY AND YZ
                             P6-BLUEPRINT

 PLOT CONTROL:               TYPE OF PLOT
 F0000.000 FIRST N VALUE     T1-TOOL CENTER
 L9999.999 LAST N VALUE      T2-TOOL PROFILE
```



```
GRAPHICS MODE     N1842.0000  ELAPSED TIME: 00:15:19.8        PART ID: DEMOPART
 X+000.0865 Y+000.0000 Z-001.3500 A+000.000  B+000.0000
```

INTRODUCTION

Graphics mode may be accessed as a primary mode. This mode lets you confirm tool center and part profile paths without generating tool motion. Prior to running a part in Graphics you must select to view tool motion in one of the primary planes, one of the simultaneous two plane views, isometric view, or blueprint view. You must also select the origin and size of the Graphics display. Once you have plotted the part, the cursor jog utility can help you locate coordinates on. A timer utility can help you approximate machining time.

DISPLAY and DATA ENTRY

Graphics mode contains two pages. Page 1, **upper screen**, contains the plot set up information. Page 2, **lower screen**, is the Graphics display screen where the plot is drawn. Use the GRAPHICS DISPLAY and SET UP VALUES soft keys to move back and forth between the screens.

GRAPHICS SETUP EVENT

As an alternate to entering the Graphics setup values in this mode, you can program a Graphics Setup event as the first event in your part program to automatically set the graph origin, size, and the other parameters described in this section. Refer to the Graphics Setup event topic later in this manual.

SET UP GRAPHICS

Whether you plot tool motion in Graphics mode or in Auto or Single Event modes, you must first set the plane select and plot size to properly show all tool motion. Complete the following entries.

P - PLANE SELECT

Choose an entry in the range of P0 - P6 to select the planes in which to plot tool motion.

SETTING THE PLOT SIZE

To configure the control to display all tool motion, you must enter the plot origin and plot size prior to running a part program. To complete these entries you must enter an X, Y and/or Z plot origin and a plot size. Two example parts and their appropriate setup values are shown on the **following page**.

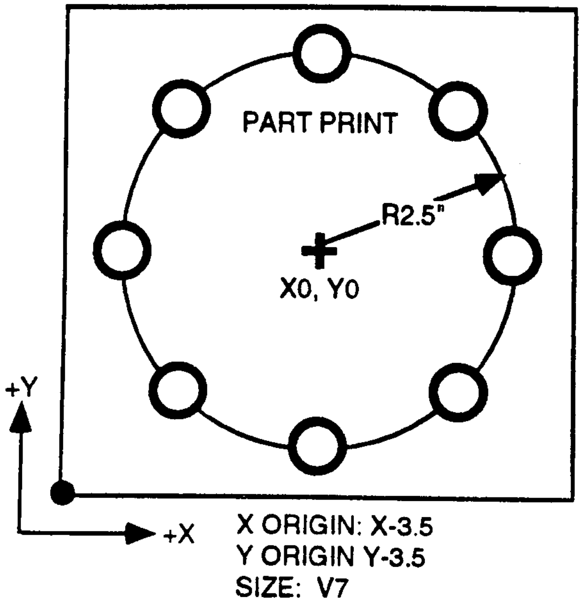
X, Y, Z - GRAPHICS ORIGIN

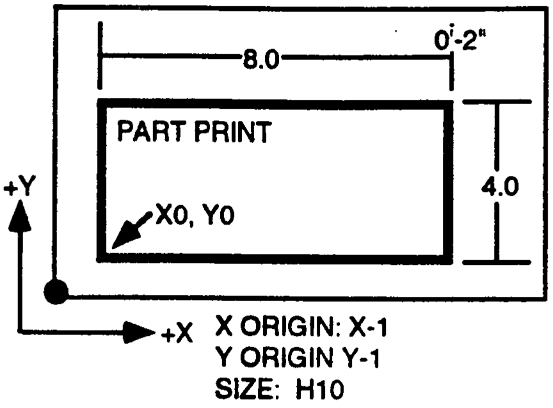
The plot origin defines the point, in part coordinates, which represents the lower left corner of the display screen. For each axis to be displayed, enter a number that is approximately 1 inch smaller than the most negative tool position you wish to plot. Portions of the part which have coordinates smaller (more negative) than the Graphics Origin will not appear on the screen.

H or V - SIZE

The SIZE entry determines how many inches or millimeters of axes travel will be shown on the display screen. Enter H to specify the number of inches or millimeters to display along the horizontal axes. Enter V to specify the number of inches or millimeters to display along the vertical axis. The control automatically sizes the other entry.

We recommend that you use V if the part's length is about the equal along both axes, or longer along the axis that is vertically aligned to the plotting window.





SETTING PLOT SIZE USING THE JOG CURSOR

As an alternate method of determining the plot origin and size, you can use the jog cursor to frame the desired tool motion then have the control calculate the X, Y, Z, and V or H entries. To use this method, follow the procedure described below.

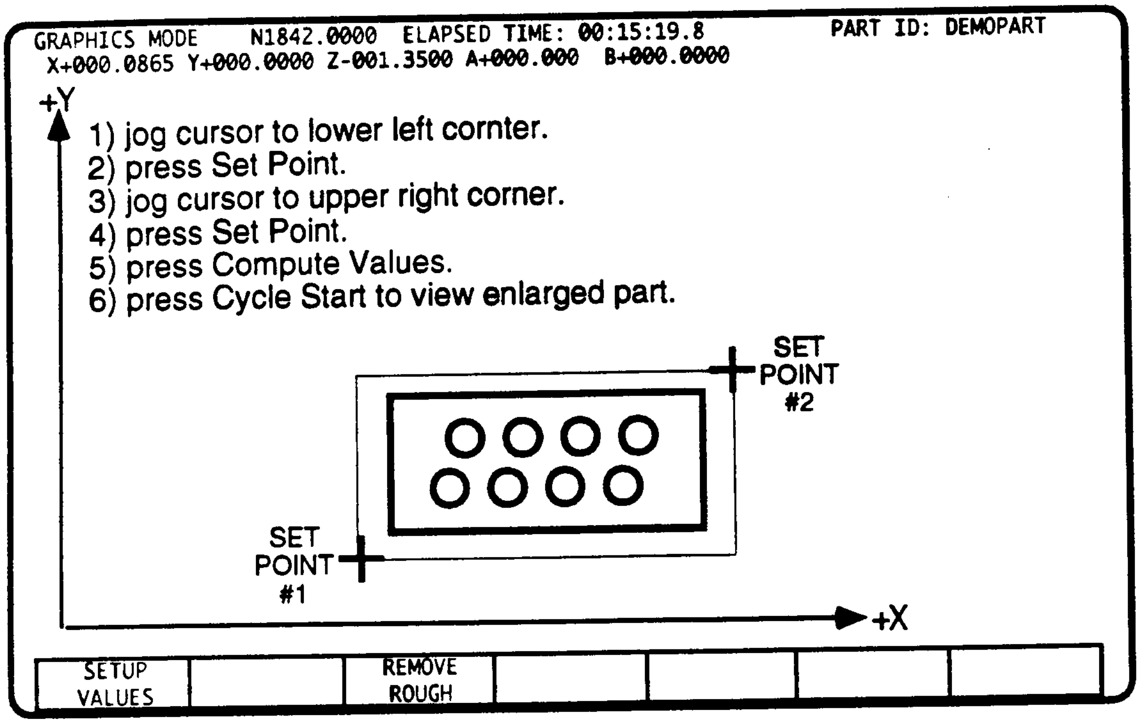
RESET VALUES

Pressing the RESET VALUES soft key will configure the monitor to display the widest possible view of the machine envelope (as defined by your Software Travel Limits). Pressing this soft key ensures that all tool motion in the selected plane will appear on the screen. A plot of a part on this screen will appear very small. After pressing the RESET VALUES soft key, you must use the jog cursor to "zoom in" on the plot.

ZOOM IN ON THE PICTURE

Use this feature to increase the size of a plot which already appears on the display screen. To zoom in you must use the jog keys to move the Graphics cursor to points near the part profile that you want to enlarge. The Graphics cursor will "jog" at the rate that is selected in the Jog mode. Thc 4 steps to enlarging a part on the screen are described and illustrated below.

1) Jog the crosshairs to the lower left corner of the detail you want to frame. Press the SET POINT soft key.
2) Jog the crosshairs to the upper right corner of the detail you want to frame. Press the SET POINT soft key a second time.
3) Press the COMPUTE VALUES soft key. This will cause the control to calculate a plot origin and plot size based on the rectangle defined by the two set point locations.
4) Press CYCLE START to execute the part in Graphics mode.



```
GRAPHICS MODE    N1842.0000  ELAPSED TIME: 00:15:19.8          PART ID: DEMOPART
 X+000.0865 Y+000.0000 Z-001.3500 A+000.000  B+000.0000
```

1) jog cursor to lower left cornter.
2) press Set Point.
3) jog cursor to upper right corner.
4) press Set Point.
5) press Compute Values.
6) press Cycle Start to view enlarged part.

T - TOOL CENTER or PART PROFILE

Select T1 to plot the tool center path, or select T2 to display the part profile path as the tool moves around the part. When plotting the part profile, the control uses tool information from the part program and Tool Tables mode to display the correct tool offset. When plotting a part that uses cutter compensation, it is often informative to plot the part with the tool center path and part profile path on the same screen. Note that the control will always display the tool center path when plotting in the Auto or Single Event mode.

F, L - PLOT CONTROL

Enter the First event and the Last N Numbers to display in this mode. The entire part program will be plotted when F0, L9999.999 is entered. You may change these entries to display any portion of a part program. To ensure that the motion shown on the monitor represents actual tool motion, the portion of the part program that you specify to plot should contain the following items.

- execute a T Code for the tool used to shape the path or profile.
- begin with an event programmed in absolute coordinates to relate the program to the plot origin.
- execute the programmed feedrate.
- activate any cutter compensation
- proceed through all events to be plotted.

EXECUTION - GRAPHICS MODE

In this mode, axis motion is inhibited and plotting rate is independent of the part program's feedrate.

The timer displays the estimated machining time assuming that your FEEDRATE OVERRIDE potentiometer is at 100%. The timer will not account for: the time it takes to complete a tool change, the time it takes for an M, S, and T code to execute, or the time it takes for an axis to accelerate and decelerate. You must dry run the part to get the part's absolute cycle time. The timer resets to zero when you press CLEAR.

Once the process is started, the plot is drawn from designated FIRST to LAST event. You need not direct or control the plotting operation once it has begun. Press CYCLE START to begin plotting.

You may halt at any time before the plot is completed by pressing MOTION STOP. The plotting will resume from the point of interruption when you press CYCLE START. You may also press CLEAR at any point along the way (plotting or halted) to erase the screen without losing your place in the plotting sequence. When halted, you may press the RESET key to reinitialize plotting to the FIRST event. If uninterrupted, the plotting will halt after the LAST event has been plotted.

After executing a plot, you may CLEAR and immediately repeat the plot or change a Page 1 entry. If you change from T1 to T2, or from T2 to T1, you can overlay a second plot on the first. This is often useful for relating tool motion to the profile.

Whenever a plot is halted, you may also move to Jog mode and select a constant jog rate. On return to Graphics mode you may now control the crosshair cursor with JOG keys, and further adjust the cursor's rate using the FEEDRATE OVERRIDE potentiometer. As you move the cursor, the X, Y, Z Position Displays will report the crosshair coordinates. No axis motion will occur. Use this utility to locate coordinates on the plot or locate a new plotting origin.

EXECUTION - AUTO MODE

When you view Graphics while cutting a part in Auto mode, axis motion is allowed (unless you have selected the Z or All Axis Inhibit with one of the TM entries in the Set Up mode) and plotting commences as you are executing the part program to machine a part. Auto mode execution also provides an actual "run time" estimate of machining time with recognition of FEEDRATE OVERRIDE control.

Assuming that you have completed the Page 1 entries in Graphic mode, you may now select the Auto mode. Note that the FIRST and LAST entries in Graphics mode will not affect execution in the Auto or Single Event modes.

Begin program execution by pressing RESET, then CYCLE START.

Plotting progresses whether or not you have selected to view the plot display screen. Press GRAPHICS DISPLAY to view the plot.

In this mode, plotting is secondary to part program execution. MOTION STOP will halt program execution as well as plotting. Pressing RESET when the control is not is cycle will reinitialize the part program to begin at the first event. Pressing clear will erase the plot. Since the graphics cursor tracks actual tool motion, the cursor "jog" utility is not available in this mode.

EXECUTION - SINGLE EVENT MODE

Plotting is similar to Auto mode execution with the exception that events are executed in sequence, one at a time. The timer reports the execution time for a single event.

GRAPHICS INFORMATION

The plot is composed of lines, arcs and circles that represent milling and drilling operations.

The following elements are shown in full intensity.

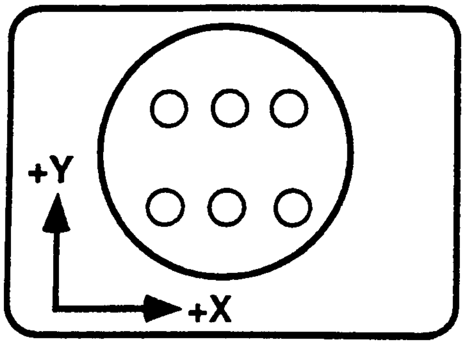
- axes vectors.
- finish pass, Frame and Circle Mill events.
- all moves at feedrate, except roughing or pocket passes of the Circle and Frame Mill event.
- hole centers, G cycles.
- hole circumference, G Cycle (if a T Code is active and a tool diameter is stored the Tool Tables mode when T2 is selected.

The following elements are shown in half intensity. You can choose whether to display these moves by pressing the VIEW ROUGH/REMOVE ROUGH soft key.

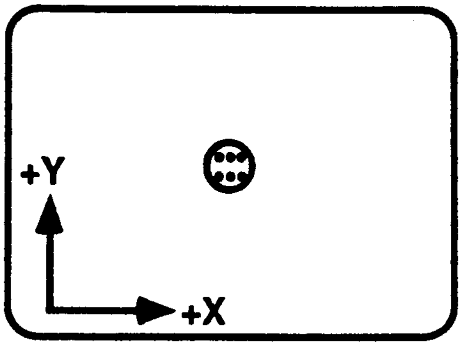
- all moves at rapid rate are shown as dotted lines.
- roughing or pocket passes of the Frame and Circle Mill events are shown as half intensity solid lines.

EXAMPLES TO HELP YOU ADJUST THE GRAPHICS DISPLAY

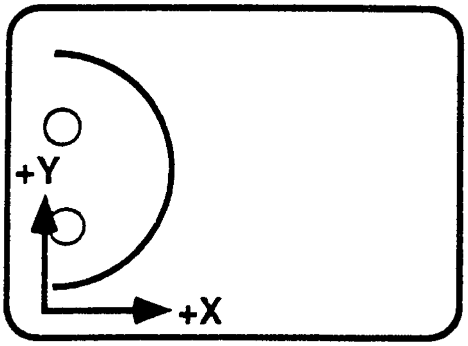
The following illustrations will help you learn how to adjust the Graphic Setup Values. The upper left illustration shows the correct Graphics Display for the part; the other five illustrations show incorrect Graphic displays, and provide corrective changes for the machine operator.



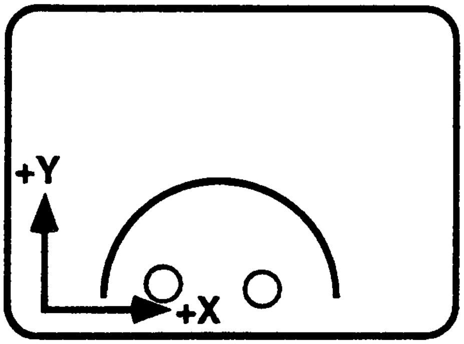
The correct display



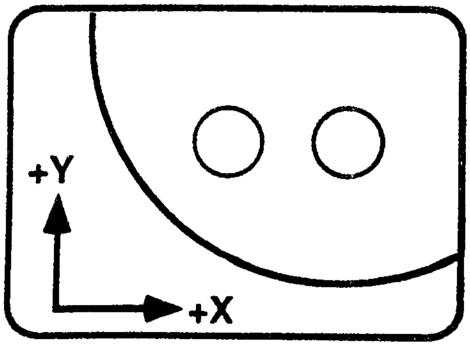
Problem: Part is too small.
Correction: Move the origin to the lower left corner of part and make the size larger.



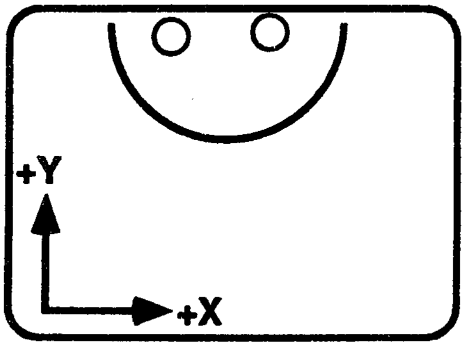
Problem: Part is too far left.
Correction: Make the X Origin a smaller number.



Problem: Part is too low.
Correction: Make the Y Origin a smaller number.



Problem: Part is too big.
Correction: Origin is OK.
Make Size smaller.

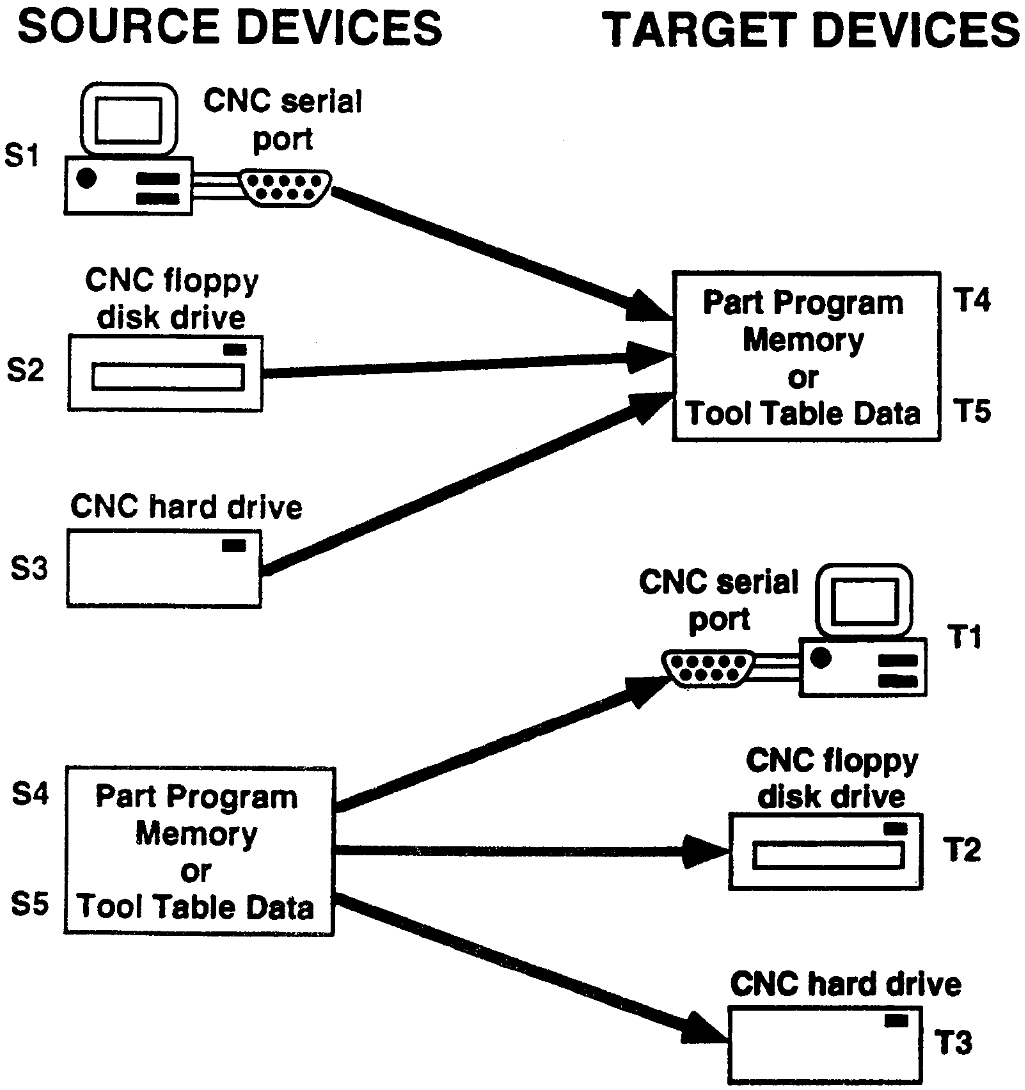


Problem: Part is too high.
Correction: Make the Y Origin a larger number.

INTRODUCTION

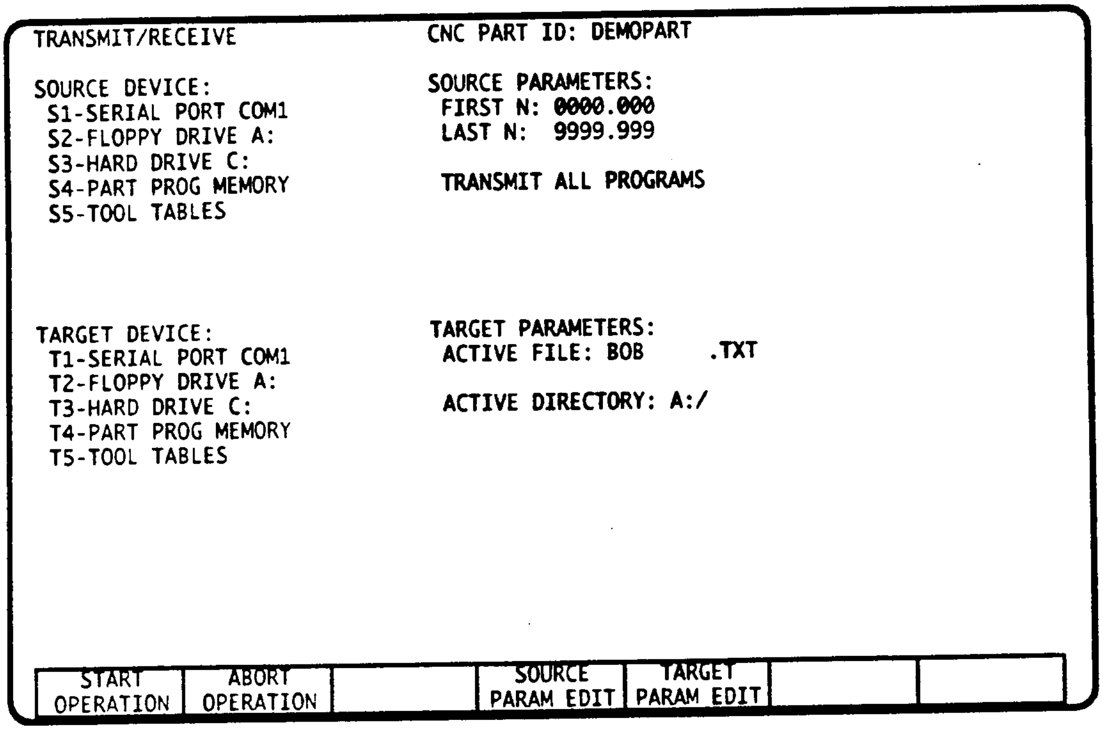
The Transmit/Receive mode is accessed as a primary or secondary mode. In general, this mode is used to transfer part programs or Tool Table data between the control and an external storage device. External storage devices include: the control's floppy disk drive[1], the control's hard disk[1] drive, or an external computer[2] which is connected to the control's serial port. All of these external devices can be used with the control's buffer input feature which allows the CNC to execute part programs whose size exceeds the control's available part program storage.

Each machine operator should back up their part programs on a regular basis.



[1] Although the floppy disk drive is cabled directly to the control, it is referred to as an external device because the control cannot edit a file that resides on the floppy disk. To edit a file on the disk, you must move it into the control's part program Catalog, make the edits on the CNC, then move the file back onto the floppy disk.

[2] Since the most common device that is used to communicate with our control is a personal computer, this manual will use the term "personal computer" to refer to any device that is capable of communicating with the control through its serial port.



```
TRANSMIT/RECEIVE                    CNC PART ID: DEMOPART

SOURCE DEVICE:                      SOURCE PARAMETERS:
 S1-SERIAL PORT COM1                 FIRST N: 0000.000
 S2-FLOPPY DRIVE A:                  LAST N:  9999.999
 S3-HARD DRIVE C:
 S4-PART PROG MEMORY                 TRANSMIT ALL PROGRAMS
 S5-TOOL TABLES



TARGET DEVICE:                      TARGET PARAMETERS:
 T1-SERIAL PORT COM1                 ACTIVE FILE: BOB     .TXT
 T2-FLOPPY DRIVE A:
 T3-HARD DRIVE C:                    ACTIVE DIRECTORY: A:/
 T4-PART PROG MEMORY
 T5-TOOL TABLES
```

OUTLINE FOR USING THE TRANSMIT/RECEIVE MODES

The general procedure for using the Transmit/Receive mode is described below. Instructions for using each type of device are provided later in this section.

Step 1 - Select a Source Device
The Source Device identifies the original location of the data you wish to transmit.

Step 2 - Edit the Source Device Parameters
The Source Device Parameters will vary with your Source Device selection in Step 1.

Step 3 - Select a Target Device
The Target Device identifies the destination of the file sent from the Source Device.

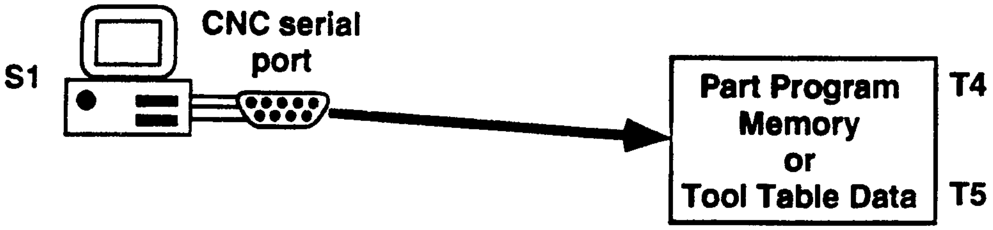
Step 4 - Edit the Source Device Parameters
The Source Device Parameters will vary with your Source Device selection in Step 1.

Step 5 - Begin the Transfer
Press the START OPERATION soft key to begin.

Computer to CNC Serial Port

to

Part Program Storage or Tool Tables



Selecting the Serial Port as the Source device indicates that you wish to move a computer file containing part programs or Tool Table data from your computer's serial port to the CNC's serial port. Your selection of the Target device specifies the type of data (part programs or Tool Table Data) you are transmitting.

The procedure below shows how to perform the transmission. Further information about working with these devices is provided later in this section.

Source Device

1) Select S1-Serial Port Com1

2) Move to your computer and select the filename you wish to transmit. Also select the communication parameters for the transfer.

3) Press the SOURCE EDIT PARAM soft key then select communication parameters which match the settings on your computer.

Read the Serial Port Information topic later in this section to learn about obtaining a communications cable and about obtaining communications software for your computer.

Target Device

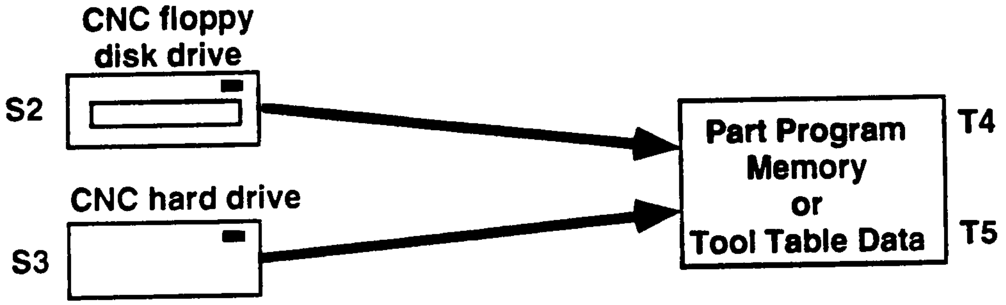
4) **Select T4 if you are transmitting part programs to the CNC; select T5 if you are transmitting Tool Table data to the CNC.**

5) **There are no Target parameters for the T4 or T5 target devices.**

6) **Press the START OPERATION soft key to put the control in 'receive" mode.**

7) **Move to your computer and start the transmission.** The CNC will scroll a list of incoming N numbers if T4 is selected; it will display a list of incoming Tool Table codes if T5 is selected. The message DONE will be displayed at the end of the transmission.

CNC Floppy Disk or Hard Drive
to
Part Program Storage or Tool Tables



Selecting the Floppy Drive or the Hard Drive as the Source device indicates that you wish to move a file containing part programs or Tool Table data into the control's memory. Your selection of the Target device specifies the type of data (part programs or Tool Table Data) you are transmitting.

The procedure below shows how to perform the transmission. Further information about working with these devices is provided later in this section.

Source Device

1) Select S2-Floppy Drive A: if you are transmitting from the control's floppy disk drive. Select S3-Hard Drive C: if you are transmitting from the control's hard disk drive.

2) Press the SOURCE EDIT PARAM soft key then select the filename you wish to transfer to the CNC. transfer. Read the Floppy or Hard Drive Information topic later in this section to learn about selecting a filename.

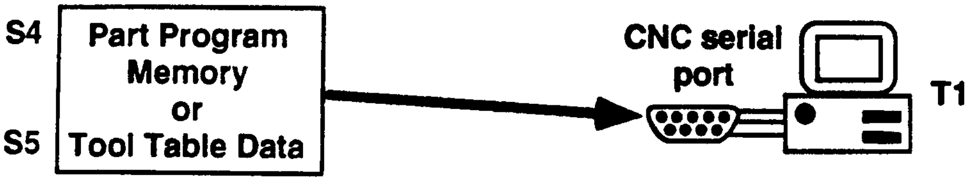
Target Device

3) Select T4 if you are transmitting part programs to the CNC; select T5 if you are transmitting Tool Table data to the CNC.

4) There are no Target parameters for the T4 or T5 target devices.

5) Press the START OPERATION soft key to start the transmission. The CNC will scroll a list of incoming N numbers if T4 is selected; it will display a list of incoming Tool Table codes if T5 is selected. A DONE message will be displayed at the end of the transmission.

Part Program Storage or Tool Tables to Serial Port Com1



Selecting Part Program Storage or Tool Tables as the Source device indicates that you wish to move the control's Part Programs or Tool Table data to the Target device. Selecting the Serial Port as the Target device indicates that you wish to send the data to your personal computer.

The procedure below shows how to perform the transmission. Further information about working with these devices is provided later in this section.

Source Device

1) Select S4 if you are transmitting part programs to the computer; select S5 if you are transmitting Tool Table data to the computer.

2) Press the SOURCE EDIT PARAM soft key then select the data you wish to transfer.

   If S4 is selected, choose whether you wish to transfer one or all part programs. Select the F through L range of N Numbers if you choose to transmit one program.

   If S5 is selected, choose which Tool Table(s) you wish to transmit.

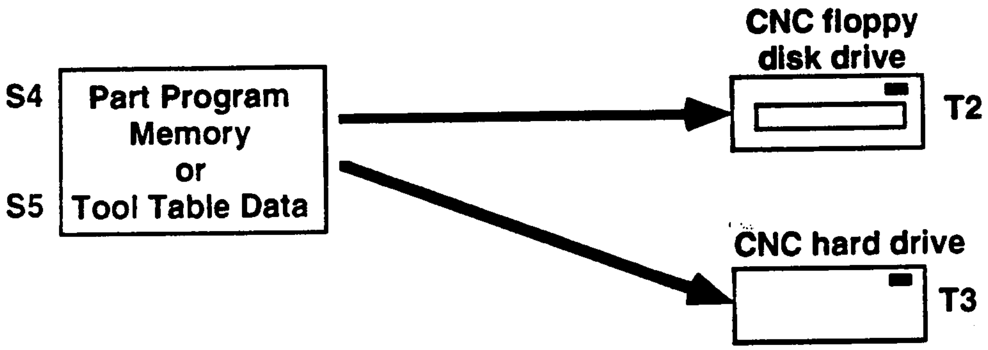
Target Device

3) Select T1 to select the serial port as the Target device.

4) Press the TARGET EDIT PARAM soft key then select communication parameters which match the parameters you have selected on your computer. Read the Serial Port Information topic later in this section to learn about obtaining a communications cable and about obtaining communications software for your computer.

5) Put your computer into "receive" mode, then press the START OPERATION soft key on the control. The CNC will scroll a list of incoming N numbers if T4 is selected; it will display a list of incoming T, H, D, or E codes. The message DONE will be displayed to signify the end of the transmission.

Part Program Storage or Tool Tables to CNC Floppy Disk or Hard Drive



Selecting the Part Program Storage or Tool Tables as the Source device indicates that you wish to move the control's part programs or Tool Table data to the selected device. Selecting T2 or T3 as the Target device indicates that you want to store the data on the control's floppy or hard disk drive.

The procedure below shows how to perform the transmission. Further information about working with these devices is provided later in this section.

Source Device

1) Select S4 if you are transmitting part programs to the computer; select S5 if you are transmitting Tool Table data to the computer.

2) Press the SOURCE EDIT PARAM soft key then select the data you wish to transfer.

If S4 is selected, choose whether you wish to transfer one or all part programs. Select the F through L range of N Numbers if you choose to transmit one program.

If S5 is selected, choose which Tool Table(s) you wish to transmit.

Target Device

3) Select T2 if you are transmitting to the control's floppy disk drive. Select T3 if you are transmitting to the control's hard disk drive.

4) Press the TARGET EDIT PARAM soft key then select the filename where the incoming data will be stored on the selected disk. Read the Floppy or Hard Drive Information topic later in this section to learn about selecting a filename.

5) Press the START OPERATION soft key to begin the transfer. The CNC will scroll a list of incoming N numbers if T4 is selected; it will display a list of incoming T, H, D, or E codes. The message done will be displayed to signify the end of the transmission.

SERIAL PORT INFORMATION

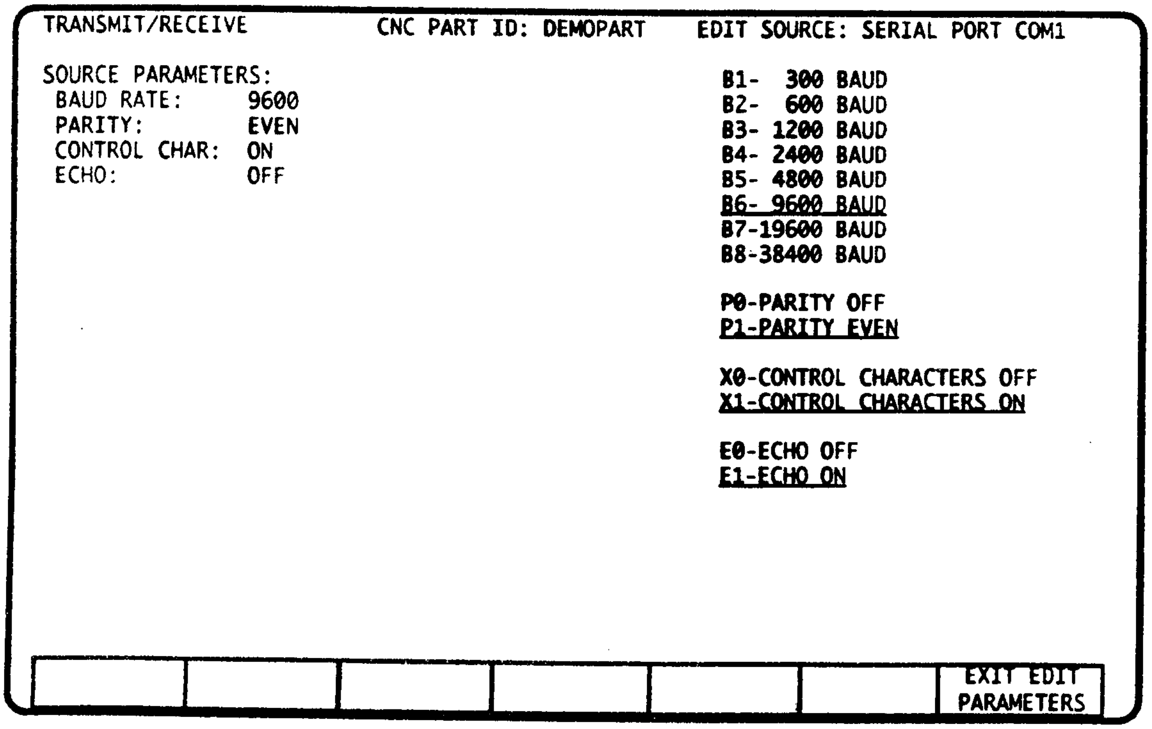
In order to use the control's serial port as the Source or Target device, you must first obtain the items listed below:

1) A cable which connects the control's serial port to your computer's serial port. Pinouts and other information that you will need to purchase or construct this cable is located in the Off-line Communications Appendix at the end of this manual. Since the CNC's serial port is PC compatible, off-the-shelf cables and adapters should be available from most computer supply companies.

2) A Communications software program for your PC. The communications program you select must be able to set the speed of the data transfer (baud rate), support even Parity transfer, and support the Xon/Xoff handshaking protocol. Autocon Technologies has developed and tested several Communications programs which are optimized for data transfer with DynaPath CNCs. Contact our Applications Engineering department for information about these products.

SERIAL PORT PARAMETERS

Prior to starting a transmission when Serial Port Com1 is selected as the Source or Target device, you must select communication parameters on the screen shown below. This screen is displayed when you press the SOURCE PARAM EDIT or TARGET PARAM EDIT soft key while Serial Port Com1 is selected. You must set each communication parameter to match similar settings in your computer's communications program. Recommended settings are shown below.



```
TRANSMIT/RECEIVE              CNC PART ID: DEMOPART    EDIT SOURCE: SERIAL PORT COM1

SOURCE PARAMETERS:                                       B1-   300 BAUD
 BAUD RATE:       9600                                   B2-   600 BAUD
 PARITY:          EVEN                                   B3-  1200 BAUD
 CONTROL CHAR:    ON                                     B4-  2400 BAUD
 ECHO:            OFF                                    B5-  4800 BAUD
                                                         B6-  9600 BAUD
                                                         B7-19600 BAUD
                                                         B8-38400 BAUD

                                                         P0-PARITY OFF
                                                         P1-PARITY EVEN

                                                         X0-CONTROL CHARACTERS OFF
                                                         X1-CONTROL CHARACTERS ON

                                                         E0-ECHO OFF
                                                         E1-ECHO ON
```

Baud rate - The baud rate specifies the speed of the data transfer. You should use the highest baud rate which is supported on both the CNC and your computer.

Control Characters On/Off - Xon/Xoff is a widely used protocol which allows the receiving device to command the sending device to temporarily halt the transmission. The ability to halt and resume the transmission is required when the receiving device must temporarily suspend the transmission while it stores data to one of its drives. It is also required to use the control's buffer input feature. Since devices such as tape readers do not support Xon/Xoff, you may need to turn Control Characters Off and implement this feature by adding additional hardware lines to your communications cable. Consult the Off-line Communications Appendix if your device is not capable of using Xon/Xoff.

Parity - Parity checking is an algorithm which will generate a communications error when the receiving device determines that the incoming data has been corrupted. You should always select Even parity on both the CNC and the computer.

Echo - This setting does not affect transmitted data. Changing the Echo setting will affect how the data appears on your computer's display screen. If double characters are displayed on the computer screen, change this setting to Echo Off, if no characters are displayed on the computer screen, change this entry to Echo On.

INFORMATION ABOUT USING THE FLOPPY or HARD DRIVE

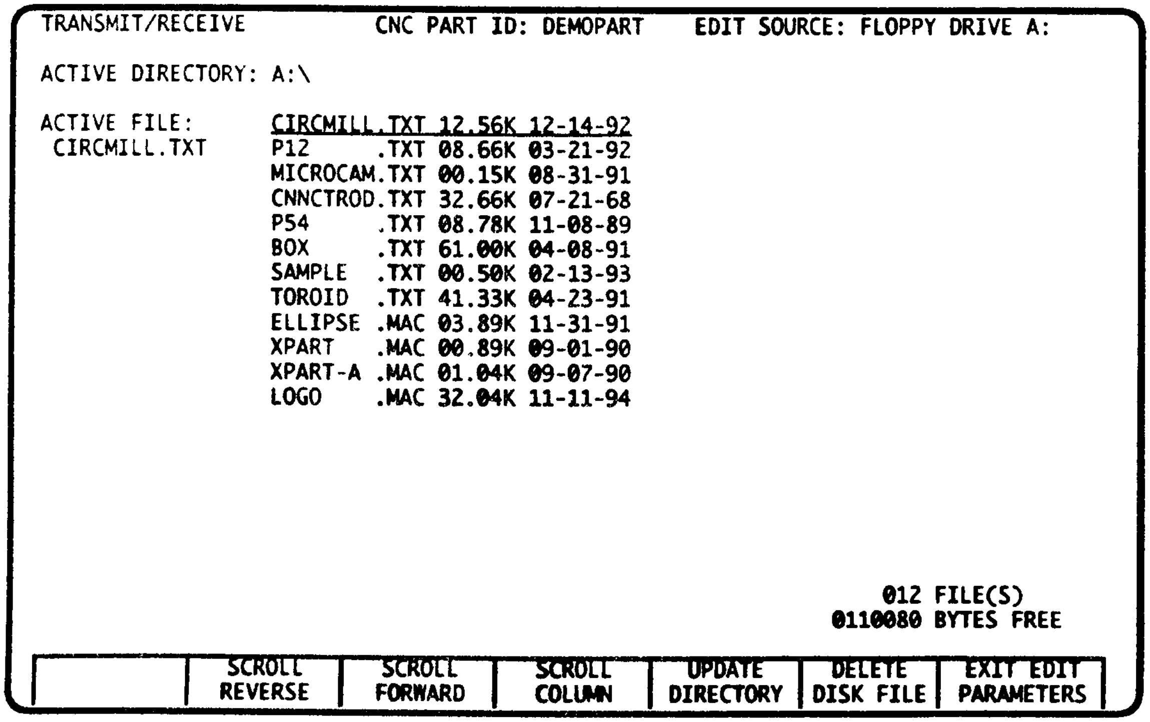
Selecting the floppy drive or hard drive as the Target or Source device requires that you select a filename on the PARAMETER EDIT screen shown below. This screen display a list of files which reside on the selected disk. You may now use one of the following procedures.

The CNC is capable of reading all files which reside in the floppy disk's root directory. Files which reside in subdirectories cannot be accessed by the CNC.

Select a filename using the scroll keys - if you wish select a filename which already exists on the diskette, use the three SCROLL soft keys to move the highlight bar over the desired file. Once the correct file is highlighted, press the ENTER key to complete the selection. The name of the currently selected file will be displayed on the left side of the screen.

Creating a new filename - You can create a new filename on a diskette by typing the desired name and pressing ENTER. The name you enter must be compatible with the filenaming scheme that is used for personal computers. Sample filenames are shown below.

Deleting a disk file - to delete a file, you must first make it the active file by highlighting it with the SCROLL keys and pressing ENTER. Once the filename is active, press the DELETE DISK FILE soft key. A message on the screen will list the filename which it will delete and request a confirmation from the operator. Press the Y key to confirm that you wish to delete the file; press the N key to cancel the delete.



```
TRANSMIT/RECEIVE           CNC PART ID: DEMOPART    EDIT SOURCE: FLOPPY DRIVE A:

ACTIVE DIRECTORY: A:\

ACTIVE FILE:        CIRCMILL.TXT 12.56K 12-14-92
 CIRCMILL.TXT       P12     .TXT 08.66K 03-21-92
                    MICROCAM.TXT 00.15K 08-31-91
                    CNNCTROD.TXT 32.66K 07-21-68
                    P54     .TXT 08.78K 11-08-89
                    BOX     .TXT 61.00K 04-08-91
                    SAMPLE  .TXT 00.50K 02-13-93
                    TOROID  .TXT 41.33K 04-23-91
                    ELLIPSE .MAC 03.89K 11-31-91
                    XPART   .MAC 00.89K 09-01-90
                    XPART-A .MAC 01.04K 09-07-90
                    LOGO    .MAC 32.04K 11-11-94

                                                                  012 FILE(S)
                                                               0110080 BYTES FREE
```

MANAGING PROGRAMS IN THE CNC MEMORY

This topic consists of a series of notes which describe how the control responds when T4-Part Program Memory is selected as the Target device.

- If you attempt to send a program that has the same part ID as a program in storage, **and the part ID is not selected in the Catalog mode**, a DUPLICATE ID error message will abort the transfer and all following data will be discarded.

- If you attempt to send a program that has the same part ID as a program in storage, **and you have preselected the same ID in the Catalog mode**, a program merge will occur. The result of a program merge is a part program which contains all events from the program on the external device, plus all events in the original program with an N Number which are not overwritten by the external program.

- If the control receives a program that does not begin with a part ID, it will automatically assign it the name P0000000. As an exception to this rule, a part program without a part ID will be stored in the currently selected part program ID if that program has a size of zero characters.

- If the transmission fails, the control will display an error message specifying the reason for the error along with the N Number and Part ID (part receive) or T, H, D, E, Q or L Code (tool receive) just prior to where the failure occurred. All data received prior to the error will be edited into the Catalog or Tool Tables; all data received after the error will be discarded by the control.

- Any operation may be canceled by pressing the ABORT OPERATION soft key. If the Target device is the part program memory or the Tool Tables, all data that was received prior to the abort will be edited into the Catalog or Tool Tables; all data received after the abort will be discarded. If the Target device is the floppy or hard disk drive, all data will be discarded.

- If you are using the Transmit/Receive mode as a Concurrent mode, you must select a dummy part ID such as NEW from the Concurrent Catalog before starting an operation. The dummy ID will be replaced by the actual program title once the transfer begins.

- The T5 - Tool Tables Target device is locked out when you are using the Transmit/Receive mode as a Concurrent mode.

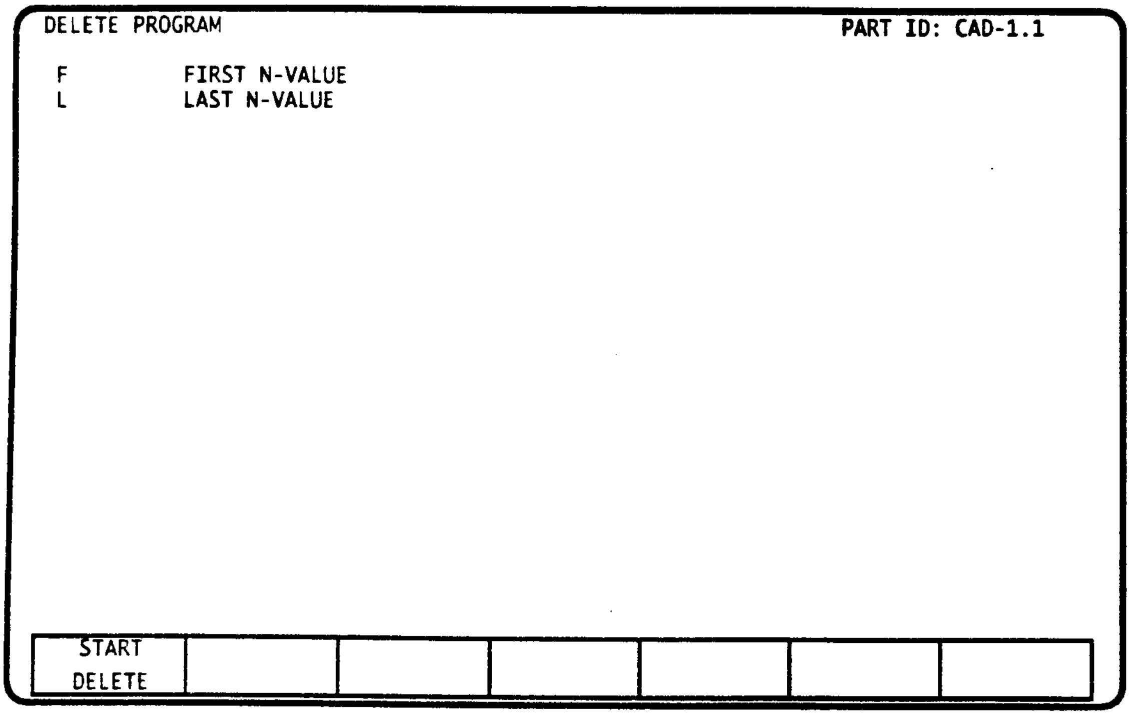
BUFFER INPUT FROM PERSONAL COMPUTER

To make use of the control's Buffer Input feature, your device must be able to halt and resume data transmission in response to ASCII control characters (Xon/Xoff) or control lines (Clear to Send/Request to Send signals).

Part programs of unlimited size may be executed with the Buffer Input feature. However, no Subroutine or Repeat events are allowed in the part program since the control discards each event that it successfully executes. Since Buffer Input executes each event in the order that it is received, N numbers are not required at the start of each event.

To use this feature select the part program ID **RS232C** in the Catalog mode[3], complete the Communications Settings described earlier. Move to Auto or Single Event mode and press CYCLE START to begin execution. You can now start sending part program data from your storage device.

[3] Caution: Some modes will be removed from the Mode Select menu when RS232C is the active part program ID. You must select another title from the Catalog to restore the missing modes.



```
DELETE PROGRAM                                          PART ID: CAD-1.1

F          FIRST N-VALUE
L          LAST N-VALUE
```

DELETE PROGRAM MODE - INTRODUCTION

Delete Program mode may be accessed as a primary or secondary mode. Choose this mode to delete one or more events from a part program.

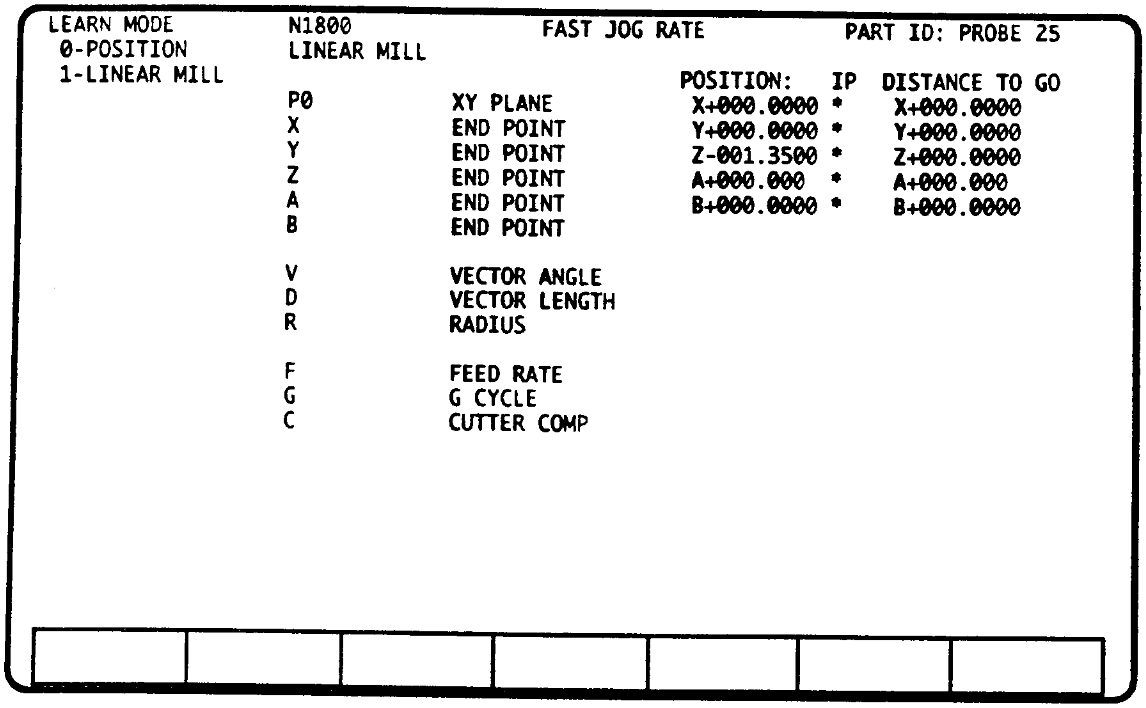
DISPLAY AND EXECUTION

A program must be selected in the Catalog mode before beginning operations in the Delete Program mode. On entry to this mode, the selected Part ID (program title) will appear on the display screen.

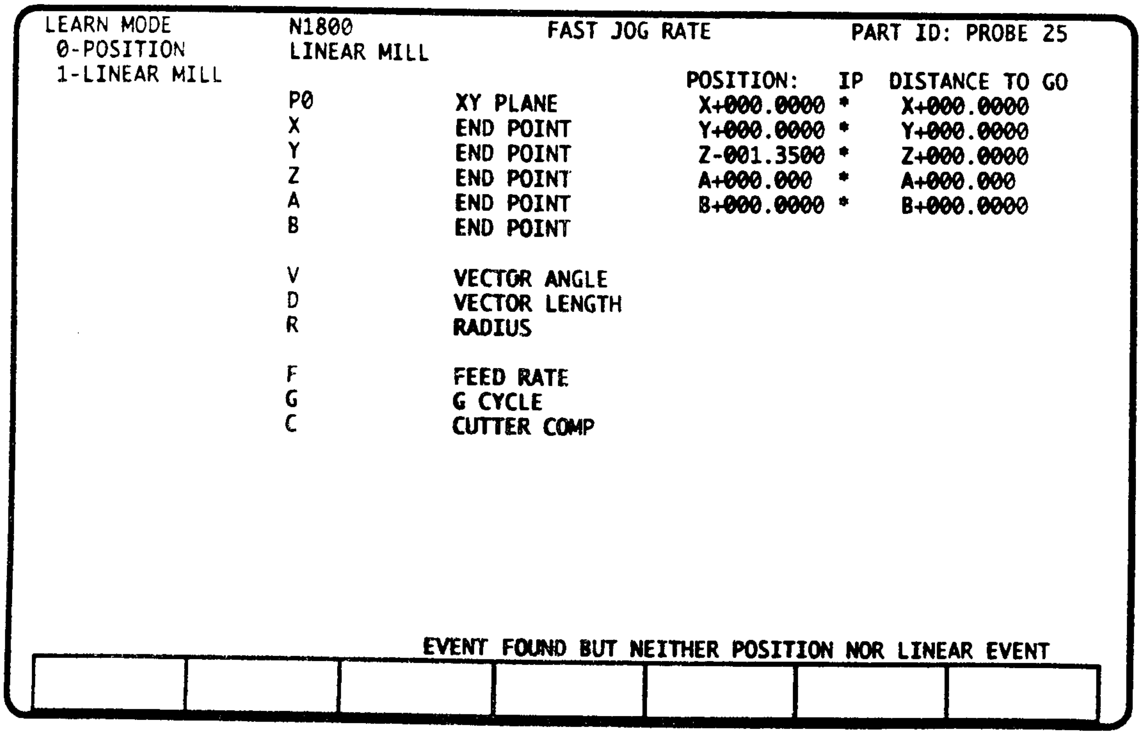
F selects the N Number of the first event that you wish to erase from program memory. L selects the N Number of the last event to be erased. L must be equal to or greater than F.

Once you have selected the range of events to be deleted, press the START DELETE soft key. A DONE message will appear indicating that all events from F to L have been erased from program storage.

Choose F and L numbers carefully. Once data is deleted in this way, it cannot be recovered.



```
LEARN MODE          N1800              FAST JOG RATE             PART ID: PROBE 25
 0-POSITION         LINEAR MILL
 1-LINEAR MILL                                        POSITION:   IP  DISTANCE TO GO
                    P0          XY PLANE               X+000.0000 *    X+000.0000
                    X           END POINT              Y+000.0000 *    Y+000.0000
                    Y           END POINT              Z-001.3500 *    Z+000.0000
                    Z           END POINT              A+000.000  *    A+000.000
                    A           END POINT              B+000.0000 *    B+000.0000
                    B           END POINT

                    V           VECTOR ANGLE
                    D           VECTOR LENGTH
                    R           RADIUS

                    F           FEED RATE
                    G           G CYCLE
                    C           CUTTER COMP
```



```
LEARN MODE          N1800              FAST JOG RATE             PART ID: PROBE 25
 0-POSITION         LINEAR MILL
 1-LINEAR MILL                                        POSITION:   IP  DISTANCE TO GO
                    P0          XY PLANE               X+000.0000 *    X+000.0000
                    X           END POINT              Y+000.0000 *    Y+000.0000
                    Y           END POINT              Z-001.3500 *    Z+000.0000
                    Z           END POINT              A+000.000  *    A+000.000
                    A           END POINT              B+000.0000 *    B+000.0000
                    B           END POINT

                    V           VECTOR ANGLE
                    D           VECTOR LENGTH
                    R           RADIUS

                    F           FEED RATE
                    G           G CYCLE
                    C           CUTTER COMP



                          EVENT FOUND BUT NEITHER POSITION NOR LINEAR EVENT
```

LEARN MODE - INTRODUCTION

Learn mode may be accessed as a primary mode. In Learn mode you can jog the tool tip to a location on an existing part and transfer the point's absolute coordinates directly into a part program Position or Linear Mill event.

DISPLAY AND EXECUTION

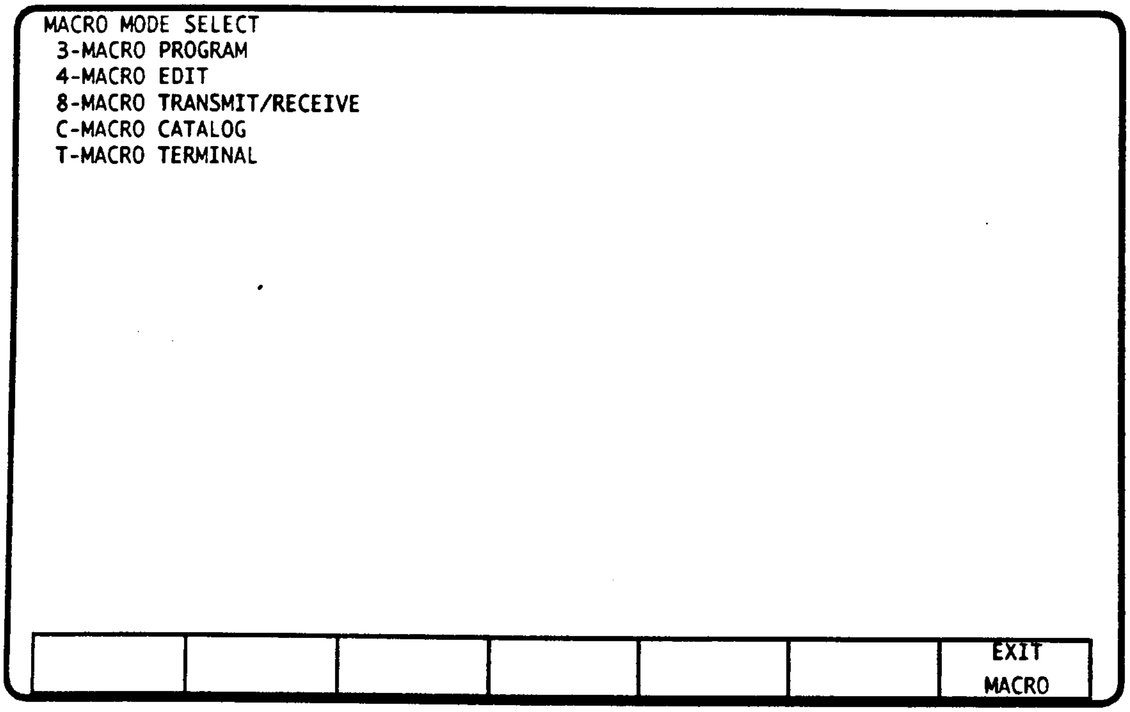
Once you have selected a program from the Catalog and selected this mode, the **upper screen** will be displayed.

You can jog the axes directly from the Learn mode. The JOG switches are enabled at the rate or increment distance that is selected in the Jog mode. The active jog rate or increment in displayed on the screen. If a NO RATE SELECTED message appears, select a rate in the Jog mode then return to the Learn mode.

To transfer axis coordinates into a part program, first jog the axes to the first point that you want to store. Next, enter the N Number for the event to be stored. A precautionary message, **lower screen,** will be displayed if an event type other that Position or Linear Mill has already been programed at the N Number that you entered. If this message does not appear, the event is vacant.

Next choose the Position or Linear Mill event type. To transfer coordinates into this event, key in the axis (X, Y, or Z) and press ENTER. The coordinate for the axis you entered will be transferred to the selected event. Repeat this sequence for each axis position you wish to record. You may also key in any event data (such as a feedrate) while you are in the Learn mode.

Repeat the process of jogging, selecting an N Number, event type, then storing the end point until the part is completely defined.



MACRO MODE - INTRODUCTION

The Macro mode is a special section of the control that is dedicated to creating, editing, and transmitting macros and Customer Prepared Cycles (CPCs). These are programs that reside in the Macro Catalog and can be called from any part program, or even another macro. The unique qualities of each type of program are:

Macro - Macros are identical to main Catalog part programs, except macros reside in the Macro Catalog.

Customer Prepared Cycle (CPC) - (sometimes referred to as an Advanced Macro or User Macro) - A CPC is an advanced form of a macro because it can accept variable data from the event that calls it into execution. This variable data to can represent part dimensions, feedrates, spindle speeds, cutting direction, etc. CPCs are programmed by combining arithmetic events[1] with normal part programming events[2] to build flexible, intelligent machining cycles.

To create, edit, or transmit a macro or CPC, you must first select the Macro mode, then choose the specific Macro mode that performs the desired task. To exit the Macro mode, press the EXIT MACRO soft key.

Each macro mode has an identical counterpart in the main Mode Select menu. To learn the basic operation of each macro mode, read the corresponding main mode section of this manual. This section only describes differences between the operation of the main and macro modes.

Information on calling a macro (or CPC) into execution is provided in the Macro Call Event section of this manual.

[1] Arithmetic evens can be used to do basic math (+, -, /, sin, cos, tan, sqrt, etc.), perform if/then branching, find active tool and fixture offsets, etc. as described in the CPC part of this manual.

[2] Normal programming events include all of the events that are available in the main Program mode. With CPCs, however, the data that programs the event may be variable information.

Information on the CPC programming language and strategies on how to write a CPC are described in the Developing a CPC part of this manual.

The Macro modes are:

3 - Macro Program mode
4 - Macro Edit mode
8 - Macro Transmit/Receive mode
C - Macro Catalog mode
T - Macro Terminal mode.

MACRO CATALOG MODE

CATALOG PARTITIONING

The control has a part program storage capacity of 180,000 characters, and 150 program titles. The number of characters and the number of program titles has been partitioned between the main and Macro Catalogs. Your machine supplier sets the size of the respective partitions.

MACRO CATALOG OPERATION

Catalog operations are similar to those in the main Catalog, but here you are performing file operations on macro programs.

Refer to the Catalog mode section of this manual for instruction on selecting, copying, renaming and deleting macro programs.

MACRO PROGRAM MODE

Macro Program mode is like main Program mode. Here you create macro programs using the same event types that are available to a main Catalog part program.

The general format for a macro program is identical to a main Catalog program. Each event or block must begin with an N Number, the event type, then the program data. An M02 or M30 End of Program M Code must be the last event in every macro program.

Refer to the Program mode section for instruction on creating part programs. Also note instructions on using the DELETE EVENT and COPY/STORE soft keys.

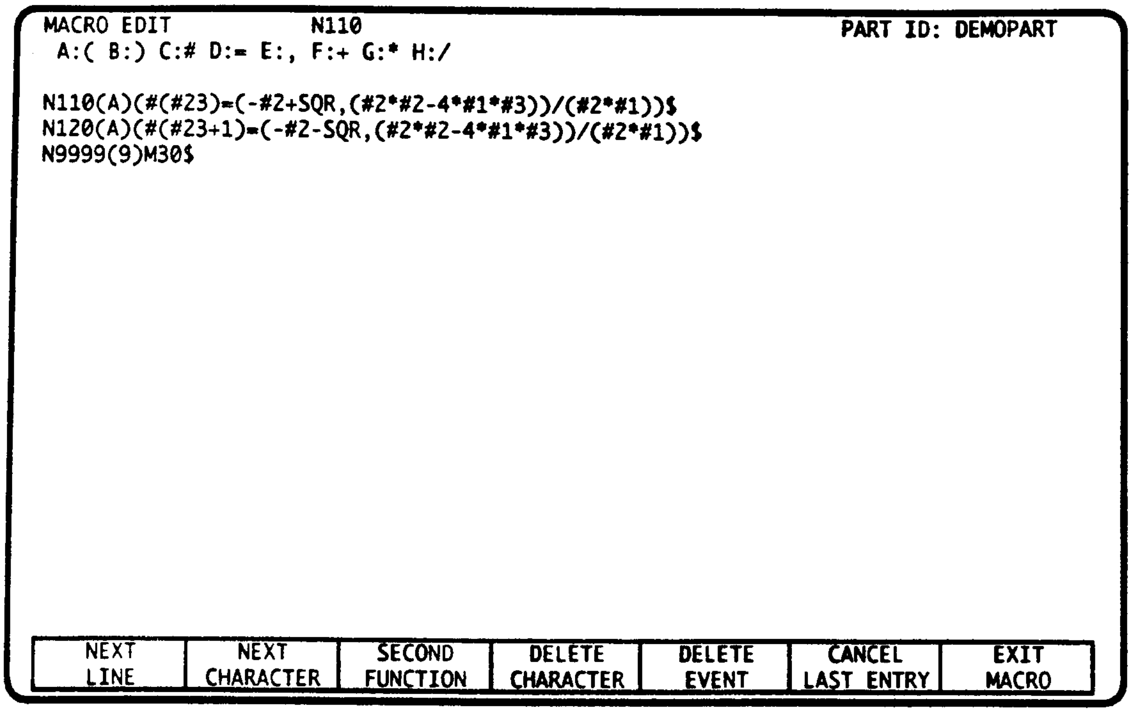
MACRO EDIT MODE

The Macro Edit mode allows Macros to be viewed, edited and created at the control[3]. Moving the cursor and editing program is described in the Edit mode section. However, the Macro Edit mode provides an additional soft key that allows you to enter 2nd function arithmetic characters (these characters are: ( ) # = , * /) from the control's keyboard.

[3] We recommend that you create macros in the Macro Program mode and use the Macro Editor mode to visually inspect your macros. When CPCs are installed, the Macro Edit provides a 2nd Function utility which is used to enter Arithmetic events from the keyboard.

2nd FUNCTION KEYS

2nd function characters are only used when editing a CPC. The 2nd function key descriptors appear on the second line of the Macro Edit display, below. Pressing the SECOND FUNCTION soft key highlights this line and activates the 2nd function keys. The line will stay highlighted and the keys will remain active until you press the SECOND FUNCTION soft key again. Entering any of the letters A-H while the 2nd function definitions are active will yield that key's 2nd function; the remaining keys, however, will be unaffected.

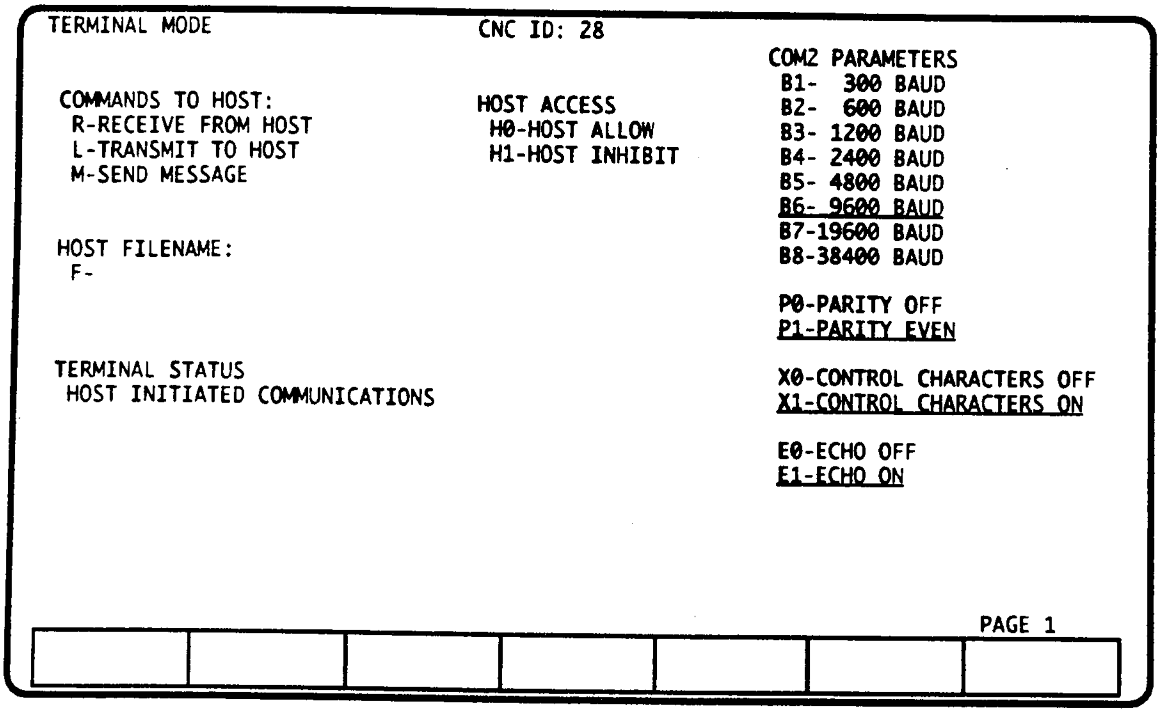


```
MACRO EDIT              N110                                      PART ID: DEMOPART
 A:( B:) C:# D:= E:, F:+ G:* H:/

N110(A)(#(#23)=(-#2+SQR,(#2*#2-4*#1*#3))/(#2*#1))$
N120(A)(#(#23+1)=(-#2-SQR,(#2*#2-4*#1*#3))/(#2*#1))$
N9999(9)M30$
```

MACRO TRANSMIT/RECEIVE MODE

Macro Transmit/Receive mode allows you to transmit and receive macros that you have created or stored on an off-line device. Your off-line device can be a serial communications device (such as a personal computer) or the control's floppy disk drive or hard disk drive.

Before receiving or transmitting a macro program, you must complete the communications parameters described in the Transmit/Receive mode section of this manual.



```
TERMINAL MODE                        CNC ID: 28
                                                          COM2 PARAMETERS
                                                           B1-  300 BAUD
 COMMANDS TO HOST:                   HOST ACCESS           B2-  600 BAUD
  R-RECEIVE FROM HOST                 H0-HOST ALLOW        B3- 1200 BAUD
  L-TRANSMIT TO HOST                  H1-HOST INHIBIT      B4- 2400 BAUD
  M-SEND MESSAGE                                           B5- 4800 BAUD
                                                           B6- 9600 BAUD
                                                           B7-19600 BAUD
 HOST FILENAME:                                            B8-38400 BAUD
  F-
                                                           P0-PARITY OFF
                                                           P1-PARITY EVEN

 TERMINAL STATUS                                           X0-CONTROL CHARACTERS OFF
  HOST INITIATED COMMUNICATIONS                            X1-CONTROL CHARACTERS ON

                                                           E0-ECHO OFF
                                                           E1-ECHO ON

                                                                    PAGE 1
```

To use the Terminal mode feature, you must provide a host computer that runs specially programmed third party software that handles DNC communications with the DynaPath control. Since DNC operations are customized for every application, you must refer to the DNC program's documentation for information on how to utilize this feature from your host computer.

Programmers who wish to develop DNC software for the DynaPath control should refer to the Off-line Communications Appendix in this manual.

TERMINAL MODE - INTRODUCTION

Terminal mode is an optional operating mode which allows bidirectional data transfer between the control and a host computer. Host-initiated transfer can be unattended on the CNC side. The data may consist of part programs, tool data, messages or file names. Up to 99 controls may be attached to one host computer. Terminal Mode will communicate with the host computer through the control's secondary serial port; its primary serial port remains free for standard Transmit and Receive mode operations.

TERMINAL MODE COMMUNICATIONS

Before communications can be established by either host or CNC, you must select the communications settings (baud rate, parity[1], etc.) for Terminal mode communications. These settings must match the communication parameters which are selected on the host computer. The Transmit/Receive mode section of this manual describes communication parameters.

[1] As a note to developers of off-line software: Parity affects part program and tool table data transfer only. Parity checking is not in affect during any ACK/NAK handshaking communications. For further information, refer to the Off-line Communications Appendix.

ENTERING TERMINAL MODE

Press T at the Mode Select screen to select the Terminal mode. The control can also switch automatically to the Terminal mode when the host initiates communications. See next topic.

AUTOMATIC ENTRY TO TERMINAL MODE

A control will be forced to enter Terminal mode when the host computer initiates communications unless any of the following host-lock-out conditions exist.

1) The control is in Program mode. In this case, the control will ignore the host's attempt to begin communication; the operator would be unaware of the host's attempt to communicate.
2) The control is in Graphics mode and is actively plotting a part when the host attempts to initiate communications. Otherwise, the control will switch to Terminal mode on host initiation.
3) The control is in Jog mode and the axes are being jogged. If the axes are stationary in Jog mode the control will switch to Terminal mode.
4) The control is in Transmit or Receive mode and a program is being transmitted through the primary serial port. If the control is in Transmit or Receive mode, but no transfer is taking place, the host can access.
5) The control is cycling in Auto or Single Event mode[2].
6) The operator sets a host access inhibit, as explained later in this section.
7) The control is off, or not cabled to the host.

Terminal mode screen, **facing page**, appears when the operator chooses Terminal mode or when communications are initiated and established by the host. If the control is running concurrently in the Auto or Single Event modes, an EXIT CONCURRENT soft key will be present; for host-initiated communications, an ABORT COMMUNICATIONS soft key be present. The CNC ID number (top, center of the Terminal mode screen) gives the host computer an "address" for each of the up-to-99 controls it can communicate with. The CNC ID number is set by your machine installer.

The LEFT ARROW key allows you to access the Terminal mode's Communications Log screen. This screen displays messages that describe recent communications activity, such as who initiated and terminated or aborted communications, what command was executed and any error that occurred if the CNC aborted communications.

TERMINAL MODE - CNC INITIATED OPTIONS

The following operations can be initiated by the CNC operator. These options allow the CNC to request part program or Tool Table data, transmit data to the host, or send a message to the host.

HOST FILENAME:

The HOST FILENAME lets you identify a file on the host computer. The HOST FILENAME may represent a file where the host computer will store incoming data, or where the host may read data that it will send to the CNC. The name you enter here does not relate to part program IDs in the Catalog mode.

To enter the filename, press F, then type in the filename[3] (up to 32 characters), then press ENTER. You can clear a filename and deselect the F option at any time by pressing F CLEAR.

[2] To allow host initiation while the control is in Auto or Single Event modes, **but not in cycle**, you must press the CONCURRENT PROGRAM soft key, enter Catalog mode and select a different program name.

[3] The filename you enter must conform to the filenaming convention used by the host computer. For example, many personal computers only allow filenames in the xxxxxxxx.xxx format.

R-RECEIVE FROM HOST

Pressing R ENTER allows the control to receive the contents of the specified filename from the host. When the R-RECEIVE FROM HOST option is highlighted, you can initiate the transfer by pressing one of the RECEIVE soft keys described below.

Pressing the RECEIVE PROGRAM soft key signals the host to begin transmitting the program (or programs) contained in the specified filename to the CNC. If no filename is present, what gets sent is determined by the host. Refer to your DNC software's documentation.

Pressing the RECEIVE TOOLS soft key causes the host to transmit Tool Table data into the CNC's Tool Tables. This will destroy the previous contents of the CNC's Tool Tables. If the data contained in the host filename does not contain tool table data you will get an OPERATION ERROR. If no filename is specified, what happens is determined by the host. Refer to your DNC software's documentation. The RECEIVE TOOLS soft key is not available from concurrent Auto or Single Event modes.

Pressing the ABORT RECEIVE soft key signals the host to prematurely terminate the transmission.

Pressing CLEAR will deselect the R option.

T-TRANSMIT TO HOST

Pressing T ENTER allows you to transmit the active part program, all programs in the Catalog, or the Tool Tables to the host. When the TRANSMIT TO HOST option is selected, the transmit soft keys described below will be displayed. If you have entered a host filename, the host will store the data you send in a file with the name specified. If you have not specified a host filename, where the data gets stored is determined by the host; refer to your DNC software's documentation.

Pressing the TRANSMIT PROGRAM soft key begins sending the active part program (the part program ID that is selected in the Catalog mode) to the host.

Pressing the XMIT ALL PROGRAMS soft key transmits all the programs in the CNC Catalog to the host.

Pressing the TRANSMIT TOOLS soft key will transmit the CNC's Tool Tables to the host.

Pressing the ABORT RECORD soft key terminates the transmission prematurely.

Pressing CLEAR will deselect the TRANSMIT option.

M-SEND MESSAGE

Press M-SEND MESSAGE to send a message (up to 32 characters long) to the host. As you enter the message, it appears on the lower left part of the screen immediately preceded by an M. Pressing the ENTER key sends the message to the host. The messages will then disappear from the Terminal mode screen (although it will be shown on the Communications Log screen). Since sending the message requires the CNC to establish and then terminate communications with the host (it happens very quickly), generally the message COMMUNICATIONS TERMINATED will appear below TERMINAL STATUS as soon as the enter key is pushed, provided no error occurs.

Messages can be removed by pressing CLEAR or by selecting another function .

TERMINAL MODE - HOST INITIATED OPTIONS

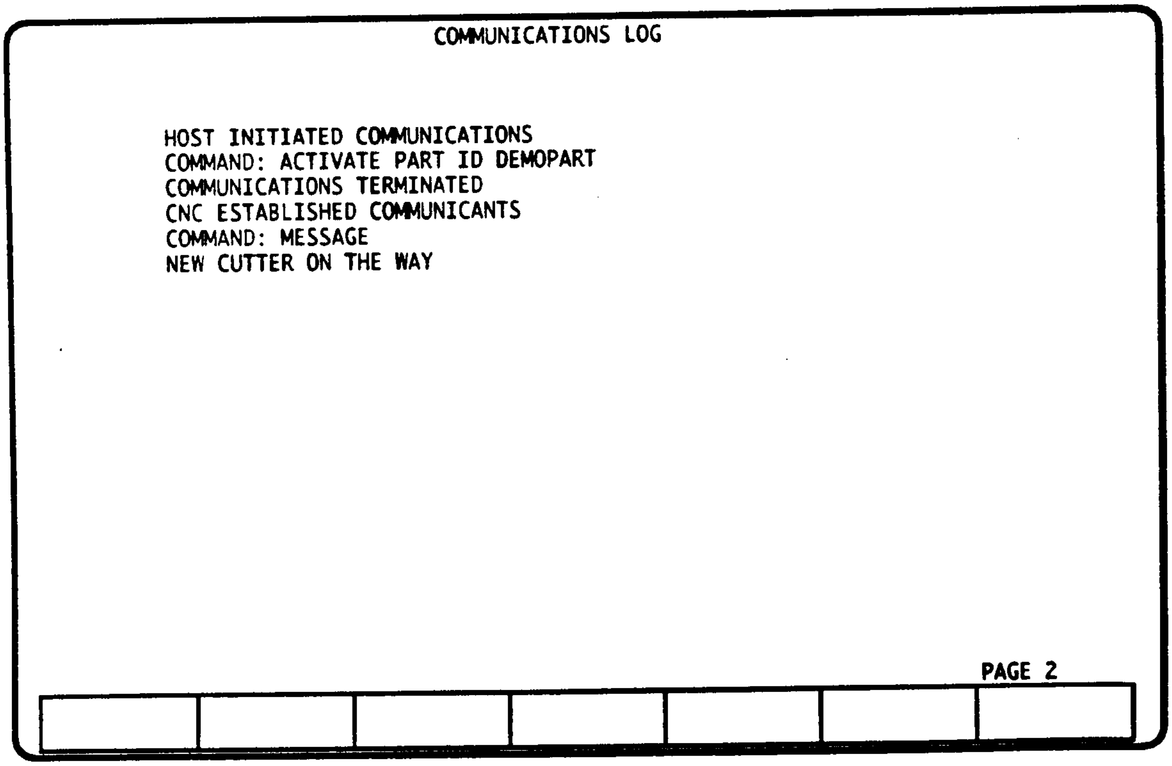
While you are operating the control, the host computer may instigate any of the following activities. Since DNC operations are customized for every application, you must refer to your DNC software's documentation for information on how to use these options from your host computer.

1) send one or more part programs or Tool Table data to the CNC.
2) receive one or more part programs or Tool Table data from the CNC.
3) activate a part program ID in the CNC's Catalog mode.
4) delete a part program ID in the CNC's Catalog mode.
5) send a message to the CNC.

The operator need not be present during any of these operations. All of these operations can be completed without operator intervention as long as the the control is not in one of the Lock-Out conditions described earlier. However, the operator can abort any operation by pressing the ABORT COMM soft key.

TERMINAL MODE STATUS

Press the LEFT ARROW key to view the Communications Log screen, **below**. This screen provides a log of the most recent communications which have occurred through the Terminal mode's communications port. Messages that may appear are listed below.



COMMUNICATIONS LOG

```
HOST INITIATED COMMUNICATIONS
COMMAND: ACTIVATE PART ID DEMOPART
COMMUNICATIONS TERMINATED
CNC ESTABLISHED COMMUNICANTS
COMMAND: MESSAGE
NEW CUTTER ON THE WAY
```

PAGE 2

Message	Comments
CNC ESTABLISHED COMMUNICATIONS	Link by CNC request
HOST ESTABLISHED COMMUNICATIONS	Link by host request
CANNOT ESTABLISH COMMUNICATIONS	No link
COMMUNICATIONS ABORTED	Xmit, Receive or Message aborted by CNC
COMMUNICATIONS TERMINATED	Xmit, Receive or Message successfully completed
HOST ABORTED COMMUNICATIONS	Xmit, Receive or Message aborted by host
DATA ERROR	Xmit/Receive error
OPERATION ERROR	Xmit/Receive error
RS232 RATE NOT SELECTED	No transmission allowed until RS232 rate selected in Xmit or Receive mode
PRT INHIBIT	Serial port in use by Macro; not available for Terminal mode
RESERVED PART ID FAULT	RS232C, P0000000 or P-1 part name selected
CONCURRENT DELETE NOT ALLOWED	Host tried to delete with control in concurrent mode
CONCURRENT RECEIVE TOOLS NOT ALLOWED	Host tried to load tools with control in concurrent mode

HOST ACCESS

This feature allows the operator to control whether or not the host can initiate communications. H0 allows the host to initiate communication. H1 prohibits the host from initiating communications. The control reverts to host allow (H0) each time the control is turned on.

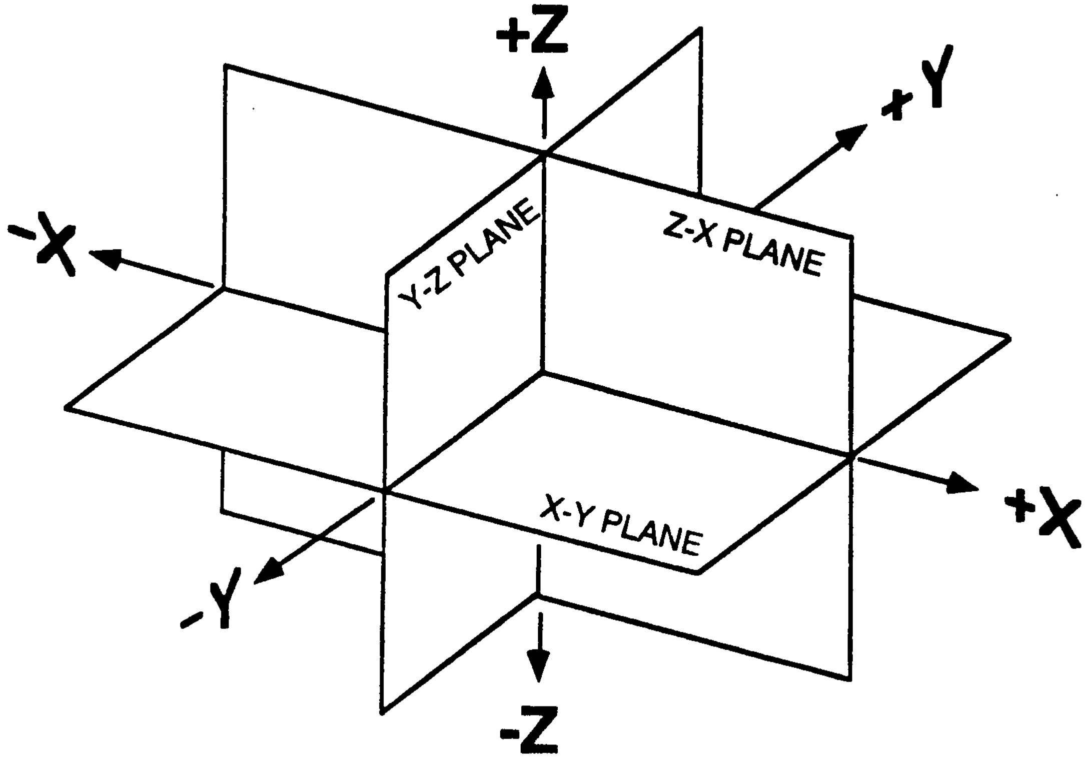
COMMUNICATION PROTOCOL

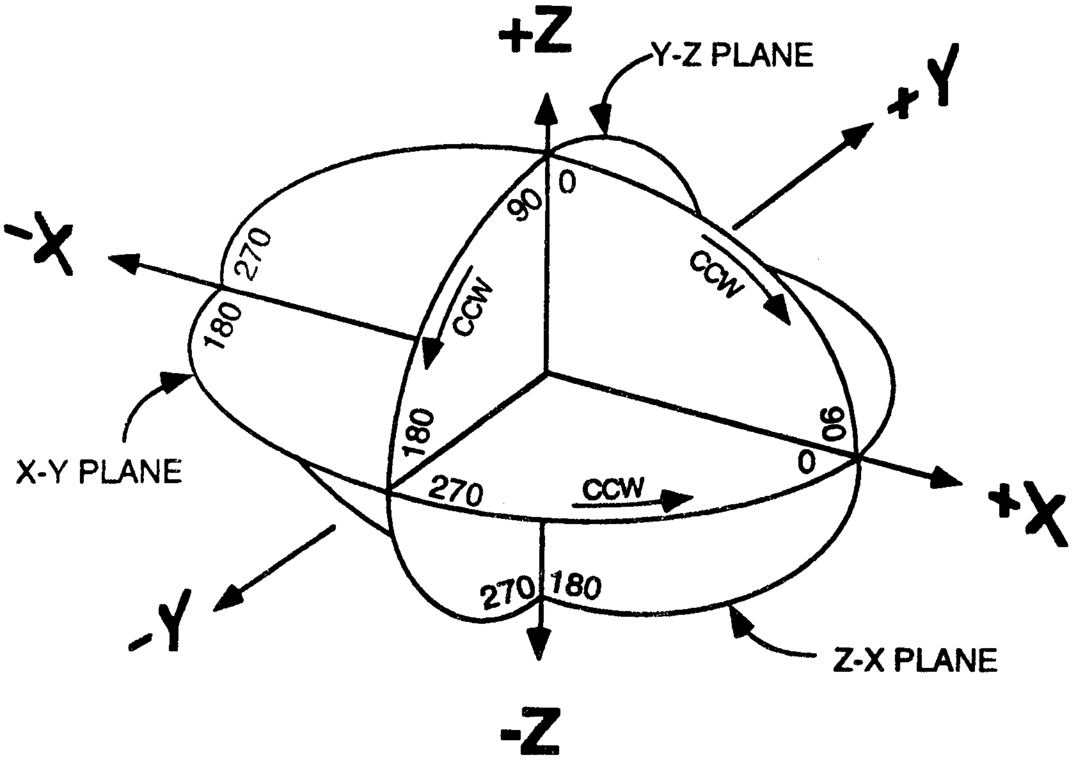
The Terminal mode uses a standard handshaking protocol called ACK/NAK. If you are developing a DNC program to utilize the control's Terminal mode, refer to the Off-line Communications Appendix for a detailed explanation of our ACK/NAK handshaking link.

PROGRAMMING INTRODUCTION

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON


+Z
+Y
-X
Y-Z PLANE
Z-X PLANE
+X
X-Y PLANE
-Y
-Z


+Z
Y-Z PLANE
+Y
-X
0
90
270
CCW
CCW
180
X-Y PLANE
180
90
0
270
CCW
+X
-Y
270
180
Z-X PLANE
-Z

PROGRAMMING - INTRODUCTION

Delta Series controls can be programmed using DynaPath's menu-driven Conversational programming events or in accordance with EIA RS-274/ISO 1056 format. Both formats may appear in the same program.

DynaPath's Conversational format offers canned cycles such as cavity milling, frame and circle milling and bolt circles to reduce part program size. Further, programming utilities such as rotates, repeats, subroutines, and macros can be programmed in Conversational events.

PROGRAM ID

Before programming, you must first select a part ID from the Catalog mode.

N NUMBER, EVENT TYPE, and EVENT DATA

You must begin each event or block in a part program with an N Number to identify the event's order of execution. After entering an N Number, press the EVENT TYPE key to view the full menu of events. Select the type of Conversational event[1] that you wish to program by keying in the event's identifying character (the identifying character is displayed on the monitor), then enter all appropriate data for the event. It is not necessary to enter leading and trailing zeros. After completing an event or block, select an new N Number and repeat this process. Refer to the Program mode section for more descriptive information. Program examples appear throughout Part 4 of this manual.

CARTESIAN COORDINATE SYSTEM

All motion events may be programmed in the XYZ Cartesian Coordinate system. Your machine supplier designates axes orientations for your machine. Axes dimensions are interpreted in inch or millimeter units depending on your selection in the Set Up mode.

Cartesian coordinates may be programmed as absolute or incremental dimensions. Absolute and incremental dimensions may be programmed in the same event.

ABSOLUTE DIMENSIONS

An absolute dimension locates a point by its signed X, Y, Z distance from Program Zero.

INCREMENTAL DIMENSIONS

An incremental dimension locates a point by its signed X, Y, Z distance from the last tool tip position. In Conversational events, incremental entries are programmed with the INCR key. When you press the incr key, a slash (/) character will be displayed after the number's last digit. A Z 2.3 INCR entry, for example, is displayed as Z2.3/ in a part program.

In EIA/ISO format, absolute dimensions are programmed with a G90 block, and incremental dimensions are programmed with a G91 block.

[1] An EIA/ISO block is Event Type E.

POLAR COORDINATE SYSTEM

Endpoints in Conversational Position, Linear Mill, Arc Mill and Repeat events may alternately be programmed in polar coordinates.

In polar events, motion is limited to a single horizontal or vertical plane. The plane is selected by entering:

P0 - XY plane (+X is the primary axis[2])
P1 - YZ plane (+Y is the primary axis[2])
P2 - ZX plane (+Z is the primary axis[2])

Within the selected plane, linear event endpoints are located by the distance D and angle V from the tool's start point. Angle V opens counterclockwise from zero degrees reference[3] about the tool's start point.

Polar Arc Mill events program the arc radius R, the angle A from the arc center to the tool's start position, and the angle B from arc center to the arc's endpoint Angles (A and B) open counterclockwise from zero degrees reference[4] about the arc's center.

CHOOSING PROGRAM FORMAT

Continue with Part 4 if you wish to review Conversational Programming. Skip to Part 5 if you prefer to use EIA RS-274/ISO 1056 program code.

MACHINE SUPPLIER OPTIONS

There are several settings that have been made by your machine supplier which affect the control's programming systems. It is very important that you read Part 1- Section 2 of this manual to learn about these settings. The major items that affect the control's programming system are:

- H and D code offsets to program the tool's length and diameter.
- auxiliary axis that is named C, V, or U.
- custom G codes in the EIA/ISO programming system.

As described in Part 1-Section 2, you can determine the state of these settings by pressing the FAULT/STATUS key then the SYSTEM SETTINGS soft key. Any item highlighted on this System Settings screen is enabled on your control.

[2] The first letter in each plane identifies the primary axis. The positive direction of the primary axis is often referred to as the zero degree reference axis. This is because the zero degree (0°) direction in the specified plane always runs parallel to, and in the same direction as the positive, primary axis.

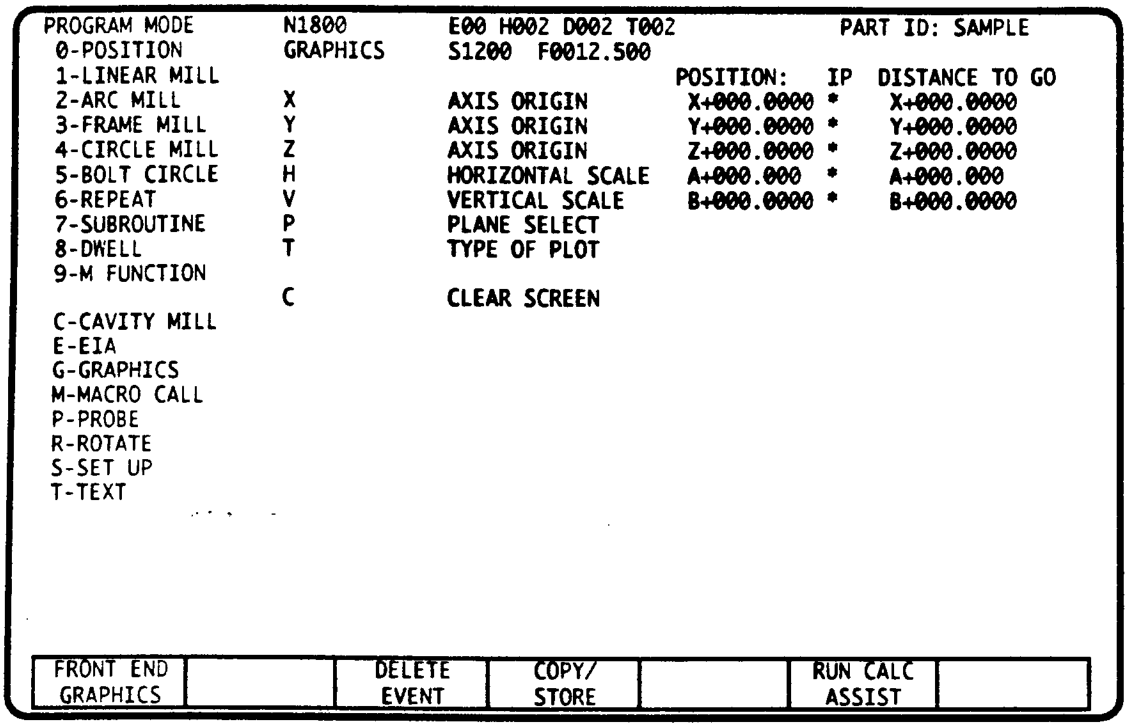
[3] The zero degree reference line starts at the tool's start point and runs parallel to the primary axis in the positive direction.

[4] The zero degree reference line starts at the arc center and runs parallel to the primary axis in the positive direction.

CONVERSATIONAL PROGRAMMING

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON



```
PROGRAM MODE       N1800      E00 H002 D002 T002          PART ID: SAMPLE
 0-POSITION        GRAPHICS   S1200  F0012.500
 1-LINEAR MILL                                 POSITION:   IP  DISTANCE TO GO
 2-ARC MILL        X          AXIS ORIGIN       X+000.0000 *    X+000.0000
 3-FRAME MILL      Y          AXIS ORIGIN       Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL     Z          AXIS ORIGIN       Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE     H          HORIZONTAL SCALE  A+000.000  *    A+000.000
 6-REPEAT          V          VERTICAL SCALE    B+000.0000 *    B+000.0000
 7-SUBROUTINE      P          PLANE SELECT
 8-DWELL           T          TYPE OF PLOT
 9-M FUNCTION
                   C          CLEAR SCREEN
 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```

GRAPHICS EVENT - INTRODUCTION

This utility event allows you to program a part's Graphics Setup information at the start of your part programs. Execution of this event causes the control to automatically size the Graphics screen as specified by the values that you program.

Execution of this event also commands the control to store your programmed values as modal settings in the Graphics mode. These values will be retained by the control until you change them with another Graphics event or you manually change the values in the Graphics mode.

Examples showing how to determine the Graphic setup values are located in the Graphics Mode section of this manual.

GRAPHICS EVENT DATA

Parameter	Description	Entry
N	Sequence number	Required
X	Axis Origin	Optional
Y	Axis Origin	Optional
Z	Axis Origin	Optional
H or V	Size of Graphics Window -do not program H and V in same event	Optional
T	Type of Plot (T1=Tool Center, T2=Part Profile)	Optional
P	Plane Select -P0 - XY plane -P1 - XZ plane -P2 - YZ plane -P3 - Isometric -P4 - XY and XZ planes -P5 - XY and YZ planes -P6 - Blueprint	Optional
C	Clear Screen (C0=do not clear, C1=clear screen)	Optional

P - PLANE SELECT

Choose an entry in the range of P0 - P6 to select the planes in which to plot tool motion.

X, Y, Z - PLOT ORIGIN

The plot origin defines the point, in **part** coordinates, which represents the lower left corner of the display screen. For each axis to be displayed, enter a number that is approximately 1 inch smaller than the most negative tool position that you wish to plot.

H or V - SIZE

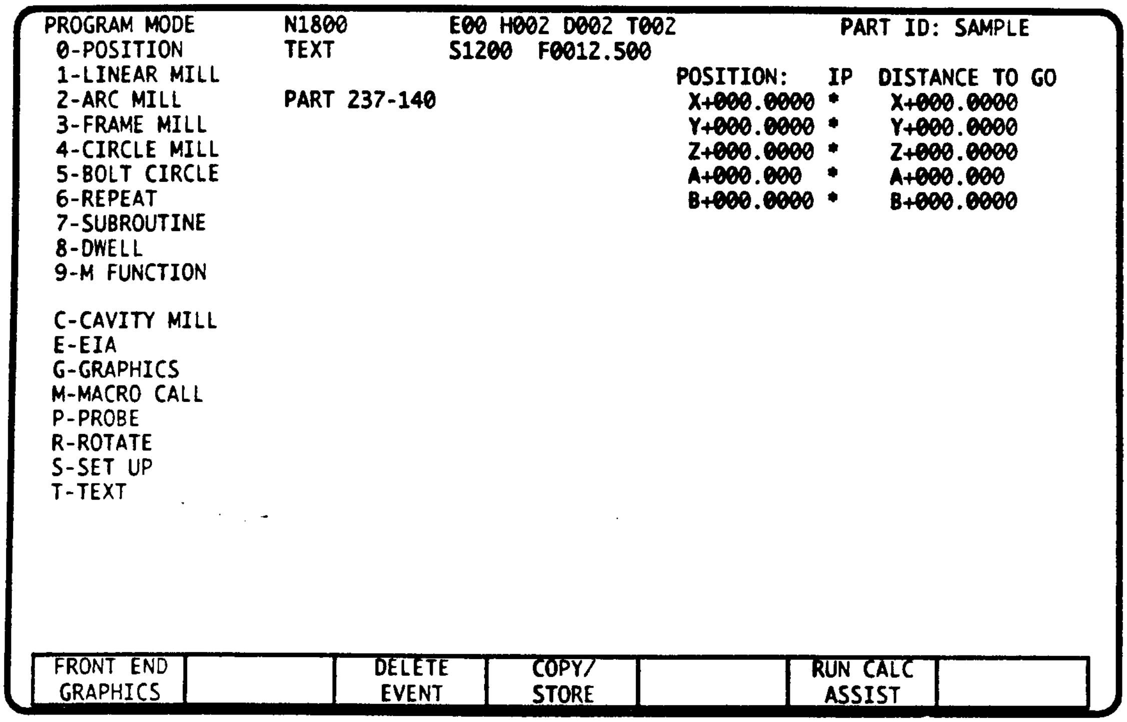
The SIZE entry determines how many inches or millimeters of axes travel will be shown on the display screen. Enter H to specify the number of inches or millimeters to display along the horizontal axes. Enter V to specify the number of inches or millimeters to display along the vertical axis. Only program one of these entries: the control automatically sizes the other entry.

T - TYPE of PLOT

The Type of Plot entry chooses whether the Graphics mode plots the tool center path or the part profile path. This entry is only used when your are plotting in the Graphics mode. The Auto mode always plots the position of the tool center.

C - CLEAR SCREEN

Programming C1 in this event will cause the control to clear the Graphics screen.



```
PROGRAM MODE        N1800         E00 H002 D002 T002            PART ID: SAMPLE
 0-POSITION         TEXT          S1200  F0012.500
 1-LINEAR MILL                                        POSITION:   IP  DISTANCE TO GO
 2-ARC MILL         PART 237-140                       X+000.0000 *    X+000.0000
 3-FRAME MILL                                          Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL                                         Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE                                         A+000.000  *    A+000.000
 6-REPEAT                                              B+000.0000 *    B+000.0000
 7-SUBROUTINE
 8-DWELL
 9-M FUNCTION

 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```

TEXT EVENT - INTRODUCTION

This utility event allows insertion of part identification numbers, operator instructions or other comments in a part program. Text messages appear with the part program listing in the Edit mode and appear when the part program executes in Auto or Single Event modes.

TEXT EVENT DATA

Parameter	Description	Entry
N	Sequence number	Required
Text	Field of up to 16 characters selected from: • 26 alpha keys • 10 digit keys • period key • minus key • INCR key (acts as a space key)	----

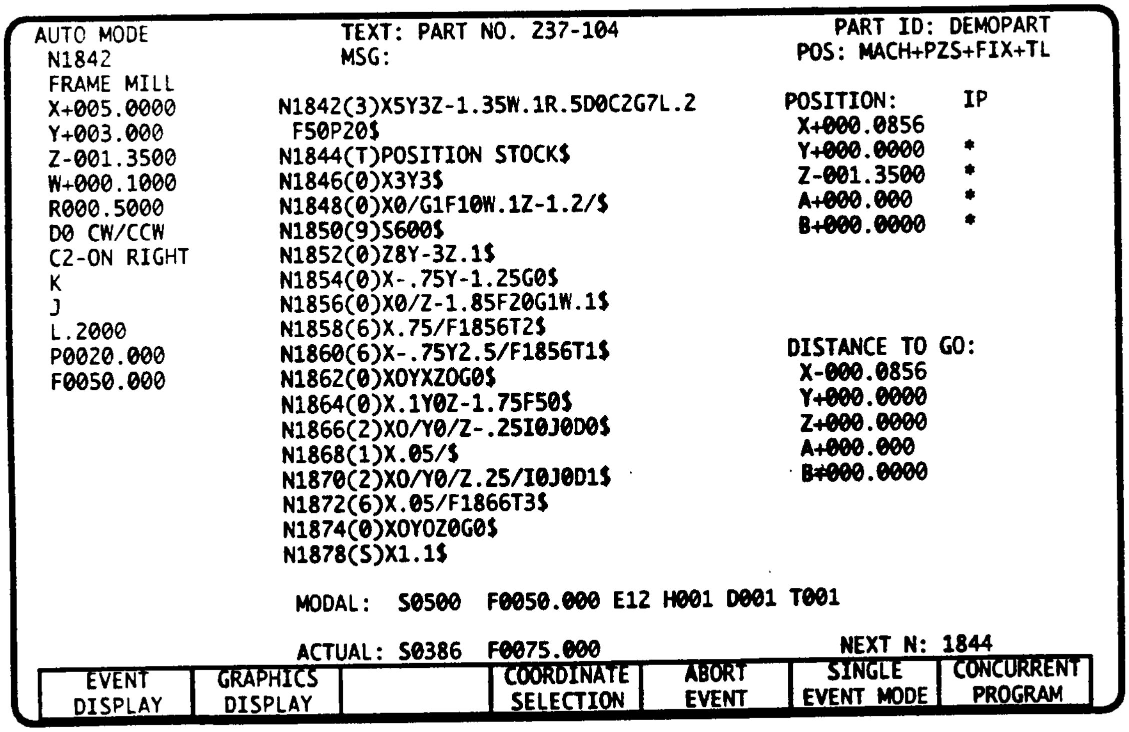
TEXT ENTRY

A message of 1 to 16 characters may be entered into this event. After selecting an N Number and EVENT TYPE T, key in the message. If you make a mistake during text entry, press CLEAR and start over. Press ENTER to store the text as shown, **facing page**. To edit a Text event, reenter the information.

TEXT DISPLAY

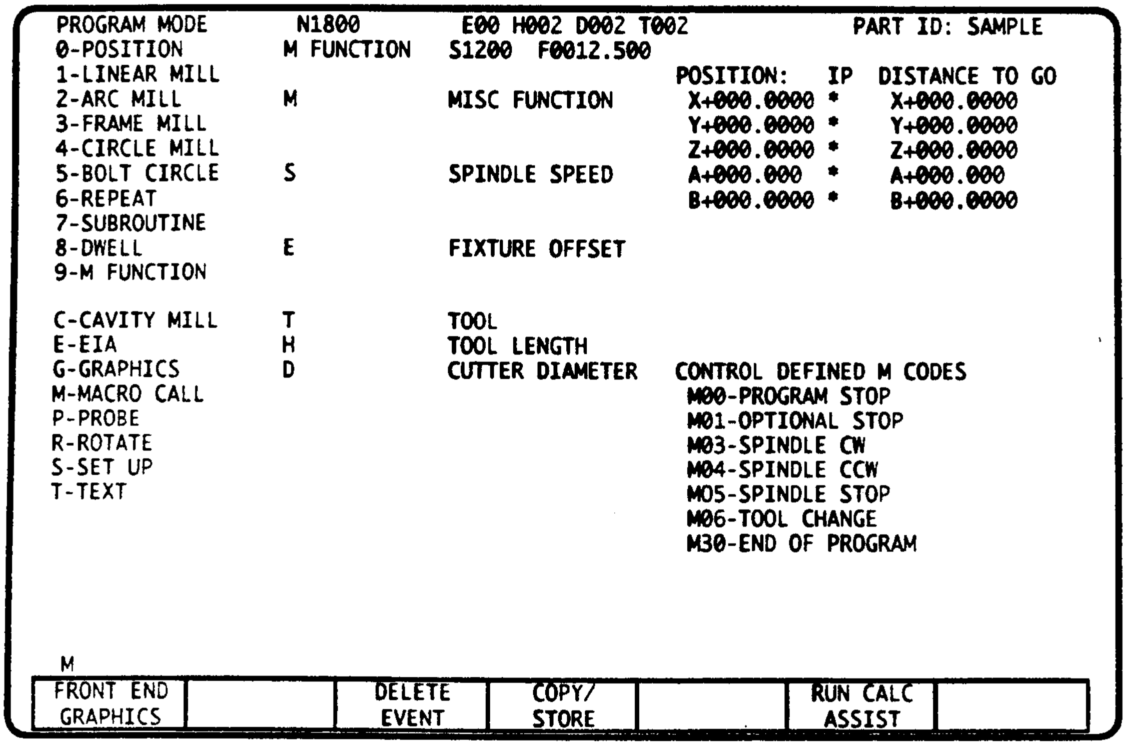
Text messages may be viewed with the program listing by moving to the Edit mode. The N Number assigns the event's location in the part program.

Execution of the Text event causes its message to be displayed in Auto or Single Event modes as shown **below**. The message remains on display as the part program continues execution until a following Text event is displayed. A text message may be removed from the display by programming a blank message.



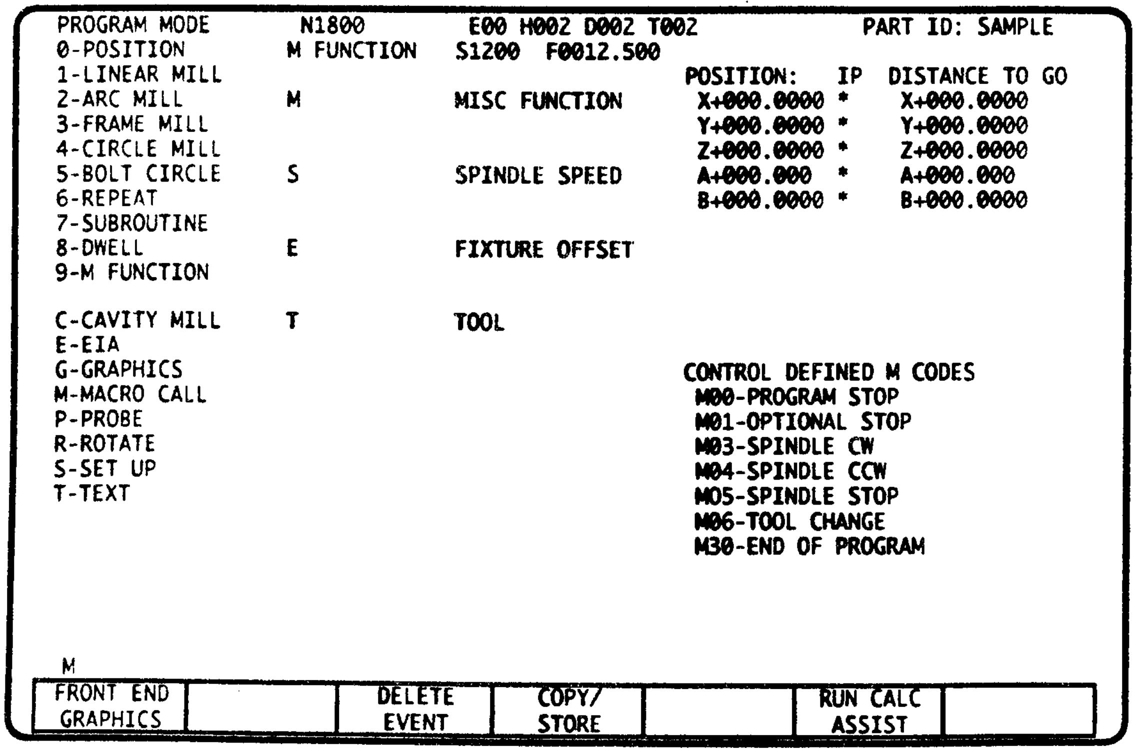
```
AUTO MODE                TEXT: PART NO. 237-104                PART ID: DEMOPART
 N1842                   MSG:                               POS: MACH+PZS+FIX+TL
 FRAME MILL
 X+005.0000          N1842(3)X5Y3Z-1.35W.1R.5D0C2G7L.2        POSITION:      IP
 Y+003.000            F50P20$                                  X+000.0856
 Z-001.3500          N1844(T)POSITION STOCK$                   Y+000.0000     *
 W+000.1000          N1846(0)X3Y3$                             Z-001.3500     *
 R000.5000           N1848(0)X0/G1F10W.1Z-1.2/$                A+000.000      *
 D0 CW/CCW           N1850(9)S600$                             B+000.0000     *
 C2-ON RIGHT         N1852(0)Z8Y-3Z.1$
 K                   N1854(0)X-.75Y-1.25G0$
 J                   N1856(0)X0/Z-1.85F20G1W.1$
 L.2000              N1858(6)X.75/F1856T2$
 P0020.000           N1860(6)X-.75Y2.5/F1856T1$               DISTANCE TO GO:
 F0050.000           N1862(0)XOYXZOG0$                         X-000.0856
                     N1864(0)X.1Y0Z-1.75F50$                   Y+000.0000
                     N1866(2)XO/Y0/Z-.25I0J0D0$                Z+000.0000
                     N1868(1)X.05/$                            A+000.000
                     N1870(2)XO/Y0/Z.25/I0J0D1$                B+000.0000
                     N1872(6)X.05/F1866T3$
                     N1874(0)XOYOZ0G0$
                     N1878(S)X1.1$

                      MODAL:  S0500  F0050.000 E12 H001 D001 T001

                      ACTUAL: S0386  F0075.000                 NEXT N: 1844
```

M FUNCTION EVENT - CONTROL WITH H and D CODES



```
PROGRAM MODE       N1800        E00 H002 D002 T002             PART ID: SAMPLE
0-POSITION        M FUNCTION   S1200   F0012.500
1-LINEAR MILL                                   POSITION:   IP  DISTANCE TO GO
2-ARC MILL        M            MISC FUNCTION     X+000.0000 *    X+000.0000
3-FRAME MILL                                     Y+000.0000 *    Y+000.0000
4-CIRCLE MILL                                    Z+000.0000 *    Z+000.0000
5-BOLT CIRCLE     S            SPINDLE SPEED     A+000.000  *    A+000.000
6-REPEAT                                         B+000.0000 *    B+000.0000
7-SUBROUTINE
8-DWELL           E            FIXTURE OFFSET
9-M FUNCTION

C-CAVITY MILL     T            TOOL
E-EIA             H            TOOL LENGTH
G-GRAPHICS        D            CUTTER DIAMETER  CONTROL DEFINED M CODES
M-MACRO CALL                                     M00-PROGRAM STOP
P-PROBE                                          M01-OPTIONAL STOP
R-ROTATE                                         M03-SPINDLE CW
S-SET UP                                         M04-SPINDLE CCW
T-TEXT                                           M05-SPINDLE STOP
                                                 M06-TOOL CHANGE
                                                 M30-END OF PROGRAM



 M
```

M FUNCTION EVENT - CONTROL WITHOUT H and D CODES



```
PROGRAM MODE       N1800        E00 H002 D002 T002             PART ID: SAMPLE
0-POSITION        M FUNCTION   S1200   F0012.500
1-LINEAR MILL                                   POSITION:   IP  DISTANCE TO GO
2-ARC MILL        M            MISC FUNCTION     X+000.0000 *    X+000.0000
3-FRAME MILL                                     Y+000.0000 *    Y+000.0000
4-CIRCLE MILL                                    Z+000.0000 *    Z+000.0000
5-BOLT CIRCLE     S            SPINDLE SPEED     A+000.000  *    A+000.000
6-REPEAT                                         B+000.0000 *    B+000.0000
7-SUBROUTINE
8-DWELL           E            FIXTURE OFFSET
9-M FUNCTION

C-CAVITY MILL     T            TOOL
E-EIA
G-GRAPHICS                                      CONTROL DEFINED M CODES
M-MACRO CALL                                     M00-PROGRAM STOP
P-PROBE                                          M01-OPTIONAL STOP
R-ROTATE                                         M03-SPINDLE CW
S-SET UP                                         M04-SPINDLE CCW
T-TEXT                                           M05-SPINDLE STOP
                                                 M06-TOOL CHANGE
                                                 M30-END OF PROGRAM



 M
```

M FUNCTION EVENT - INTRODUCTION

The M Function event is used for:

- commanding tool changes
- activating tool length and diameter offsets
- commanding spindle speeds
- activating Fixture Offsets
- programming miscellaneous functions such as: spindle (CW, CCW, or Off), program halts, end of program, coolant flow, or any other function that has been designed by your machine supplier.

One of the first events in every part program should be an M Function event (or a series of M Function events) that commands a tool change, activates the tool's length and diameter offsets, commands a spindle speed, and activates the fixture offset (E00-E32) that you wish to use. The last event in every program must be an M Function event that programs an M02 or M30 end of program command.

The degree that spindle and tool change operations are automated will vary according to your machine's capabilities and design. Your control may be equipped with additional M codes that your machine supplier has designed to operate a specific tool changer or any other mechanism that is unique to your machine.

Your machine supplier has selected the method that your control will use to program tool length and diameter offsets. Part 1-Section 2 describes how to determine which method has been selected for your control. The two methods are:

- **T, H, and D codes** - Separate T, H, and D codes are used to identify, respectively, the physical tool, the tool's length, and the tool's diameter.

- **T code only** - The active T code specifies the physical tool that is in the spindle, the tool's length offset, and the tool's diameter offset.

Your machine supplier is the best source of information for instructions on programming tool changes and activating tool length and diameter offsets.

M FUNCTION DATA

Parameter	Description	Entry
N	Sequence Number	Required
M	M Code (miscellaneous)	Optional
S	S Code (spindle speed)	Optional
E	E Code (fixture offset)	Optional
T	T Code (physical tool)	Optional
H	H Code (tool length offset)[1]	Optional
D	D Code (tool diameter offset)[1]	Optional
C	C Code (provided with spindle orient feature)[2]	Optional

[1] H and D codes will only be present in this event if your control uses H and D codes to activate tool length and diameter offsets. Refer to Part 1-Section 2 to learn more about this issue.

[2] If a C code appears in the M Function event, consult your machine supplier for instructions about its use.

ORDER OF OUTPUT TO MACHINE INTERFACE

One M, S, and T code may be programmed in the same M Function event. If all three are programmed, the output order to the machine interface is shown below.

1) M Code (unless M00, M01, M02, M06 or M30 is programmed)[3,4]
2) S Code
3) T Code[4]

When M00, M01, M02, M06, or M30 is programmed, the output order is S, T, M.

After one of these codes is output, the CNC waits for a "Done" signal from the machine before it will output the next code or transfer to the next part program event.

An E Code may be programmed in any M Function event. Since E Codes are not output to the machine interface and do not become active until a following motion command event, they are not included in this list. H and D codes are not included in this list for the same reason.

M CODES

Miscellaneous functions are programmed by M Codes in the range of M00-M99. Some systems permit entry of three digit M Codes in the range of M000-M999. Autocon has designated the following M Codes to perform the specific functions listed below.

M00 - PROGRAM STOP

M00 halts program execution and tool motion until you press CYCLE START. This command does not turn off the spindle.

M01 - OPTION STOP

The Optional Stop feature can be enabled by your machine supplier. If the control's Option Stop switch is ON, program execution halts when the control executes an M01 command. If the control's Option Stop switch is OFF, program execution does not halt because the M01 command is ignored. The Optional Stop feature is turned on and off with a Status screen soft key[5]. The on/off status of the feature is also listed on the Status screen. To view this screen, press the FAULT/STATUS key twice.

If the Optional Stop feature is not enabled, or is turned off, an M01 command has no affect.

[3] Any M code that is not contained in this list may be output either first or last when M, S, and/or T codes are programmed in the same event. Check with your machine supplier about deferred M codes.

[4] Macro/CPC M Codes or Macro/CPC T Codes are advanced features that may be enabled by your machine supplier. Refer to Part 1 - Section 2 to determine whether these features have been enabled. In general, when the corresponding feature is enabled, the control will execute an assigned Macro Catalog part program upon execution of a Macro/CPC M Code or a Macro/CPC T Code. Technical information about these features is provided later in this section.

[5] Alternately, your machine supplier may provide an OPTIONAL STOP switch to turn this feature on and off. The on/off status of the feature appears on the Status Display screen. Press the FAULT/STATUS key twice to view this screen.

M02 - END OF PROGRAM

M02 has the exact same function as M30. Refer to M30 description below.

M03 - SPINDLE CLOCKWISE

If the spindle is controlled by part program commands, this code commands the spindle to turn clockwise.

M04 - SPINDLE COUNTERCLOCKWISE

If the spindle is controlled by part program commands, M04 commands the spindle to turn counterclockwise. If your spindle is not bidirectional M04 will act as M03.

M05 - SPINDLE STOP

If the spindle is controlled by part program commands, this code commands the spindle to stop.

M06 - TOOL CHANGE

Due to the diversity of spindle control and tool change applications, you should consult your machine supplier for tool change instructions. The information for the M06 - Tool Change command describes the minimum steps necessary to complete a manual tool change.

In general, your tool change sequence must include the following steps.

1) Position the spindle to provide clearance for the tool change.
2) Stop the spindle.
3) Inhibit execution of the part program until the tool change is complete.
4) Change the tool.
5) Activate the new tool's length and diameter offsets.
6) Start the spindle.
7) Allow the spindle to reach operating speed.
8) Transfer to the following part program event.

Unless the machine supplier alters the following sequence, the control's execution of an M06 event is as follows;

1) A T Code, when programmed in this event, executes first to cue a tool.[6]
2) The machine rapids the axes to the tool change point. Either the Home position (as defined in the Set Up mode) or a precise point that has been defined by your machine supplier will serve as the tool change position.
3) The control outputs M05. If the spindle is controlled, it will turn off.
4) The control outputs M06. If an auto tool changer is present, it may use this signal to begin the tool exchange.

If the tool will be manually loaded, program an M00 - Program Halt following the M06 to inhibit the program from transferring to the next event. After the tool change is complete press CYCLE START. If your cycle halts twice during the tool change, or you have a TOOL CHANGE COMPLETED pushbutton on your Operator's panel, the M00 command is redundant and need not be programmed. See the example below.

[6] If your control does not use H and D codes, the T code also specifies the Tool Table length and diameter offsets for the tool. If your control uses H and D codes, you should also program an H and D code in this event to identify the tool's length and diameter offsets.

(9) M06 T04 H04 D04 - rapid to the tool change position, identify tool number and tool offset entries.

(9) M00 - halt program transfer. Change tool. If the spindle is manually controlled, set its speed and turn the spindle on now. Press CYCLE START to resume operations. This event may be redundant, see above text.

(9) M03 S1000 - If the spindle is controlled by the CNC, this event will turn on the spindle clockwise at 1000 RPM.

(8) L2 - a dwell may be programmed to allow the spindle to reach operating speed before the next feed move begins.

M08 - COOLANT ON

Outputs a signal to turn coolant on. Your machine supplier determines whether this signal will be used.

M09 - COOLANT OFF

Outputs a signal to turn coolant off. Your machine supplier determines whether this signal will be used.

M30 - END OF PROGRAM

Enter this M Code as the last event in your part program. The control will execute the following series of commands.

1) The tool rapids to Home position.[7,8]
2) The control outputs M05. If the spindle is controlled, it will turn off.
3) The control outputs the M30 code.
4) The control performs a control reset. The control reset has the same effect as pressing the RESET key when the control is not in cycle. The specific actions that occur on a control reset are listed in the Auto mode section of this manual.

MACHINE SUPPLIER ASSIGNED M CODES

Your machine supplier may assign the following M Codes.

M07, M10-M29, M31-M99; and M100 to M999 if 3 digit M Codes are available.

Execution of one of these codes causes a BCD signal to appear at the machine tool interface. The machine supplier may design logic to decode the signal and actuate custom features or controls. Some applications will report the function of these M Codes on one of the ISO ASSIST pages. Refer to the machine supplier's specifications for use of these codes.

An advanced feature called "Macro/CPC M codes" allows an M code to call a specific Macro Catalog part program into execution.

[7] Coordinates for the Home position are entered in the Set Up mode.

[8] Machine supplier option: some machines will not return the axes to the Home position when an M30 or M02 code is output. In the this case the axes will remain in their last programmed position. Any tool clearance that is needed for an automatic pallet or fixture changer must be programmed before the M30 command.

MACRO/CPC M CODES

You only need to read this topic if the Macro/CPC M Code feature has been enabled by your machine supplier. To determine if this feature is used, move to the control's System Settings screen as described in Part 1-Section 2; it is enabled when the SPECIAL M CODE MACRO item is highlighted. The next line on the display provides a list of special M codes that are designated for use with this feature. As described below, the function of a Macro/CPC M Code depends on whether it is programmed in a Main Catalog or Macro Catalog part program.

Main Catalog part program - execution of a Main Catalog part program event that contains an M code which has been designated as an Macro/CPC M Code will cause the control to automatically execute the Macro Catalog part program that is assigned to the M code. The title of the Macro/CPC that is assigned to the M code is (Mxx), where xx is the number of the M code. Example: If M80 has been designated as a Macro/CPC M Code, the event

```
N100 (9) M80
```

will call the Macro/CPC program titled M80 into execution. A program of this title must reside in the Macro Catalog. You should not edit or delete the program without explicit instructions from your machine builder.

Macro/CPC M Codes always executes last when they are programmed in an event that contains any combination of M, S, T, E, H, and D codes.

Macro Catalog part program - When executed from a Macro/CPC part program, Macro/CPC M Codes are strobed to the machine interface in the exact same manner as a "normal" M code. No Macro/CPC will be called into execution.

S CODES - SPINDLE SPEED CONTROL

S Codes command the spindle speed. Most systems will have a four digit S Code (Sxxxx). Some systems permit a five digit S Code (Sxxxxx) with a maximum entry of S65535.

MANUAL SPINDLE CONTROL

When the spindle is not controlled by the CNC, programmed S Codes can prompt the operator to choose the correct RPM range for a following machining sequence. Before running a part program, the operator can preview the program to identify the required spindle speeds. The S Code that is currently active (i.e. the last executed S Code) appears on Auto mode's Event Display and Position Display screens. These displays can prompt the operator to choose the correct spindle speed when an M00 halts program execution.

PROGRAMMED SPINDLE CONTROL

The machine supplier may provide a programmable spindle. In this case the S Code may be used to program the spindle speed directly as Revolutions Per Minute.

The spindle will not respond to an S Code unless an M03 or M04 command is active. After the control sends an analog signal to the spindle the control may dwell until the spindle reaches the programmed speed. If this is not the case, you can program a following Dwell event to allow the spindle to reach operating speed.

T CODES - INTRODUCTION

The T code topic has been split into two separate descriptions depending on whether your control has been configured to use H and D codes. Refer to Part 1-Section 2.

T, H, and D CODES

Read this section only if your control uses H and D codes. The following three codes are required to activate a tool. If your control has enabled the Macro/CPC T Code feature (refer to Part 1-Section 2), disregard this section; your machine supplier must document the use of T codes for your control.

T code - 2 through 6 digits - The number programmed identifies the physical tool that must be put in the spindle. This number will be strobed to the machine-tool interface to command the tool changer. If your machine does not have a tool changer, this number will be displayed on the Auto and Single Event mode screens to inform the operator which tool to put in the spindle. If a Q-Lookup table is present in the Tool Tables, the programmed T code will be converted by the Q table as described in the Tool Tables Mode sections of this manual. When Tool Life Management is active, the tool number you program identifies the tool group from which the control will choose the proper tool to output to the machine-tool interface.

H code - tool length offset 2 or 3 digits - The number programmed identifies the H table tool length offset for the active tool. Activation of this offset does not cause any axes motion; instead the control stores a Z axis offset that it adds to the next event that programs a Z axis endpoint. For this reason, the first motion command following a tool code should program an absolute Z endpoint which positions the tool tip relative to the workpiece. The active H code can be canceled with an H00 command.

D code - tool diameter offset 2 or 3 digits - The number programmed identifies the D table tool length offset for the active tool. Activation of this offset does not cause any axes motion; The active D code can be canceled with an D00 command.

T CODES - CONTROLS WITHOUT H and D CODES

Read this section only if your control does not use H and D codes. If your control has enabled the Macro/CPC T Code feature (refer to Part 1-Section 2), disregard this section; your machine supplier must document the use of T codes for your control.

2 or 3 DIGIT T CODE FORMAT

The T code number selects the tool's length and diameter offsets in the Tool Table, and is output to the machine interface. Activation of this offset does not cause any axes motion; instead the control stores a Z axis offset that it adds to the next event that programs a Z axis endpoint. For this reason, the first motion command following a tool code should program an absolute Z endpoint which positions the tool tip relative to the workpiece. Also, the control will compensate for the tool's radius during any compensated milling sequence. The control will also compensate for the tool's diameter during the execution of a Frame Mill, Circle Mill, and Cavity Mill event. Most applications allow you to cancel the active tool offsets with an T00 command.

If a Q-Lookup table is present in the Tool Tables, the programmed T code will be converted by the Q table as described in the Tool Tables Mode sections of this manual. When Tool Life Management is active, the tool number you program identifies the tool group from which the control will choose the proper tool to output to the machine-tool interface.

4 DIGIT T CODE FORMAT and a SWAPPING TOOL LOOKUP TABLE

This is a special method of programming T codes on a machine that is equipped with an automatic, random tool changer. Read this section only if your control uses a 4 digit, a swapping lookup table, but does not use H and D codes. Refer to Part 1-Section 2 to learn about these features. You should also read the previous 2 or 3 Digit T Code Format topic to familiarize yourself with the manner in which the control activates tool offsets.

A four digit T Code is represented by TXXxx, where XX identifies the physical tool and xx designates the tool length and cutter diameter Tool Table offsets that will be used.

The first digit pair, XX is programmed in the range of 01-99. In some auto tool changer applications this digit pair will begin to cue the corresponding tool into position for a following M06 tool exchange.

The second digit pair, xx, designates which of the 99 Tool Table offsets will be used for the active tool.

E CODES - FIXTURE OFFSETS

E Codes assign a fixture offset to be used for a following series of machining events. E Codes range from E00 - E32. The dimensions for fixture offsets E01-E32 are stored in the Tool Tables mode. These values represent the distance that each axis is to be offset from the machine's datum[9]. Executing an E Code, causes part zero to be offset from the datum. Examples which illustrate the uses of Fixture Offsets are provided in the Fixture Offset Table section in Part 2 of this manual.

E Code execution does not cause any axes motion -instead the control stores an offset for each axis (X,Y,Z) that it will add to the part program's next motion command. The manner in which the axes move to a point relative to the fixture's part zero is chosen by your machine supplier through the selection of Immediate or Deferred offset activation.

If the control has Immediate offset activation, the first motion command that follows an E Code will cause all axes to move to a point relative to the fixture's part zero.

If the control has Deferred offset activation, the first motion command that follows an E Code will cause only the axes that are programmed in the event to move to a point relative to the fixture's part zero. Any axis that is not programmed in the event remain stationary until its endpoint is programmed in a following event.

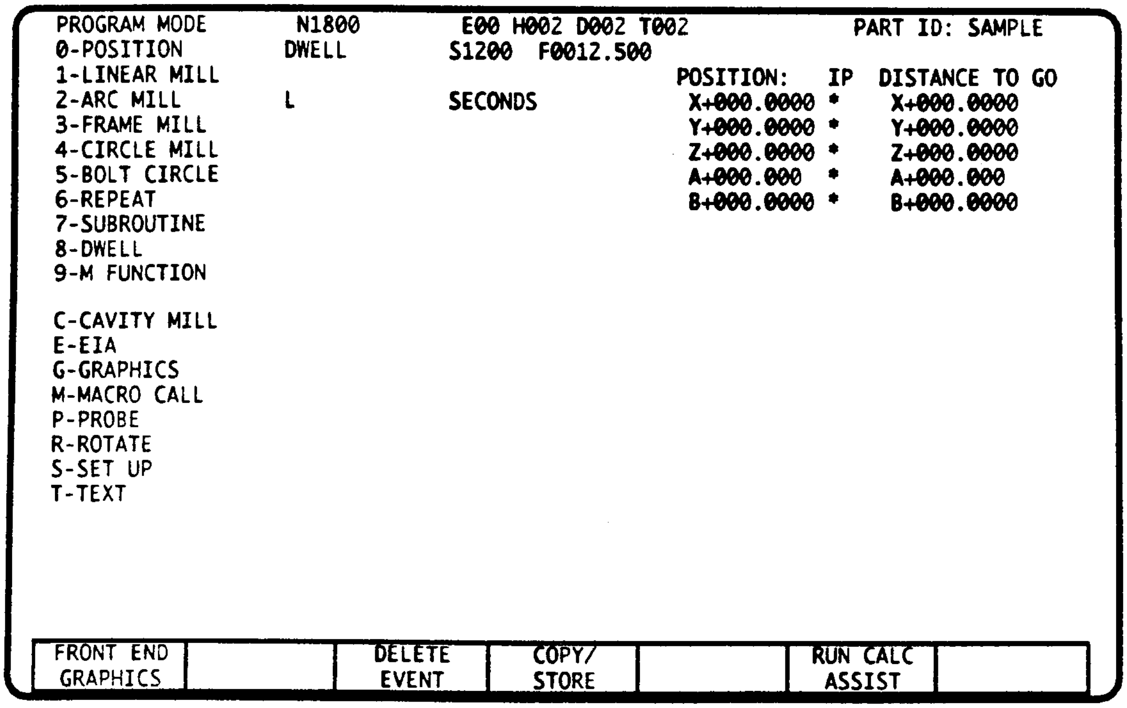
Refer to the Immediate or Deferred Offsets topic in the Fixture Offset Table section of this manual.

Execution of E00 causes the control to cancel any active fixture offset.

Once executed, an E Code is not canceled by an M02[10], M06, M30[10] command, or by pressing RESET.

[9] The datum is set with a ZERO SET operation in the Jog mode. If no ZERO SET is performed, the control assumes that machine coordinate zero is the current datum.

[10] Your machine supplier may choose to configure the control so that E Codes are canceled on M02 or M30 execution. After a full part program cycle, check your Auto mode display to see if E Codes are canceled by M02 or M30.



```
PROGRAM MODE       N1800        E00 H002 D002 T002             PART ID: SAMPLE
0-POSITION        DWELL         S1200  F0012.500
1-LINEAR MILL                                     POSITION:    IP  DISTANCE TO GO
2-ARC MILL        L             SECONDS            X+000.0000 *    X+000.0000
3-FRAME MILL                                       Y+000.0000 *    Y+000.0000
4-CIRCLE MILL                                      Z+000.0000 *    Z+000.0000
5-BOLT CIRCLE                                      A+000.000  *    A+000.000
6-REPEAT                                           B+000.0000 *    B+000.0000
7-SUBROUTINE
8-DWELL
9-M FUNCTION

C-CAVITY MILL
E-EIA
G-GRAPHICS
M-MACRO CALL
P-PROBE
R-ROTATE
S-SET UP
T-TEXT
```

DWELL EVENT - INTRODUCTION

This event causes part program execution to be inhibited for a specified interval of time. A dwell can synchronize part program execution to outside influences as, for example, the time a spindle might require to reach its operating speed.

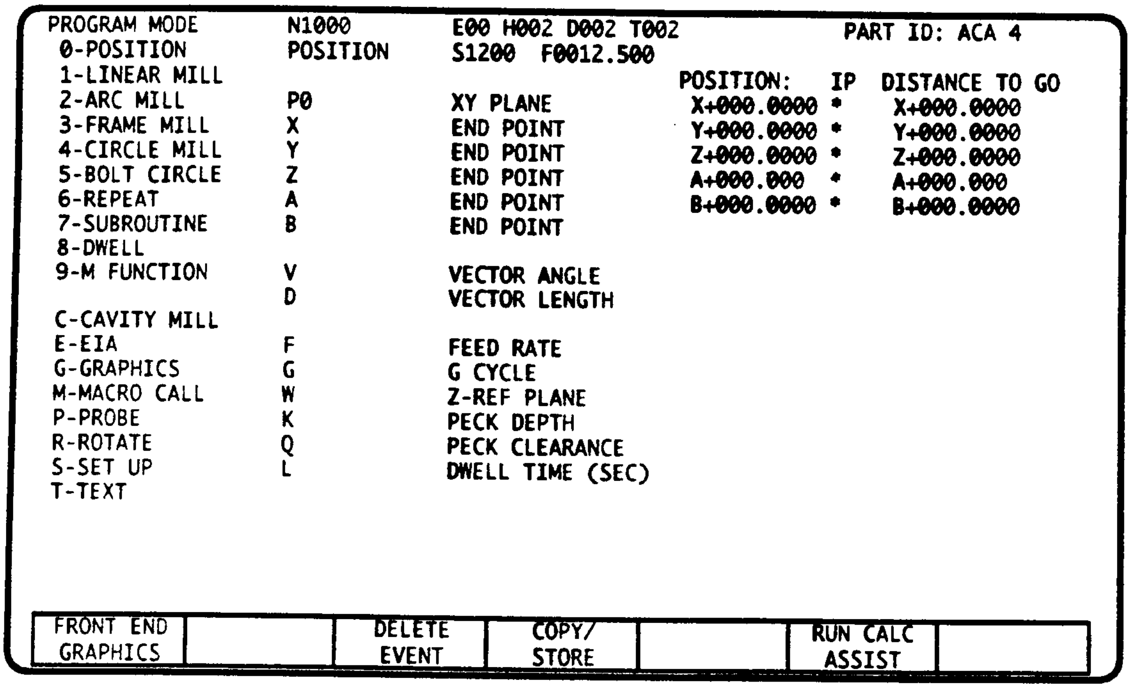
Warning: A Dwell event should never be used to give the operator time to complete an operation within the machining envelope while the spindle is turned on, or when the CNC is in cycle. The proper way to allow for such an adjustment is to program an M05 spindle stop command followed by an M00 program halt command in successive M Function events.

DWELL EVENT DATA

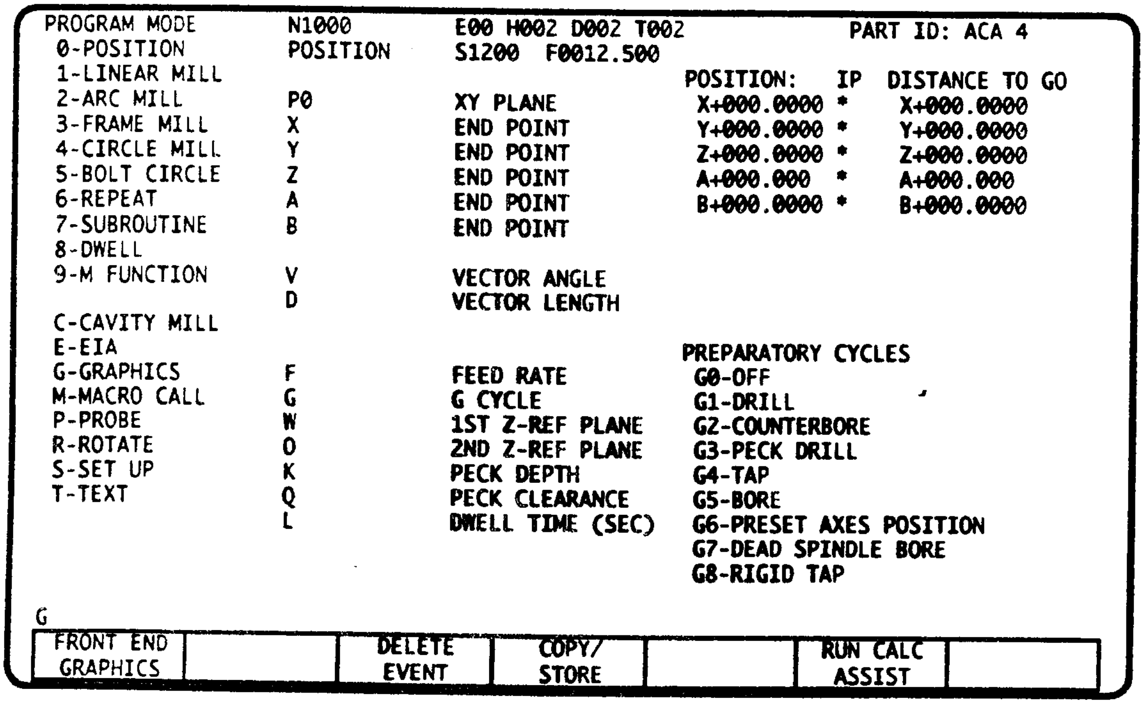
Parameter	Description	Entry
N	Sequence Number	Required
L	Dwell in seconds	Required

L - DWELL

L programs a dwell which can range from 0.1 to 999.9 seconds.



```
PROGRAM MODE       N1000       E00 H002 D002 T002             PART ID: ACA 4
 0-POSITION        POSITION    S1200  F0012.500
 1-LINEAR MILL                                  POSITION:   IP  DISTANCE TO GO
 2-ARC MILL        P0          XY PLANE          X+000.0000 *    X+000.0000
 3-FRAME MILL      X           END POINT         Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL     Y           END POINT         Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE     Z           END POINT         A+000.000  *    A+000.000
 6-REPEAT          A           END POINT         B+000.0000 *    B+000.0000
 7-SUBROUTINE      B           END POINT
 8-DWELL
 9-M FUNCTION      V           VECTOR ANGLE
                   D           VECTOR LENGTH
 C-CAVITY MILL
 E-EIA             F           FEED RATE
 G-GRAPHICS        G           G CYCLE
 M-MACRO CALL      W           Z-REF PLANE
 P-PROBE           K           PECK DEPTH
 R-ROTATE          Q           PECK CLEARANCE
 S-SET UP          L           DWELL TIME (SEC)
 T-TEXT
```



```
PROGRAM MODE       N1000       E00 H002 D002 T002             PART ID: ACA 4
 0-POSITION        POSITION    S1200  F0012.500
 1-LINEAR MILL                                  POSITION:   IP  DISTANCE TO GO
 2-ARC MILL        P0          XY PLANE          X+000.0000 *    X+000.0000
 3-FRAME MILL      X           END POINT         Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL     Y           END POINT         Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE     Z           END POINT         A+000.000  *    A+000.000
 6-REPEAT          A           END POINT         B+000.0000 *    B+000.0000
 7-SUBROUTINE      B           END POINT
 8-DWELL
 9-M FUNCTION      V           VECTOR ANGLE
                   D           VECTOR LENGTH
 C-CAVITY MILL
 E-EIA                                          PREPARATORY CYCLES
 G-GRAPHICS        F           FEED RATE         G0-OFF
 M-MACRO CALL      G           G CYCLE           G1-DRILL
 P-PROBE           W           1ST Z-REF PLANE   G2-COUNTERBORE
 R-ROTATE          O           2ND Z-REF PLANE   G3-PECK DRILL
 S-SET UP          K           PECK DEPTH        G4-TAP
 T-TEXT            Q           PECK CLEARANCE    G5-BORE
                   L           DWELL TIME (SEC)  G6-PRESET AXES POSITION
                                                 G7-DEAD SPINDLE BORE
                                                 G8-RIGID TAP
G
```

POSITION EVENT - INTRODUCTION

This event positions the tool at rapid rate to an endpoint designated by Cartesian or polar coordinates. Programmable G Codes can add drill, tap, counterbore, peck, bore, or rigid tapping (with option) cycles to this event. A special G6 code lets you shift part coordinates.

POSITION EVENT DATA (refer to the **upper** screen display)

Parameter	Description	Entry
N	Sequence number	Required
		Cartesian Endpoint
X or X/	X axis endpoint	Endpoint, see topic below
Y or Y/	Y axis endpoint	Endpoint, see topic below
Z or Z/	Z axis endpoint or G cycle depth	Endpoint, see topic below
		Polar Endpoint
P	Plane select P0=XY, P1=YZ, P2=ZX	Select P0 for G1-G5, or G7-G8
V	Vector Angle	Endpoint, see topic below
D	Distance	Endpoint, see topic below

G CYCLES and G CYCLE PARAMETERS (refer to the **lower** screen display)

G	G Cycle	G0-G5, G7-G8 Modal; G6 Nonmodal
F	Feedrate	Modal
K or K/	Peck	Modal
Q	Peck Clearance	Modal
W	1st Z Reference Plane	Modal
O	2nd Z Reference Plane	Modal
L	Dwell	Modal

G Cycle	Description	Required Parameters
G0	Turn off G Cycles	none
G1	Drill	W, Z, F
G2	Counterbore	W, Z, F, L
G3	Peck Drill	W, Z, F, K or K/, Q
G4	Tap	W, Z, F, L
G5	Bore	W, Z, F
G7	Dead Spindle Bore	W, Z, F
G8	Rigid Tap (option)	W, Z, F (see text)
G6	Load Position	see text

CARTESIAN or POLAR ENDPOINT DESCRIPTION

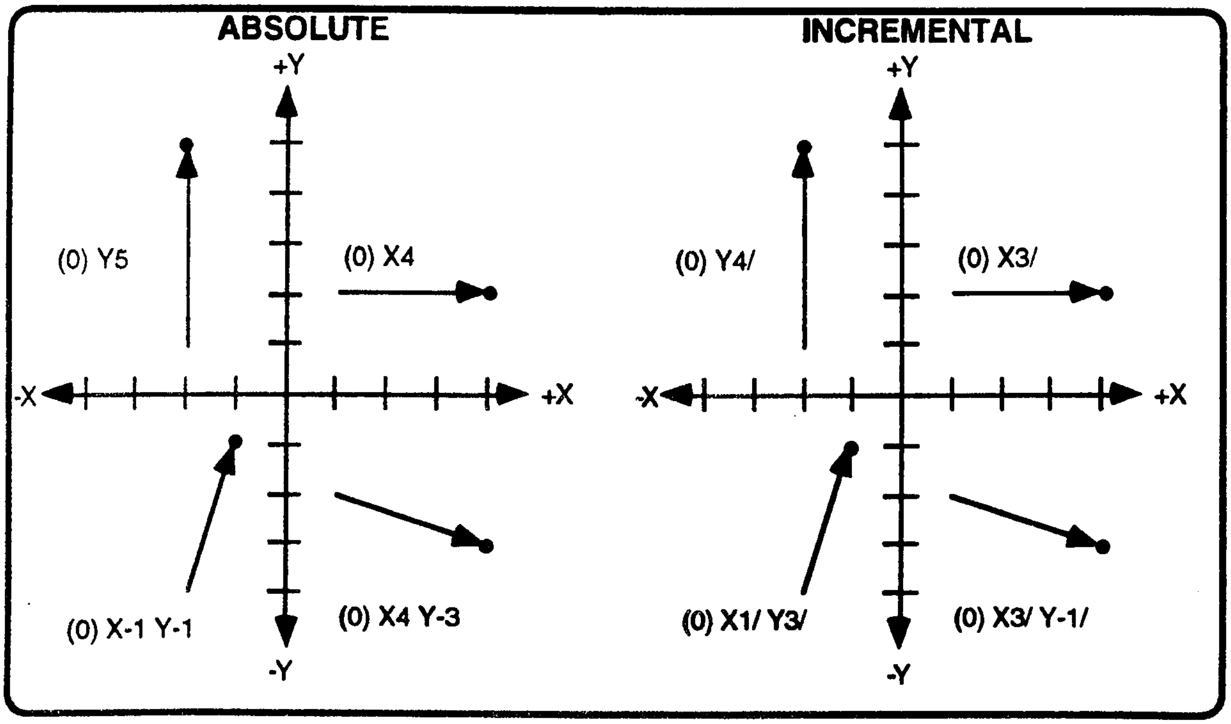
The G0 cycle, active at startup, is used for simple positioning. The event's endpoint can be completely described with Cartesian or polar entries, or with one Cartesian endpoint (X, Y, or Z) and an angle. If you run your part program through Calc-Assist, further endpoint descriptions can be programmed by combining Cartesian and polar data. Calc-Assist is explained in Part 5 of this manual.

G0 - POSITION WITH G CYCLES OFF

G0 is always active at start up. This causes the control to position without adding a fixed cycle. G0 remains in effect until a G1-G5 or G7-G8 executes in a Position or Bolt Circle event. To return to normal positioning after G1-G5 or G7-G8 events, program G0 in the next Position event. Executing an M02 or M30 - End of Program M code or pressing RESET when the control is not in cycle will also reestablish the G0 condition.

X, Y, Z - CARTESIAN ENDPOINT DESCRIPTION

Enter dimensions for each axis that must move to reach the endpoint. An axis endpoint may be programmed as a signed absolute or signed incremental entry. Examples of absolute and incremental Position events are shown below.

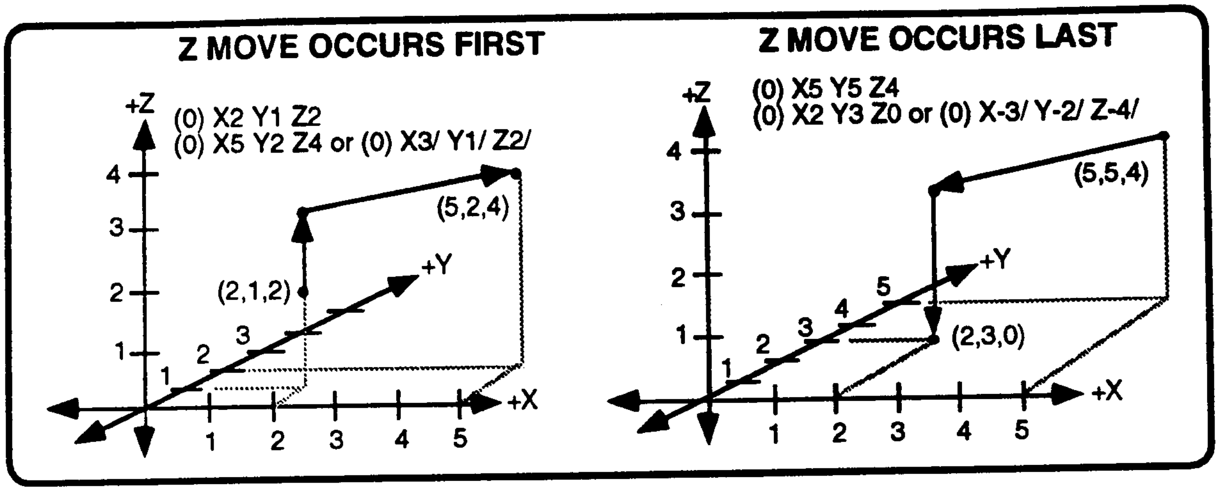


Absolute and incremental entries may be programmed in the same event. The move from X1, Y-2 to X4, Y-3 (lower right quadrant in upper illustrations), for example, can be programmed with any of the four events listed below.

(0) X4 Y-3	X and Y are absolute dimensions
(0) X4 Y-1/	X is absolute; Y is incremental
(0) X3/ Y-3	X is incremental; Y is absolute
(0) X3/ Y-1/	X and Y are incremental dimensions

If XY motion is programmed, the tool moves at rapid rate in a direct path from start point to the XY endpoint. In most systems, when Z is programmed with another axis, the Z motion occurs first for rapid Z motion away from the work, and last for rapid Z motion towards the work. Refer to **lower** examples.

Some controls are set to command simultaneous XYZ tool motion when a G0 Position event executes. However, all controls will move the Z axis independent of all other axes when a quill cycle (G1-G5, G7-G8) is modal, or is programmed in the event.



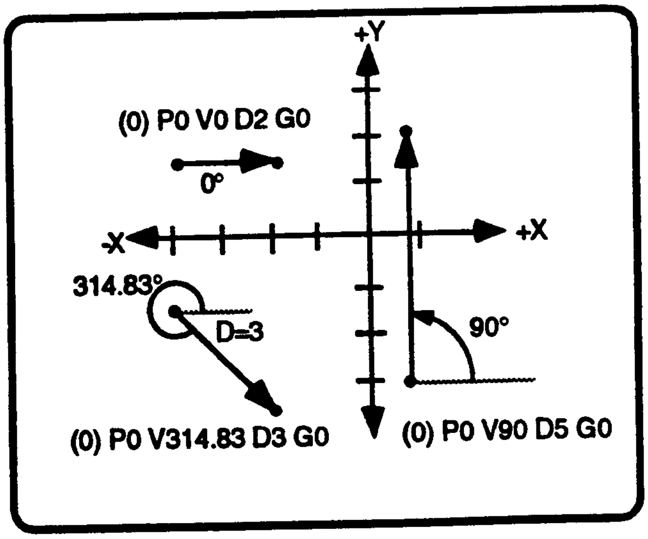
P , V, D - POLAR ENDPOINT DESCRIPTION

P - PLANE SELECT

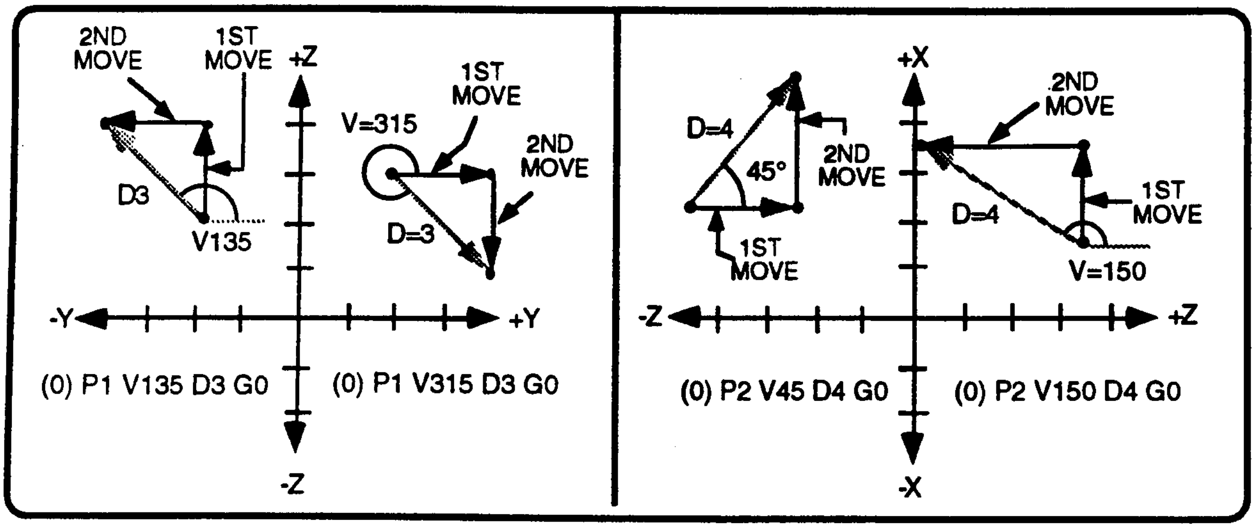
Polar coordinates define a move which occurs only in the selected plane. P is initially set to the XY plane P0 in each new event. Enter P1 to position in the YZ plane, or P2 to position in the ZX plane. Refer to examples, **following page.**

D, V - POLAR ENDPOINT DESCRIPTION

The tool moves in the selected plane P a distance D in direction V. D is the unsigned length of the move. V is an angle that opens counterclockwise about the start point from standard position. Standard position is drawn from the tool's start point in the +X direction in P0, +Y direction in P1, and +Z direction in P2. The counterclockwise direction is determined by looking at the plane from the positive axis that is perpendicular to the selected plane (you will be looking in perpendicular axis' negative direction). Example: to determine counterclockwise in the XY plane, you must view the plane from the +Z axis, looking in the -Z direction.



When a polar move is in the P0 (XY) plane, the tool moves directly along the programmed path at rapid rate. Although a YZ or ZX polar move is similarly programmed, Z axis motion does not occur simultaneously with the other axes.[1]



QUILL CYCLES G1-G5, G7-G8

A Position event with Cartesian or Polar coordinates or a Bolt Circle event may program G1-G5 or G7-G8 cycles The G cycle is modal, repeating in a following Position or Bolt Circle event if the following event does not program G.

The G cycles are:

G1 - drill
G2 - counterbore
G3 - peck drill

G4 - tap
G5 - bore

G7 - dead spindle bore
G8 - rigid tapping (option)

If a G1-G5 or G7-G8 Position event programs XY motion, the XY motion occurs first (to position the tool over the center of the hole) then the Z axis advance and G cycle motion occur. **If no XY motion is required, enter X0/ or Y0/ to allow execution of the G cycle.**[2]

MODAL EIA/ISO QUILL CYCLES

Press the ISO ASSIST key to display a list of quill cycles available in the EIA event (they are generally in the G73-G89 range). There are some cycles in the EIA event that have capabilities beyond the G cycles in the Position event. It is important to note that the EIA quill cycles and the Position event quill cycles are interactively modal. In other words, a quill cycle established in an EIA event will execute in a following Position event and vice versa. Program G0 in the Position event if you must cancel the active quill cycle.

[1] Your machine supplier may configure your system to simultaneously rapid X, Y, and Z during G0 Position events. Z axis motion will always occur independent of other axis motion when a quill cycle (G1-G5, G7-G8) executes. Refer to Part 1-Section 1.

[2] While a quill cycles is active, execution of a Position event that does not contain an X or Y endpoint will not cause any tool motion. All modal values programmed in this event, however, will be updated by the control and will be used in the next Position event that programs an X or Y endpoint. This programming method lets you change one or more modal G cycle parameters without commanding any tool motion.

PARAMETERS USED IN G CYCLES

W - 1st Z AXIS REFERENCE PLANE

W is a Z axis reference plane that is parallel to the XY plane. Program W as the signed, absolute coordinate that locates the reference plane slightly above the part surface (we typically suggest a value of .100" or 2.0 mm assuming that the top of the work surface has been set to Z0). Z axis advance motion will slow from rapid to feedrate F as soon as the tool tip reaches W. The tool also returns to this plane at the end of each G cycle whenever a 2nd reference plane (letter O) is not active. W is modal with all following events that require a reference plane W.

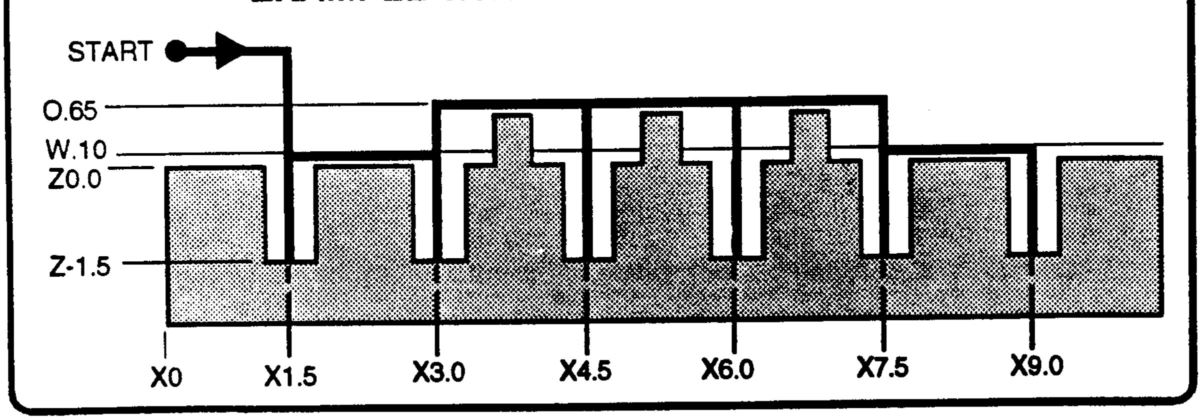
Note: This entry is programmed with the letter "I" if your control is equipped with an auxiliary axis that is programmed with the letter "W". Refer to Part 1-Section 2.

O - 2nd Z AXIS REFERENCE PLANE

O is a 2nd Z axis reference plane parallel to the XY plane. Program O as the signed, absolute coordinate that locates the Z axis position where you want the tool to return to at the end of every G cycle. O is an optional entry; if not programmed, it will default to the value of the W plane.

The easiest way to program the 2nd reference plane is to establish a rule that you will **always** program W in **every** Position or EIA event that programs the letter O. Although you are not required to follow this rule, it will make your program easier to understand since you will not have to scan through your entire program to learn the relationship between the two reference planes. Complete rules for programming the 2nd reference plane appear after the example below.

EXAMPLE WITH 2nd REFERENCE PLANE



```
N010  (9)   T02 H02 D02 M06 E01      • use tool #2 and fixture offset #1
N020  (9)   M03 S1250                • turn spindle on
N030  (0)   X0 Y1 Z2 G0              • position tool above work
N040  (0)   X1.5 G1 Z-1.5 F12 W.1    • drill hole, 2nd ref. plane defaults to .10"
N050  (0)   X3.0 O.65 W.1            • drill hole #2, tool returns to O.65"
N060  (0)   X4.5                     • drill hole #3, tool returns to O.65"
N070  (0)   X6.0                     • drill hole #4, tool returns to O.65"
N080  (0)   X7.5 O.1 W.1             • drill hole #5, set O to .10"
N090  (0)   X9.0                     • drill hole #6, tool returns to O.1
N100  (9)   M30                      • end of program
```

Rules for programming a 2nd reference plane - if you decide not to follow the rule of programming W in every event that programs the 2nd reference plane, you need to be aware of the following rules that govern the O plane.

- The O reference plane is canceled when an M02, M30, Position event G0, or EIA event G80 command executes, or when the RESET button is pressed when the control is not in cycle. The control cancels the 2nd reference plane by setting it equal to the modal value of the 1st reference plane W.
- Until O is established for the first time in a part program (or established for the first time following a G0/G80 quill cycle cancel command), the control will always set the 2nd reference plane equal to the 1st reference plane. The 2nd reference plane is considered to be established when it has been programmed to a different value than W.
- Once O is established, it will be maintained independent of the W plane (i.e., programming W does not affect O; programming O does not affect W). As an exception to this rule, programming W and O to the same value will, in effect, cancel O.

F - FEEDRATE

The feed portion of a quill cycle occurs at feedrate F. Feedrate F is modal with all events that program feedrate.

Z - DEPTH

Z is the end depth of the G cycle. Enter Z as a signed Z axis coordinate or Z/ as a signed distance from W. Z or Z/ is modal with following G1-G5 or G7-G8 events.

L - DWELL

Tap and counterbore cycles can program L seconds of dwell. L is modal with following G2 and G4 Position and Bolt Circle events.

K or K/ - PECK

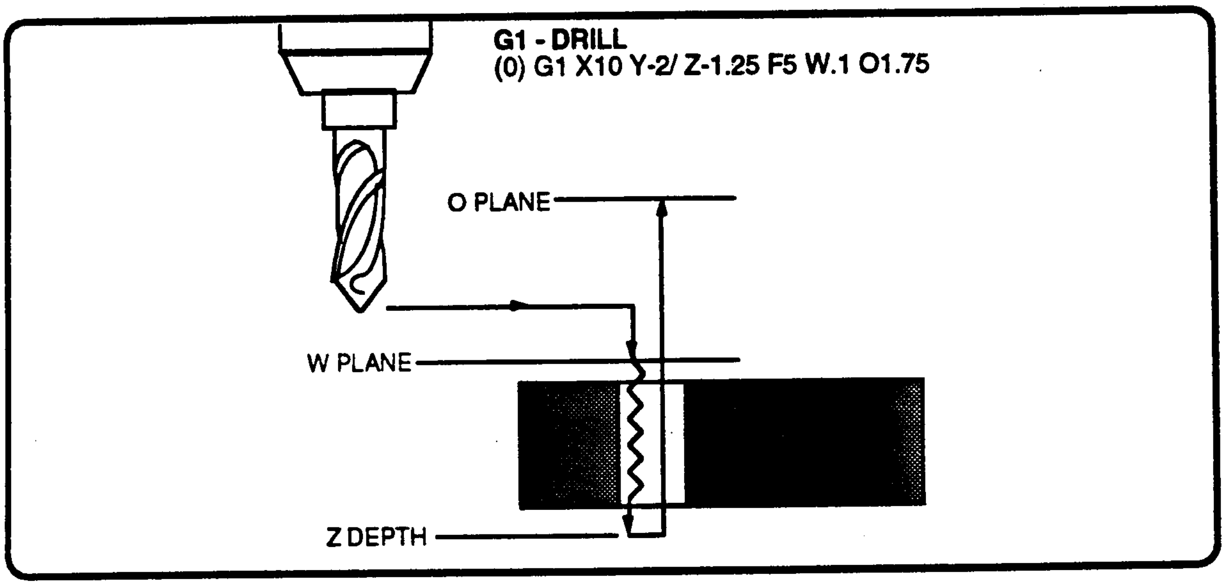
In the G3 cycle, a unit depth of K is drilled per peck. K is unsigned. If K is absolute, each retract motion will return the tool tip to W. If K is incremental, each retract motion will return the tool tip to Q. K or K/ is modal with following Position and Bolt Circle peck drill events.

Q - PECK CLEARANCE DIMENSION

Q is a clearance dimension above each K/ peck in the G3 peck cycle. If Q is not programmed, the default value is 0.1" (2.5mm). Q is used only when peck depth K is programmed as an incremental dimension.

G1 - DRILL CYCLE

The drill tip slows from rapid to feedrate F at reference plane W, continues the feed to Z, then rapids to the O plane. (See illustration **below**).

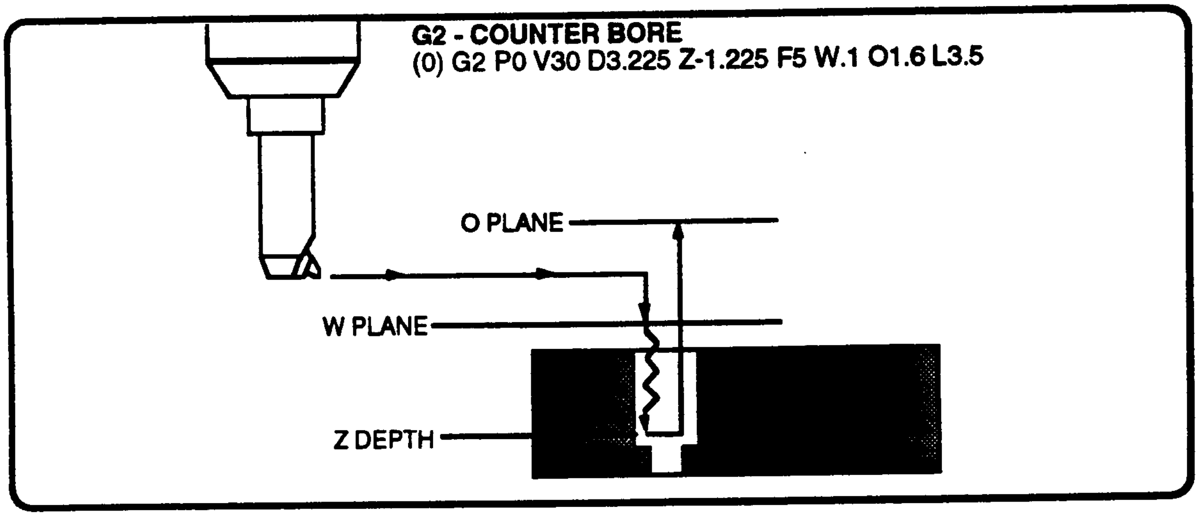


G1 - DRILL

```
(0) G1 X10 Y-2/ Z-1.25 F5 W.1 O1.75
```

G2 - COUNTERBORE CYCLE

The counterbore tool tip slows from rapid to feedrate F at reference plane W, continues the feed to Z, dwells L seconds, then rapids to the O plane (See illustration, **below**).

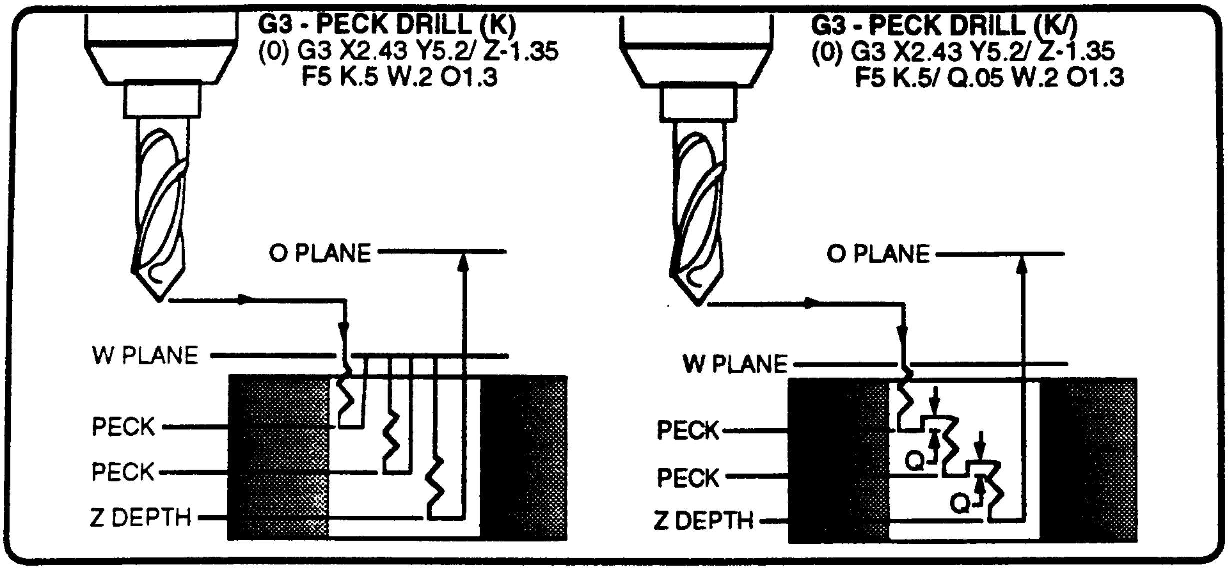


G2 - COUNTER BORE

```
(0) G2 P0 V30 D3.225 Z-1.225 F5 W.1 O1.6 L3.5
```

G3 - PECK DRILL CYCLE

The drill tip slows from rapid to feed F at reference plane W. Drill feeds K units from W into the part and retracts to finish the first peck. The retract motion is to W if K is absolute, or to clearance Q, if K is incremental. The next infeed begins at Q and continues through a second unit of K before another retract motion occurs. Each following peck similarly advances until the tip arrives at Z depth. The tool then rapids to O. If K is incremental and Q is not programmed, the backup will be .1" (2.5 mm) (See illustrations, **below**).

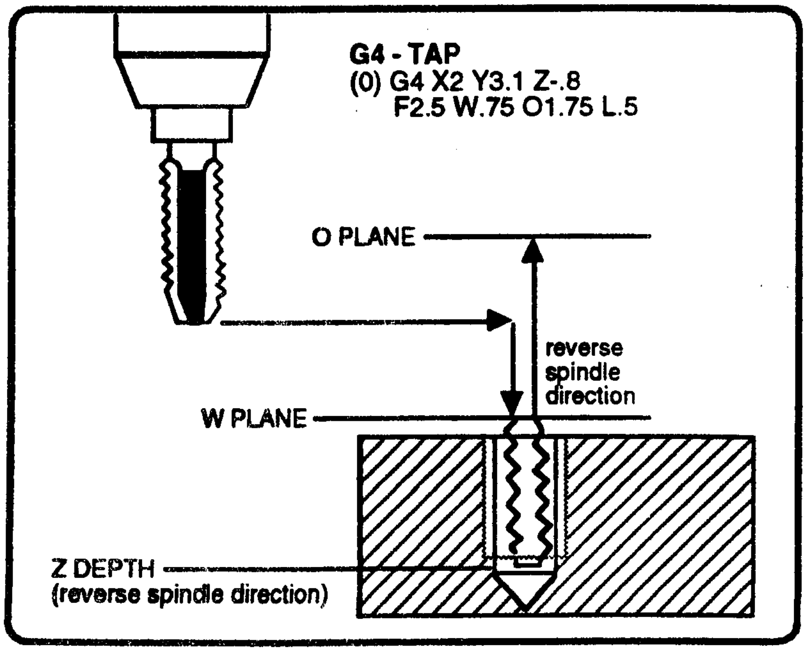


G3 - PECK DRILL (K)
(0) G3 X2.43 Y5.2/ Z-1.35
F5 K.5 W.2 O1.3

G3 - PECK DRILL (K/)
(0) G3 X2.43 Y5.2/ Z-1.35
F5 K.5/ Q.05 W.2 O1.3

G4 - TAP CYCLE

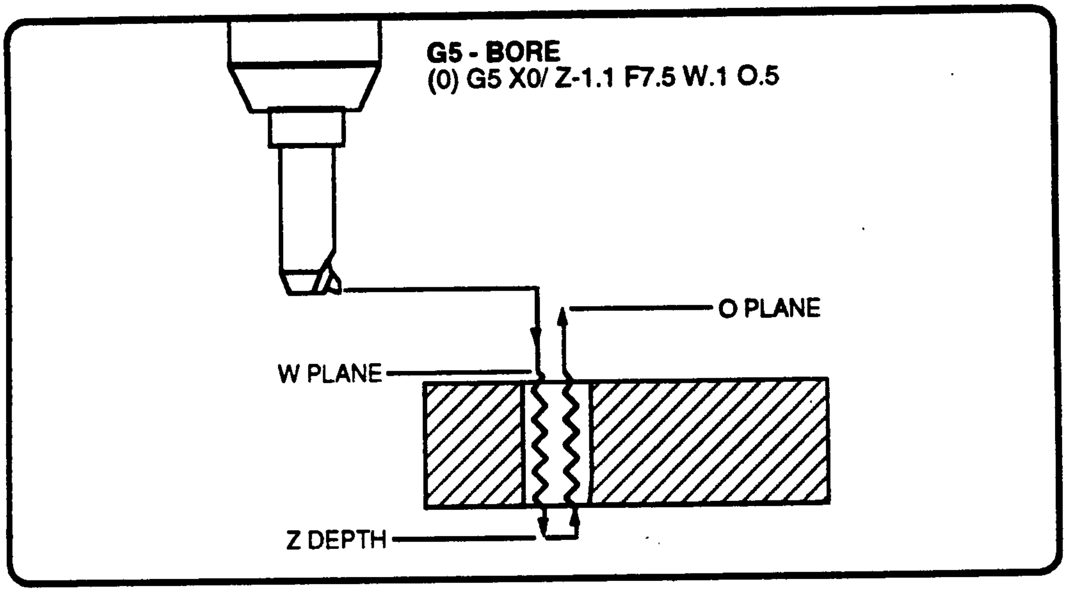
You need a floating tap holder and programmable reversible spindle. The spindle speed to feedrate ratio must be proportional to tap threads per inch (or mm). Spindle speed is set prior to the event. The tap slows to feed F at reference plane W, feeds to Z, then the spindle will reverse direction. L seconds of added dwell may be programmed if more reversal time is required. The tap will then feed back to W where the spindle reverses again, restoring your initial rotary direction. If a 2nd reference plane is active, the tool will now rapid to the O plane. If spindle reversal or spindle speed is not programmable, we recommend that you use the G5 cycle with a reversing tap holder.



G4 - TAP
(0) G4 X2 Y3.1 Z-.8
F2.5 W.75 O1.75 L.5

G5 - BORE CYCLE

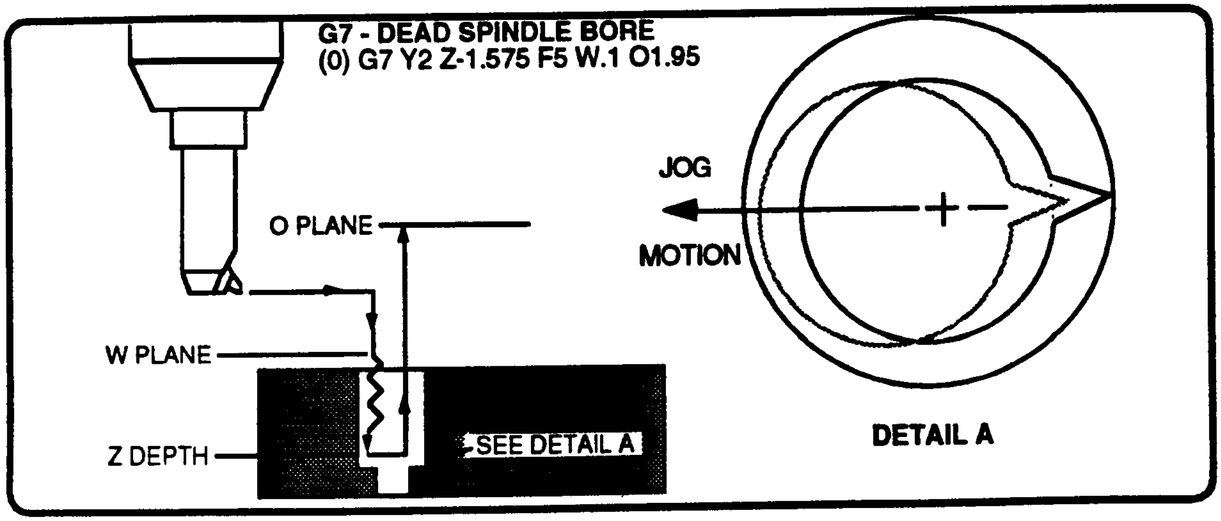
The bore tool tip slows to F at reference plane W, feeds to Z, feeds back to W, and rapids to the 2nd reference plane O. (See illustration **below**).



G7 - DEAD SPINDLE BORE CYCLE

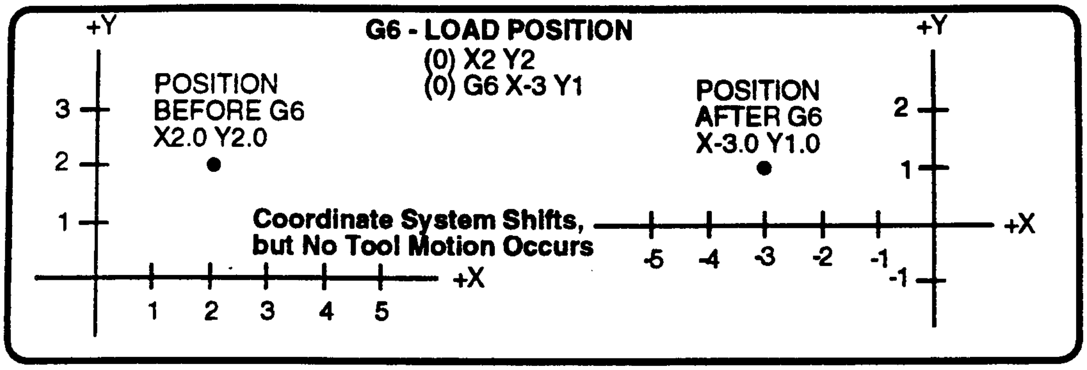
The bore tool tip slows to F at reference plane W, then feeds to Z (see illustration **below**). The cycle sends out M05 to halt the spindle, then M00 to halt the program. You can then press RETRACT to withdraw the tool from the hole. If the halted spindle turns freely, you can turn the tip to point along the X or Y axis. Then move to Jog mode and back the tip away from the hole's circumference (the clearance will depend on size of the the boring bar). Next, jog in Z, or move to Auto mode and press RETRACT to withdraw the tool from the hole.

To continue the cycle in Auto mode, press CYCLE START. The tool will rapid (+ or - direction) in Z to the O plane. If X or Y was jogged during the halt, the tool then rapids back to the XY hole center. The spindle then turns on in the original direction to end the cycle. If necessary, you can program a following Dwell event to give the spindle time to accelerate to the programmed speed



G6 - LOAD POSITION

A G6 event shifts axis origins and coordinates without causing any machine motion (See illustration below). The G6 event assigns its entered X, Y, Z coordinates to the current tool position. X, Y or Z entries must be absolute. Any axis not programmed keeps its last dimension. The axis shift is effective until replaced by a later G6 event or it is canceled by an EIA/ISO G99 block, End of Program M Code, or RESET when the control is not in cycle. A G6 event has no effect on the last G0-G5 or G7-G8 condition. The last G Code condition in this range remains modal.


G6 - LOAD POSITION
(0) X2 Y2
(0) G6 X-3 Y1
POSITION BEFORE G6 X2.0 Y2.0
POSITION AFTER G6 X-3.0 Y1.0
Coordinate System Shifts, but No Tool Motion Occurs
+Y
+X
1
2
3
4
5
-5
-4
-3
-2
-1
+Y
+X

A, B, C, U, V, W - AUXILIARY AXIS POSITION PROGRAMMING

Your machine supplier labels the optional Auxiliary axes A, B, C, U, V, or W.

A linear auxiliary axis endpoint may be a signed absolute or incremental entry.

A rotary axis endpoint is programmed in decimal degrees as an absolute or incremental angle. Use a + sign for ascending degrees rotation, or a - sign for descending degrees rotation, to a maximum 999.999 degrees incremental. (There is no windup.) The rotary display rolls over at the 359.999/000.000 degree division in either direction.

If a 1 or 5 degree (minimum increment) index axis is installed, a FORMAT ERROR alert appears if you try to enter an endpoint between the smallest index increments.

For G1-G5 or G7-G8, the G cycle will not begin until the X,Y and auxiliary axes (linear or rotary) are in position. A G cycle is always performed by the Z axis.

G8 - RIGID TAPPING

Rigid Tapping is an optional feature which allows tapping with a rigid tap holder. Your machine supplier can enable this feature if your machine's spindle can provide digital feedback signals to the control.

You must program the spindle speed in a prior M Function event. The control will calculate the tapping feedrate by using your last spindle speed and thread dimensions you program.

Rigid Tapping is a modal cycle which is programmed in Position or Bolt Circle event. A quick way to determine if your control has Rigid Tapping is by looking at the Position or Bolt Circle event screen. Rigid Tapping is installed if G8 RIGID TAP is listed on the screen when you press the G key.

REQUIREMENTS

To use the Rigid Tapping feature, you must have:

- A machine equipped with a DC spindle.
- The Rigid Tapping option.

F - TAP THREAD DIMENSIONS - PITCH OR LEAD

Program F to specify the pitch or lead of your tap. When programming in inches, enter a sign positive F value to define the tap's pitch in threads/inch. When programming in millimeters, program a sign negative F value to define the tap's lead in millimeters/thread.

The Lead/Pitch dimension F is modal to Rigid Tapping cycles **only**, and **does not** cancel the last F feedrate. To maintain program clarity, you should program the Lead/Pitch F entry in every Rigid Tapping event.

The modal feedrate F will be reactivated when a non-Rigid Tap event or block executes.

FEEDRATE FORMULAS

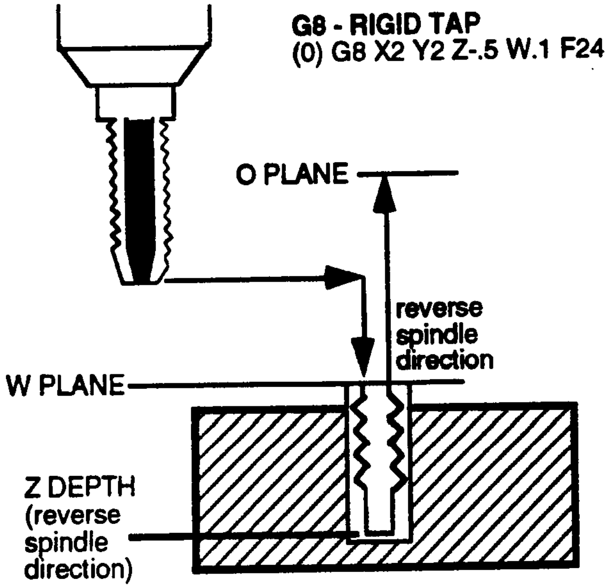
The control uses the following equations to calculate the tapping feedrate. The spindle speed in revs/min must be programmed in a prior M Function event.

Inch feedrate: +F defines the pitch in threads/inch:

Z feedrate = (spindle revs/min) / (threads/in) = in/min

Millimeter feedrate: -F defines the lead in mm/thread:

Z feedrate = (spindle revs/min)*(mm/thread) = mm/min



ORDER OF G8 CYCLE MOTION

1) Spindle direction and speed must be set in a previous M Function event.
2)X, Y, and any auxiliary axes rapid to their programmed endpoints.
3)The Z axis rapids to the reference plane W. During or immediately after this move, the spindle will stop then resume at the original speed and direction.
4) Z feeds to the programmed Z depth. The FEEDRATE OVERRIDE Pot is disabled at this time.
5) When Z reaches it's programmed depth, the spindle reverses direction.
6) Z retracts at feed to the reference plane W.
7) An M05 is output to stop the spindle.
8)M03 or M04 is output to start the spindle in original direction.
9) Tool rapids to O plane.

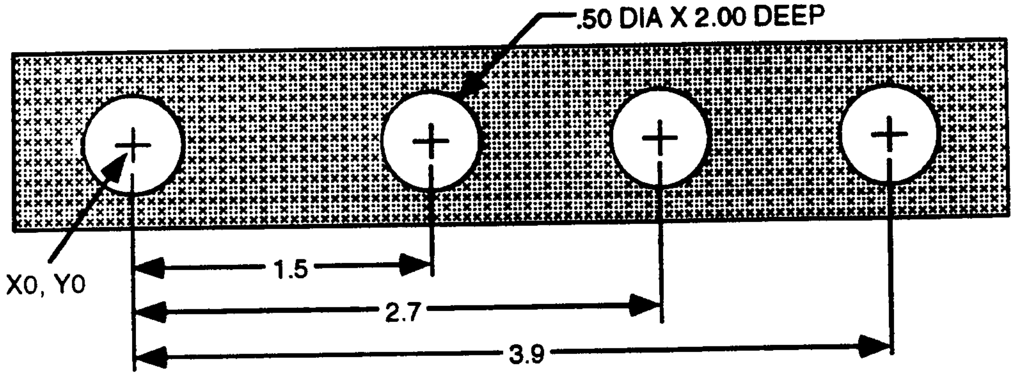
CANCELING RIGID TAPPING

The G8 Rigid Tapping cycle is canceled when G0-G5 or G7 executes in a Position or Bolt Circle event, M02 or M30 executes, or you press the RESET button when the control is not in cycle.

POSITION EVENT EXAMPLE

The **following** example drills four holes using the G1-Drill cycle in several Position events. Both part programs will cut the same part. The left program uses absolute dimensions to locate successive holes; the right program uses incremental dimensions to locate the holes. The program begins by activating the offsets for tool number 1 and turning the spindle on. Next, N20 positions the tool over the first hole. N30 then drills the first hole. Notice that the command X0/ causes no tool motion in XY, but allows the G1 cycle to be performed in Z. Next, N40 positions the tool over and drills the second hole. Since G1 is modal, all following Position events will perform the drill cycle programmed in N30. N40 and N50 drill the third and fourth holes. N60 ends the program with an M30 command. The M30 command automatically cancels the drill cycle.

Since the work surface is at a uniform height and there are no clamps to be concerned about, this program does not use a 2nd reference plane. The control will automatically set it equal to W.

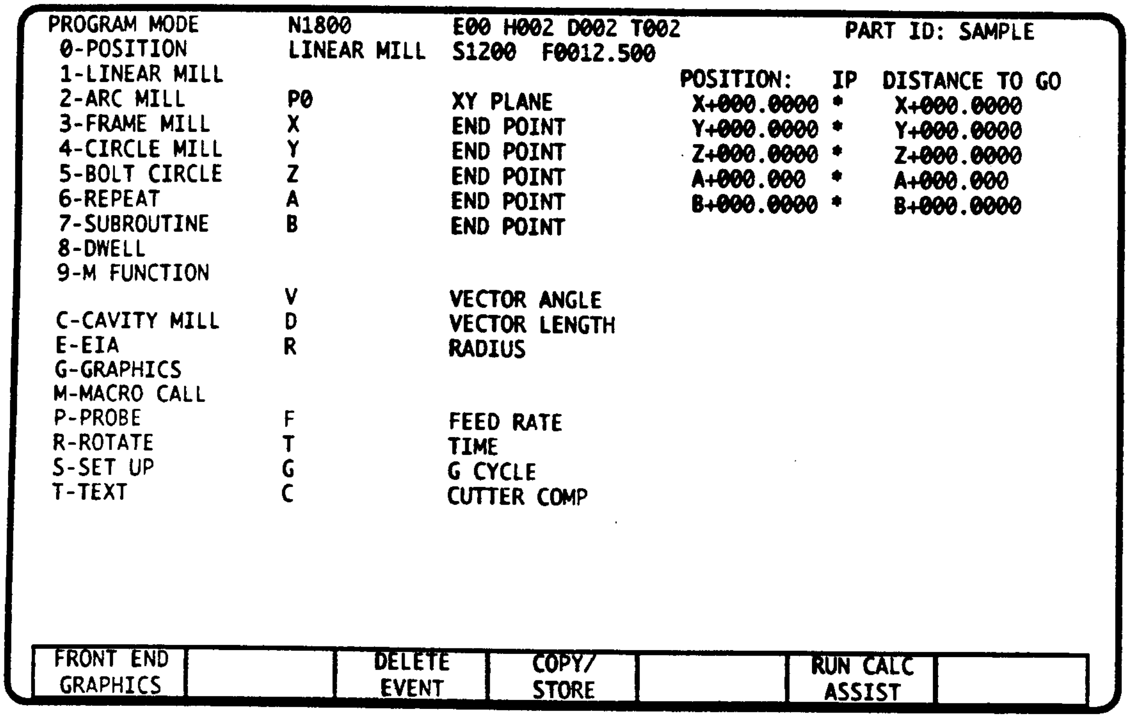


Absolute Dimensions

```
N05  (G)  X-1 Y-1 Z-2.5 H8
N10  (9)  T1 H1 D1 M3 S1600
N20  (0)  X0 Y0 Z.5 G0
N30  (0)  X0/ G1 Z-2 F10 W.1
N40  (0)  X1.5
N50  (0)  X2.7
N60  (0)  X3.9
N70  (9)  M30
```

Incremental Dimensions

```
N05  (G)  X-1 Y-1 Z-2.5 H8
N10  (9)  T1 H1 D1 M3 S1600
N20  (0)  X0 Y0 Z.5 G0
N30  (0)  X0/ G1 Z-2 F10 W.1
N40  (0)  X1.5/
N50  (0)  X1.2/
N60  (0)  X1.2/
N70  (9)  M30
```



```
PROGRAM MODE       N1800         E00 H002 D002 T002          PART ID: SAMPLE
 0-POSITION        LINEAR MILL   S1200  F0012.500
 1-LINEAR MILL                                     POSITION:    IP  DISTANCE TO GO
 2-ARC MILL        P0            XY PLANE           X+000.0000 *     X+000.0000
 3-FRAME MILL      X             END POINT          Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL     Y             END POINT          Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE     Z             END POINT          A+000.000  *     A+000.000
 6-REPEAT          A             END POINT          B+000.0000 *     B+000.0000
 7-SUBROUTINE      B             END POINT
 8-DWELL
 9-M FUNCTION
                   V             VECTOR ANGLE
 C-CAVITY MILL     D             VECTOR LENGTH
 E-EIA             R             RADIUS
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE           F             FEED RATE
 R-ROTATE          T             TIME
 S-SET UP          G             G CYCLE
 T-TEXT            C             CUTTER COMP
```

LINEAR MILL EVENT DATA

(Refer to the **upper** screen display.)

Parameter	Description	Entry
N	Sequence Number	Required
P	Plane Select P0=XY, P1=YZ, P2=ZX	Defaults to P0
		Cartesian Endpoint
X or X/	X axis endpoint	Endpoint, see topic below
Y or Y/	Y axis endpoint	Endpoint, see topic below
Z or Z/	Z axis endpoint	Endpoint, see topic below
		Polar Endpoint
V	Vector angle	Endpoint, see topic below
D	Vector distance	Endpoint, see topic below
F	Direct feedrate on cutting path	Modal
T	Inverse Time (-1/minutes)	Optional, not modal
R	Auto Radius	Optional
C	Cutter Compensation C0=off, C1=left, C2=right	Modal
G	Wait for In Position G blank=don't wait, G0=wait	Defaults to G blank

LINEAR MILL EVENT - INTRODUCTION

This event programs a linear move at a specified feedrate or period of time. The start point is the tool's position at the start of the event. The endpoint can be programmed with Cartesian coordinates (absolute or incremental), polar coordinates, or one Cartesian coordinate and the angle. This event can be used to turn cutter radius compensation on and off, blend an auto radius with a following Linear Mill event, and specify a "wait for in position" between successive events. The feedrate can be be entered in ipm, mmpm, or minutes.

CARTESIAN or POLAR ENDPOINT DESCRIPTION

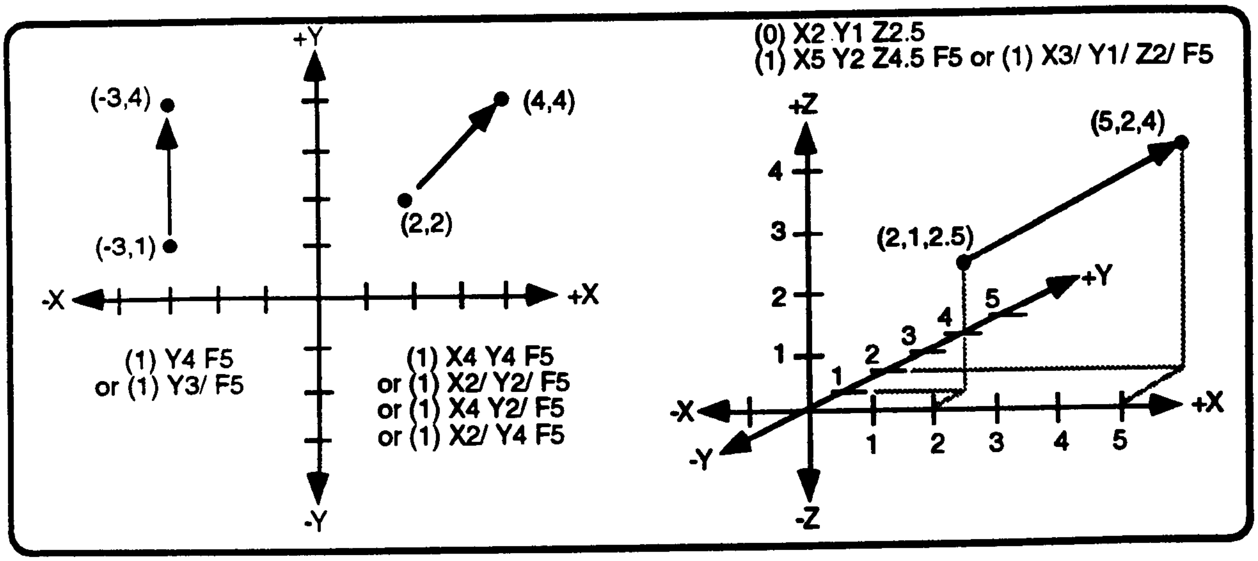
The endpoint can be completely described with Cartesian coordinates or Polar coordinates. A two axis motion can be programmed with one Cartesian endpoint and an angle. If you run your part program through Calc-Assist (refer to the Calc-Assist part to this manual), further linear moves can be programmed with various combinations of Cartesian and Polar data.

PLANE SELECT

Plane Select establishes the plane for cutter radius compensation offset and chooses the plane for polar entries. The default is to P0, the XY plane. The standard plane for use of cutter radius compensation is the XY plane, but in cases where a ballnose end mill or other special cutter is used, compensation may be set to the YZ or ZX plane.

X, Y, Z ENDPOINT - CARTESIAN ENTRY

Enter dimensions for each axis that must move to reach the milling endpoint. Each axis endpoint may be programmed as an absolute or incremental entry. Absolute and incremental entries may appear in the same event. The resulting motion will be a move at feedrate F from the tool's start point straight to your programmed endpoint. Refer to the illustrations, below.

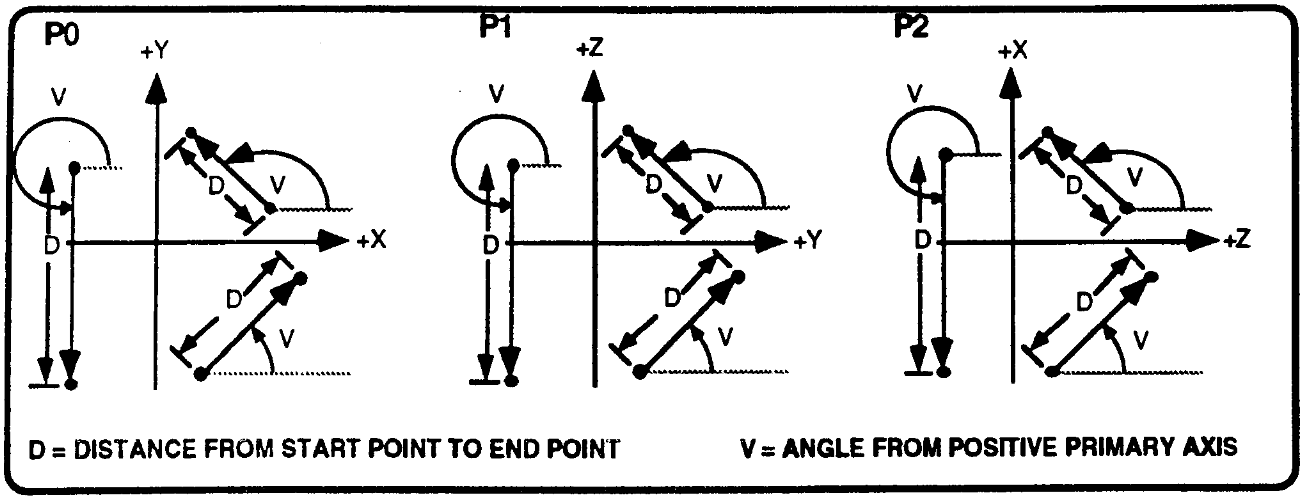


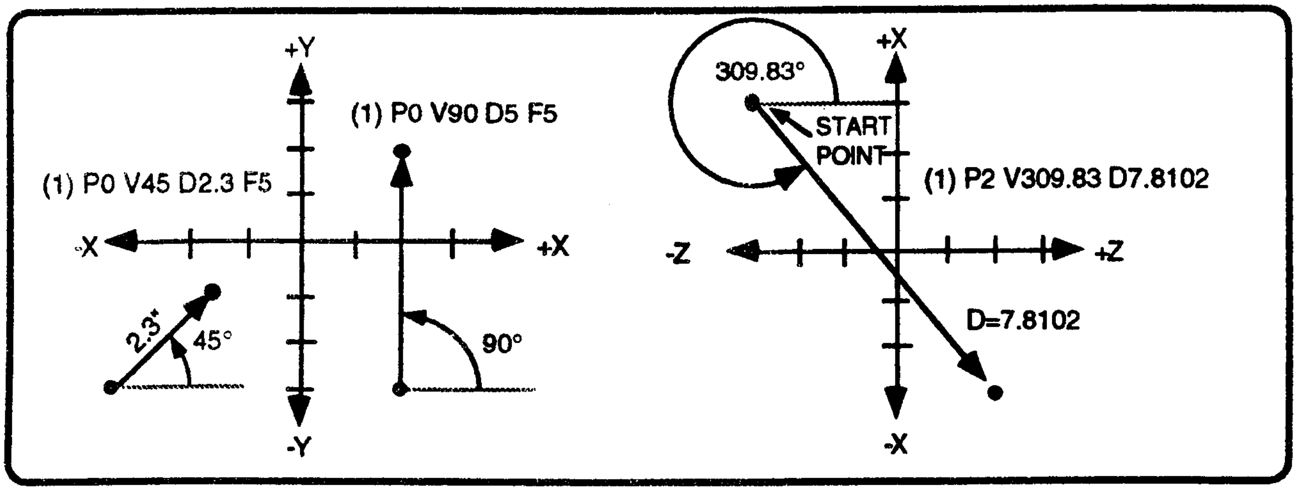
P, V, D - POLAR ENDPOINT DESCRIPTION

Polar coordinates (V and D) define a one or two axis move that occurs within the selected plane (P0, P1, P2). D is the absolute, unsigned distance of the move, measured along angle V from the tool's start point to the tool's endpoint. V is the angle of the move, measured from a zero degree reference line[1] that starts from the tool's start point and opens counterclockwise[2] to the tool's endpoint.

You can program a three axis move using polar coordinates by defining the move in the selected plane P with angle V and distance D, then entering the endpoint of the out of plane axis with a Cartesian coordinate.

Note: The angle V entry is programmed with the letter "J" if your control is equipped with an auxiliary axis that is programmed with the letter "V". Refer to Part 1-Section 2.


P0
+Y
V
D
V
D
+X
D
V
P1
+Z
V
D
V
D
+Y
D
V
P2
+X
V
D
V
D
+Z
D
V
D = DISTANCE FROM START POINT TO END POINT
V = ANGLE FROM POSITIVE PRIMARY AXIS



[1] Zero degree reference lines run parallel to, and in the same direction as, the positive primary axis in the selected plane. The primary axis in the XY plane is X, in the YZ plane is Y, and in the ZX plane is Z. The other axis in each plane is referred to as the secondary axis.

[2] Counterclockwise is the direction that provides the shortest angle from the positive primary axis to the positive secondary axis in the selected plane.

F - FEEDRATE

Feedrate is modal with all following event types which program feedrate F for their respective operations. In this event, feedrate F programs the direct feed along the linear path.

T - INVERSE TIME

As an alternative to programming the feedrate F for an event, you can specify the number of minutes that are required to complete the event's contouring motion. This is most useful when linear and rotary axis feeds are programmed together.

To program the feed time, enter the negative inverse[3] of the time (in minutes) required to complete the contour. For example: enter T-0.5 to program a feed that must take 2 minutes to complete; enter T-6.0 to program a feed that must take 10 seconds (1/6 minutes) to complete. This entry format allows you the fine resolution and range you may require to program feed from .006 seconds (program T-9999.99) to 100 minutes (program T-.01).

When an Inverse Time event executes, the control will ensure that all axes feed for the specified duration and arrive simultaneously at their programmed endpoints. The combined feedrate for the move is displayed in the Auto and Single Event modes.

T is not modal; it must be programmed in every block which uses Inverse Time. The modal feedrate F will be reactivated on the next event that does not program Inverse Time.

F and T may not be programmed in the same event. One entry erases the other.

A, B, C, U, V, W - AUXILIARY AXES PROGRAMMING

When one, two, or three auxiliary axes are installed with the system, your machine supplier names the axis identifiers (A, B, C, U, V, W) and defines the axis type as rotary or linear.

When contouring auxiliary axes are programmed with the primary (X, Y, Z) axes in this event, the control's modal feedrate calculation is restricted to the linear path of the primary axes. On execution, auxiliary axes begin motion and reach their endpoints simultaneously with all other axes. The primary axes (X, Y, Z) will be slowed, if necessary, so as not to exceed your machine supplier's assigned maximum contouring rate for a rotary contouring axis.

If only auxiliary contouring axes are programmed in this event, the feedrate is not modal and must be programmed in each similar event. The chart below shows how the programmed feedrate is interpreted by the control depending on the types of contouring axes installed in your system.

1 linear axis programmed	F is ipm (mmpm).
1 rotary axis programmed	F is deg/min.
2 linear axes programmed	F is interpolated in ipm (mmpm).
2 rotary axes programmed	F is interpolated in deg/min.
1 linear, 1 rotary programmed	F commands linear axis, rotary feed is calculated to reach endpoint with linear axis.

[3] The inverse of a number (x) is equal to 1 divided by the number, or 1/x. The Inverse Time T you enter must be sign negative.

A linear contouring axis is programmed with a signed absolute or signed incremental endpoint. In a very few applications, a slaved axis may be ganged with Z. A slaved linear axis moves in tandem with Z axis milling motion, and may not be independently programmed.

A rotary index axis may not be programmed in the Linear Mill event.

A rotary contouring axis is programmed in decimal degrees as a signed absolute or signed incremental angle. Signed absolute angles may range from -360° to +360°, not inclusive. The sign represents the direction of the move. Use a + sign for ascending degrees rotation (e.g. from 0° to 30°), or a - sign for descending degrees rotation (e.g. from 30° to 0°). The number represents the absolute angle for the rotary axis. Example:

```
N10 (0) A0 X0 Y0 Z.1        - start position for all axes
N20 (1) A30 F360            - move axis 30° from 0° to 30°
N30 (1) A60                 - move axis 30° from 30° to 60°
N40 (1) A-50                - move axis 10° from 60° to 50°
N50 (1) A0                  - move axis 310° from 50° to 360°
```

If you wanted N50 to move the A axis directly to 0°, program A-0 (instead of A+0).

Signed incremental angles may range from -999.999/ to +999.999/.

CUTTER RADIUS COMPENSATION

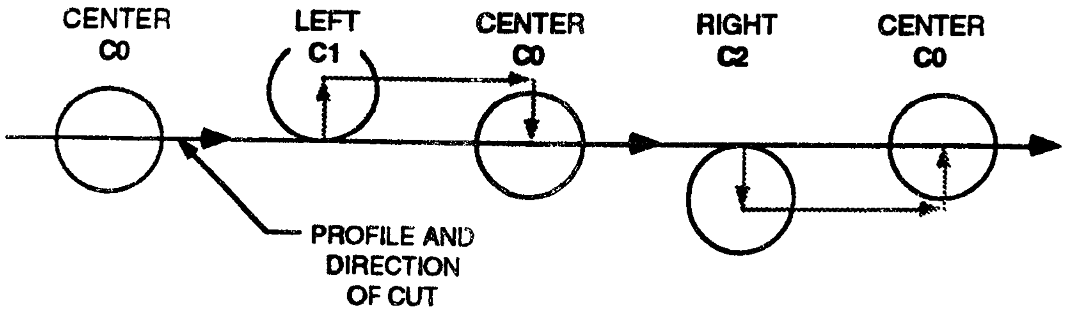
Cutter Radius Compensation (CRC) is a utility that simplifies profile programming by allowing you to program part coordinates without considering the tool's size. When you cycle a part with CRC properly programmed, the control will automatically offset the programmed path to compensate for the active tool's radius, and command the tool to move along the path that smoothly blends each event with the following Linear or Arc Mill event. CRC may only be used in an uninterrupted sequence of Linear Mill and Arc Mill events.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

C - TURN CRC ON OR OFF

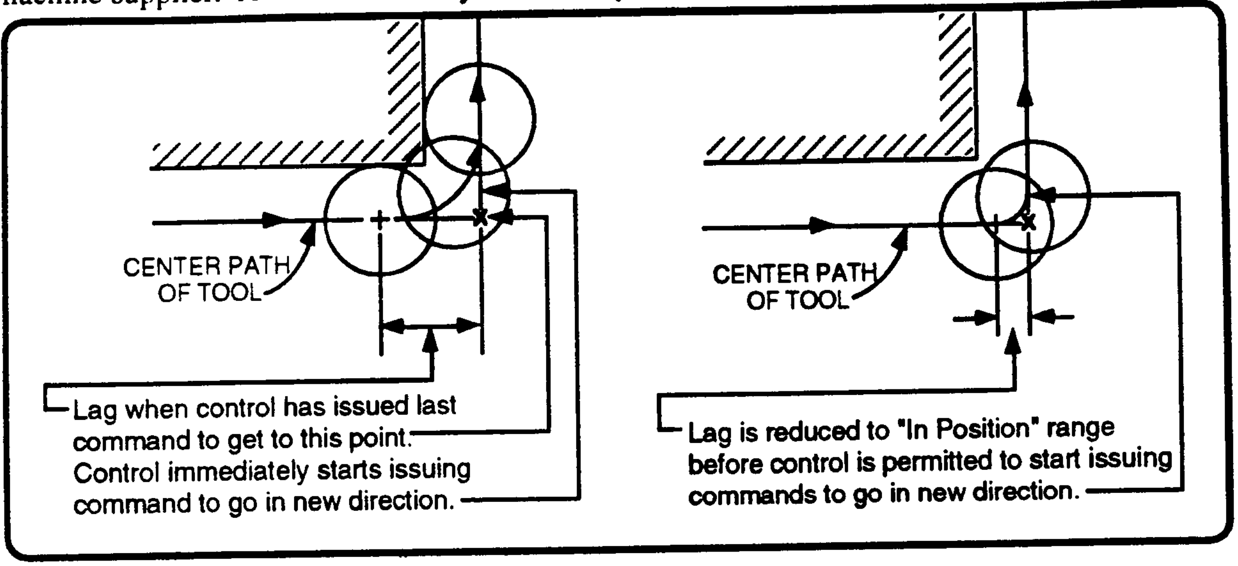
C0 cancels any active cutter compensation. C1 turns cutter compensation on to the left of the profile; C2 turns cutter compensation on to the right of the profile. Turning cutter compensation on or off will offset the tool a distance equal to the tool's radius in a direction that is perpendicular to the profile. CRC is modal with all following Linear and Arc Mill events. Refer to the lower illustration.

Cutter compensation is described in Part 4 - Section 8 of this manual.



G0 - WAIT FOR IN POSITION

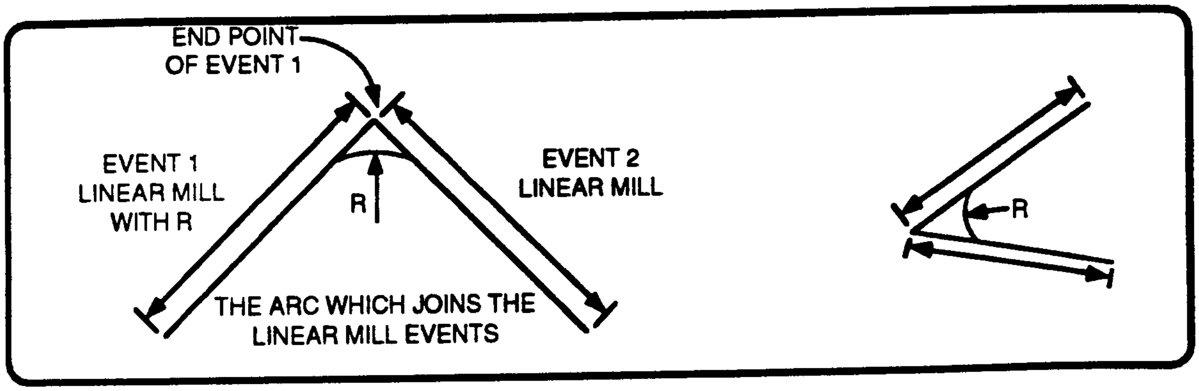
You would normally program a series of linear and arc moves without using the G0 entry. However, if a high feedrate shows signs of corner rounding in a test part, program G0-Wait for In Position to restore the corner, **below**. G0 is nonmodal and operative only in the event in which it is programmed. If G is left blank (not programmed), the next event begins right after the last feed increment is output to the servos. Note that the next motion output can occur while there is still following error to be resolved. If G0 is programmed, the next event will not begin until the following error is reduced to less than the "In Position" range established by the machine supplier. A noticeable delay will usually occur during the Wait for In Position.


CENTER PATH OF TOOL
Lag when control has issued last command to get to this point.
Control immediately starts issuing command to go in new direction.
CENTER PATH OF TOOL
Lag is reduced to "In Position" range before control is permitted to start issuing commands to go in new direction.

R - AUTO RADIUS

R may be programmed to mill an arc profile joining two Linear Mill events. Refer to the following illustration. The end point for the first event is programmed where the intersection of the two linear profiles would normally occur. The first event also prpgrams a radius dimension R to establish an arc that joins the two profiles. The second event is programmed in the normal manner. Refer to the figures **below**.

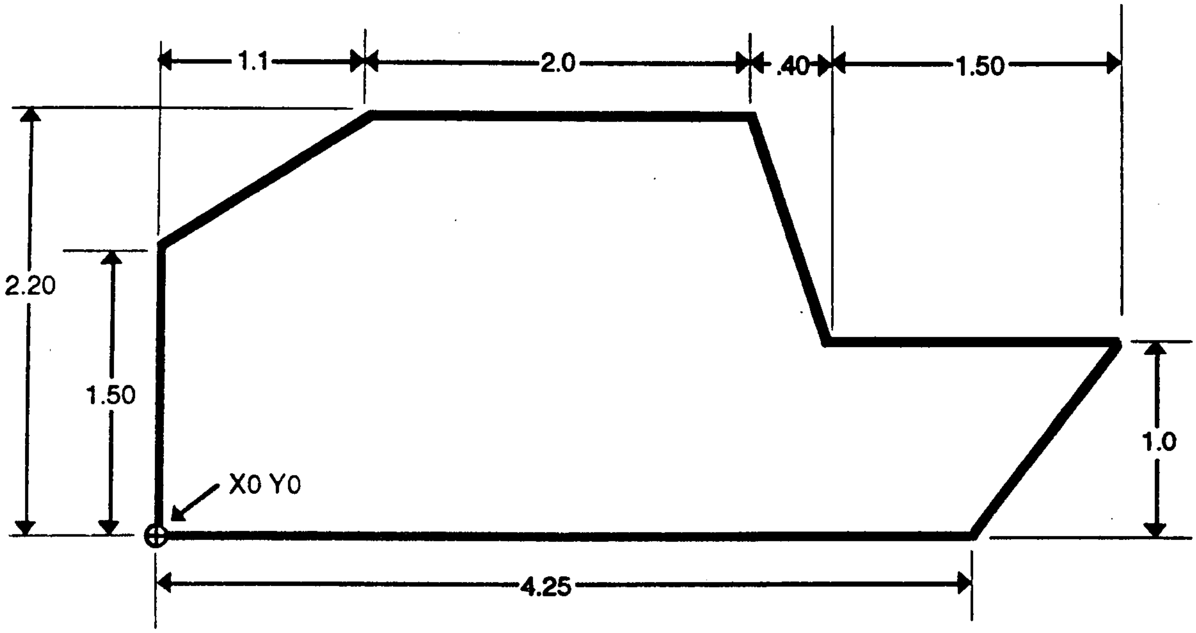
If desired, a second Linear Mill event may program an auto radius to join with a third, and so on, as long as both Linear Mill events are in the same plane. In the Linear Mill events to be joined, no entry is permitted for the out of plane axis. No other event types may be programmed between Linear Mill events which are joined in this way. When a Linear Mill event programs G0 "Wait for In Position" as well as auto radius R, the wait occurs between the first linear motion and the arc motion.


END POINT OF EVENT 1
EVENT 1 LINEAR MILL WITH R
R
EVENT 2 LINEAR MILL
THE ARC WHICH JOINS THE LINEAR MILL EVENTS
R

EXAMPLES:

Note that the following two examples are only intended to demonstrate different methods of using Linear Mill events; they are not intended to be complete part programs since they do not include a Z axis depth nor do they include commands to identify a tool number or spindle speed.

The following example shows different methods of programming the same part. The **left** solution uses only absolute dimensions, the **middle** solution uses only incremental dimensions, and the **right** solution combines absolute and incremental dimensions in the same program (sometimes within the same block). The effort required for the **right** solution is less than the other two solutions since all dimensions are taken directly from the part print. The other two solutions require some minor calculations by the programmer.



ABSOLUTE ONLY

```
N.5  (G)  X-1 Y-1 H9
N01  (0)  X0 Y0 Z.1
N02  (1)  Y1.5
N03  (1)  X1.1 Y2.2
N04  (1)  X3.1
N05  (1)  X3.5 Y1.0
N06  (1)  X5.0
N07  (1)  X4.25 Y0
N08  (1)  X0
N09  (9)  M30
```

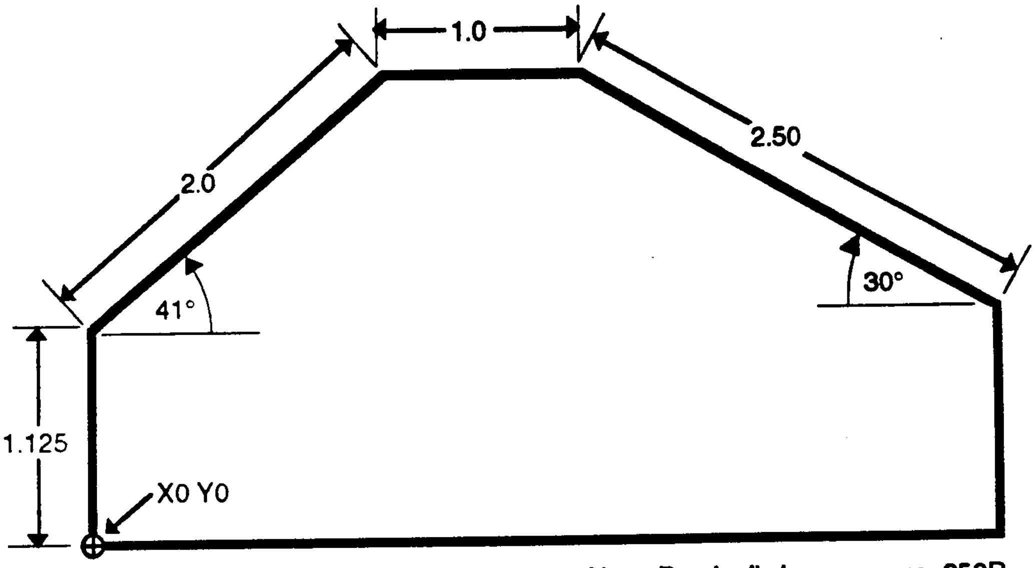
INCREMENTAL ONLY

```
N.5  (G)  X-1 Y-1 H9
N01  (0)  X0 Y0 Z.1
N02  (1)  Y1.5/
N03  (1)  X1.1/ Y.7/
N04  (1)  X2.0/
N05  (1)  X.4/ Y-1.2/
N06  (1)  X1.5/
N07  (1)  X-.75/ Y-1.0/
N08  (1)  X-4.25/
N09  (9)  M30
```

MIX ABS. and INC

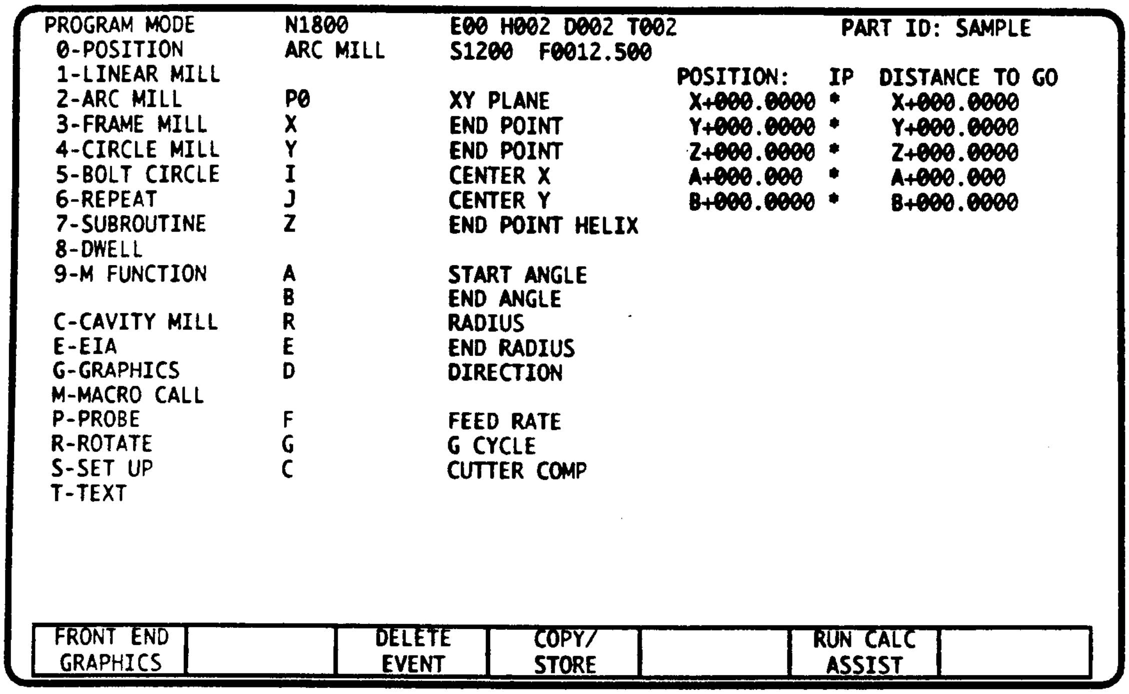
```
N.5  (G)  X-1 Y-1 H9
N01  (0)  X0 Y0 Z.1
N02  (1)  Y1.5
N03  (1)  X1.1 Y2.2
N04  (1)  X2.0/
N05  (1)  X.4/ Y1
N06  (1)  X1.5/
N07  (1)  X4.25 Y0
N08  (1)  X0
N09  (9)  M30
```

The example **below** demonstrates how to use polar dimensions and the control's auto-radius feature within a part program. Recall that the auto-radius is added to an event to command the control to automatically blend a radius between consecutive Linear Mill events.



Note: Break all sharp corners .050R

```
N.5  (G) X-1 Y-1 V4        - Graphics Setup event
N01  (0) X0 Y-.5 Z.1       - position to start point below part
N02  (1) Y1.125 R.05       - program first move and .05 blend with next event
N03  (1) V41 D2 R.05       - polar move at 41° with a length of 2.0 inches
N03  (1) V0 D1 R.05        - polar move at 0.0°, with a length of 1.0 inch
N04  (1) V330 D2.5 R.05    - polar move at 330°. (angle is measured from tool's start point.)
N05  (1) Y0 R.05           - Cartesian endpoint definition
N06  (1) X0 R.05           - Cartesian endpoint definition.
N07  (1) Y.5 R.25          - Cartesian endpoint to break X0, Y0 corner then begin .25"
                             tangential exit with next event.
N08  (1) X-.5              - Tangential exit from part.
N99  (9) M30               - End of Program
```



```
PROGRAM MODE        N1800        E00 H002 D002 T002                PART ID: SAMPLE
 0-POSITION         ARC MILL     S1200   F0012.500
 1-LINEAR MILL                                     POSITION:   IP  DISTANCE TO GO
 2-ARC MILL         P0           XY PLANE           X+000.0000 *    X+000.0000
 3-FRAME MILL       X            END POINT          Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL      Y            END POINT          Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE      I            CENTER X           A+000.000  *    A+000.000
 6-REPEAT           J            CENTER Y           B+000.0000 *    B+000.0000
 7-SUBROUTINE       Z            END POINT HELIX
 8-DWELL
 9-M FUNCTION       A            START ANGLE
                    B            END ANGLE
 C-CAVITY MILL      R            RADIUS
 E-EIA              E            END RADIUS
 G-GRAPHICS         D            DIRECTION
 M-MACRO CALL
 P-PROBE            F            FEED RATE
 R-ROTATE           G            G CYCLE
 S-SET UP           C            CUTTER COMP
 T-TEXT
```

ARC MILL EVENT - INTRODUCTION

This event mills an arc which may extend to 360°. The tool's start position locates the arc's start point. Since the control uses the start position in arc calculations, the tool must be located on the arc path at the start of the event.

You can program the arc in Cartesian coordinates (absolute or incremental) or polar coordinates. Cartesian coordinates require endpoint, arc center, direction, and feed. Absolute and incremental dimensions can be programmed in the same event.

Polar coordinates require arc radius, start angle, end angle, direction, and feed.

This event can be used to turn cutter radius compensation on and off, cut a Z advance helix and a variety of cam profiles, and specify a "wait for in position" after the event.

ARC MILL EVENT DATA

This event is initially set to program arc motion in the XY plane. Program P1 or P2 to change the arc motion to the YZ or ZX planes. After you select the plane (P0, P1, or P2), the Arc Mill event will only allow endpoint and center coordinates that are located in that plane.

Parameter	Description	Entry
N	Sequence Number	Required
P	Plane Select (P0=XY, P1=YZ, P2=ZX)	Defaults to P0
		Cartesian Arc
X or X/	X axis endpoint	Required in P0 or P2
Y or Y/	Y axis endpoint	Required in P0 or P1
Z or Z/	Z axis endpoint (Z axis advance in P0)	Required in P1 or P2
I or I/	X axis arc center	Required in P0 or P2
J or J/	Y axis arc center	Required in P0 or P1
K or K/	Z axis arc center	Required in P1 or P2
D	Arc Direction (D0=CW, D1=CCW)	Defaults to D0
		Polar Arc
A	Start Angle	Required for polar
B	End Angle	Required for polar
R	Arc Radius	Required for polar
D	Arc Direction (D0=CW, D1=CCW)	Defaults to D0
E	End Radius (Cam Mill)	Optional
		Miscellaneous entries
F	Direct feedrate on cutting path	Modal
C	Cutter Compensation (C0=off, C1=left, C2=right)	Modal
G	Wait for In Position (G blank=don't wait, G0=wait)	Defaults to G blank
Z	Z Axis Advance (P0 helix)	Optional

CARTESIAN or POLAR ENDPOINT DESCRIPTION

An arc can be completely described by programming Cartesian or polar coordinates. If you run your part program through Calc-Assist, further arc descriptions can be programmed by combining Cartesian and polar coordinates (refer to the Calc-Assist part of this manual).

P - PLANE SELECT

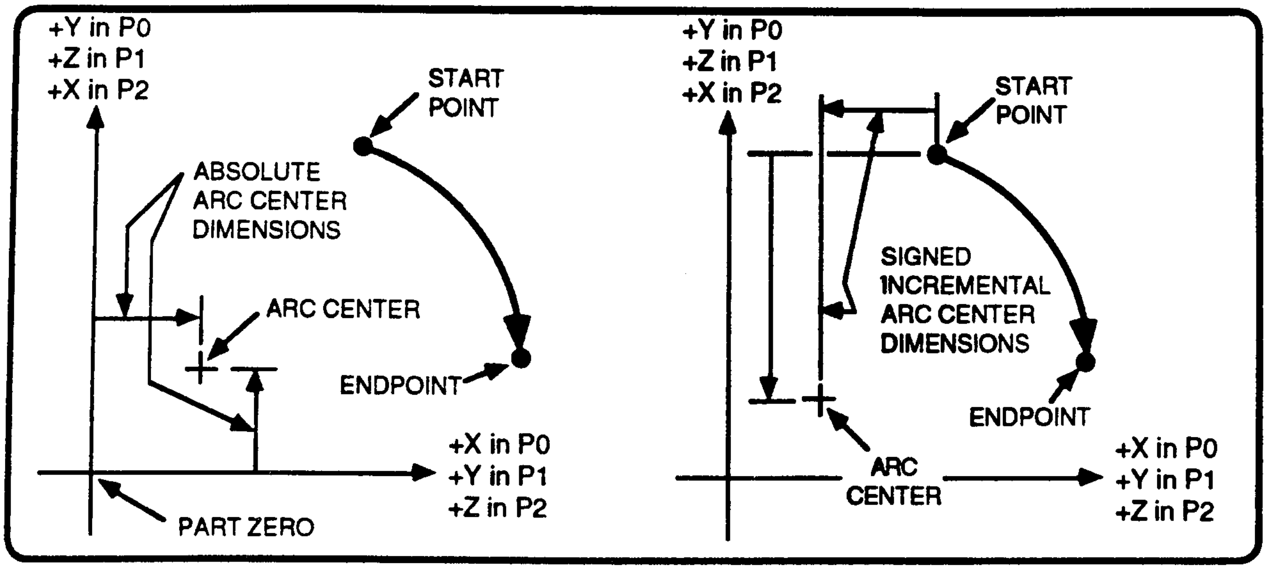
Plane Select identifies which pair of axes must move to mill the arc, and also chooses the plane for cutter offset when you are using cutter radius compensation. P0, the XY plane, is initially selected in each new event.

D - ARC DIRECTION

The control initially selects D0 (clockwise) milling motion around the arc center. The cutting direction (CW or CCW) around an arc is established by looking toward the arc plane from the positive axis that is not included in the the arc's plane (e.g. an arc's direction in the XY plane is properly referenced by viewing it from the +Z axis).

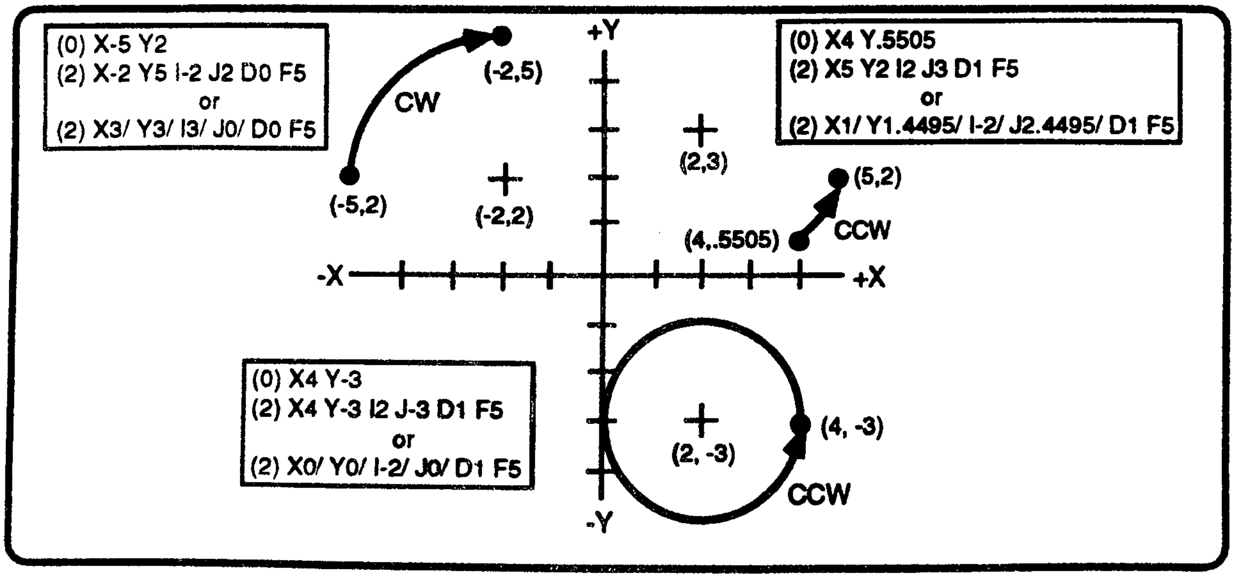
I, J, K ARC CENTER - CARTESIAN ENTRY

Enter both arc axis dimensions in the plane to define the arc center. An absolute or incremental entry is required for each axis. An absolute entry is the arc center's signed axis coordinate. An incremental entry is the signed distance from the start point to the arc center along the axis. Absolute and incremental entries may appear in the same event. In the plane P0, program arc center I on the X axis and arc center J on the Y axis. In plane P1, program arc center J on the Y axis and arc center K on the Z axis. In plane P2, program arc center K on the Z axis and arc center I on the X axis. Refer to illustrations below.



X, Y, Z ENDPOINT - CARTESIAN ENTRY

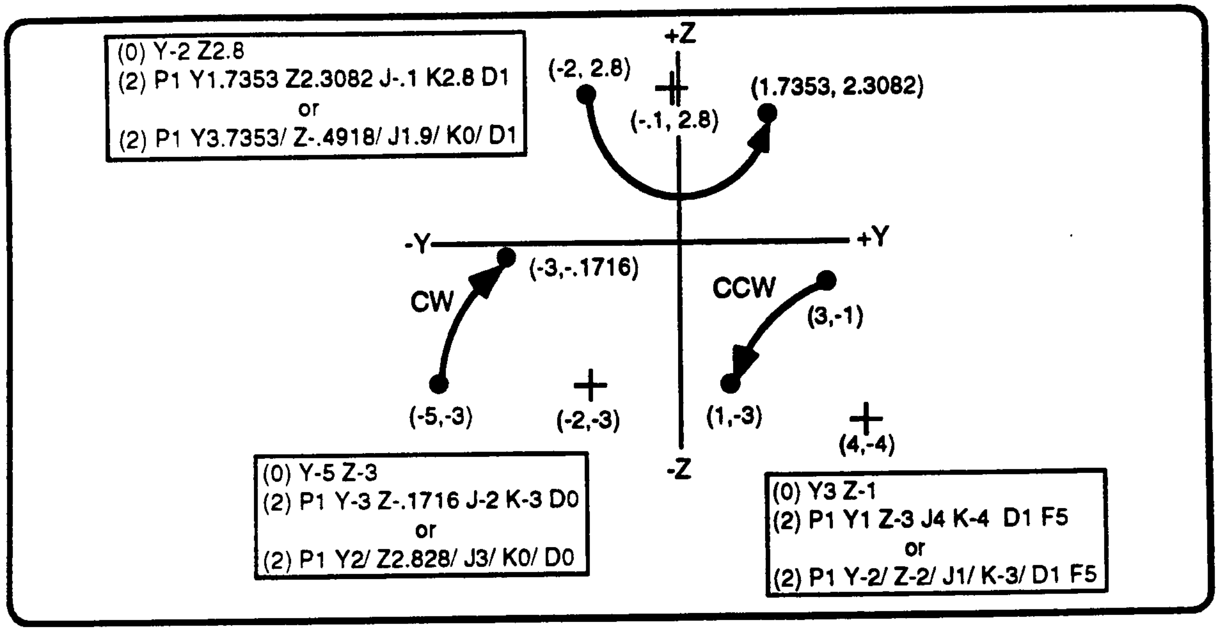
Enter both axis dimensions in the plane to define the arc endpoint. An absolute or incremental entry is required for each axis. An absolute entry is the endpoint's signed axis coordinate. An incremental entry is the signed distance from the start point to the endpoint along the axis. Absolute and incremental entries may appear in the same event. In plane P0, program the X and Y endpoints. In P1, program Y and Z. In P2, program Z and X. Refer to illustrations, below and following page.



```
(0) X-5 Y2
(2) X-2 Y5 I-2 J2 D0 F5
or
(2) X3/ Y3/ I3/ J0/ D0 F5
```

```
(0) X4 Y.5505
(2) X5 Y2 I2 J3 D1 F5
or
(2) X1/ Y1.4495/ I-2/ J2.4495/ D1 F5
```

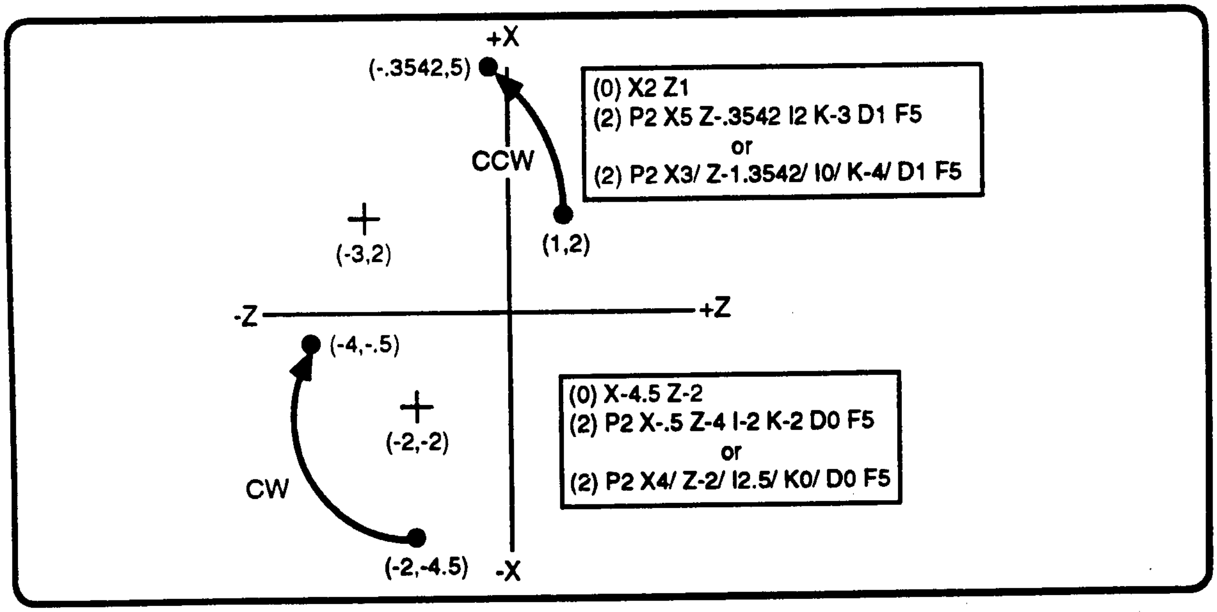
```
(0) X4 Y-3
(2) X4 Y-3 I2 J-3 D1 F5
or
(2) X0/ Y0/ I-2/ J0/ D1 F5
```



```
(0) Y-2 Z2.8
(2) P1 Y1.7353 Z2.3082 J-.1 K2.8 D1
        or
(2) P1 Y3.7353/ Z-.4918/ J1.9/ K0/ D1
```

```
(0) Y-5 Z-3
(2) P1 Y-3 Z-.1716 J-2 K-3 D0
        or
(2) P1 Y2/ Z2.828/ J3/ K0/ D0
```

```
(0) Y3 Z-1
(2) P1 Y1 Z-3 J4 K-4  D1 F5
        or
(2) P1 Y-2/ Z-2/ J1/ K-3/ D1 F5
```



```
(0) X2 Z1
(2) P2 X5 Z-.3542 I2 K-3 D1 F5
        or
(2) P2 X3/ Z-1.3542/ I0/ K-4/ D1 F5
```

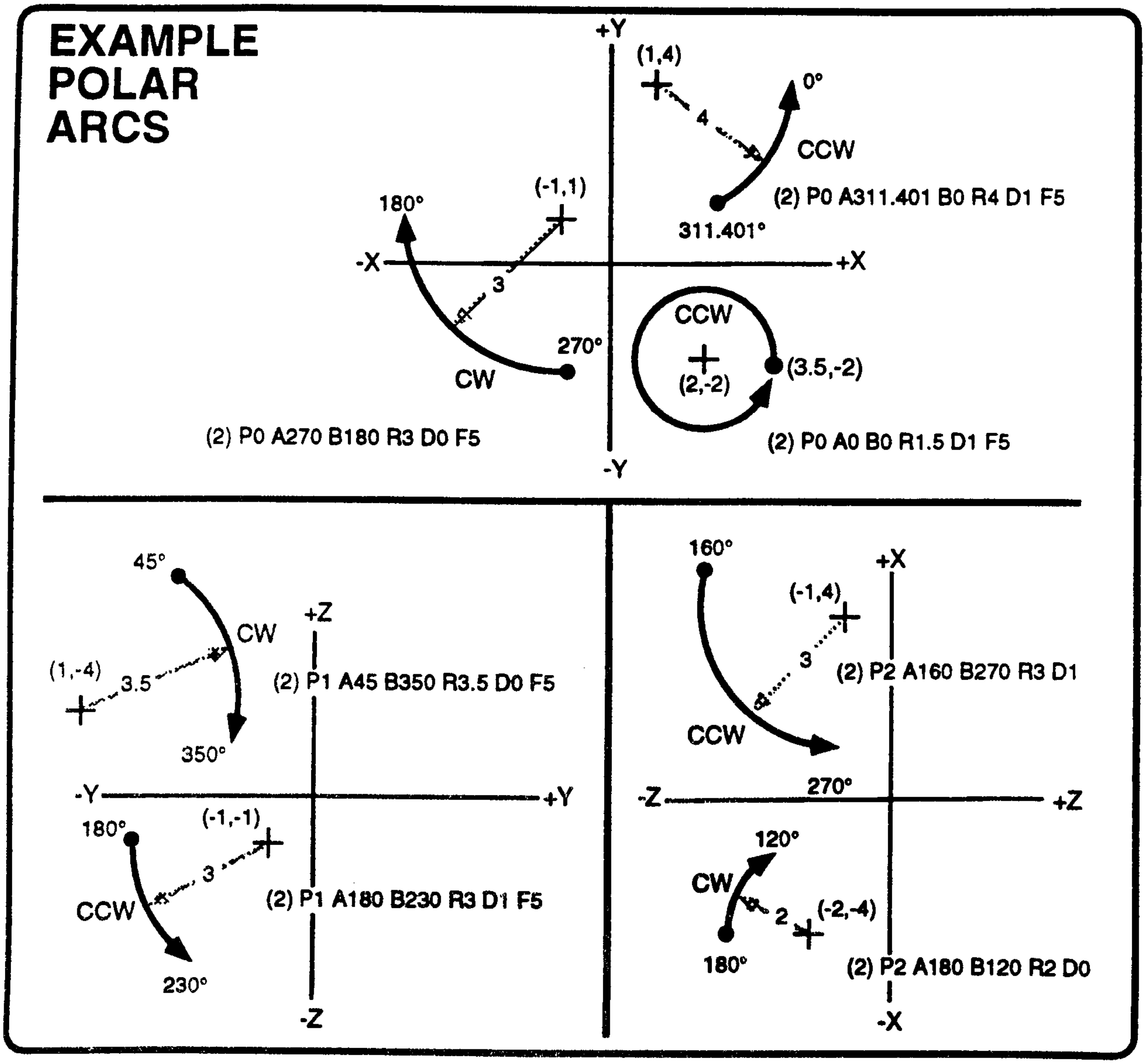
```
(0) X-4.5 Z-2
(2) P2 X-.5 Z-4 I-2 K-2 D0 F5
        or
(2) P2 X4/ Z-2/ I2.5/ K0/ D0 F5
```

R ARC RADIUS - POLAR ENTRY

The polar arc radius is the unsigned distance from the arc center to the tool's start point.

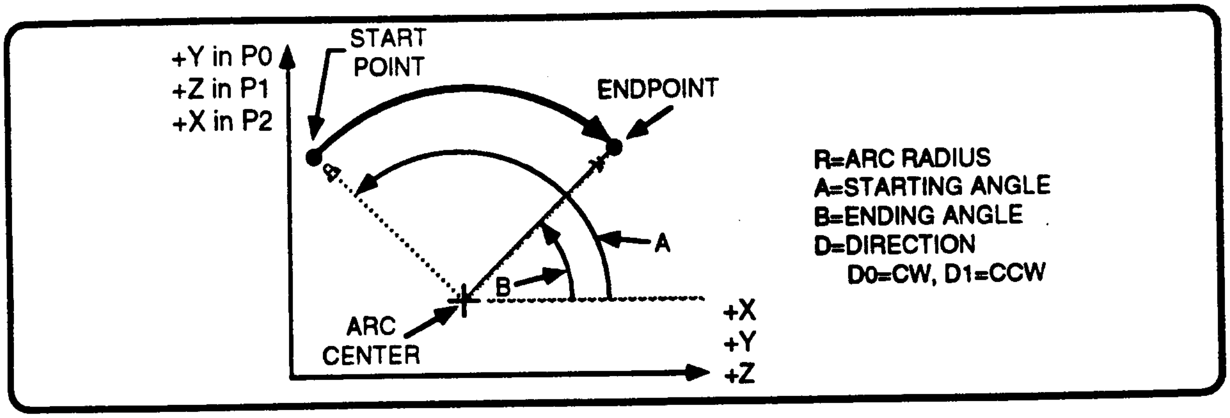
A, B START AND END ANGLES - POLAR ENTRY

Enter the start and end angles in the selected plane. Start angle A locates the tool's start point on the arc described by radius R. End angle B similarly defines the tool's endpoint. Both angles open counterclockwise[1] from a zero degrees reference line[2] that originates at the arc center. Refer to **illustrations, below and previous page.**



[1] Counterclockwise is the direction that provides the shortest angle from the positive primary axis to the positive secondary axis in the selected plane.

[2] Zero degree reference lines run parallel to, and in the same direction as, the positive primary axis in the selected plane. The primary axis in the XY plane is X, in the YZ plane is Y, and in the ZX plane is Z. The other axis in each plane is referred to as the secondary axis.



F - FEEDRATE

In this event, feedrate F programs direct feed along the arc path. Feedrate is modal with all following event types which program feedrate F.

CUTTER RADIUS COMPENSATION

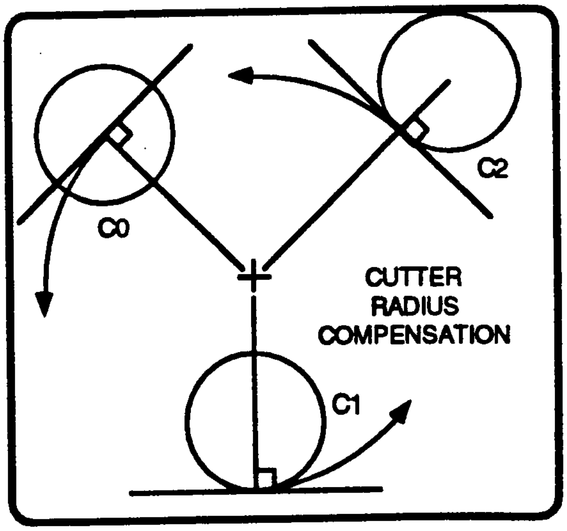
Cutter Radius Compensation (CRC) is a utility that simplifies profile programming by allowing you to program part coordinates without considering the tool's size. When you cycle a part with CRC properly programmed, the control will automatically offset the programmed path to compensate for the active tool's radius, and command the tool to move along the path that smoothly blends each event with the following Linear or Arc Mill event. CRC may only be used in an uninterrupted sequence of Linear Mill and Arc Mill events.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

C - TURN CRC ON OR OFF

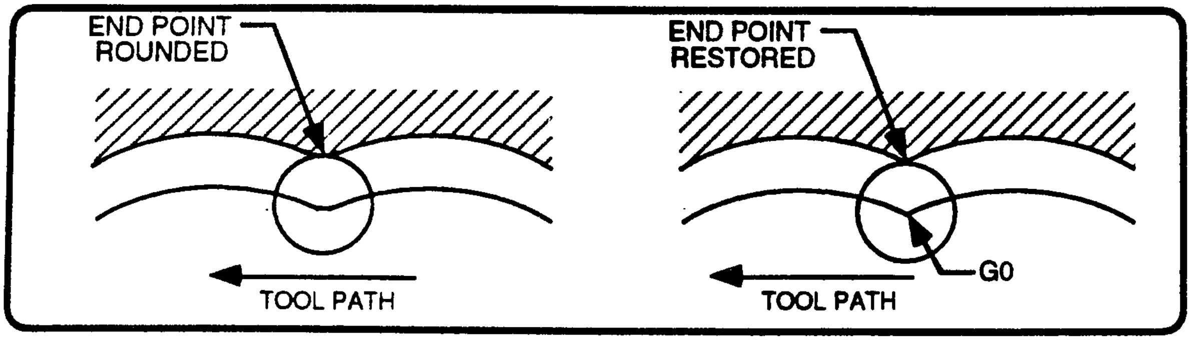
C0 cancels any active cutter compensation. C1 turns cutter compensation on to the left of the profile; C2 turns cutter compensation on to the right of the profile. Turning cutter compensation on or off will offset the tool a distance equal to the tool's radius in a direction that is perpendicular to the profile. CRC is modal with all following Linear and Arc Mill events. Refer to the illustration at the **right**.

Cutter compensation is described in Part 4 - Section 8 of this manual.



G0 - WAIT FOR IN POSITION

You would normally program a series of linear and arc moves without using the G0 entry. However, if a high feedrate shows signs of corner rounding in a test part, program G0-Wait for In Position to restore the corner, **below**. G0 is nonmodal and operative only in the event in which it is programmed. If G is left blank (not programmed), the next event begins right after the last feed increment is output to the servos. Note that the next motion output can occur while there is still following error to be resolved. If G0 is programmed, the next event will not begin until the following error is reduced to less than the "In Position" range established by the machine supplier. A noticeable delay will usually occur during the Wait for In Position. Refer to the **lower** illustrations.



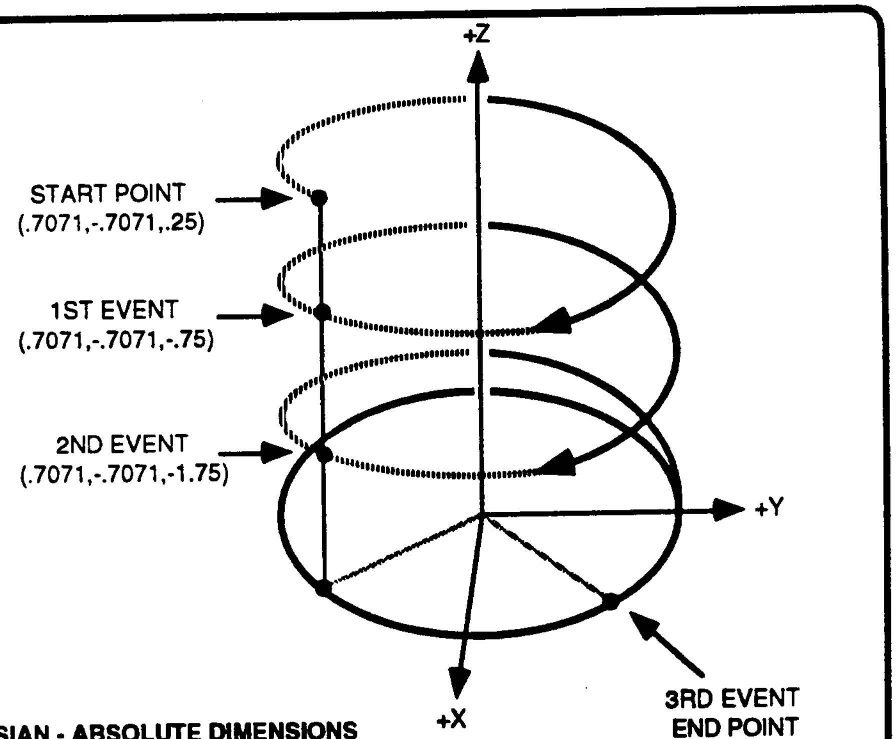
Z - XY PLANE HELIX

When the P0 (XY) plane is selected, a helix may cut by programming the arc as described earlier with the addition of a Z axis advance.

Since the Arc Mill event is limited to a 360° arc, three events will be required to mill the example program. The advance is 2.75 inches in Z over the same number of revolutions, or a lead of 1 inch per revolution.

The start point is (.7071, -.7071, 0.25), .25 inches above the surface.
The endpoint is (.7071, 0.7071, -2.5).
The tool moves in the clockwise direction.

Refer to the **following** program example.



CARTESIAN - ABSOLUTE DIMENSIONS

```
N10 (2) P0 X.7071 Y-.7071  I0 J0 Z-.75 D0 F10
N20 (2) P0 X.7071 Y-.7071 I0 J0 Z-1.75 D0 F10
N30 (2) P0 X.7071 Y.7071 I0 J0 Z-2.5D0 F10
```

CARTESIAN - INCREMENTAL DIMENSIONS

```
N10 (2) P0 X0/ Y0/ I-.7071/ J.7071/ Z-1/ D0 F10
N20 (2) P0 X0/ Y0/ I-.7071/ J.7071/ Z-1/ D0 F10
N30 (2)P0 X0/ Y1.4121/ I-.7071/ J.7071/ Z-.75/ D0 F10
```

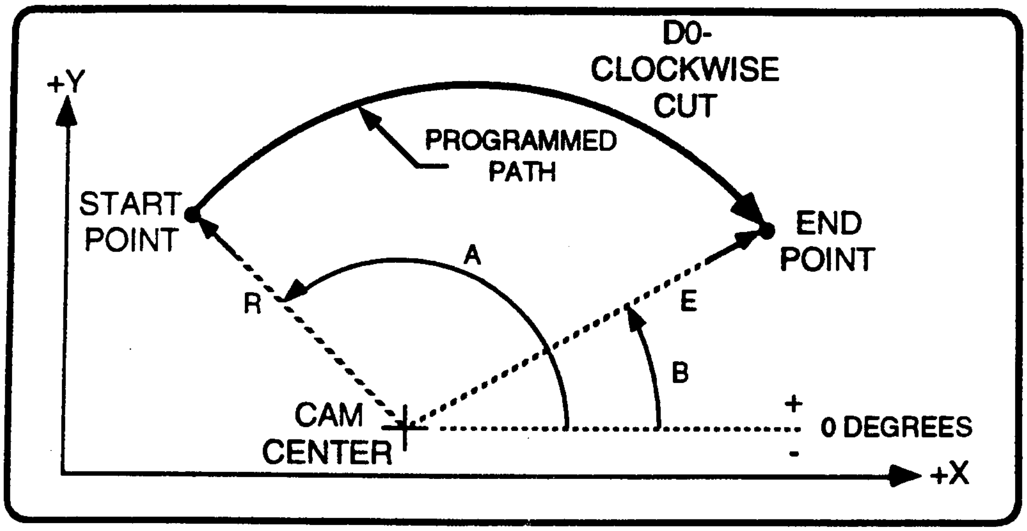
POLAR DIMENSIONS

```
N10 (2) P0 A315 B315 R1 Z-.75 (or Z-1/) D0 F10
N20 (2) P0 A315 B315 R1 Z-1.75 (or Z-1/) D0 F10
N30 (2) P0 A315 B45 R1 Z-2.5 (or Z-.75/) D0 F10
```

E - CAM MILL END RADIUS - POLAR ENTRY

When end radius E is programmed in a polar Arc Mill event, a curve of uniformly changing arc radius results[3]. This capability is useful in programming a variety of cam profiles. The event must use A, B and R entries. End radius E is available with P0, P1 or P2 plane select. Start radius R and end radius E must each be greater than zero, and the included angle between start angle A and end angle B must be less than 360°. Cam profiles may only be programmed in the XY plane; no lead in Z may be programmed with a cam profile.

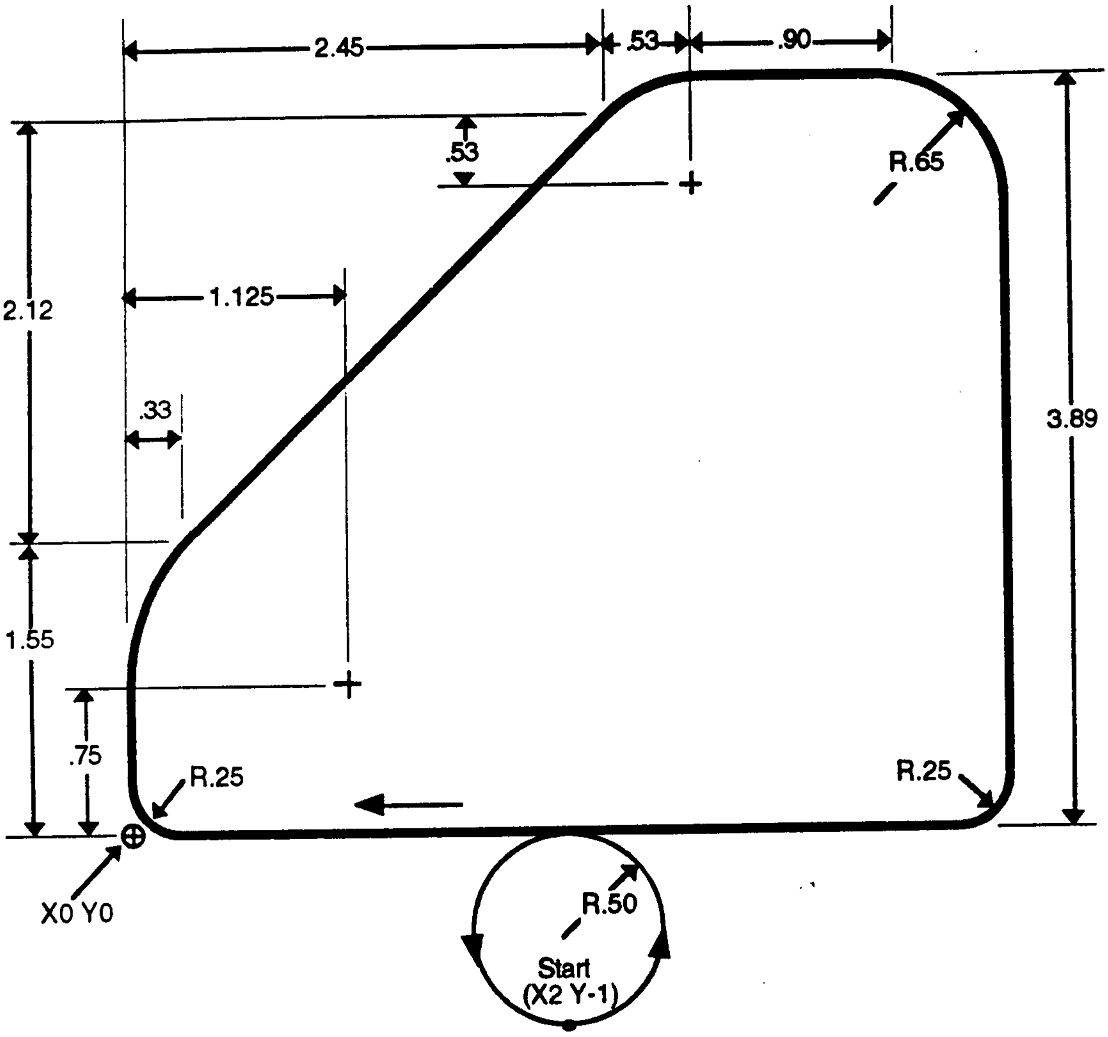
Refer to the illustration **below**.



[3] The change in radius is directly proportional to the change in degrees. The resulting curve is properly called an involute.

EXAMPLE:

Note that the following example is only intended to demonstrate methods of using Arc Mill events; it is not intended to be a complete part program since it does not include a Z axis depth nor does it include commands to identify a tool number or spindle speed.



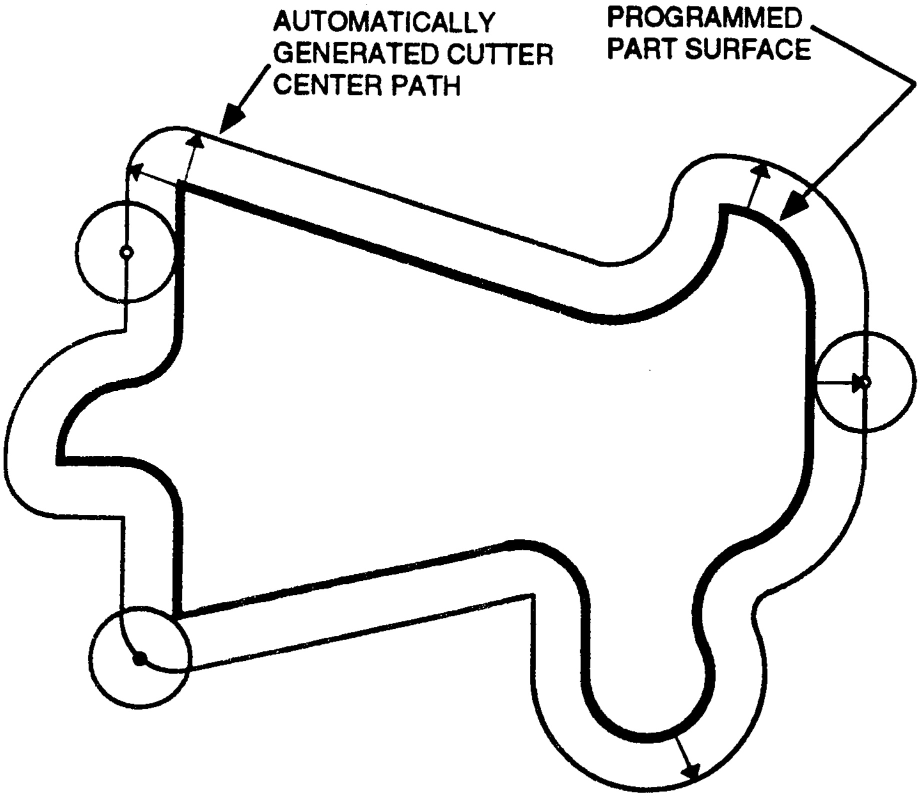
```
N.5   (G)   X-1 Y-1 V6                      - Graphics Setup event
N01   (0)   X2 Y-1 Z.1                      - convenient start position
N02   (2)   A270 B90 R.5 D1                 - tangential arc to enter profile
N03   (1)   X.25                            - Linear Mill to start of arc
N04   (2)   X0 Y.25 I.25 J.25 D0            - Cartesian arc
N05   (1)   Y.75                            - Linear Mill event
N06   (2)   X.33 Y1.55 I1.125 J.75 D0       - Cartesian arc
N07   (1)   X2.45 Y3.67                     - Linear Mill event
N08   (2)   X.53/ Y3.89 I.53/ J-.53/        - Cartesian arc
N09   (1)   X.90/                           - Linear Mill event
N10   (2)   A90 B0 R.65 D0                  - Polar arc
N11   (1)   Y.25                            - Linear Mill event
N12   (2)   A0 B270 R.25 D0                 - Polar arc
N13   (1)   X2                              - Linear Mill event
N14   (2)   A90 B270 R.5 D1                 - Polar arc to tangentially exit profile
N99   (9)   M30                             - end of program command
```

CUTTER RADIUS COMPENSATION

This section describes cutter radius compensation (CRC) for Conversational Linear Mill, Arc Mill events and EIA/ISO G01, G02, G03, G21, G22, G23 blocks. Cutter radius compensation for the Frame Mill, Circle Mill, and Cavity Mill events is internal to these events and requires no special operations.

This feature is designed for part surface programming. You can program the part using dimensions read directly from the part's drawing, and, if cutter compensation has been programmed correctly, the control will generate a tool path to ensure that the cutter's edge moves along the programmed surface. Refer to illustration, **below**.

Before attempting to program a cutter compensation sequence, you should be familiar with the M Function, Linear Mill, and Arc Mill events. If programming in EIA/ISO format, you should be familiar with the corresponding G01, G02, G03, G21, G22, and G23 blocks.



USING CUTTER RADIUS COMPENSATION

To use CRC, your part program must comply with the following rules. These rules are described in greater detail throughout this section.

1) Prior to using CRC, the part program must execute a T Code in an M Function event (or EIA/ISO block) to identify the tool that will cut the part. The tool's diameter must be stored in the Tool Tables.

2) After you position the tool to the start of the sequence, program at least one cutter compensation event[1], in the compensation plane, with compensation off (C0, G40).

   EXAMPLE: N130 (1) X0/ C0 Z-3 P0

   In event N130 above, the compensation will be in the XY plane (P0, G17). A Z axis move is optional, but permitted as long as an XY command is programmed. The XY command (X0/) is not required to actually move the tool.

3) Program a cutter compensation event[1], in the compensation plane, with CRC on to the left (C1, G41) or right (C2, G42). To choose the direction, look from behind the tool as it makes the first cut and determine if the tool center should be to the left (C1, G41) or right (C2, G42) of the path. This event will command a two step move. First, the tool will feed perpendicular to the programmed path a distance equal to the active tool's stored radius. Second, the edge of the tool will feed along the programmed line or arc.

   EXAMPLE: N140 (1) X4/ C1

4) With compensation turned on, all following events must be cutter compensation events[1] with the same plane select. Text, Repeat, M Function[2], Subroutine, Dwell, Macro Call, Rotate, and Set Up events are not permitted. Compensation is modal, you only have to program it in the turn on event. Out of plane moves (Z axis in this example) are allowed as long as they are accompanied with a move in the compensation plane (XY).

5) To turn off CRC, program a cutter compensation event[1] in the compensation plane with compensation off (C0, G40). This event will command a feed perpendicular to the programmed path a distance equal to the active tool's stored radius.

   EXAMPLE: N240 (1) X0/ C0

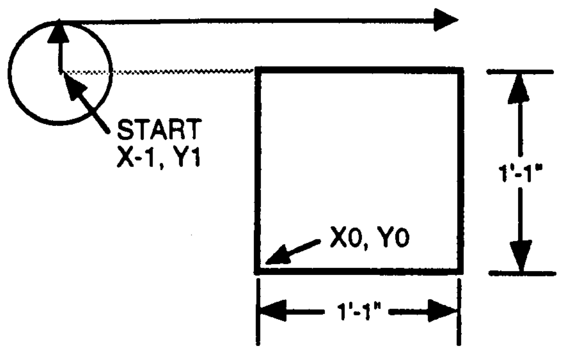
LEAD IN EVENTS

The tool's position at the start of a cutter compensation sequence should be selected carefully.

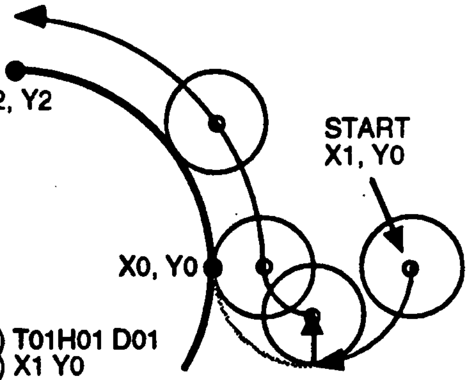
The tool should be positioned at least one tool diameter away from the part profile, and, if possible, be at the Z cutting depth. The compensation turn on sequence will be easiest if the tool center's start position is in-line with the first feed move. The illustrations below show examples of turning CRC on with Linear Mill events left and Arc Mill events right. In both examples, notice how the start point is more than one tool diameter away from the part surface. Further notice the ease in turning CRC on with Linear Mill events when the start point is carefully chosen.

[1] Cutter compensation events are: Conversational Linear Mill and Arc mill events and EIA/ISO G01, G02, G03, G21, G22, and G23 blocks.

[2] As an exception of this rule, intermediate M Function events that do not generate axis motion are allowed.



```
N10 (9) T01 H01 D01
N20 (0) X-1 Y1
N30 (1) X0/ C0
N40 (1) X1 C1
```



```
N10 (9) T01H01 D01
N20 (0) X1 Y0
N30 (2) A0 B270 R.5 D0 C0
N40 (2) A270 B180 R.5 D0 C2
N50 (2) A0 B90 R2 D1
```

MAINTAINING COMPENSATION

Once cutter compensation is turned on, there must be a **continuous** sequence of Linear and Arc Mill events, and G01, G02, G03, G21, G22, or G23 blocks. If this sequence is interrupted by a different event type, the look ahead is interrupted (see following Look Ahead Logic topic). All compensated events must have the same plane select

The event should also refrain from executing T Codes (to change an offset value) when cutter compensation is active.

LEAD OUT EVENTS

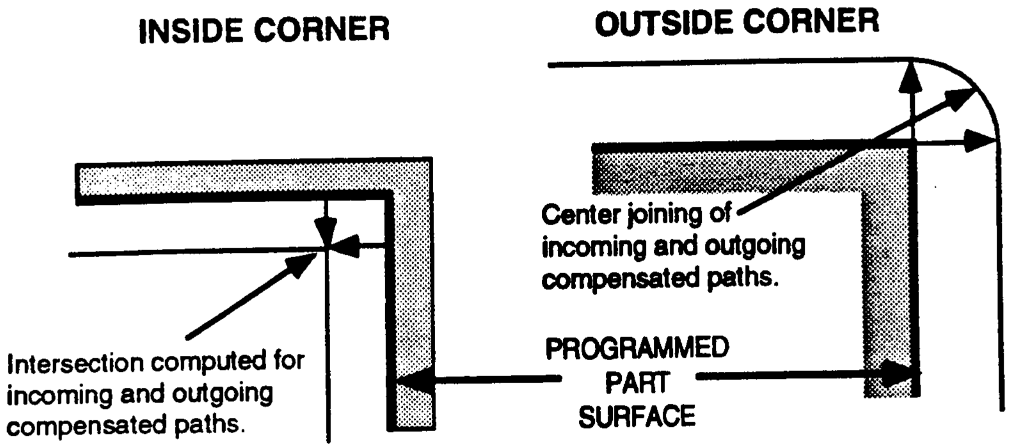
Turn CRC off after the tool has cleared the part by at least one tool diameter. The event that turns CRC off must command at least one axis in the compensated plane but does not have to program tool motion (example: (1) X0/ C0). On execution, the turn off event will cause the tool to feed perpendicular to the previous event a distance equal to the stored tool radius. If axis motion is programmed in this event, the motion will occur after the perpendicular turn off move.

LOOK AHEAD LOGIC

When cutter compensation is active, Look Ahead logic dictates the tool path as the control joins successive events. While executing one event, Look Ahead logic preprocesses the following event to obtain the data required to smoothly join the two events. As a result of preprocessing the following event, the control determines if it should connect the two events with an inside or outside join.

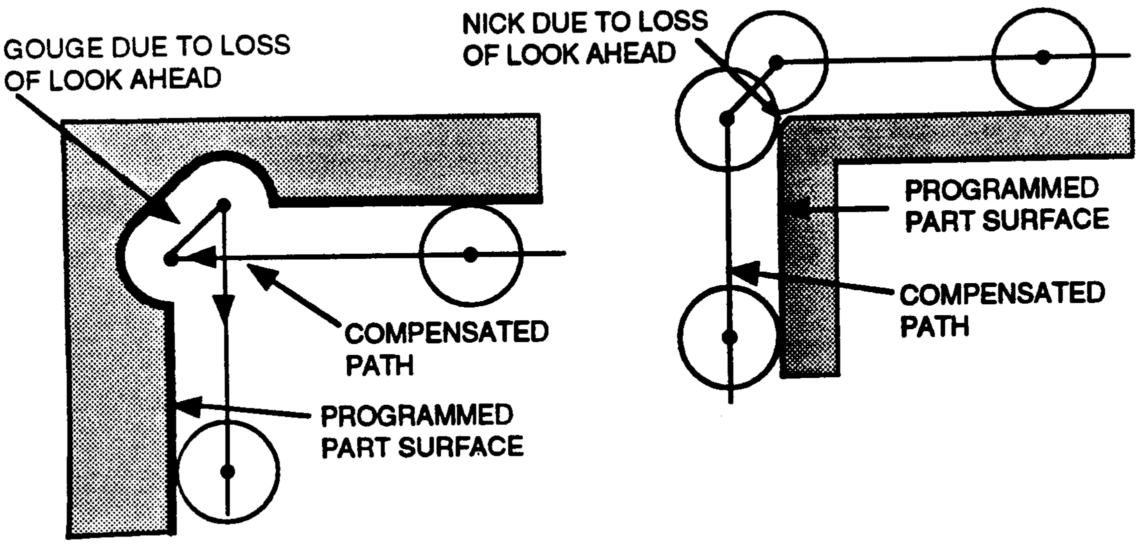
Inside joins, **below left**, occur when the tool is on the inside of a corner or intersection. In other words, the angle between the two events is less than 180°. Such events are joined with two intersecting linear events.

Outside joins, **below right**, occur when the tool is on the outside of a corner or intersection. In other words, the angle between the two events is greater than 180°. Such events are "connected" with an arc, generated by the control, that ensures that the cutter's edge is always tangent to the profile. No dwell mark will occur.


INSIDE CORNER
OUTSIDE CORNER
Center joining of incoming and outgoing compensated paths.
Intersection computed for incoming and outgoing compensated paths.
PROGRAMMED PART SURFACE

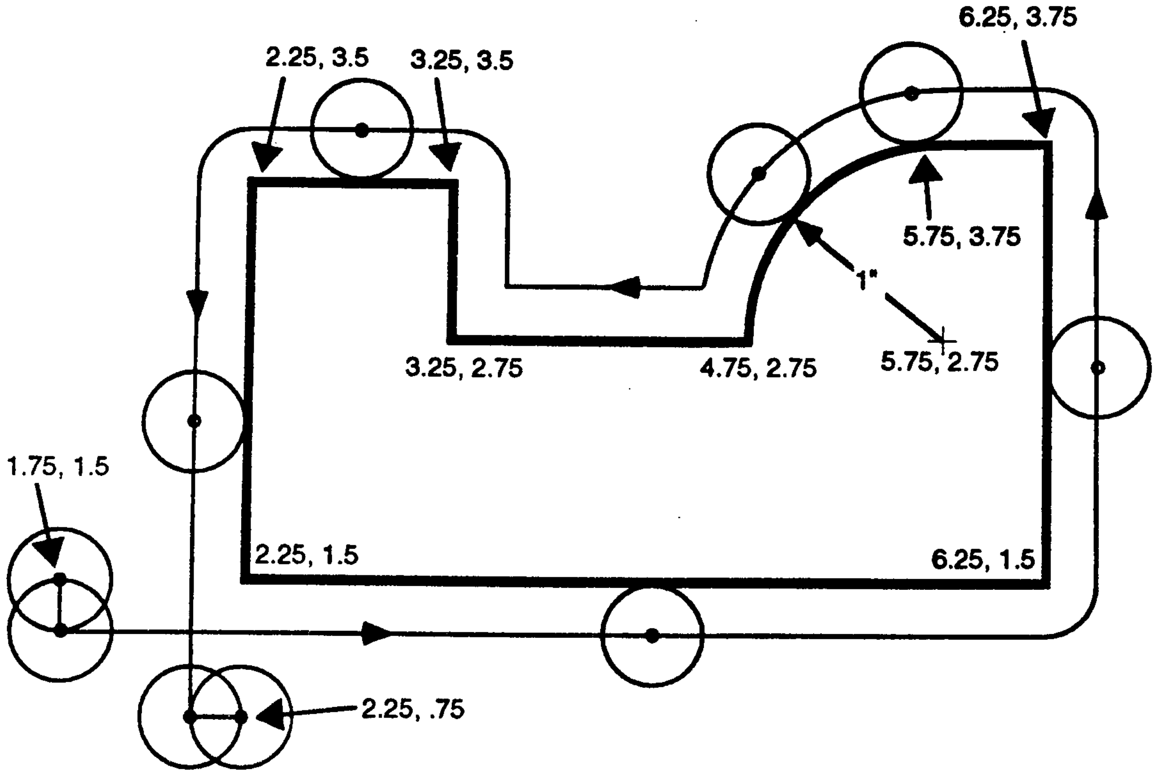
LOSS OF LOOK AHEAD

Loss of Look Ahead occurs when the following event does not provide any endpoint data in the plane of compensation. When this occurs, the control will maintain left or right compensation, but move the tool center to the end of the programmed event. Frequently, the loss of Look Ahead will cause a gouge on inside joins, or a nick on inside joins, **below**. These blemishes are caused when you break the rule that all events in a CRC sequence must be cutter compensation events[1].



EXAMPLE

The example **below** shows three different methods for programming cutter compensation. All three programs define the part's profile; a 0.500" cutter diameter has been stored for the first tool in the Tool Tables.



CONVERSATIONAL

```
N010 (9) T01 H01 D01 E01
N020 (9) M03 S1250
N030 (0) X1.75 Y1.5 Z-1.25

N040 (1) X0/ C0 F15
N050 (1) X6.25 C2 F15
N060 (1) Y3.75
N070 (1) X5.75
N080 (2) R1 A90 B180 D1
N090 (1) X3.25
N100 (1) Y3.5
N110 (1) X2.25
N120 (1) Y.75
N130 (1) X0/ C0
N999 (9) M30
```

EIA/ISO ABSOLUTE

```
N005 G17
N007 G90
N010 T01 H01 D01 E01
N020 M03 S1250
N030 G00 X1.75 Y1.5 Z-1.25
N035 G01
N040 G40 X1.75 F15
N050 G42 X6.25 F15
N060 Y3.75
N070 X5.75
N080 G03 X4.75 Y2.75 I5.75 J2.75
N090 G01 X3.25
N100 Y3.5
N110 X2.25
N120 Y.75
N130 G40 X2.25
N999 M30
```

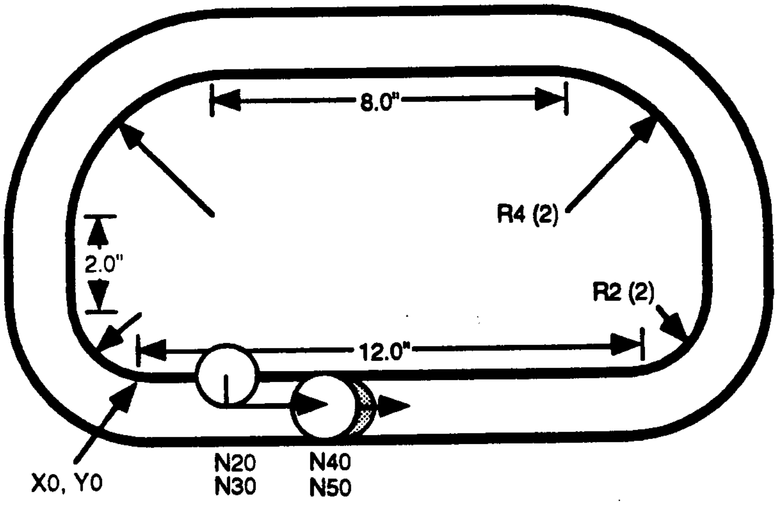
EIA/ISO INCREMENTAL

```
N005 G17
N007 G90
N010 T01 H01 D01 E01
N020 M03 S1250
N030 G00 X1.75 Y1.5 Z-1.25
N035 G01
N036 G91
N040 G40 X0 F15
N050 G42 X4.5 F15
N060 Y2.25
N070 X-.5
N080 G03 X-1 Y-1 I0 J-1
N090 G01 X-1.5
N100 Y.75
N110 X-1
N120 Y-2.75
N130 G40 X0 F100
N999 M30
```

EXAMPLE 2 - CHANNEL and SLOT CUTTING

Many profiles that fall into the category of channels or slots can be easily programmed with the Frame Mill event. However, if you need to cut an irregular profile that does not allow clearance for you to preposition the tool to Z depth, this example shows how to plunge into the workpiece while CRC is active without gouging the profile.

To begin, N20 positions the tool center in line with the profile and above the workpiece. N30 cancels compensation as required by the CRC turn on rules. N40 turns cutter comp on to the right and begins feeding the tool along the profile (the tool is still above the workpiece). N50 feeds the tool to the Z depth. Although N50 programs an out of plane move (which breaks the "Maintaining Compensation" rules), the part will not be gouged as long as the next event feeds the tool in the same direction as the event that immediately precedes the out of plane command, and the following event commands a move which is longer than the tool's radius. To satisfy this condition, N60 continues the linear feed to X12. Events N70-N140 complete the profile. N150 retracts the tool. N160 turns CRC off. Finally, N170 ends the part program. The retract move (N150) will not gouge the part as long as the retract occurs at least one tool radius from the endpoint of the linear path that ends at X12.



```
N10   (9)  T01 M03 S1200         - tool number 1, turn on spindle
N20   (0)  X2 Y0 Z.1             - position in line with, and above profile
N30   (1)  X0/ C0                - comp off
N40   (1)  X1/ C2 F15            -  comp on right begin linear mill
N50   (1)  Z-1                   - feed to depth
N60   (1)  X12                   - continue linear mill
N70   (2)  R2 A270 B0 D1         - profile
N80   (1)  Y2/                   - profile
N90   (2)  R4 A0 B90 D1          - profile
N100  (1)  X-8/                  - profile
N110  (2)  R4 A90 B180 D1        - profile
N120  (1)  Y-2/                  - profile
N130  (2)  R2 A180 B270 D1       - profile
N140  (1)  X4/                   - profile
N150  (1)  Z.1                   - retract
N160  (1)  X1/ C0                - turn comp off above the workpiece
N170  (9)  M30                   - end of program
```

NEGATIVE RADIUS CUTTER COMPENSATION

All information up to this point has described positive cutter compensation. Positive compensation is used when you program part dimensions, store the cutting tool's diameter in the Tool Tables, and let the control generate the cutting path.

Negative cutter compensation is used when the part program defines a tool center path for a part. Negative cutter compensation lets you change cutter diameters without forcing you to edit the part program's dimensional data. This is accomplished by entering a positive or negative tool diameter in the Tool Tables.

Negative cutter compensation is used in the specific circumstance which is described below.

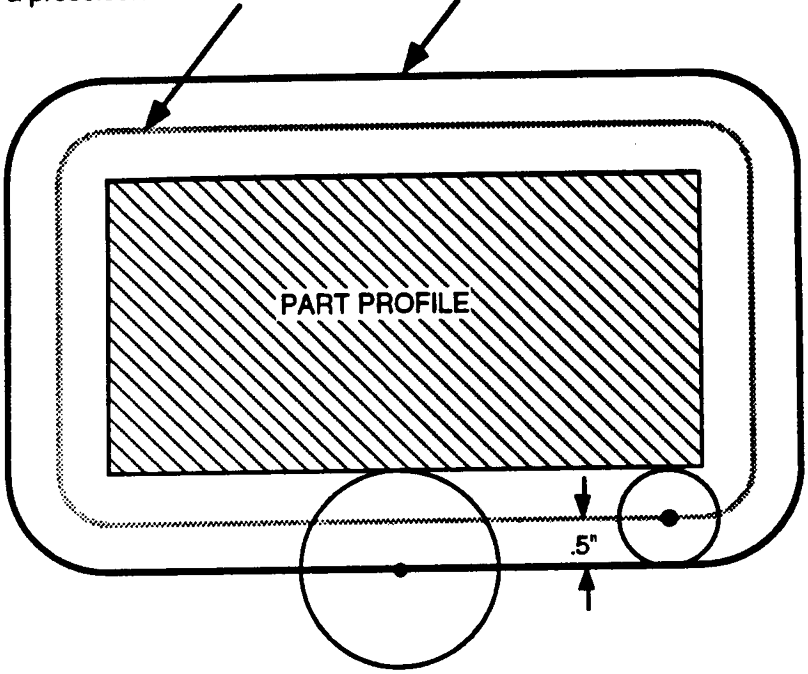
1) The part program must program a tool center path (i.e. no cutter compensation is used). This type of program is generated with the knowledge that a predetermined tool diameter will cut the part. Such programs are typically generated by CAD/CAM programs that do not insert CRC commands into the program code. The CAD program generates a tool center path to cut the part.

2) You must edit the part program to add T Codes which identify the the tool's diameter, and add CRC turn on, and turn off sequences. The resultant part program must comply with the rules describe earlier in this section. Choose the cutter compensation direction by relating the tool center position (left or right) to the actual part profile, not the existing tool center path.

3) Store the tool's offset diameter in the Tool Tables. This diameter is not the physical diameter of the tool, but the difference between the diameters of the original tool and the tool that will cut the part.

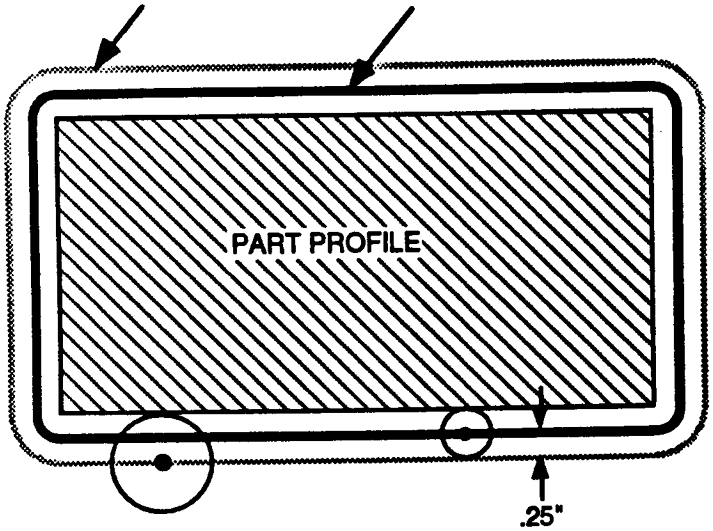
EXAMPLE: a program was generated on a CAD/CAM system to cut the part with a .500" diameter tool. You now want to cut the part with different size tools. The table below shows the diameter that you should enter in the Tool Table for each tool size.

INTENDED TOOL DIA.	CUTTING TOOL DIA.	ENTER IN TOOL TABLES
.500"	.250"	-.250"
.50000"	.500"	0.000"
.5"	1.00"	+.500"



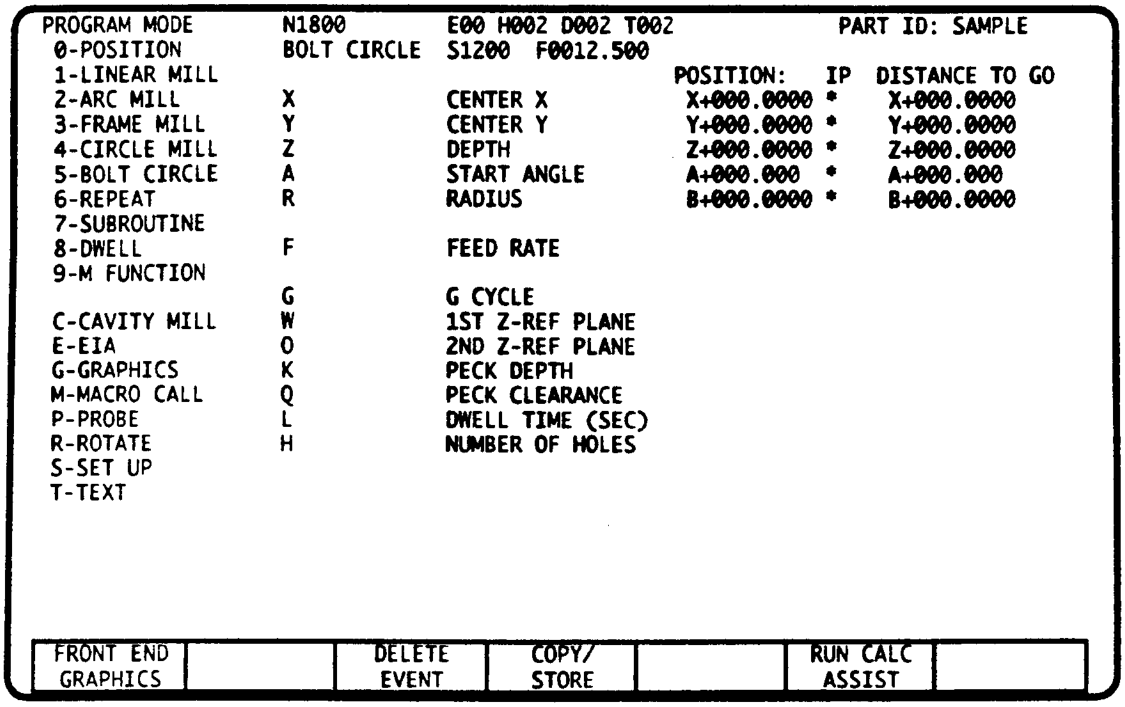
Part program path to cut the rectangle with a preselected .5" cutter.

CNC generated path to use 1" cutter. Store the diameter difference in the Tool Table (1-.5=.5).

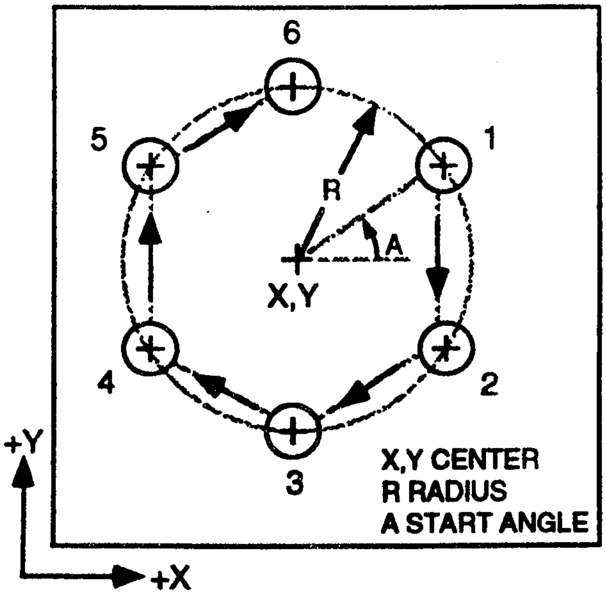


Part program path to cut the rectangle with a preselected .5" cutter.

CNC generated path to use .25" cutter. Store the diameter difference in the Tool Table (.25-.5=-.25).



```
PROGRAM MODE         N1800         E00 H002 D002 T002            PART ID: SAMPLE
 0-POSITION          BOLT CIRCLE   S1200  F0012.500
 1-LINEAR MILL                                       POSITION:   IP  DISTANCE TO GO
 2-ARC MILL          X             CENTER X           X+000.0000 *    X+000.0000
 3-FRAME MILL        Y             CENTER Y           Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL       Z             DEPTH              Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE       A             START ANGLE        A+000.000  *    A+000.000
 6-REPEAT            R             RADIUS             B+000.0000 *    B+000.0000
 7-SUBROUTINE
 8-DWELL             F             FEED RATE
 9-M FUNCTION
                     G             G CYCLE
 C-CAVITY MILL       W             1ST Z-REF PLANE
 E-EIA               O             2ND Z-REF PLANE
 G-GRAPHICS          K             PECK DEPTH
 M-MACRO CALL        Q             PECK CLEARANCE
 P-PROBE             L             DWELL TIME (SEC)
 R-ROTATE            H             NUMBER OF HOLES
 S-SET UP
 T-TEXT
```



BOLT CIRCLE EVENT - INTRODUCTION

This event programs a bolt hole circle in the XY plane. On execution, the control will perform the programmed or modal G Cycle (drill, tap, bore, etc.) at each hole around the circle. Z clearance for XY positioning must be established prior to the start of this event.

The bolt circle is defined with the circle's center coordinates, radius, and the quill cycle to be performed. The bolt hole pattern is defined with the angular orientation (relative to the center) of the first hole, the number of evenly spaced holes around the complete circle, and the number of G cycles from the first hole moving clockwise around the bolt circle.

BOLT CIRCLE DATA

Parameter	Description	Entry
N	Sequence Number	Required
X	Bolt circle center, X	Required
Y	Bolt circle center, Y	Required
Z	Depth	G Cycle Modal
A	Start angle (first hole)	Required
R	Bolt circle radius	Required
H	Number of Holes	Required

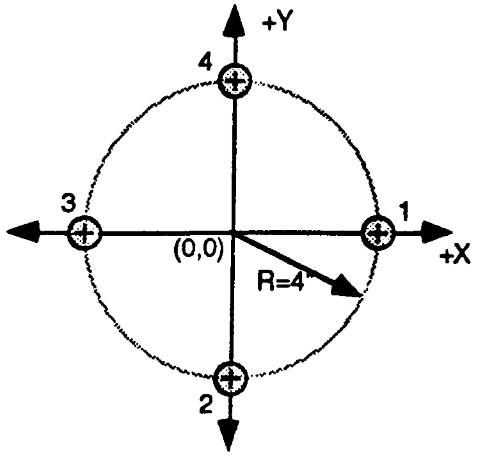
G CYCLES and G CYCLE PARAMETERS

G	G Cycle	G0-G5, G7-G8 Modal
F	Feedrate	- G Cycle Modal
K or K/	Peck	- G Cycle Modal
Q	Peck Clearance	- G Cycle Modal
W	1st Z Reference Plane	- G Cycle Modal
O	2nd Z Reference Plane	- G Cycle Modal
L	Dwell	- G Cycle Modal

G Cycle	Description	G Cycle Entries
G0	Turn off G cycles	none
G1	Drill	W, Z, F
G2	Counterbore	W, Z, F, L
G3	Peck Drill	W, Z, F, K or K/, Q
G4	Tap	W, Z, F, L
G5	Bore	W, Z, F
G7	Dead Spindle Bore	W, Z, F
G8	Rigid Tap (option)	W, Z, F (see text)

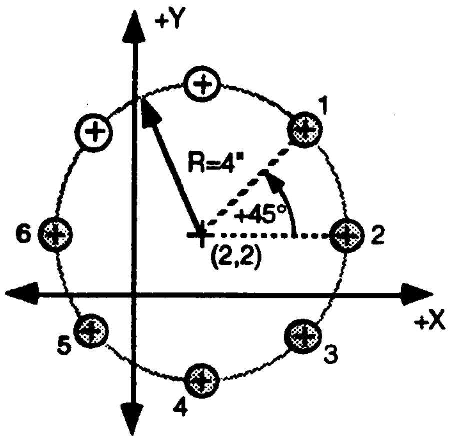
G1 - DRILL

(5) X0 Y0 Z-1 A0 R4 F4 G1 W.1 H04



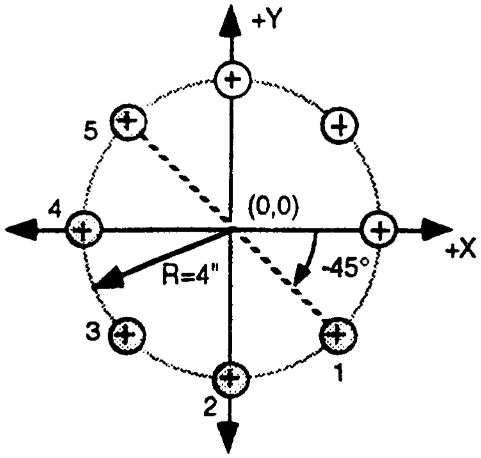
G2 - COUNTERBORE

(5) X2 Y2 Z-1.5 A45 R4 F5 G4 W.1 L1.5 H08.06



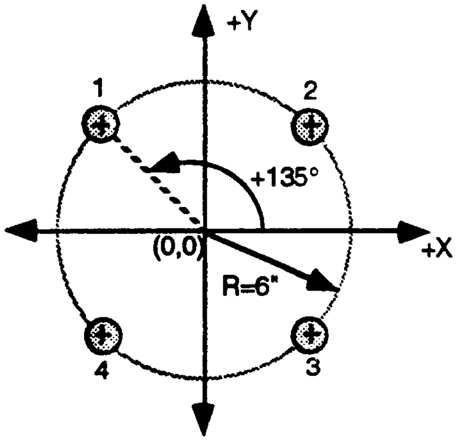
G3 - PECK DRILL

(5) X0 Y0 Z-1 A315 R4 F7.5 G3 W.25 K.4 H08.05



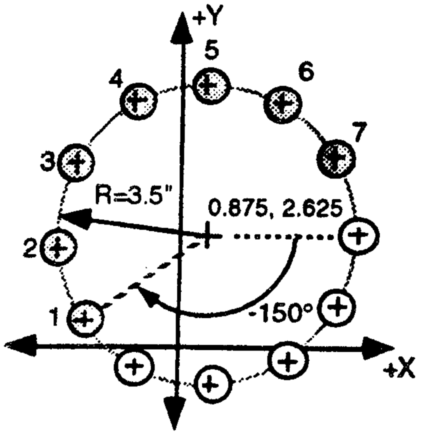
G4 - TAP

(5) X0 Y0 Z-1 A135 R6 F6.5 G4 W.2 L2 H04



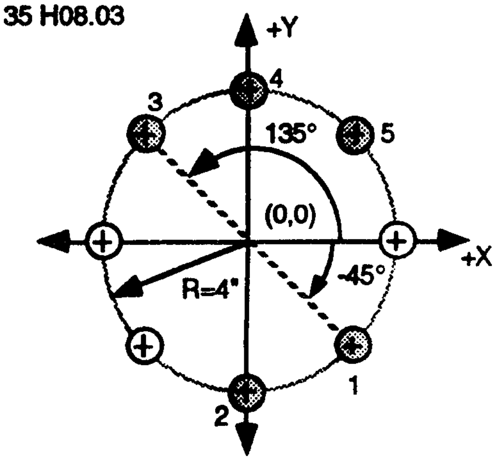
G5 - BORE

(5) X0.875 Y2.625 Z-2.5 A-150 R3.5 F5 G5 W.1 H12.07



G1 - SKIP

(5) X0 Y0 Z-1.75 A-45 R4 F7.5 G1 W.25 H08.02 then COPY/STORE to next event and edit... A135 H08.03



X, Y - BOLT CIRCLE CENTER

Enter the bolt hole circle's X and Y center coordinates. The tool does not move to this position, but the control uses it in the process of calculating hole centers. The circle center may be programmed as signed absolute X and Y axis coordinates, or as the signed incremental distance from the XY tool position at the start of the event. An entry is not required for an axis if it is on its XY circle center coordinate prior to the start of the event.

R - BOLT CIRCLE RADIUS

Radius R is the distance from the XY center to the center of each hole on the bolt circle. R is entered as an unsigned absolute value.

A - START ANGLE OF FIRST HOLE

Start angle A is the first hole center's angular position on the bolt circle. The tool initially rapids in XY from its last programmed position to a point on center with this hole to begin a G cycle sequence clockwise around the bolt circle. By locating this first hole, start angle A also establishes the relative locations of all equidistant holes on the bolt circle.

Start angle A is entered in signed absolute degrees, referenced from the +X axis (0 degrees). If angle A is positive, the angle opens counterclockwise from standard position. If A is negative, the angle opens clockwise from standard position.

H - NUMBER OF HOLES

The characters Hxx.xx designate the 4 digit H entry format. The first two digits set the number of equally spaced holes on the bolt circle, and are followed by a (.) period. The second pair of digits specify how many holes will be drilled, bored, etc. The G cycle begins at the first hole and always moves clockwise around the bolt hole circle for the number of holes entered by the second digit pair. The entry H12.12, for example, assigns 12 hole positions and directs the G cycle to be performed at all holes. H12.12 may also be entered as H12.00 or H12 with the same results. The entry H12.07 assigns 12 hole positions, but only performs the G cycle on the first seven holes.

Each intermediate XY rapid move between holes is at the 2nd reference plane (O) coordinate. If O has not been programmed, it will default to the modal 1st reference plane (W) value. At the event's conclusion, the tool will be centered over the last hole drilled and the Z axis will be at the O plane.

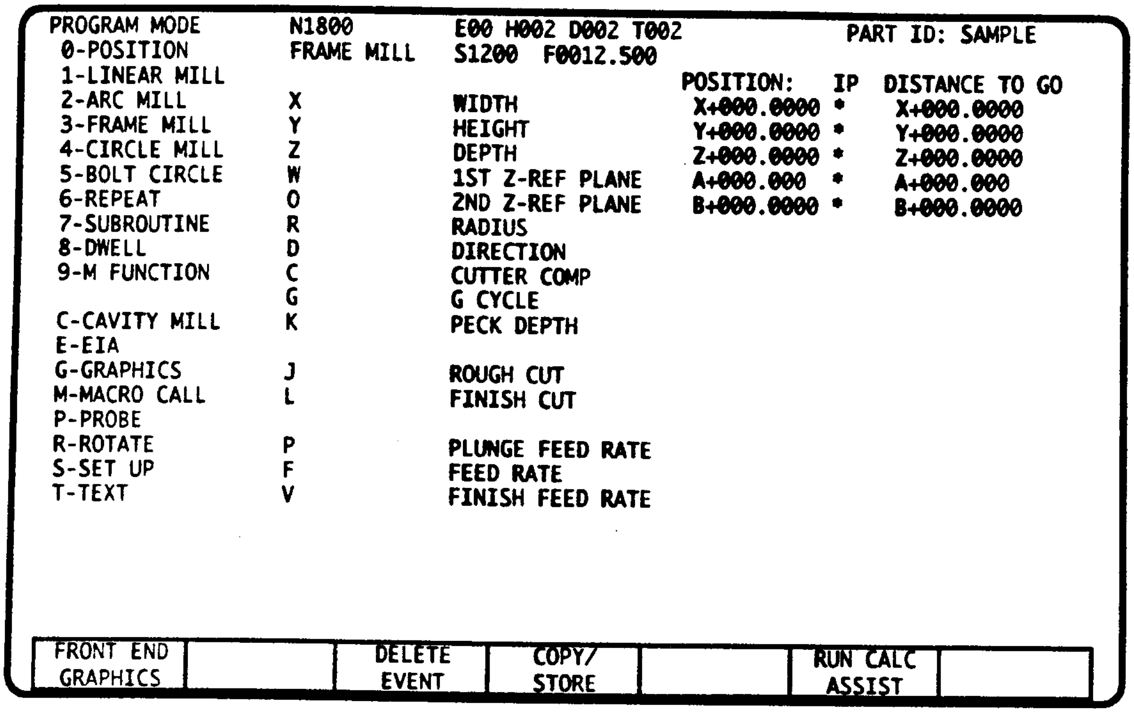
G0-G5, G7-G8 - PERFORM G CYCLE

The Bolt Circle event runs the G0-G5 and G7-G8 cycles as described for the Position event. For a complete description of tool motion associated with each G cycle and W, O, Z, F and L entries, refer to the specific G cycles in the Position Event section.

Note: The W entry is programmed with the letter "I" if your control is equipped with an auxiliary axis that is programmed with the letter "W". Refer to Part 1-Section 2.

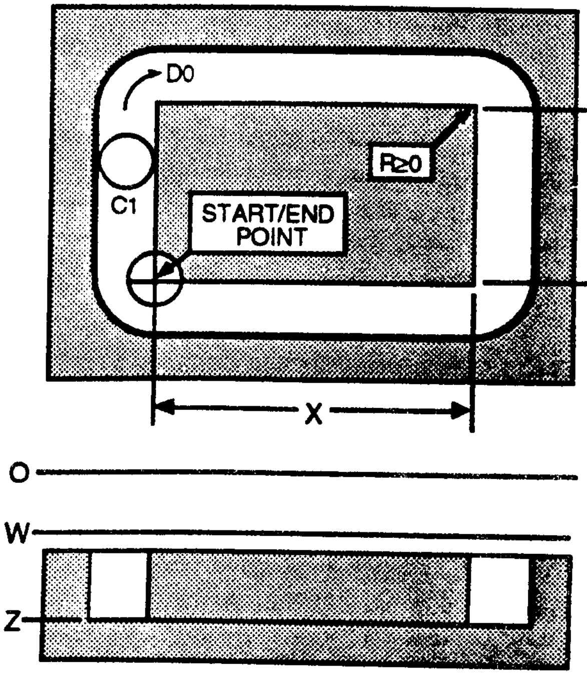
SKIP HOLE

"Skip hole" patterns may be performed by programming two or more Bolt Circle events in sequence. It is usually easiest to program the first event by choosing a start hole A that permits as many holes as possible (H) to run the G cycle moving in the clockwise direction and halting before the first "skip" hole. Then use the COPY/STORE key to copy the event to the next N number. In the copy event edit A and H to complete your hole pattern.

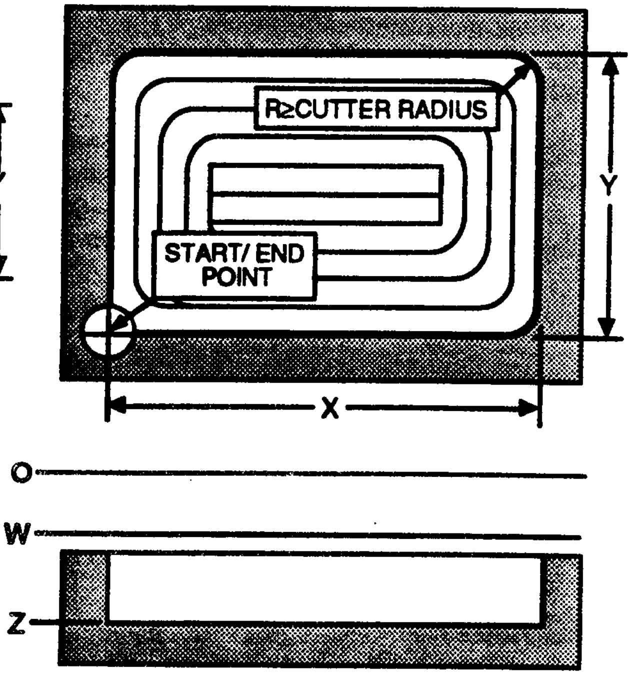


```
PROGRAM MODE       N1800        E00 H002 D002 T002                PART ID: SAMPLE
 0-POSITION        FRAME MILL   S1200  F0012.500
 1-LINEAR MILL                                   POSITION:   IP  DISTANCE TO GO
 2-ARC MILL         X           WIDTH             X+000.0000 *    X+000.0000
 3-FRAME MILL       Y           HEIGHT            Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL      Z           DEPTH             Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE      W           1ST Z-REF PLANE   A+000.000  *    A+000.000
 6-REPEAT           O           2ND Z-REF PLANE   B+000.0000 *    B+000.0000
 7-SUBROUTINE       R           RADIUS
 8-DWELL            D           DIRECTION
 9-M FUNCTION       C           CUTTER COMP
                    G           G CYCLE
C-CAVITY MILL       K           PECK DEPTH
E-EIA
G-GRAPHICS          J           ROUGH CUT
M-MACRO CALL        L           FINISH CUT
P-PROBE
R-ROTATE            P           PLUNGE FEED RATE
S-SET UP            F           FEED RATE
T-TEXT              V           FINISH FEED RATE

 FRONT END |        | DELETE | COPY/ |        | RUN CALC |
 GRAPHICS  |        | EVENT  | STORE |        | ASSIST   |
```

G0 - FRAME (OUTSIDE)
(3) G0 X3 Y1.7 Z-1 W.1 O.8 R0 D0 C1 F4.5



G7 - FRAME (POCKET)
(3) G7 X3 Y1.7 Z-1 W.1 O.8 R.2 D1 C1 F10



FRAME MILL EVENT - INTRODUCTION

This event cuts a rectangle to a specified depth. As shown, **facing page**, you can choose to mill around the rectangle's perimeter, or remove (pocket) all material within the rectangle's perimeter. The tool center position at the start of this event locates the frame's lower left corner. The rectangle is oriented parallel to the X and Y axes. It is possible to mill a rectangle that is not parallel to the X and Y axes as described at the end of this section.

FRAME MILL DATA

Parameter	Description	Entry
N	Sequence Number	Required
G	G Cycle G0=frame, G7=pocket	Defaults to G0
X	Frame Width	Required
Y	Frame Height	Required
Z	Frame Depth	Required
W	1st Z Reference Plane	Modal
O	2nd Z Reference Plane	Modal
R	Frame Corner Radius	Required

D	Milling Direction
	D0=rough CW/finish CW
	D1=rough CCW/finish CCW
	D2=rough CW/finish CCW
	D3=rough CCW/finish CW
		Defaults to D0

C	Cutter Radius Compensation C0=off, C1=left, C2=right	Required
K	Peck Mill	Optional
J	Corner Roughing Width (G7 only)	Optional
L	Finish Dimension L=finish bottom and sides, L/= finish sides only	Optional
F	Feedrate	Modal
P	Plunge Feedrate P=Plunge, P/= P plunge & 1st feed (G7 only)	Defaults to F
V	Finish Feedrate	Defaults to F

REQUIRED ENTRIES

Program G0 to mill around the frame's perimeter, or G7 pocket the frame. A pocketing operation removes all material within the rectangle. Describe the frame's shape with width X, height Y, depth Z, corner radius R and reference planes W and O. Next program direction D to choose conventional or climb milling for the rough/finish passes. After selecting the direction, choose cutter compensation direction C to place the cutter inside, outside, or on the frame's profile. The compensation direction (left, right, or center) is based on the roughing direction.

OPTIONAL ENTRIES

After completing the required entries, choose your machining parameters with peck depth K, finish dimension L, and roughing width J (G7 only). An optional 2nd reference plane can be programmed for deep frame milling. You can program up to three independent feedrates with plunge feedrate P, roughing feedrate F, and finish feedrate V. Any feedrate that is not programmed defaults to the modal feedrate.

FRAME GEOMETRY - REQUIRED ENTRIES

All Frame Mill events **must** program these entries. Refer to the illustrations on page 1.

G0 or G7 - PERFORM THE FRAME or POCKET G CYCLE

G selects a frame or pocket mill cycle. When not programmed, G defaults to G0.

X, Y - FRAME WIDTH AND HEIGHT

X defines the frame width parallel to the X axis. Y defines the frame height parallel to the Y axis. Enter X and Y as unsigned values. X and Y must be entered in all Frame Mill events

Z - DEPTH

Z is the frame's final depth. Depth is entered as the signed Z axis coordinate, or the signed incremental distance from the W reference plane. Z must be entered in all Frame Mill events.

R - CORNER RADIUS

R is the frame's corner radius. For G0 frames with center or outside cutter compensation, the corner radius must be greater than or equal to zero. For G0 frames with inside cutter compensation, and all G7 frames, the sum of the cutter's radius[1] and the finish dimension L must be less than the corner radius R. R is an unsigned entry, and must be entered in all Frame Mill events.

W - 1st Z AXIS REFERENCE PLANE

W is a reference plane, parallel to the XY plane, and typically .1" (2 mm) above the part surface. Z advance slows from rapid to a programmed feedrate when the cutter crosses W. W is a signed absolute coordinate, and is modal with all following event types which use reference plane W.

Note: This entry is programmed with the letter "I" if your control is equipped with an auxiliary axis that is programmed with the letter "W". Refer to Part 1-Section 2.

O - 2nd Z AXIS REFERENCE PLANE

O is a 2nd Z axis reference plane parallel to the XY plane. Program O as the signed, absolute coordinate that locates the Z axis position where you want the tool to return to prior to making the final XY rapid move to the frame's start/end point. O is an optional entry; if not programmed, it will default to the value of the W plane.

D - MILLING DIRECTION

The four D entries let you specify the cutting direction for roughing and finishing the frame. If a finish dimension L is programmed, D0 and D1 maintain the same direction, while D2 and D3 reverse it before the finish pass. If L is not programmed, the finish direction is ignored. When not programmed, D defaults to D0.

C - CUTTER RADIUS COMPENSATION

C selects the cutter's orientation to the frame profile. C0 sets the cutter center on the profile. C1 or C2 selects a radius offset left or right of the profile[1]. Choose left or right compensation by looking along the roughing direction D and selecting the direction that puts the tool on the correct side of the profile. G0 frame cycles allow outside, tool center, or inside milling. G7 pocket cycles must choose D then a C entry which logically places the tool inside the profile.

[1] The active tool's diameter must be stored in the Tool Tables mode, and must be made active by executing its tool diameter code in a prior M Function event.

Both cycles automate the cutter's positioning to the profile. C must be entered in all Frame Mill events.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

OPTIONAL or MODAL ENTRIES

F - FEEDRATE

F programs the roughing feedrate. If P or V are not programmed, their respective operations will default to feedrate F. F is modal with all following event types which use feedrate F.

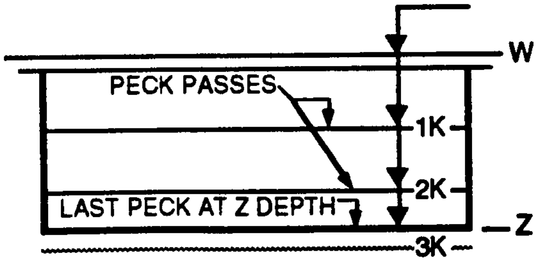
P - PLUNGE RATE

P is a Z axis feedrate which is operative as the cutter plunges from reference plane W to depth Z. A Frame Mill event with P not programmed plunges at feedrate F. When a G7 cycle programs P as "incremental", the plunge rate also controls the first two XY pocking moves (where the cutter is under full load).

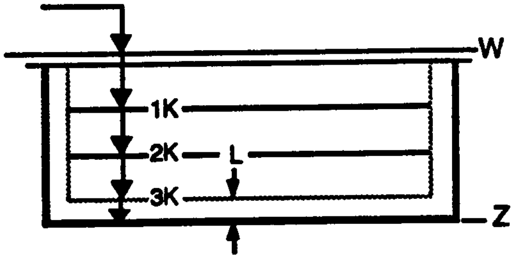
K - PECK INTERVAL

K creates a deep milling routine by specifying a maximum depth per plunge. The control will calculate the number of frame mill passes that are required to reach the programmed depth. Passes are performed at each successive interval of K from W, with the last pass interval reduced to your final Z depth. When L (absolute) requests a bottom finish, the final plunge depth is further reduced by finish dimension L to leave room for the bottom finish pass. K must be programmed in every Frame Mill event that requires a deep milling cycle, and is entered as an unsigned value. Refer to the illustrations below.

FRAME WITH PECK, WITHOUT FINISH PASS



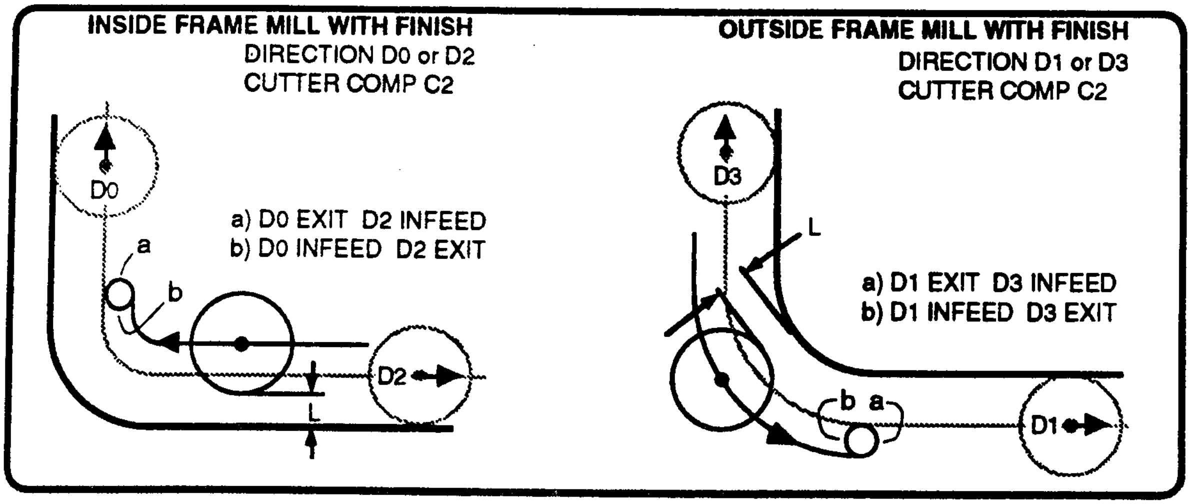
FRAME WITH PECK and FINISH PASS



L, V - FINISH DIMENSION and FEEDRATE

L is an unsigned finish dimension. When entered as an absolute value, L adds bottom and side finish passes to the milling cycle. When L is incremental, only the side pass is performed. The side finish feedrate V is entered as an unsigned absolute value, and if not programmed, defaults to feedrate F.

The finish cut occurs in the direction specified with direction D. You can select to rough and finish in the same direction, or reverse cutting direction between passes. To eliminate dwell marks at the start of the finish pass, the tool tangentially feeds into the part along an arc with a diameter equal to the finish dimension L. The tool also tangentially exits the part along this arc at the end of the finish pass. Sample tangential entry and exit moves are shown below.



J - CORNER ROUGHING WIDTH (G7 only)

J is used for G7 pocket milling cycles. J controls the maximum width of material that is removed on each pocket pass. Program J only when you wish to reduce the amount of material the cutter will remove per outward pass from the default value. If J is not entered, the control sets a default value equivalent to the cutter diameter -.010" (-.254 mm) in the corner. Cross-path width is approximately equal to .7J.

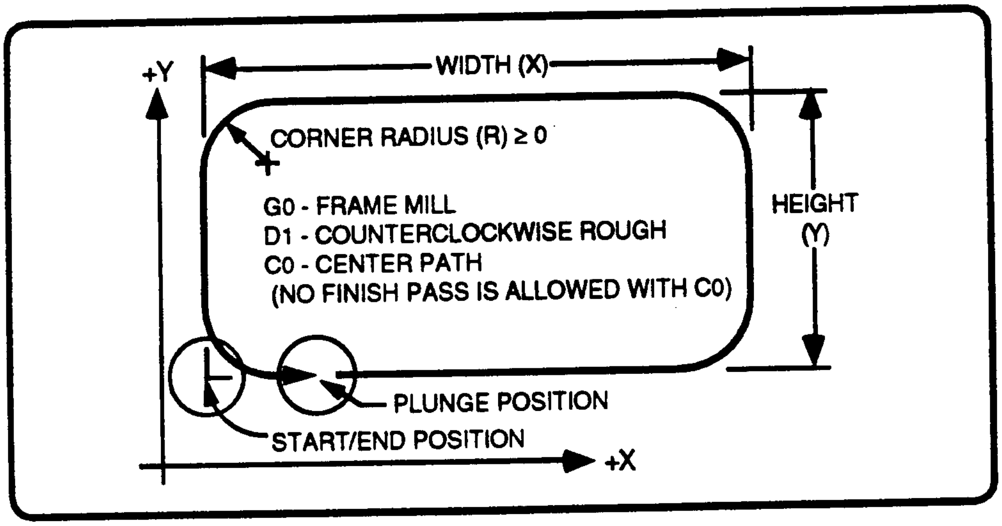
J should not exceed the cutter diameter. Be aware that the control will not use J directly but will calculate, under the constraint of J, the number of outward passes required to pocket the frame. To insure that all passes are of uniform width, the control rounds its calculations up to the next integer number of passes, and proportionally reduces the roughing width to fit the number of passes.

G0 FRAME MILL TYPES

There are three basic frames which can be programmed with G0 Frame Mill events. Each frame is described below. Note that the combination of cutting direction (D0-D3) and the cutter compensation direction (C0-C2) dictates which type of frame is cut.

TOOL CENTER FRAME MILL

Enter C0 to program a tool center frame mill. When a tool center frame event executes, the tool's center travels around the frame's profile. Only one pass occurs at Z depth unless peck depth K causes multiple passes at increasing Z depths. No finish pass is allowed with tool center frame mills. The corner radius R for this type of frame may be any value greater than or equal to zero.



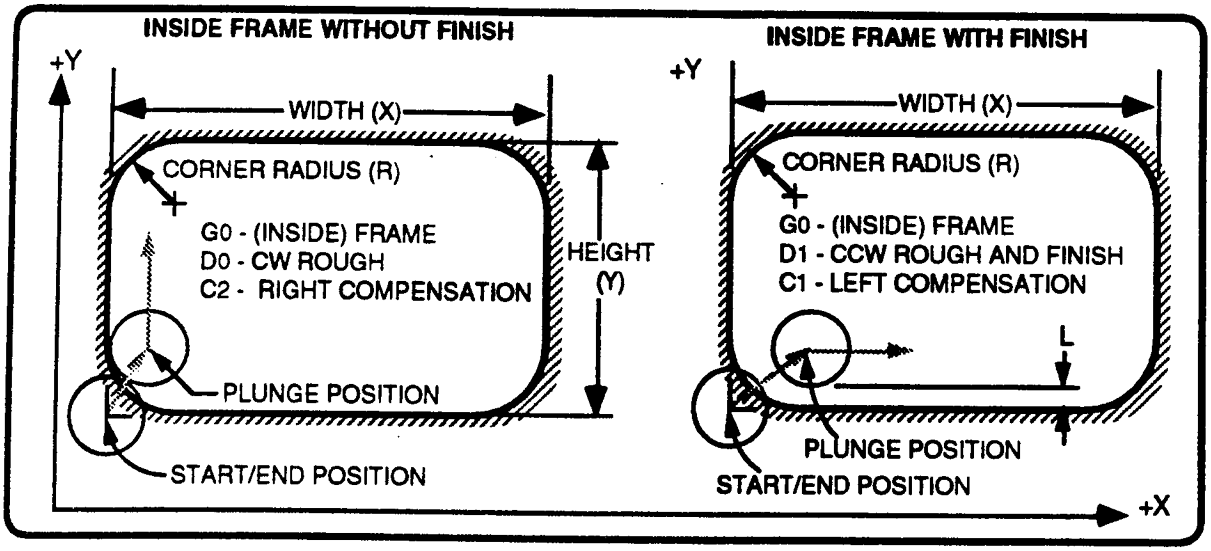
TOOL CENTER FRAME TOOL MOTION

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point determined by the control (see illustration, above). The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool center mills around the frame at feedrate F. If a peck is programmed, the tool feeds at P to the next depth, then repeats this step until the final depth is reached.
- When the tool completes the pass at Z depth, it rapids to O, then rapids to the frame's lower left corner. End of event.

G0 - INSIDE FRAME MILL

Inside frame mills, **below**, maintain your programmed X, Y and R dimensions on the inside of the rectangular frame. When L is not programmed, the tool will make one roughing path around the frame. When L is programmed, the tool will make additional finish passes around the frame to remove L inches (or millimeters) from the sides and bottom[2]. The total width of the material removed during this event is equal to the tool's diameter plus the programmed finish dimension L.

You can program inside frame mill cycles with any of the following combinations of rough/finish and cutter compensation directions: (G0, D0, C2), or (G0, D2, C2), or (G0, D1, C1), or (G0, D3, C1).



INSIDE FRAME TOOL MOTION - WITHOUT FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The plunge location is on the inside wall of the frame, but the exact location varies with the roughing direction D, the cutter comp direction C, and the cutter's diameter. Refer to the illustration **above left**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds once around the frame at feedrate F. Repetitive passes occur at increasing Z depths if a peck cycle has been programmed.
- The Z axis rapids to O, then XY rapid to the frame's lower left corner. End of event.

INSIDE FRAME TOOL MOTION - WITH FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The plunge location is L units inside the frame, but the exact location varies with the roughing direction D, the cutter comp direction C, the finish dimension L, and the cutter's diameter. Refer to the illustration **above right**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.

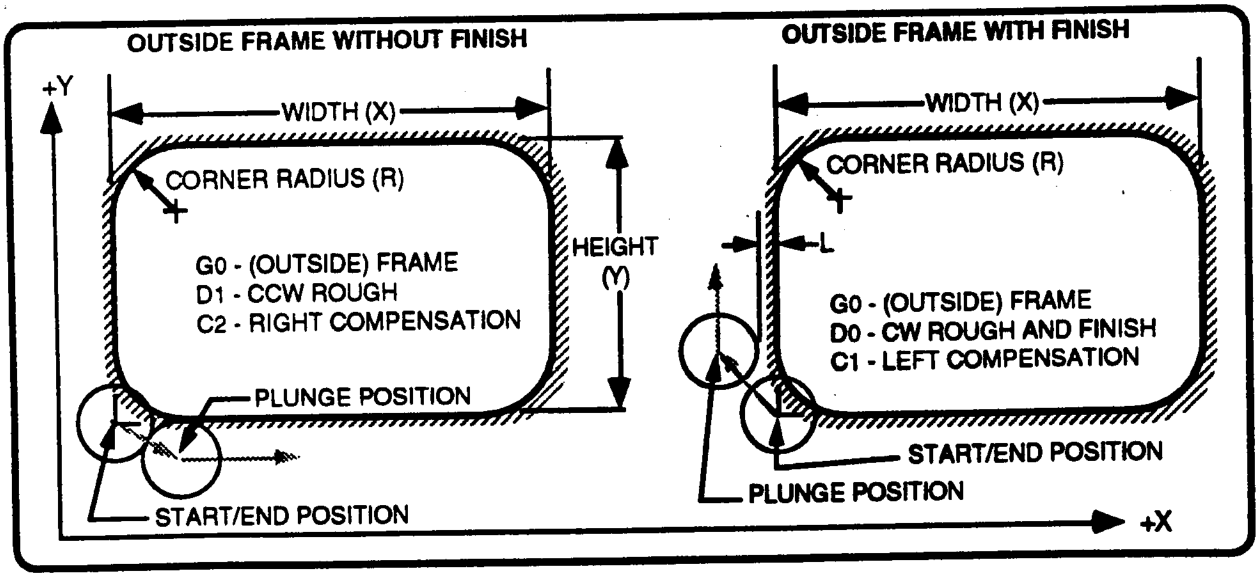
[2] A finish pass will occur on the bottom only when L is programmed as an absolute value. The bottom finish will occur at feedrate F, not V.

- The tool makes one roughing pass, at feedrate F, leaving L inches (or millimeters) to be removed on the finish pass. If a peck is programmed, repetitive roughing passes occur until the correct depth is reached.
- If L absolute requests a bottom finish, the tool feeds L units to depth Z, and makes one bottom finish pass at feedrate F.
- The tool makes a final finish pass at feedrate V to remove L units from the frame. The tool tangentially feeds in and out of the part at the start and end of the finish pass. The side finish pass occurs at Z depth; the finish pass does not peck.
- The Z axis rapids to O, then XY rapid to the frame's lower left corner. End of event.

G0 - OUTSIDE FRAME MILL

Outside frame mills, **below**, maintain your programmed X, Y and R dimensions on the outside of the rectangular frame. When L is not programmed, the tool makes one roughing path around the frame; when L is programmed, the tool makes additional finish passes around the frame to remove L inches (or millimeters) from the sides and bottom[2]. The total width of the material removed during this event is equal to the tool's diameter plus the programmed finish dimension L.

You can program outside frame mill cycles with any of the following combinations of rough/finish and cutter compensation directions: (G0, D0, C1), or (G0, D2, C1), or (G0, D1, C2), or (G0, D3, C2).



OUTSIDE FRAME TOOL MOTION - WITHOUT FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The plunge location is on the outside wall of the frame, but the exact location varies with the roughing direction D, the cutter comp direction C, and the cutter's diameter. Refer to the illustration **above left**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds once around the frame at feedrate F. Repetitive passes occur at increasing Z depths if a peck cycle has been programmed.
- The Z axis rapids to O, then XY rapid to the frame's lower left corner. End of event.

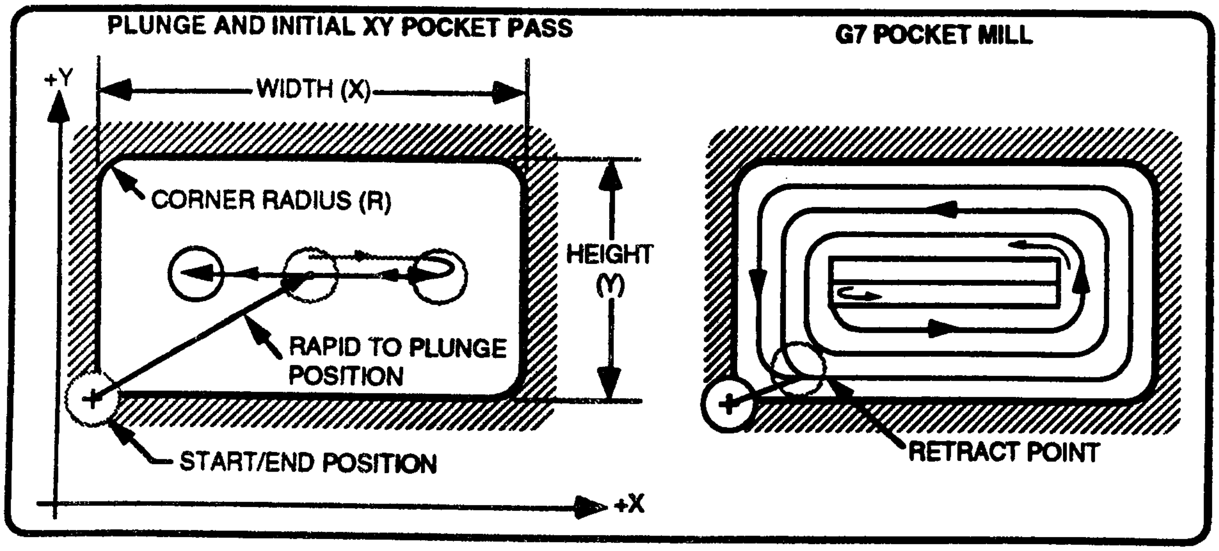
OUTSIDE FRAME TOOL MOTION - WITH FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The plunge location is L units outside the frame, but the exact location varies with the roughing direction D, the cutter comp direction C, the finish dimension L, and the cutter's diameter. Refer to the illustration **above right**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool makes one roughing pass, at feedrate F, leaving L inches (or millimeters) to be removed on the finish pass. If a peck is programmed, repetitive roughing passes occur until the correct depth is reached.
- If L absolute requests a bottom finish, the tool feeds L units to depth Z, and makes one bottom finish pass at feedrate F.
- The tool makes a final finish pass at feedrate V to remove L units from the frame. The tool tangentially feeds in and out of the part at the start and end of the finish pass. The side finish pass occurs at Z depth; the finish pass does not peck.
- The Z axis rapids to O, then XY rapid to the frame's lower left corner. End of event.

G7 POCKET MILL

Frame Mill events with G7 programmed, **below**, remove all material within the boundaries of the frame defined by X, Y, and R. If desired, you can limit the tool overlap distance by programming the roughing width J. You can also choose to make the first two XY feed motions at feedrate P by programming P as an incremental number.

You can program pocket mills with the following combination of cutting directions and cutter compensation directions: (G7, D0, C2), or (G7, D2, C2), or (G7, D1, C1), or (G7, D3, C2). Any other combination of C and D will cause a fault at run time. A finish dimension L and/or a peck depth K can be added to this cycle.



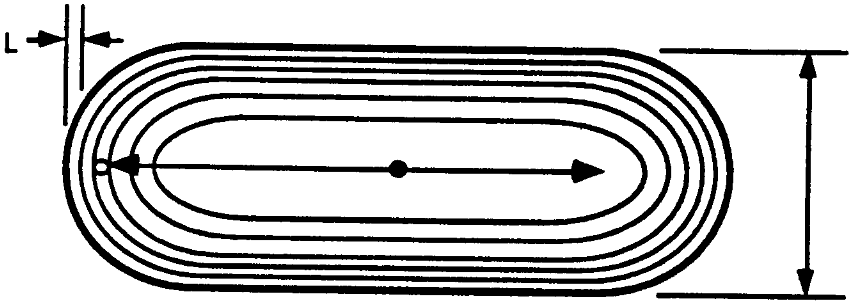
G7 POCKET MILL TOOL MOTION WITHOUT FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The location of the tool plunge is the geometric center of frame. Refer to the illustrations **above**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds in +X, reverses direction and feeds in -X to create a slot. If P incremental is programmed, this move occurs at feedrate P.
- The tool feeds outwards in a spiral rectangular pattern until all of the material is removed from within the frame. Repetitive passes occurs at increasing Z depths if a peck cycle has been programmed. Before each peck, the Z axis retracts .1", then XY rapid to the plunge position.
- The tool retracts at rapid to O.
- The tool rapids in XY back to the frame's lower left corner. End of event.

G7 POCKET MILL TOOL MOTION WITH FINISH PASS

- A prior event must move the tool center to the XY coordinate which locates the frame's lower left corner.
- The X and Y axes rapid to the plunge point. The location of the tool plunge is the geometric center of frame. Refer to the illustrations **above**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds in +X axis, reverses direction and feeds in -X to create a slot. If P incremental is programmed, this initial move occurs at feedrate P.
- The tool feeds outwards in a spiral rectangular pattern until all but the finish dimension has been removed from within frame. Repetitive passes occurs at increasing Z depths if a peck cycle has been programmed. Before each peck, the Z axis retracts .1", then XY rapid to the plunge position.
- If L absolute requests a bottom finish, the tool feeds L units to depth Z, and makes a bottom finish pass at feedrate F.
- The tool makes a final finish pass at feedrate V to remove L inches (or millimeters) from the frame. The tool tangentially feeds in and out of the part at the start and end of the finish pass. The side finish pass occurs at Z depth; the finish pass does not peck.
- The tool retracts to O.
- The tool rapids back to the frame's lower left corner. End of event.

G7 - NARROW FRAME MILL



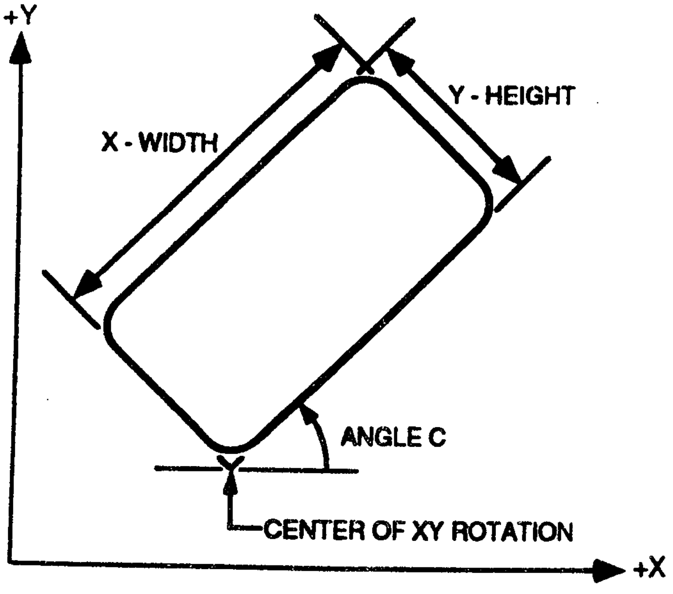
NARROW FRAME MILL

The G7 pocket mill allows the width and height of a frame to be as narrow as twice the corner radius R. Narrow frames with X width or Y height within the range of 2R to 2(R+L) are milled from the center outward, but the feed to the next outward pass is parallel to the larger of the two dimensions.

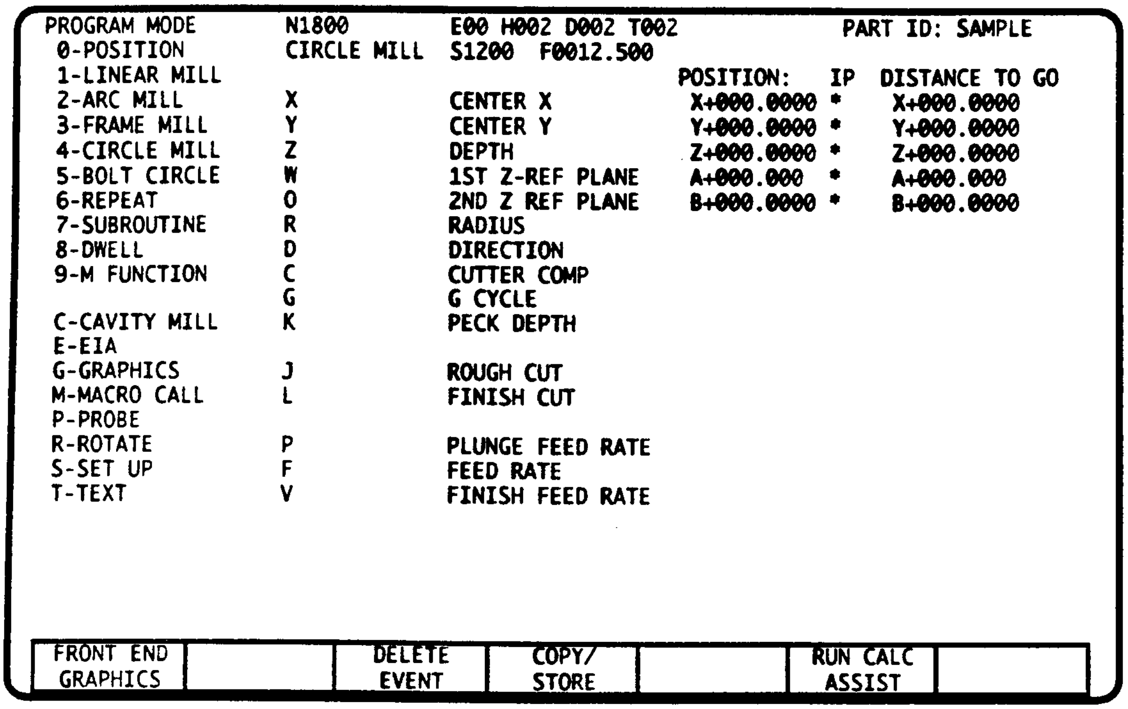
FRAME ROTATION - G0 or G7

The frame's orientation is parallel to X and Y, but the XY orientation may be altered before milling the frame. After a Position event has moved the cutter center over the frame's lower left corner, program a Rotate event with the frames lower left corner as the XY center of rotation coordinates, and the frames rotary orientation as angle C. The following Frame Mill event is then programmed with its base as the X width and its side as the Y height.

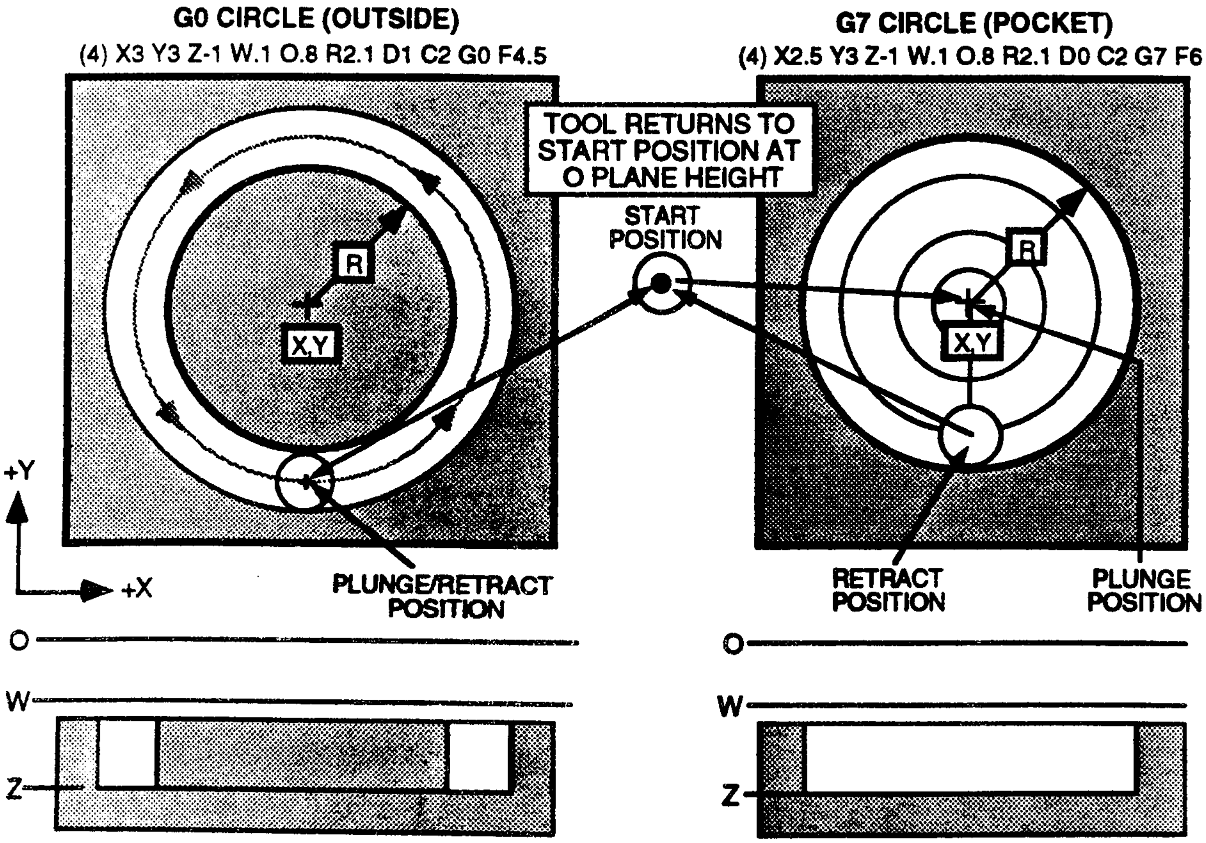
Axes rotation will be active until it is canceled by a following Rotate event with C0 programmed, or an M30/M02 M Function event executes. Refer to the Rotate Event section for further details on axes rotation.



ROTATING FRAME ORIENTATION



```
PROGRAM MODE        N1800         E00 H002 D002 T002              PART ID: SAMPLE
 0-POSITION         CIRCLE MILL   S1200  F0012.500
 1-LINEAR MILL                                    POSITION:   IP  DISTANCE TO GO
 2-ARC MILL         X             CENTER X         X+000.0000 *    X+000.0000
 3-FRAME MILL       Y             CENTER Y         Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL      Z             DEPTH            Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE      W             1ST Z-REF PLANE  A+000.000  *    A+000.000
 6-REPEAT           O             2ND Z REF PLANE  B+000.0000 *    B+000.0000
 7-SUBROUTINE       R             RADIUS
 8-DWELL            D             DIRECTION
 9-M FUNCTION       C             CUTTER COMP
                    G             G CYCLE
 C-CAVITY MILL      K             PECK DEPTH
 E-EIA
 G-GRAPHICS         J             ROUGH CUT
 M-MACRO CALL       L             FINISH CUT
 P-PROBE
 R-ROTATE           P             PLUNGE FEED RATE
 S-SET UP           F             FEED RATE
 T-TEXT             V             FINISH FEED RATE
```



G0 CIRCLE (OUTSIDE)
(4) X3 Y3 Z-1 W.1 O.8 R2.1 D1 C2 G0 F4.5
G7 CIRCLE (POCKET)
(4) X2.5 Y3 Z-1 W.1 O.8 R2.1 D0 C2 G7 F6
TOOL RETURNS TO START POSITION AT O PLANE HEIGHT
START POSITION
R
X,Y
R
X,Y
+Y
+X
PLUNGE/RETRACT POSITION
RETRACT POSITION
PLUNGE POSITION
O
W
Z
O
W
Z

CIRCLE MILL EVENT - INTRODUCTION

This event cuts a circle to a specified depth. As shown, **facing page**, you can choose to mill around the circle's perimeter, or remove (pocket) all material within the circle's perimeter. It is not necessary to position to the circle prior to this event unless O reference plane clearance cannot be assured along a direct path for tool return between the circle and the tool's start position. **This clearance must be observed since the cutter returns to the O plane and then rapids in a direct path to its starting XY coordinates at the event's conclusion.**

CIRCLE MILL DATA

Parameter	Description	Entry
N	Sequence Number	Required
G	G Cycle G0=circle, G7=pocket	Defaults to G0
X	Circle Center, X	Required
Y	Circle Center, Y	Required
Z	Depth	Required
W	1st Z Reference Plane	Modal
O	2nd Z Reference Plane	Modal
R	Radius	Required

D	Milling Direction
	D0=rough CW/finish CW
	D1=rough CCW/finish CCW
	D2=rough CW/finish CCW
	D3=rough CCW/finish CW
		Defaults to D0

C	Cutter Radius Compensation C0=off, C1=left, C2=right	Required
K	Peck Mill	Optional
J	Roughing Width (G7 only)	Optional
L	Finish Dimension L=finish bottom and sides, L/= finish sides only	Optional
P	Plunge Feedrate	Defaults to F
F	Feedrate	Modal
V	Finish Feedrate	Defaults to F

REQUIRED ENTRIES

Program G0 to mill around the circle's perimeter, or G7 to pocket the circle. A pocketing operation removes all material within the circle. Describe the circle's shape with center X and Y, radius R, depth Z, and reference planes W and O. Next program direction D to choose conventional or climb milling for the rough/finish passes. After selecting the direction, choose the cutter compensation direction C to place the cutter inside, outside, or on the circle's profile. The compensation direction (left, right, or center) is based on the roughing direction.

OPTIONAL ENTRIES

After completing the required entries, choose your machining parameters with peck depth K, finish dimension L, and roughing width J (G7 only). You can program up to three independent feedrates with plunge feedrate P, roughing feedrate F, and finish feedrate V. Any feedrate that is not programmed defaults to the modal feedrate.

CIRCLE GEOMETRY - REQUIRED ENTRIES

All Circle Mill events **must** program these entries. Refer to illustrations on **page 1.**

G0 or G7 - PERFORM THE CIRCLE or CIRCLE POCKET G CYCLE

G selects a circle or pocket circle mill cycle. G0 is the default cycle.

X, Y - CIRCLE CENTER

X and Y are the circle's center coordinates. Enter the signed absolute X and Y axes coordinates, or the signed incremental distance from the cutter's position at the start of the event. Absolute and incremental entries may appear in the same event. X and Y must be entered in all Circle Mill events.

Z - DEPTH

Z is the final depth the cutter reaches while milling the circle. Depth is entered as the signed Z axis coordinate, or the signed incremental distance from the W reference plane. Z must be entered in all Circle Mill events.

R - RADIUS

R is the circle's radius. R is an unsigned entry, and is required in all Circle Mill events.

W - 1st Z AXIS REFERENCE PLANE

W is a reference plane, parallel to the XY plane, and typically .1" (2 mm) above the part surface. Z advance slows from rapid to a programmed feedrate when the cutter crosses W. W is a signed absolute coordinate, and is modal with all following event types which use reference plane W.

Note: This entry is programmed with the letter "I" if your control is equipped with an auxiliary axis that is programmed with the letter "W". Refer to Part 1-Section 2.

O - 2nd Z AXIS REFERENCE PLANE

O is a 2nd Z axis reference plane parallel to the XY plane. Program O as the signed, absolute coordinate that locates the Z axis position where you want the tool to return to prior to making the final XY rapid move to the tool's position prior to the Circle Mill event. O is an optional entry; if not programmed, it will default to the value of the W plane.

D - MILLING DIRECTION

The four D entries specify the cutting direction for roughing and finishing the circle. D0 and D1 maintain the same cutting direction while D2 and D3 reverse it before the finish pass. If L is not programmed, the finish direction is ignored. When not programmed, D defaults to D0.

C - CUTTER RADIUS COMPENSATION

C selects the cutter's orientation to the circle's profile. C0 sets the cutter center on the profile. C1 or C2 selects a radius offset left or right of the profile[1]. Choose the left or right offset by looking along the roughing direction D and selecting the direction that puts the tool on the correct side of the profile. G0 cycles allow outside, tool center, or inside milling. A G7 pocket cycle must choose D then a C entry that places the tool inside the profile. Both cycles automate the cutter's positioning to the profile.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

OPTIONAL or MODAL ENTRIES

F - FEEDRATE

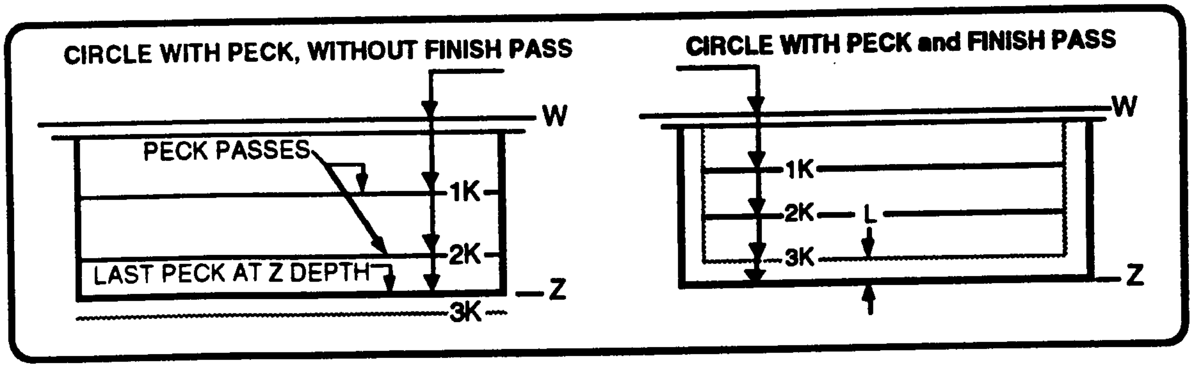
F programs the roughing feedrate. If P or V is not programmed, their respective operations will default to feedrate F. F is modal with all following event types which use feedrate F.

P - PLUNGE RATE

P is a Z axis feedrate which is operative as the cutter plunges from reference plane W to depth Z. A Circle Mill event with P not programmed plunges at feedrate F.

K - PECK INTERVAL

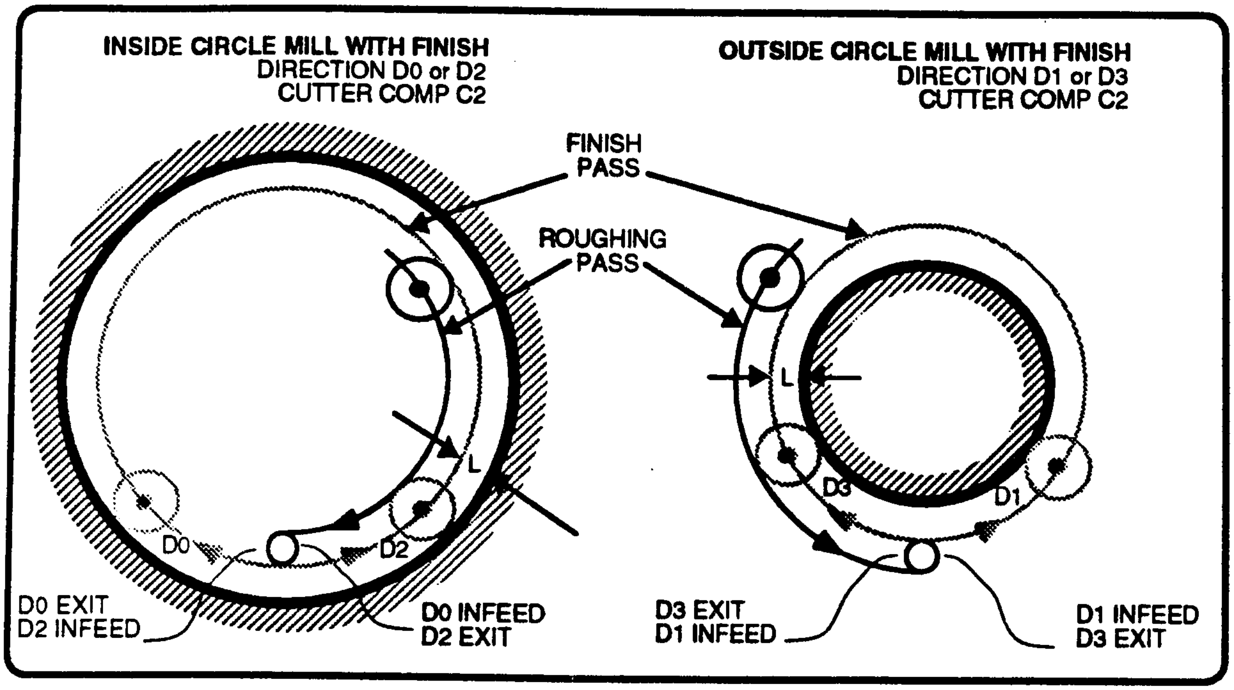
K creates a deep milling routine by specifying a maximum depth per plunge. The control will calculate the number of circle mill passes that are required to reach the programmed depth. Passes are performed at each successive interval of K from W, with the last pass interval reduced to your final Z depth. When L (absolute) requests a bottom finish, the final plunge depth is further reduced by finish dimension L to leave room for the bottom finish pass. K must be programmed in every Circle Mill event that requires a deep milling cycle, and is entered as an unsigned value. Refer to the illustrations below.


CIRCLE WITH PECK, WITHOUT FINISH PASS
W
PECK PASSES
1K
2K
LAST PECK AT Z DEPTH
Z
3K
CIRCLE WITH PECK and FINISH PASS
W
1K
2K
L
3K
Z

[1] The active tool's diameter must be stored in the Tool Tables mode, and must be made active by executing its tool diameter code in a prior M Function event.

L, V - FINISH DIMENSION AND FEEDRATE - G0 or G7 OPTIONS

L is an unsigned finish dimension. When entered as an absolute value, L adds bottom and side finish passes to the milling cycle. When L is incremental, only the side pass is performed. The side finish feedrate V is entered as an unsigned absolute value, and if not programmed, defaults to feedrate F.



The finish cut occurs in the direction specified with direction D. You can select to rough and finish in the same direction, or reverse cutting direction between passes. To eliminate dwell marks at the start of the finish pass, the tool tangentially feeds into the part along an arc with a diameter equal to the finish dimension L. The tool also tangentially exits the part along this arc at thc end of the finish pass. Sample tangential entry and exit moves are shown above.

J - ROUGHING WIDTH - G7 OPTION

J controls the maximum width of material that is removed on each pocket pass. J is a maximum roughing width that will not be exceeded on any pass while milling the pocket. Program J only when you wish to reduce the amount of material the cutter will remove per outward pass from the default value. If J is not entered, the control sets a default value equivalent to the cutter diameter -.010" (-.254 mm).

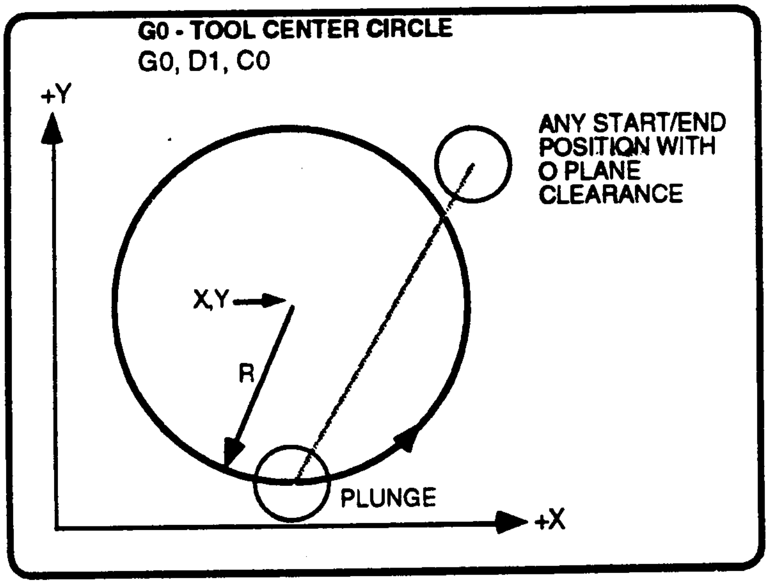
J should not exceed the cutter diameter. Be aware that the control will not use J directly but will calculate, under the constraint of J, the number of outward passes required to pocket the circle. To insure that all passes are of uniform width, the control rounds its calculations up to the next integer number of passes, and proportionally reduces the roughing width to fit the number of passes.

G0 CIRCLE MILL TYPES

There are three basic circles which can be programmed with G0 Circle Mill events. Each circle is described below. Note that the combination of cutting direction (D0-D3) and the cutter compensation direction (C0-C2) dictates which type of circle is cut.

TOOL CENTER CIRCLE MILL

Enter C0 to program a tool center circle mill. When this type of Circle Mill event executes, the tool's center travels around the circle at radius R. Only one pass occurs at Z depth unless peck depth K causes multiple passes at increasing Z depths. No finish pass is allowed with tool center circle mills.



TOOL CENTER CIRCLE TOOL MOTION

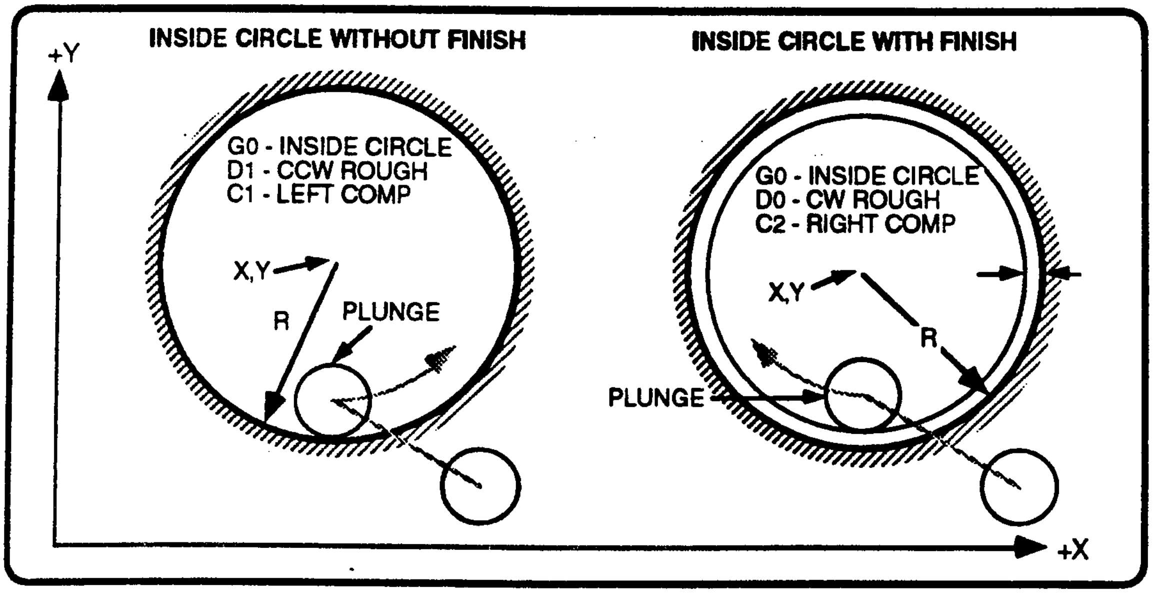
- The X and Y axes rapid to the plunge point determined by the control (see illustration, **above**). The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool center mills around the circle at feedrate F. If a peck is programmed, the tool feeds at P to the next depth, then repeats this step until the final depth is reached.
- When the tool has completed the roughing pass, it rapids to O, then XY rapid back to the tool's XY start point. End of event.

GO - INSIDE CIRCLE MILL

Inside circle mills, below, maintain your programmed radius R on the inside of the circle. When L is not programmed, the tool will make one roughing path around the circle. When L is programmed, the tool will make additional finish passes around the circle to remove L inches (or millimeters) from the sides and bottom[2].

The total width of the material removed during an inside circle mill is equal to the tool's diameter plus the programmed finish dimension L.

You can program inside circle mill cycles with any of the following combinations of rough/finish directions and cutter compensation directions: (G0, D0, C2), or (G0, D2, C2), or (G0, D1, C1), or (G0, D3, C1).



INSIDE CIRCLE TOOL MOTION - WITHOUT FINISH PASS

- The X and Y axes rapid to the plunge point. The plunge location is on the inside wall of the circle. Refer to the illustration above left.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds once around the circle at feedrate F. Repetitive passes occur at increasing Z depths if a peck cycle has been programmed.
- The tool rapids to O, then rapids back to the tool's XY start point. End of event.

INSIDE CIRCLE TOOL MOTION - WITH FINISH PASS

- The X and Y axes rapid to the plunge point. The plunge location is L units inside the circle's wall. Refer to the illustration above right.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool makes one roughing pass, at feedrate F, leaving L inches (or millimeters) to be removed on the finish pass. If a peck is programmed, repetitive roughing passes occur until the correct depth is reached.
- If L absolute requests a bottom finish, the tool feeds L units to depth Z, and makes one bottom finish pass at feedrate F.

[2] A finish pass will occur on the bottom only when L is programmed as an absolute value. The bottom finish will occur at feedrate F, not V.

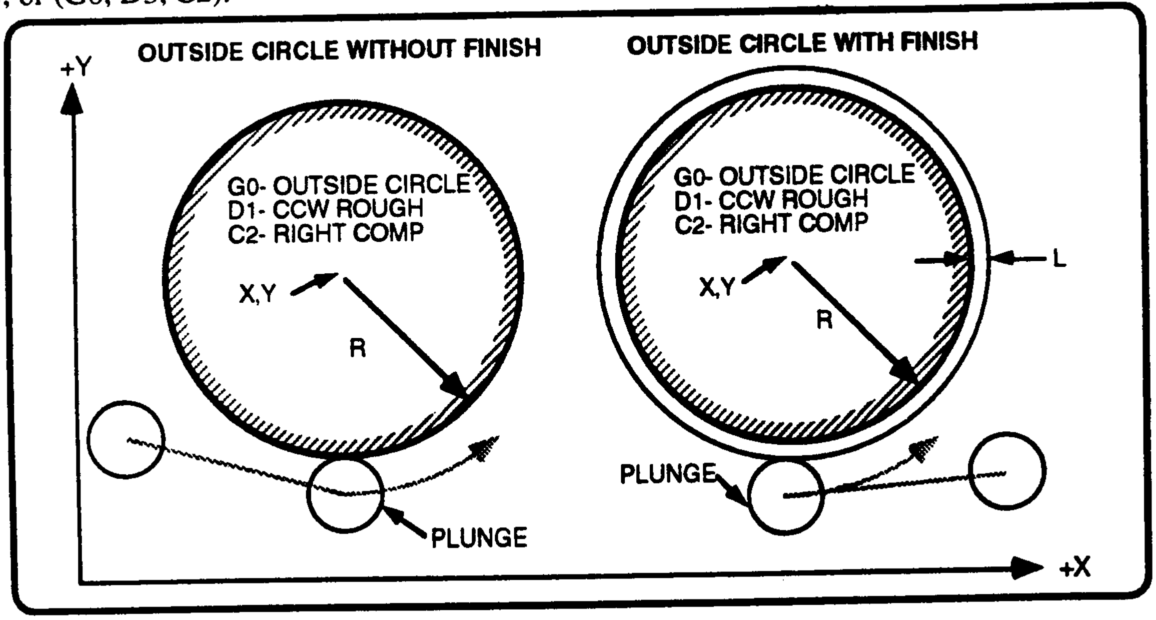
- The tool makes a final finish pass at feedrate V to remove L units from the sides of the circle. The tool tangentially feeds in and out of the part at the start and end of the finish pass. The side finish pass occurs at Z depth; the finish pass does not peck.
- The tool rapids to O, then rapids back to the tool's XY start point. End of event.

G0 - OUTSIDE CIRCLE MILL

Outside circle mills, **below**, maintain your programmed radius R on the outside of the circle's profile. When L is not programmed, the tool will make one roughing path around the circle; when L is programmed, the tool will make additional finish passes around the circle to remove L inches (or millimeters) from the sides and bottom^2.

The total width of the material removed during this event is equal to the tool's diameter plus the programmed finish dimension L.

You can program outside circle mill cycles with any of the following combinations of rough and finish directions, and cutter compensation direction: (G0, D0, C1), or (G0, D2, C1), or (G0, D1, C2), or (G0, D3, C2).



OUTSIDE CIRCLE TOOL MOTION - WITHOUT FINISH PASS

- The X and Y axes rapid to the plunge point. The plunge location is on the outside wall of the circle. Refer to the illustrations **above left**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool feeds once around the circle at feedrate F. Repetitive passes occur at increasing Z depths if a peck cycle has been programmed.
- The tool rapids to O, then rapids back to the tool's XY start point. End of event.

OUTSIDE CIRCLE TOOL MOTION - WITH FINISH PASS

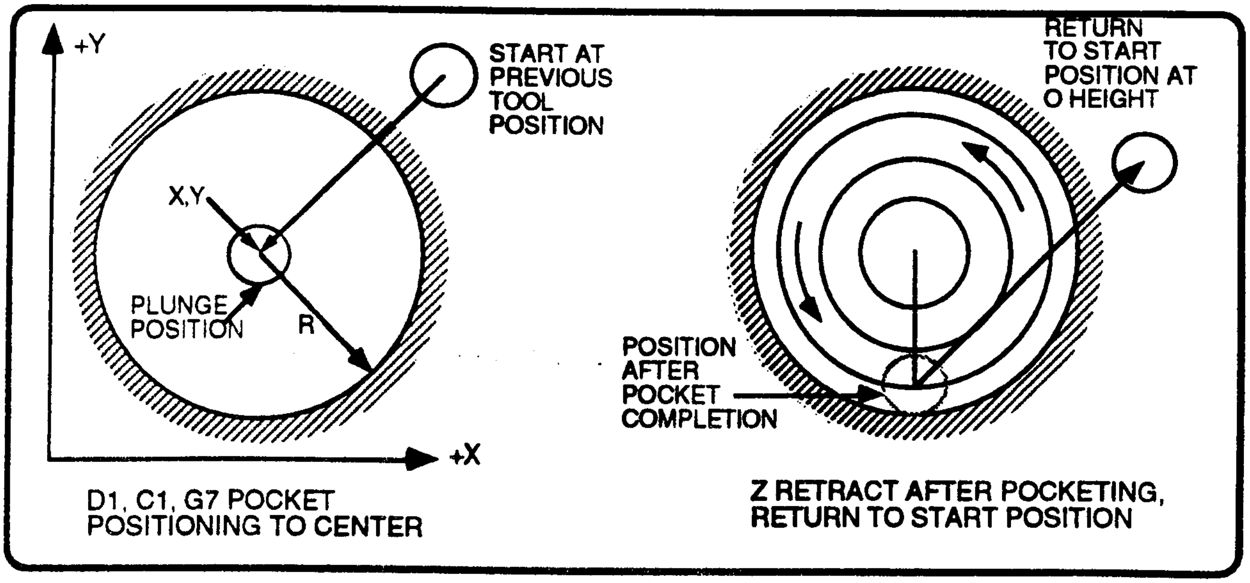
- The X and Y axes rapid to the plunge point. The plunge location is L units outside of the circle's profile. Refer to the illustrations **above right**.
- The Z axis rapids to the W plane, slows to feedrate P until the tool reaches the programmed depth, or peck increment.
- The tool makes one roughing pass, at feedrate F, leaving L inches (or millimeters) to be removed on the finish pass. If a peck is programmed, repetitive roughing passes occur until the correct depth is reached.

- If L absolute requests a bottom finish, the tool feeds L units to depth Z, and makes one bottom finish pass at feedrate F.
- The tool makes a final finish pass at feedrate V to remove L units from the side of the circle. This occurs in one pass; the finish pass does not peck. The tool tangentially feeds in and out of the part at the start and end of the finish pass. The side finish pass occurs at Z depth; the finish pass does not peck.
- The Z axis rapids to O, then rapids back to the tool's XY start point. End of event.

G7 - CIRCLE POCKET MILL

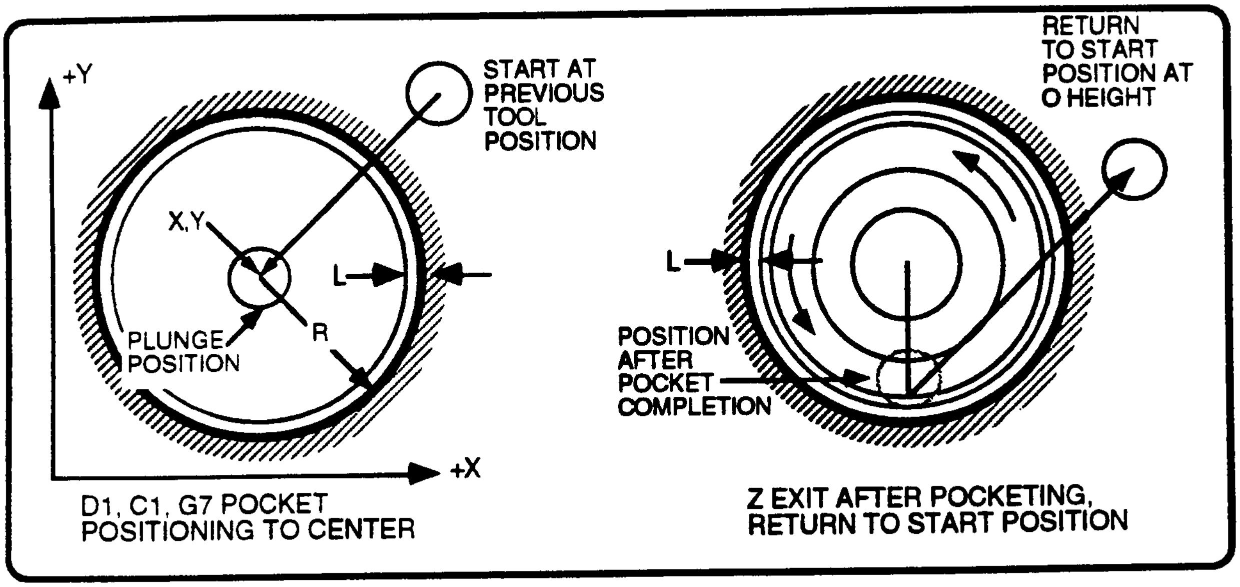
Circle Mill events with G7 programmed, below, will remove all material within the boundaries of the circle defined with X,Y and R. If desired, you can limit the tool overlap distance by programming the roughing width J.

You can program pocket mills with the following combination of cutting directions and cutter compensation directions: (G7, D0, C2), or (G7, D2, C2), or (G7, D1, C1), or (G7, D3, C1). Any other combination of C and D will cause a fault at run time. A finish dimension L and/or peck depth K can be added to this cycle.


+Y
START AT PREVIOUS TOOL POSITION
X,Y
PLUNGE POSITION
R
+X
D1, C1, G7 POCKET POSITIONING TO CENTER
RETURN TO START POSITION AT O HEIGHT
POSITION AFTER POCKET COMPLETION
Z RETRACT AFTER POCKETING, RETURN TO START POSITION

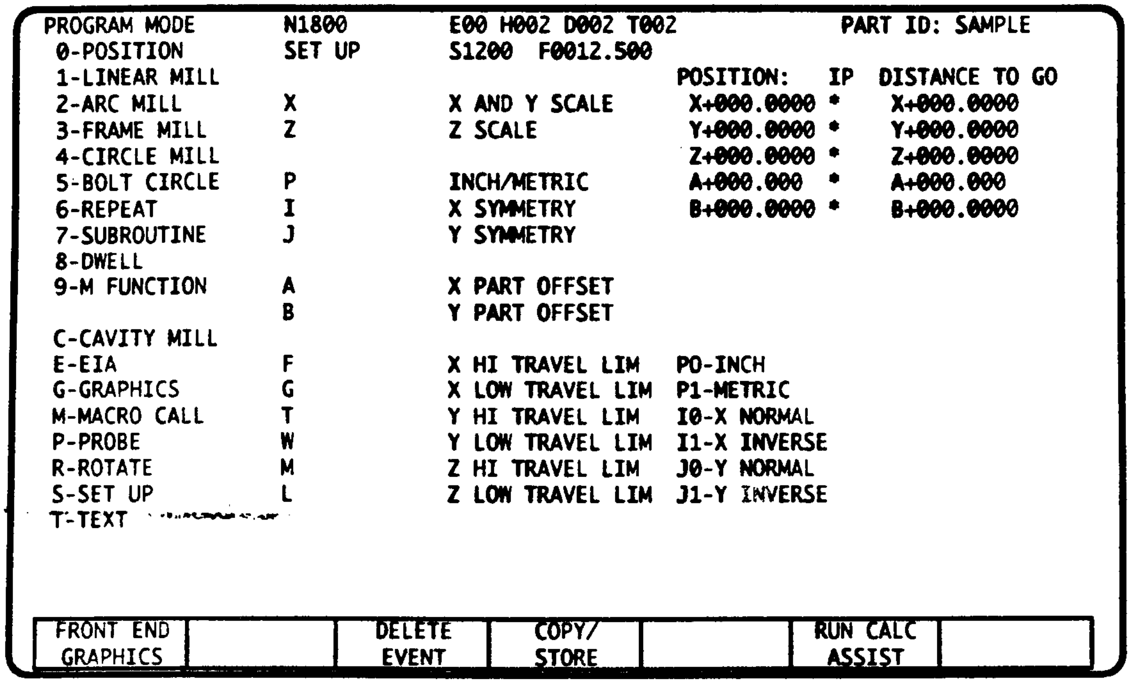
G7 POCKET MILL TOOL MOTION WITHOUT FINISH PASS

- The X and Y axes rapid the circle's center. Refer to the illustrations above.
- The Z axis rapids to W then feeds at P to depth Z, or peck increment K.
- The tool feeds a distance equal to the calculated roughing width in the - Y direction. Remember the control optimizes the roughing width J dimension to ensure constant metal removal.
- The tool cuts a circle, at feedrate F, around the arc center.
- The tool then moves outward and cuts another circle. This step repeats until the the tool reaches the radius R.
- If a peck is programmed, the tool retracts .1" in Z then rapids back to the center and repeats the process until the final Z depth is reached.
- The tool rapids to O, then rapids back to the tool's XY start point. End of event.


+Y
START AT PREVIOUS TOOL POSITION
X,Y
L
R
PLUNGE POSITION
+X
D1, C1, G7 POCKET POSITIONING TO CENTER
RETURN TO START POSITION AT O HEIGHT
L
POSITION AFTER POCKET COMPLETION
Z EXIT AFTER POCKETING, RETURN TO START POSITION

G7 POCKET MILL TOOL MOTION WITH FINISH PASS

- The X and Y axes rapid the circle's center. Refer to the illustrations above.
- The Z axis rapids to W then feeds at P to depth Z, or peck increment K.
- The tool feeds a distance equal to the calculated roughing width in the - Y direction. Remember the control optimizes the roughing width J dimension to ensure constant metal removal.
- The tool cuts a circle, at feedrate F, around the arc center.
- The tool then moves outward and cuts another circle. This step repeats until the the tool is L inches (or millimeters) from radius R.
- If a peck is programmed, the tool retracts .1" in Z then rapids back to the center and repeats the process until the the tool is L inches (or millimeters) from the Z depth. If L is programmed as an incremental dimension, no bottom finish pass occurs; therefore, the peck continues to depth Z.
- If L is programmed as an absolute dimension, the control will feed to the final Z depth and complete a pocket at feedrate F.
- The tool now tangentially feeds L units into the part, and make one more circular pass. At the end of the pass, the tool tangentially exits from the wall.
- The tool rapids to O, then rapids back to the tool's XY start point. End of event.



```
PROGRAM MODE      N1800       E00 H002 D002 T002              PART ID: SAMPLE
 0-POSITION       SET UP      S1200  F0012.500
 1-LINEAR MILL                                  POSITION:   IP  DISTANCE TO GO
 2-ARC MILL       X           X AND Y SCALE      X+000.0000 *    X+000.0000
 3-FRAME MILL     Z           Z SCALE            Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL                                   Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE    P           INCH/METRIC        A+000.000  *    A+000.000
 6-REPEAT         I           X SYMMETRY         B+000.0000 *    B+000.0000
 7-SUBROUTINE     J           Y SYMMETRY
 8-DWELL
 9-M FUNCTION     A           X PART OFFSET
                  B           Y PART OFFSET
 C-CAVITY MILL
 E-EIA            F           X HI TRAVEL LIM   P0-INCH
 G-GRAPHICS       G           X LOW TRAVEL LIM  P1-METRIC
 M-MACRO CALL     T           Y HI TRAVEL LIM   I0-X NORMAL
 P-PROBE          W           Y LOW TRAVEL LIM  I1-X INVERSE
 R-ROTATE         M           Z HI TRAVEL LIM   J0-Y NORMAL
 S-SET UP         L           Z LOW TRAVEL LIM  J1-Y INVERSE
 T-TEXT
```

SET UP EVENT DATA

Parameter	Description	Entry	
N	Sequence Number		Required
P	Inch/Metric Selection	P0 = Inch P1 = Millimeters	Optional
X	XY Axis Scale Factor		Optional
Z	Z Axis Scale Factor		Optional
I	X Axis Symmetry	I0 = Normal I1= Inverse	Optional
J	Y Axis Symmetry	J0 = Normal J1 = Inverse	Optional
A	X Axis Part Offset		Optional
B	Y Axis Part Offset		Optional
F	X Axis High (+) Travel Limit		Optional
G	X Axis Low (-) Travel Limit		Optional
T	Y Axis High (+) Travel Limit		Optional
W	Y Axis Low (-) Travel Limit		Optional
M	Z Axis High (+) Travel Limit		Optional
L	Z Axis Low (-) Travel Limit		Optional

SET UP EVENT - INTRODUCTION

The Set Up event lets you program the following options which affect the operation of all following part program events.

- switch the control to INCH or METRIC mode.
- set an XY plane, and/or Z axis scale factor.
- invert symmetry of the X and/or Y axis.
- shift the part zero location.
- assign Software Travel Limits.

Executing an event that contains one or more of these options has the same effect as using the keyboard to change the corresponding options in the Set Up mode. This is important because all settings, except scale factor, remain in effect until they are changed back to their default settings with another Set Up event, or by Set Up mode keyboard entry. As a rule of thumb, you should always program a Set Up event with your default settings immediately before the M30 End of Program M Function event.

P - INCH/METRIC SELECTION

The entry lets you switch the control into INCH or METRIC mode. This will cause the control to interpret all part program dimensions in the selected units. Set Up mode entries, Tool Table offsets, Position Display registers, Jog mode displays, and other stored machine dimensions will be converted to the selected units.

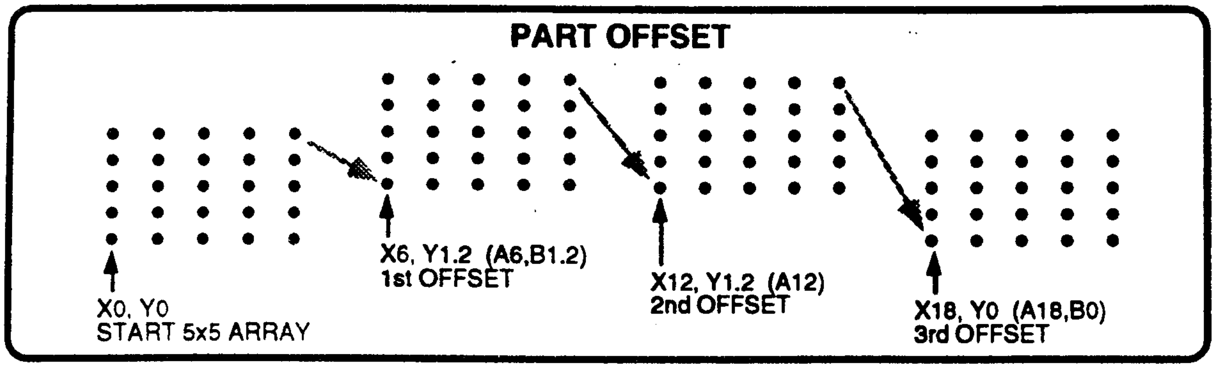
You may not program a part offset (A, B) or travel limit entries (F, G, T, W, M, L) in the same event that programs inch/millimeter (P0, P1).

A, B - PART OFFSET

A and B respectively enter X and Y part offsets. Part offsets let you shift part zero away from the datum. If a Fixture Offset (E Code) is active, the part offset will be further offset from the zero point that is established by the fixture.

Part offsets allow multiple parts on the same workpiece, or multiple fixtures on a machine. A and B are signed absolute entries representing the offset coordinates from the datum. Executing A and/or B in a Set Up event does not cause tool motion. However, X and Y will move to a point relative to the part offset when the next motion command executes.

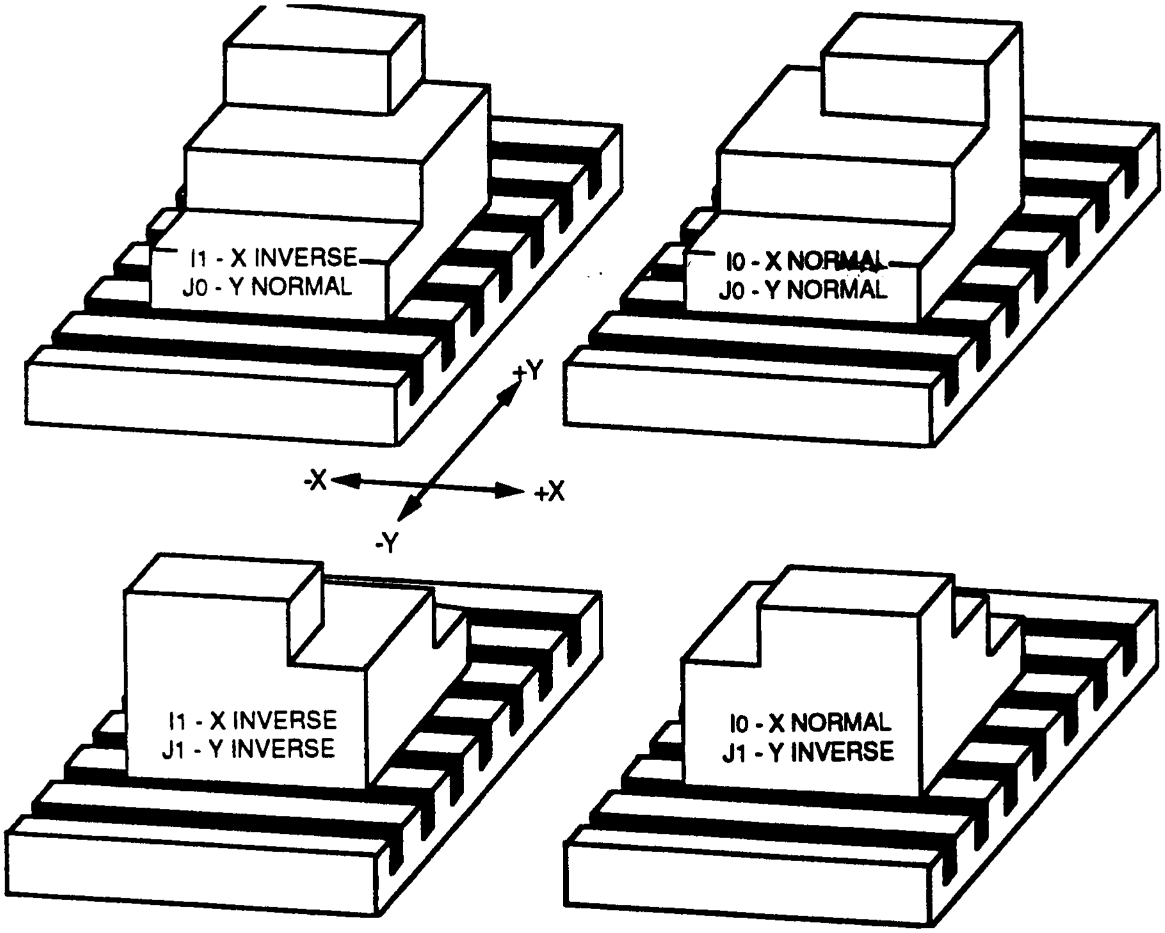
The lower illustration shows a 5 X 5 drill pattern which has been reproduced at three designated offsets. The part program executes a Subroutine event to recall the pattern after each Set Up event repositions the X0, Y0 point.



```
N210  (S)  A0, B0                  - cancel any previous part offset
N220  (0)  X0, Y0, Z.1, G0         - move to absolute X0, Y0
N230  (0)  X0/, Z-.5, G1, W.1, F10 - drill first hole
N240  (6)  X1/, F230, T4           - step 1" +X, drill hole, 4 times
N250  (6)  X.6/, F230, T4          - step .6" +Y, repeat 4 times
N260  (S)  A6, B1.2                - 1st part offset
N270  (7)  F220, L250              - recall sequence
N280  (S)  A12                     - 2nd part offset
N290  (7)  F220, L250              - recall sequence
N300  (S)  A18, B0                 - 3rd part offset
N310  (7)  F220, L250              - recall sequence
N320  (S)  A0, B0                  - return part offsets to default setting
```

I, J - AXIS SYMMETRY

Axis Symmetry allows you to cut the "mirror image" of a part. When this event commands a change in X or Y axis symmetry, the control will internally interpret dimensional signs for the selected axis (or axes) as being reversed. With I1 - X Axis symmetry, for example, an X4 command moves the tool to X-4, X-3/ moves the tool 3 units in the + X direction. To use axis symmetry, program I and/or J before the series of events that are to be mirrored. The illustrations, below, show a part which has been cut with different combinations of X and Y symmetry.



A second application of axis symmetry allows you to program a quarter or half section of a part and then recall it with a Subroutine event after this event activates a change in symmetry.

Manual jog and the Position Display registers in the Auto and Single Event modes will retain an unmirrored sign convention.

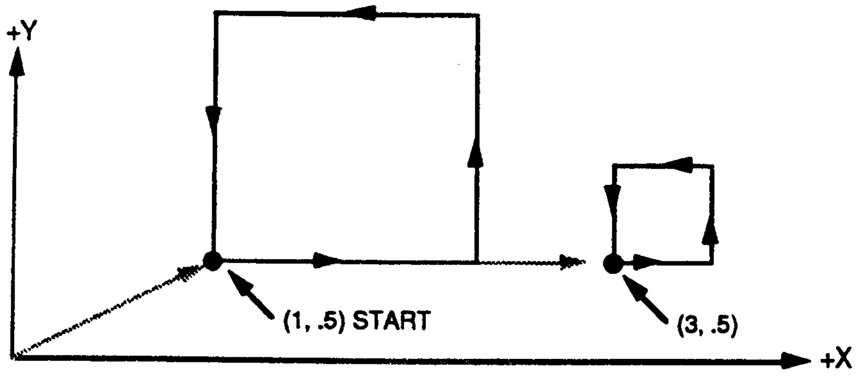
X, Z - SCALE FACTOR

X enters a scale factor that will operate on following X and Y axes commands. Z similarly programs a scale factor to be applied to following Z axis commands. You can program scale factors as absolute or incremental values. (See example **below**.)

Absolute entries become the active scale factor. Absolute entries may range from .0001 through 99.9999.

Incremental entries are algebraically added to the last scale factor. For this reason, incremental scale factors are programmed as signed values which may range from -99.9998 to +99.9998 as long as the resultant scale factor remains in the range .0001 to 99.9999. Incremental scaling is often used in conjunction with Subroutine or Repeat events to gradually enlarge or reduce a pattern of machining motions before each repetition.

ABSOLUTE AND INCREMENTAL SCALING



```
N320   (0)  X1, Y.5, Z.1                 - position to start of first square
N330   (S)  X.5/                         - set XY scale to 1.5
N340   (1)  Z-1, F5                      - plunge to Z-1 (Z move not affected by scale)
N350   (7)  F9001, L9005                 - call mill routine
N360   (0)  Z.1                          - withdraw in Z
N370   (S)  X-.5/                        - return to scale of 1
N380   (0)  X3                           - position to start of second square
N390   (S)  X.5                          - set XY scale to 0.5
N400   (1)  Z-1, F5                      - plunge to Z-1
N410   (7)  F9001, L9005                 - call mill routine
N420   (0)  Z.1                          - withdraw in Z
N430   (S)  X1                           - return to scale of 1
N600   (9)  M30                          - end program

N9001  (T)  Mill Routine                 - events called by N0350 and N0410
N9002  (1)  X1/, F10
N9003  (1)  Y1/
N9004  (1)  X-1/
N9005  (1)  Y-1/
```

SCALE INFLUENCE

Scaling causes the part coordinate system to expand or shrink about part program (X0, Y0, Z0) zero. **The tool does not move on execution of the scale factor.** However, all following part program endpoints and dimensions (see list below for exceptions) will be multiplied by the active scale factor.

The following program segment shows how the scale factor affects part program dimensions. The start point of the sequence (with a scale factor of X1, Z1) is X8, Y8, Z8.

	endpoint			
part program	**X**	**Y**	**Z**	**comments**
N20 (0) X8, Y8, Z8	X8	Y8	Z8	- position, scale is 1.0
N30 (S) X.5	X8	Y8	Z8	- set scale to .5, no motion
N40 (0) X4	X2	Y8	Z8	- X move is scaled (4 x .5 = 2)
N50 (0) Y4	X2	Y2	Z8	- Y move is scaled (4 x .5 = 2)
N60 (0) Z4	X2	Y2	Z4	- Z move is not scaled
N70 (0) X6/	X5	Y2	Z4	- X move is scaled (6/ x .5 = 3/)
N80 (1) X4, Y10/, Z2	X2	Y7	Z2	- (4 x .5 = 2), (10/ x .5 = 5/)

USING SCALING

Since scaling acts about part zero (0,0,0) it is often desirable to move the "scaling center" so that scaling may occur about another point. This can be accomplished by programming a G6 Position event, an G92 EIA/ISO event, or a part offset in the Set Up event prior to scaling the part program.

As an alternative, you may wish to program all following motions as incremental commands to avoid having to shift the part zero point. The Box Mill routine, **previous page**, shows how incremental commands react to part scaling.

If the tool is not located at 0,0,0 when the scale event executes, and you wish to scale around the coordinate 0,0,0, then you should program a Position event to locate the tool with respect to the scaled coordinated system. For example, in the chart above you should program (0) X8, Y8, Z8 before programming the scalable sequence.

SCALED AND NONSCALED ENTRIES

Scales are operative on the following entries:

XY	- axis end point, bolt center or center of rotation	D	- cavity end radius
		S, T	- cavity side dimensions
Z	- axis end point, G cycle depth	J, K	- cavity horizontal dimensions
W	- reference plane	L	- linear cavity length
R	- arc radius or cavity radius	I	- linear cavity increment
I, J, K	- arc center		

Dimensional entries for the **Frame Mill, Circle Mill, Cavity Mill, and EIA/ISO** events are also scaled.

Scales are not operative on the following entries:

L	- finish dimension or seconds of dwell	A, B, C	- angles
		H	- bolt circle, number of holes
K	- peck increment	F	- feedrate

Scales do not influence other parameters such as;

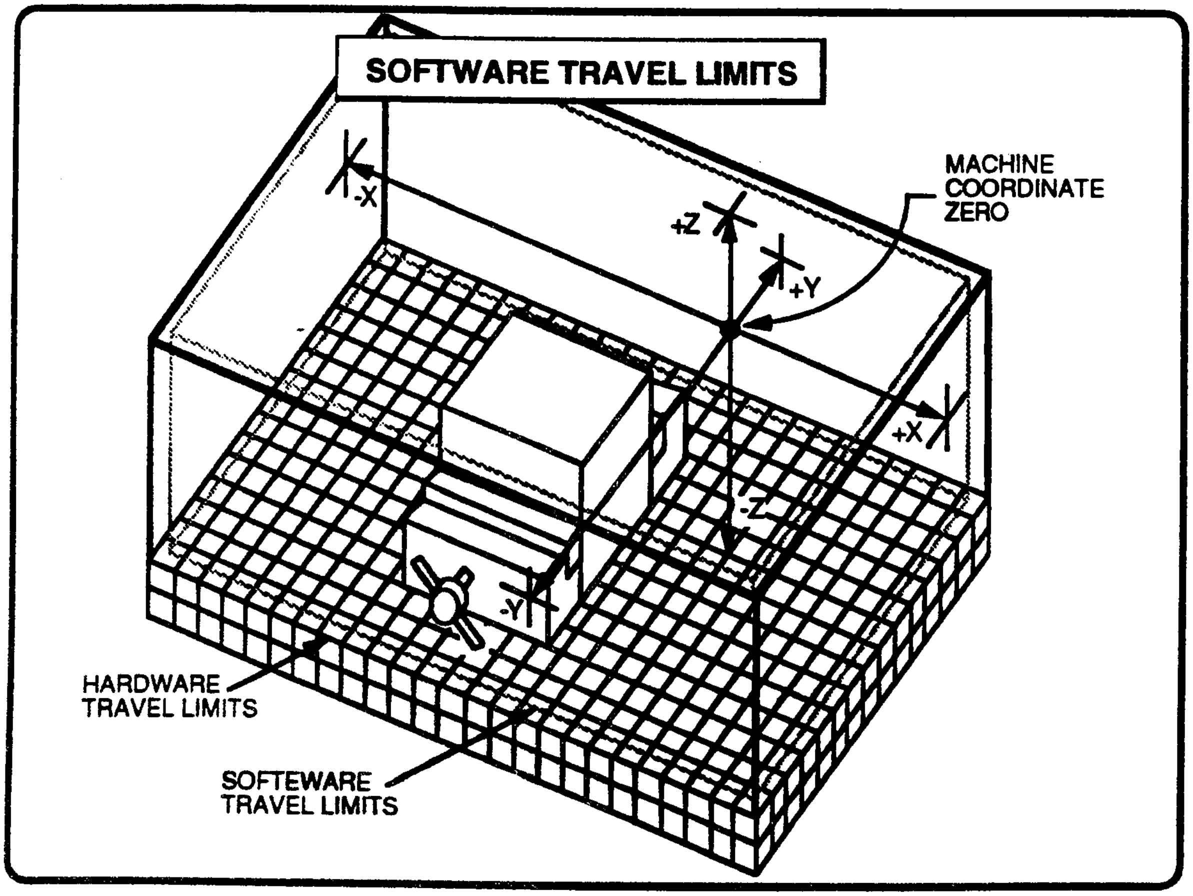
stored cutter diameter
stored tool length compensations
software travel limits
Axis Reversal Compensation
Part Offsets
Home or Tool Change Position

X and Z SCALE RESTRICTIONS

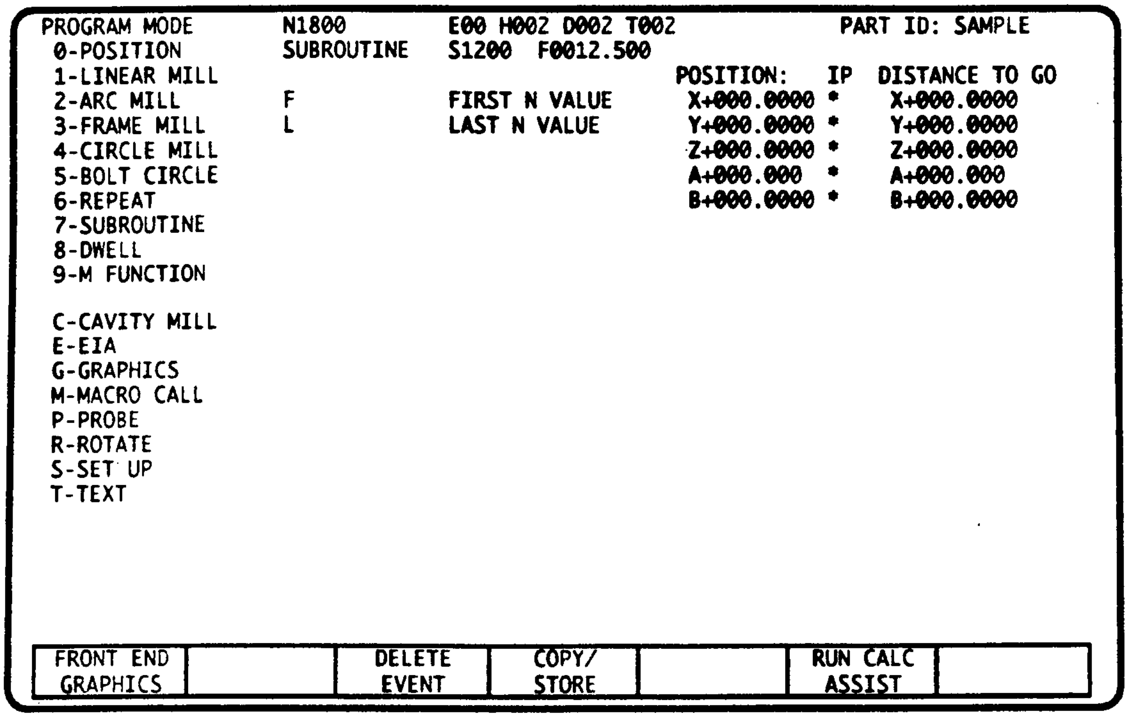
The X and Z scale factors must be equal if following arc motions will be programmed in the YZ or ZX planes.

F, G, T, W, M, L - TRAVEL LIMITS

These dimensions describe the limits of high (+) and low (-) axis travel within the work envelope. The control can monitor part program commands to halt axis travel outside of this envelope without removing power from the drives. Software travel limits are dimensioned from the machine coordinate zero as shown **below**. Programmed travel limits allow running changes in the definition of the work envelope. This feature can be used to protect expensive tooling or multiple fixtures by predetermination of safe zones for axis travel at various stages of the part program execution. The actual dimensions must be established by jogging an axis to the limit point **after** you reference the axes with a Reference Zero or Machine Zero operation and **before** a program Zero Set operation is performed. The control displays axes positions in machine coordinates prior to the Zero Set.



Note: Before executing a Set Up event that alters the software travel limits, make sure that you have a written record of the machine's default limits. You must restore the default limits before ending the program.



```
PROGRAM MODE        N1800         E00 H002 D002 T002           PART ID: SAMPLE
 0-POSITION         SUBROUTINE    S1200  F0012.500
 1-LINEAR MILL                                 POSITION:   IP  DISTANCE TO GO
 2-ARC MILL         F             FIRST N VALUE  X+000.0000 *    X+000.0000
 3-FRAME MILL       L             LAST N VALUE   Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL                                   Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE                                   A+000.000  *    A+000.000
 6-REPEAT                                        B+000.0000 *    B+000.0000
 7-SUBROUTINE
 8-DWELL
 9-M FUNCTION

 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```

SUBROUTINE EVENT - INTRODUCTION

The Subroutine event can be used as a tool for modular construction and testing of part programs. Use of this event can also reduce memory storage and programming effort for parts that require repetitive operations.

The Subroutine event programs First and Last event numbers representing an uninterrupted sequence of events. Execution of a Subroutine event causes the specified F-L series of events to be performed. Once performed, the program continues execution with the event that directly follows the Subroutine event. Refer to the program flow illustrations, **following page**.

SUBROUTINE EVENT DATA

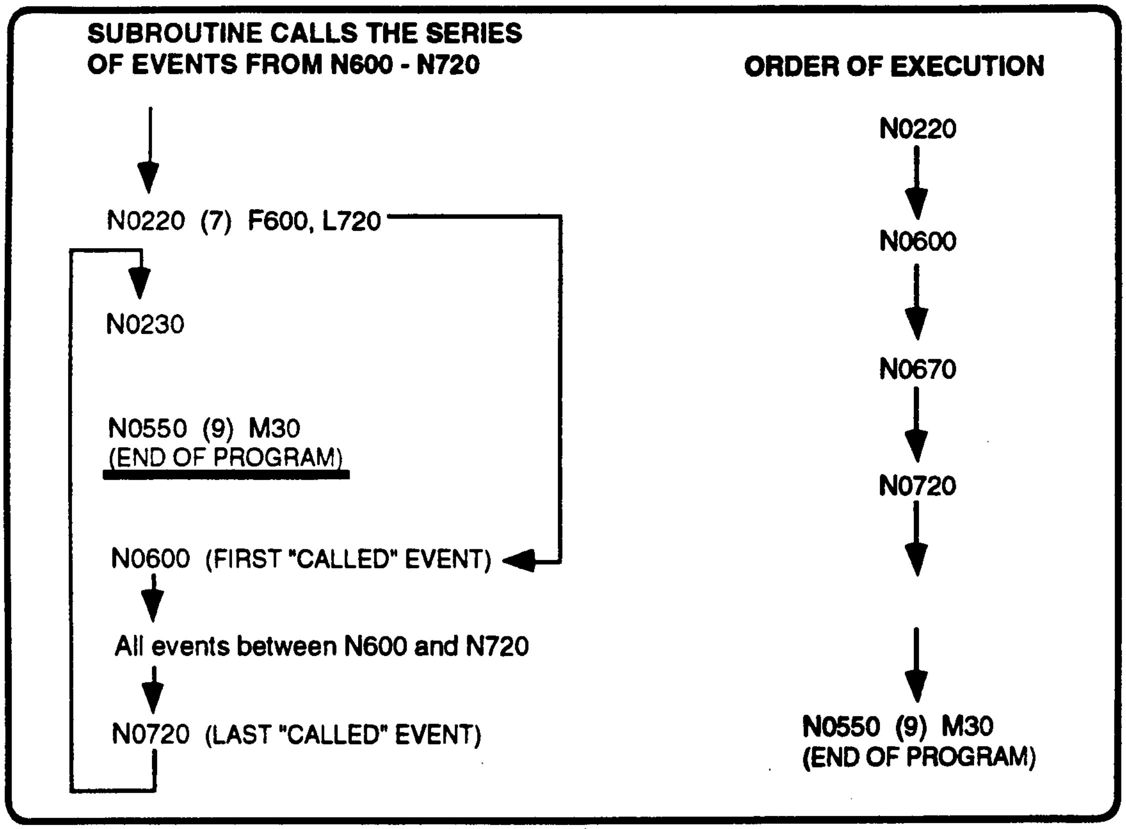
Parameter	Description	Entry
N	Sequence number	Required
F	First event to execute	Required
L	Last event to execute	Required

F - FIRST EVENT

F is the N number of the first event that will execute. This event may be located anywhere within the part program, even after an M02 or M30 - End of Program.

L - LAST EVENT

L is the sequence number of the last event that will execute. If L is greater than F, the events starting with F and continuing through L will execute. The L sequence number may not be smaller than F. If L equals F, only the single "called" event will execute.

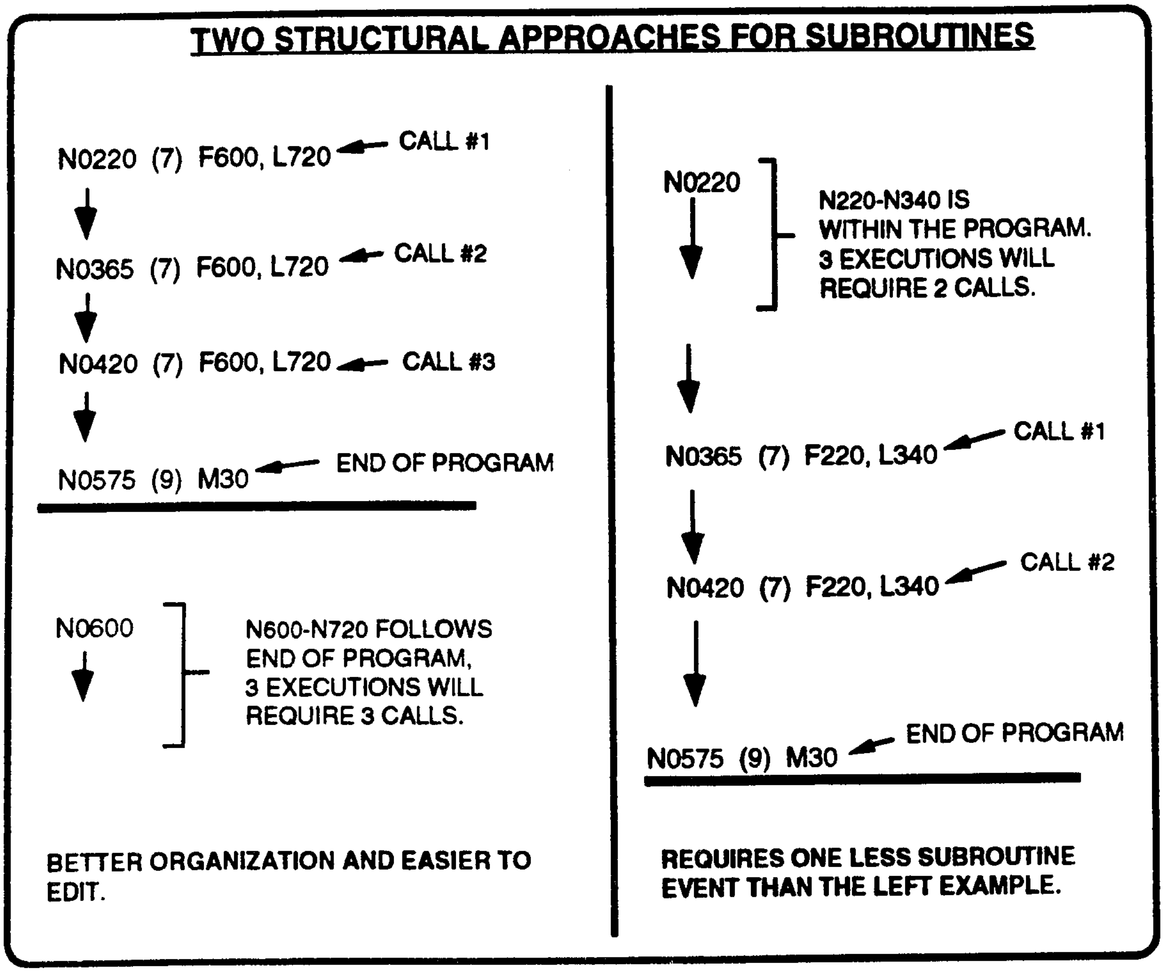

SUBROUTINE CALLS THE SERIES OF EVENTS FROM N600 - N720
ORDER OF EXECUTION
N0220 (7) F600, L720
N0230
N0550 (9) M30
(END OF PROGRAM)
N0600 (FIRST "CALLED" EVENT)
All events between N600 and N720
N0720 (LAST "CALLED" EVENT)
N0220
N0600
N0670
N0720
N0550 (9) M30
(END OF PROGRAM)

PROGRAMMING THE SUBROUTINE EVENT

The program sequence numbers designated by the Subroutine event's F and L entries must identify "stored" events. A series of events called by the Subroutine event may precede or follow the Subroutine event. A run time error will occur if the Subroutine event's N number does not fall within the programmed F - L range.

In typical part program arrangement, the series of events called by one or more Subroutine events appears after the M02 or M30 - End of Program command **(following page, left)**. Since the "called" series of events are outside of the part program's range, they will be performed only when "called", and not as a result of the normal sequential execution of the program.

In another type of arrangement, the "called" series of events is located at the first logical point of execution in the part program **(following page, right)**. Subsequent repetitions of the "series" are called by later Subroutine events. In comparison, one less Subroutine event is required by this approach at the possible expense of some part program clarity.

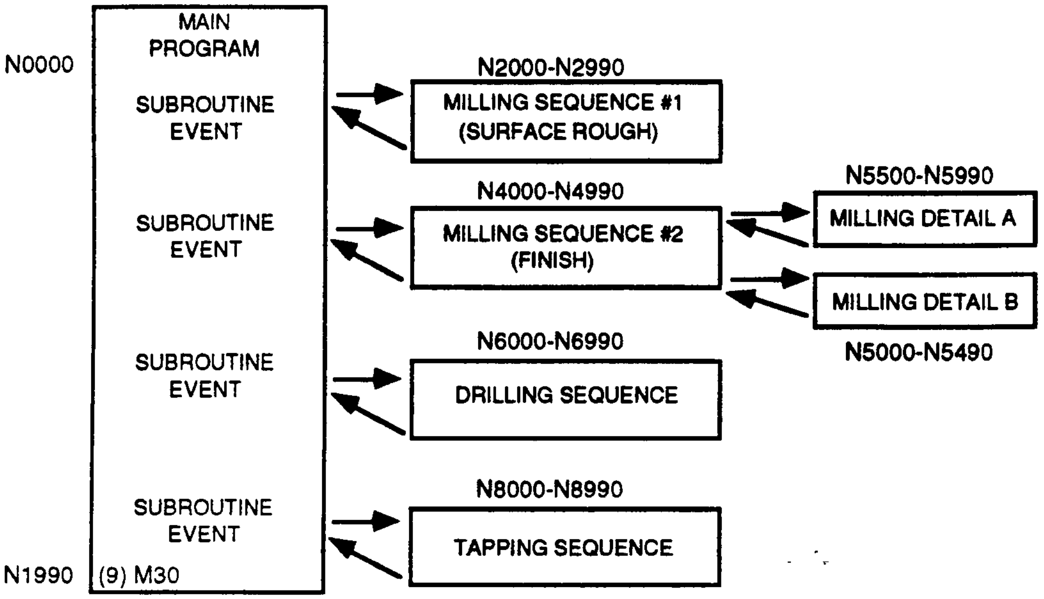

TWO STRUCTURAL APPROACHES FOR SUBROUTINES
N0220 (7) F600, L720
CALL #1
N0365 (7) F600, L720
CALL #2
N0420 (7) F600, L720
CALL #3
N0575 (9) M30
END OF PROGRAM
N0600
N600-N720 FOLLOWS END OF PROGRAM, 3 EXECUTIONS WILL REQUIRE 3 CALLS.
BETTER ORGANIZATION AND EASIER TO EDIT.
N0220
N220-N340 IS WITHIN THE PROGRAM. 3 EXECUTIONS WILL REQUIRE 2 CALLS.
N0365 (7) F220, L340
CALL #1
N0420 (7) F220, L340
CALL #2
N0575 (9) M30
END OF PROGRAM
REQUIRES ONE LESS SUBROUTINE EVENT THAN THE LEFT EXAMPLE.

ILLUSTRATIONS and PROGRAM EXAMPLES

A Subroutine event can considerably reduce programming effort and memory storage for repetitive sequences by initially programming the sequence end then addressing it as often as required via the Subroutine event. Often the called routine will be constructed as a series of incremental feeds fixed to a relocatable starting point.

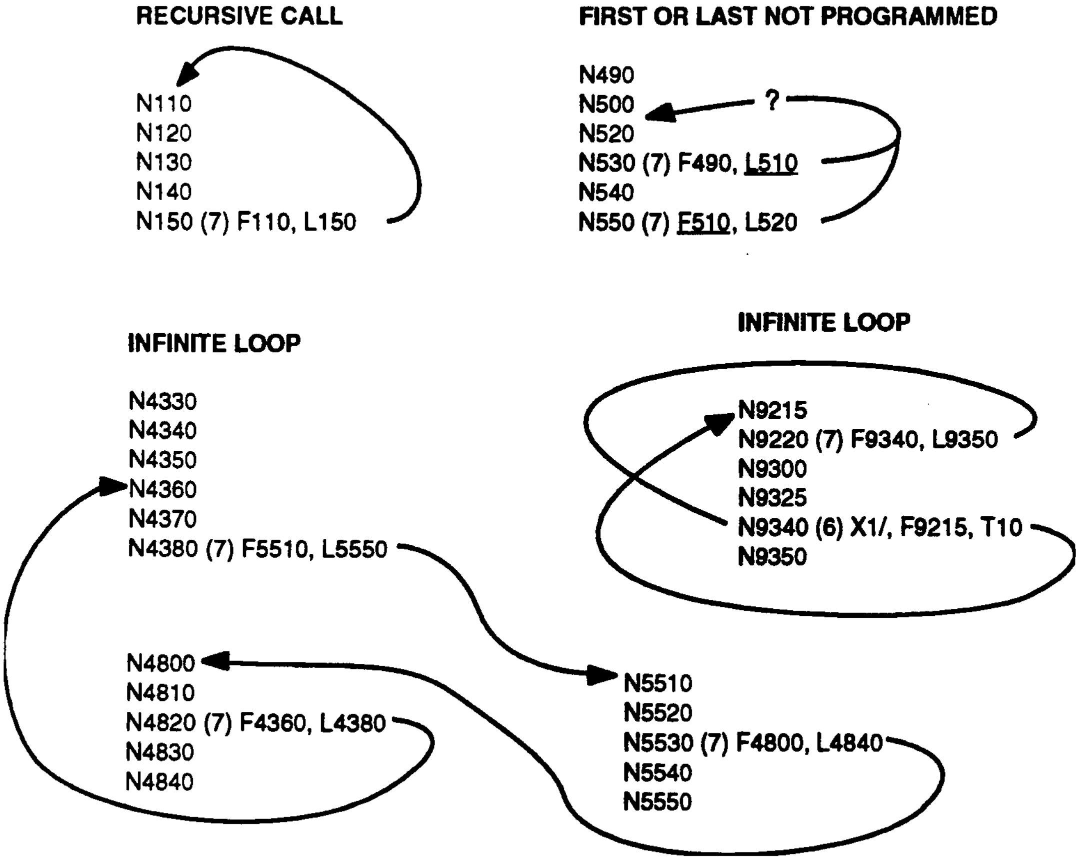
Subroutine events can also permit modular construction of a part program. This approach groups machining tasks into a series of modules which may be individually programmed and tested. The modules are later called in logical order by a series of Subroutine events within a main "calling" program. Modular construction adds clarity and organization to the part program while avoiding the tedium of moving and rearranging program data. Refer to the following illustration.

SIMPLIFIED LAYOUT FOR A SERIES OF OPERATIONS



SUBROUTINE ERRORS

Due to its nature, the Subroutine event is susceptible to several types of programming errors. The most common errors are illustrated **below** and will result in run time errors.



RECURSIVE CALL

```
N110
N120
N130
N140
N150 (7) F110, L150
```

FIRST OR LAST NOT PROGRAMMED

```
N490
N500          ?
N520
N530 (7) F490, L510
N540
N550 (7) F510, L520
```

INFINITE LOOP

```
N4330
N4340
N4350
N4360
N4370
N4380 (7) F5510, L5550

N4800
N4810
N4820 (7) F4360, L4380
N4830
N4840

N5510
N5520
N5530 (7) F4800, L4840
N5540
N5550
```

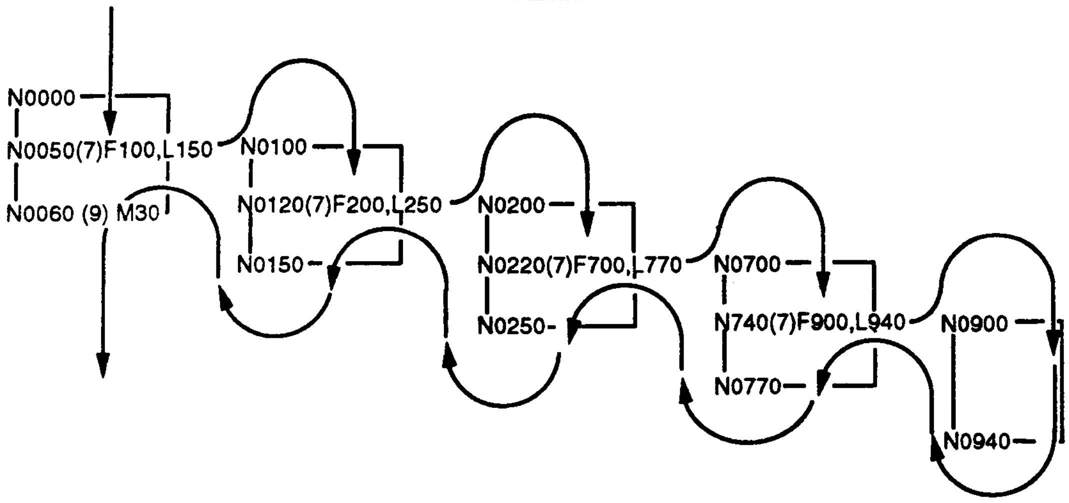
INFINITE LOOP

```
N9215
N9220 (7) F9340, L9350
N9300
N9325
N9340 (6) X1/, F9215, T10
N9350
```

NESTING

A series of events called by a Subroutine event may contain another Subroutine event. A Subroutine event contained in a called series of events is said to be "nested". When executed, the nested Subroutine event will delay execution to call a second series of events. The control can follow and retrace its path through as many as four calling Subroutine events, nested as shown below.

SUBROUTINE NESTING



*AS MANY AS FOUR RETURN ADDRESSES MAY BE CALLED.

SUBROUTINE and REPEAT EVENTS

Subroutine and Repeat events may be programmed in various combinations to perform extremely complex machining sequences. A Subroutine event may call one or more Repeat events, or a Repeat event may include one or more Subroutine events in its range of events. Nesting may continue as long as each event type does not exceed its nesting limit, and the construction avoids infinite loops.

As a practical consideration, the use of a Subroutine event is often preferred to the Repeat event. While it is simpler to repeat a single event 99 times than enter 99 Subroutine events to achieve the same effect, the Subroutine event is often able to avoid many of the Repeat event's restrictions.

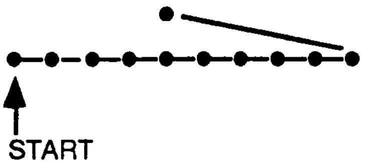
A Repeat event usually relies on preparatory activity such as an incremental step, rotation or scaling between each repeat sequence. Each preparatory activity, like the milling or drilling sequence that follows it, must mimic the last, and the series of repetitions must occur without interruption.

Some parameters may not allow multiple repetitions within the Repeat event's structure. Examples include programmed part offsets, X and Y symmetry, travel limits changes, inch/metric changes and nonincremental changes in scale or angle of rotation. The Subroutine event can save considerable programming effort and program storage by recalling a machining sequence after one of these types of preparatory activities.

EXAMPLES

The following part program segment demonstrates an interruption of a series with repetitions. One way to program the hole pattern would to be to treat the 10 hole linear array as a drill cycle with nine repeats, then position and drill the remaining hole, this example is shown, **lower left.**

REPEAT

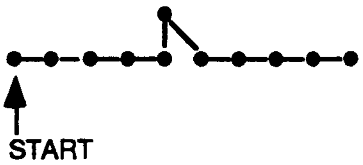


```
N 10 (0) X0/,G1,W.1,F5
N 20 (6) X5/,F10,T9
N 30 (0) X-25/,Y5/
```

ALTHOUGH PART PROGRAM IS SHORTER, EXECUTION TIME IS LONGER

60.61" XY TOOL MOTION
95.7 SECONDS EXECUTION

REPEAT AND SUBROUTINE



```
N 10 (0) X0/,G1,W.1,F5
N 20 (6) X5/,F10,T4
N 30 (0) Y5/
N 40 (0) X5/,Y-5/,G0
N 50 (7) F10,L20
```

NOW PART PROGRAM IS LONGER, BUT EXECUTION TIME IS SHORTENED

52.07 XY TOOL MOTION
91.7 SECONDS EXECUTION

The **upper right** example treats the first five holes as a drill and repeat, drills the odd hole, then positions and calls the earlier drill and repeat sequence using a Subroutine event. The two additional events resulted in a negligible increase in program storage while eliminating 8.54 inches of XY tool travel and 4 seconds of cycling time.[1]

[1] Notice that 5 holes are drilled when the Subroutine event at N50 calls events N10 and N20. Only 4 holes will be drilled if N50 programs F20 and L20.

WHEEL STRUT - EXAMPLE

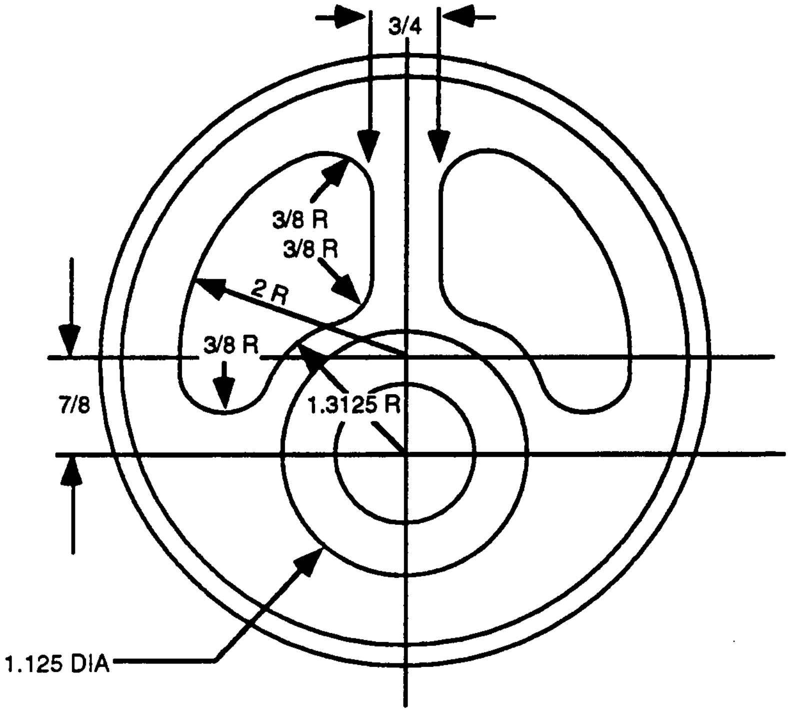
The following part program segment is used to mill the finish profile for a wheel's strut. The left recess is profiled with a series of linear and arc milling motions. The milling direction is counterclockwise with cutter radius compensation to the left. The plunge occurs midway along the linear path. Events N40 - N140 mill the left profile then return the cutter to the part's centerline (X0).

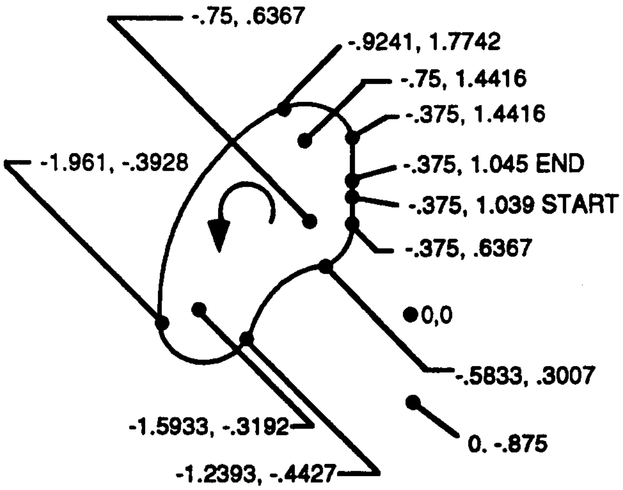
The right recess is then milled by enabling X axis symmetry, and then recalling the events N10 - N140 with a Subroutine event. Notice that cutter compensation is turned on and off within the range of events that is called by the Subroutine event. Also notice that X axis symmetry is turned off in the block (N180) the precedes the M30 End of Program command. Refer to the illustration on the **following page**.

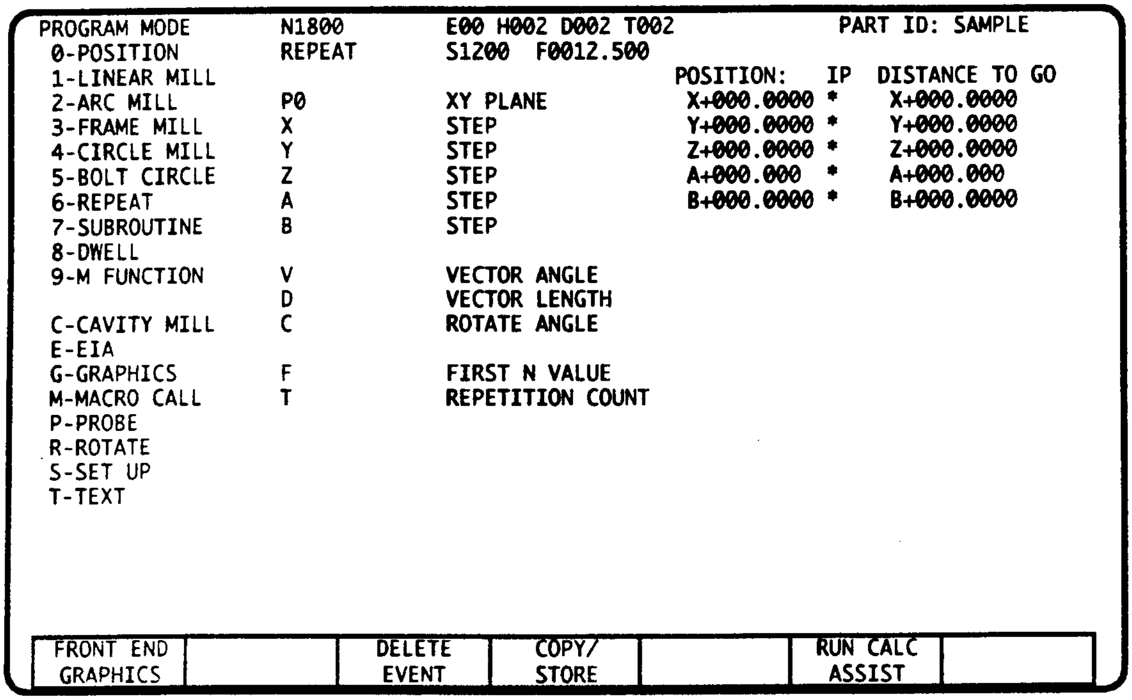
WHEEL STRUT PROGRAM

```
N00   (9)  M03 T1 H1 D1 S1000                     - tool 1, turn on spindle
N10   (0)  X-.375 Y.839 Z1                        - position to start point
N20   (1)  Y.939 F20 C0                           - dummy move to prepare cutter comp
N30   (1)  Y1.039 C1                              - comp on left. Tool is still above work[2]
N40   (1)  Z-1.625                                - feed to Z depth
N50   (1)  Y1.4416                                - begin profile with comp on left
N60   (2)  X-.9241 Y1.7742 I-.75 J1.4416 D1       -
N70   (2)  X-1.9610 Y-.3928 I0 J0 D1              -
N80   (2)  X-1.2393 Y-.4427  I-1.5933 J-.3192 D1  -
N90   (2)  X-.5833 Y.3007  I0 J-.875 D0           -
N100  (2)  X-.375 Y.6367  I-.75 J.6367 D1         -
N110  (1)  Y1.045                                 - end profile
N120  (1)  Z1                                     - retract in Z
N130  (1)  Y1.050                                 - continue linear move
N140  (1)  Y1.055 C0                              - turn cutter compensation off[2]
N150  (0)  X0                                     - position to X0
N160  (S)  I1                                     - turn on X symmetry
N170  (7)  F10 L140                               - Subroutine N10-N140
N180  (S)  I0                                     - turn X symmetry off
N190  (9)  M30                                    - end of program
```

[2] The Z infeed and retract moves do not cause the control to lose "look ahead" and gouge the part because they occur in the middle of linear moves. The infeed sequence begins at Y.939. N30 then makes a short Y axis move and turns cutter compensation on while the tool is above the work. N40 now plunges the tool to a depth of Z-1.625". To ensure that the tool does not gouge the part, the following move (N50) must continue to feed the tool in the same direction as N30. The retract sequence at N110-140 uses similar logic.


3/4
3/8 R
3/8 R
2 R
3/8 R
7/8
1.3125 R
1.125 DIA


-.75, .6367
-.9241, 1.7742
-.75, 1.4416
-.375, 1.4416
-1.961, -.3928
-.375, 1.045 END
-.375, 1.039 START
-.375, .6367
0,0
-.5833, .3007
0. -.875
-1.5933, -.3192
-1.2393, -.4427



```
PROGRAM MODE       N1800        E00 H002 D002 T002            PART ID: SAMPLE
 0-POSITION        REPEAT       S1200  F0012.500
 1-LINEAR MILL                                    POSITION:   IP  DISTANCE TO GO
 2-ARC MILL        P0           XY PLANE           X+000.0000 *    X+000.0000
 3-FRAME MILL      X            STEP               Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL     Y            STEP               Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE     Z            STEP               A+000.000  *    A+000.000
 6-REPEAT          A            STEP               B+000.0000 *    B+000.0000
 7-SUBROUTINE      B            STEP
 8-DWELL
 9-M FUNCTION      V            VECTOR ANGLE
                   D            VECTOR LENGTH
 C-CAVITY MILL     C            ROTATE ANGLE
 E-EIA
 G-GRAPHICS        F            FIRST N VALUE
 M-MACRO CALL      T            REPETITION COUNT
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```

REPEAT EVENT - INTRODUCTION

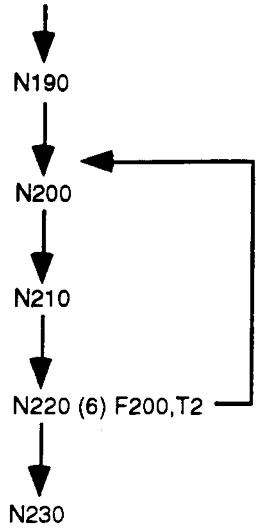
The Repeat event can reduce programming effort and memory storage by allowing multiple repetitions of a series of program events. The range of events to be repeated begins with a specified N number and is concluded by the Repeat event. This event may program as many as 999 repetitions.

If desired, you can program a preparatory position move, usually incremental, that will occur before each repetition. The step may be programmed with Cartesian or polar coordinates. Alternately you can program axis rotation that will occur before each repetition. A Rotate event must designate the center of rotation before any Repeat event axis rotation occurs.

REPEAT EVENT DATA

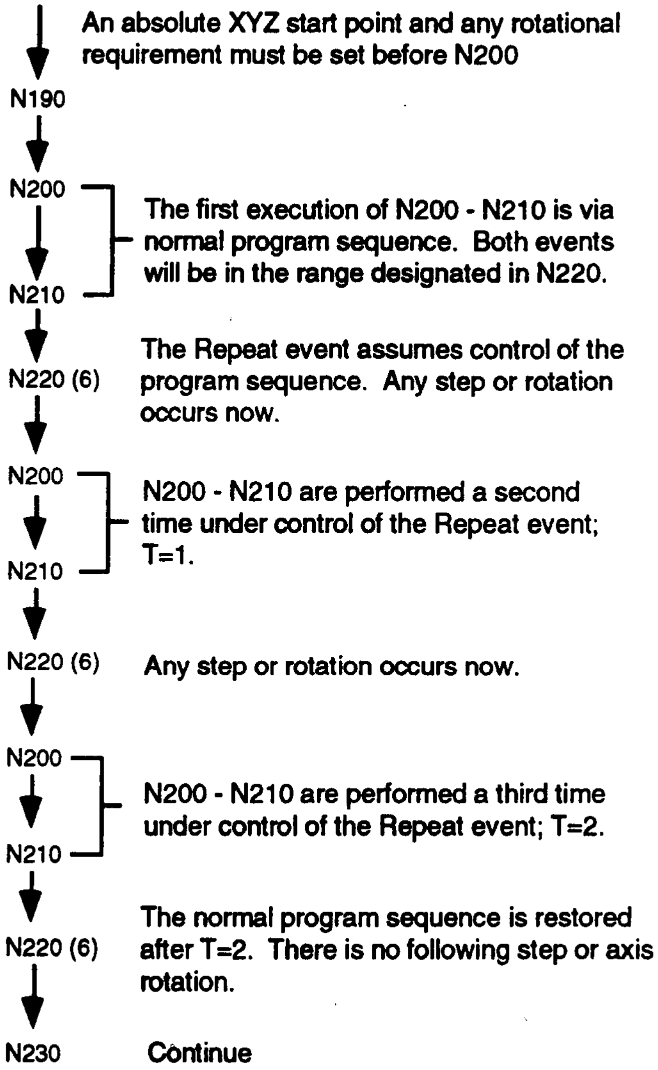
Parameter	Description	Entry
N	Sequence Number	Required
F	First Event to be repeated	Required
T	Number of Repetitions	Required
		Cartesian Step
X or X/	Preparatory Step, X	Optional (See Text)
Y or Y/	Preparatory Step, Y	Optional (See Text)
Z or Z/	Preparatory Step, Z	Optional (See Text)
		Polar Step
P	Plane Select (P0=XY, P1=YZ, P2=ZX)	Default to P0
V	Vector Angle, Preparatory Step	Optional
D	Distance, Preparatory Step	Optional
		Rotation
C	Preparatory axis Rotation	Optional (See Text)

REPEAT EVENT RECALLS N200-N210



The Repeat event may also program a step or axis rotation.

EXECUTION ORDER



An absolute XYZ start point and any rotational requirement must be set before N200

The first execution of N200 - N210 is via normal program sequence. Both events will be in the range designated in N220.

The Repeat event assumes control of the program sequence. Any step or rotation occurs now.

N200 - N210 are performed a second time under control of the Repeat event; T=1.

Any step or rotation occurs now.

N200 - N210 are performed a third time under control of the Repeat event; T=2.

The normal program sequence is restored after T=2. There is no following step or axis rotation.

Continue

F - FIRST EVENT TO BE REPEATED

F defines the first N Number in the repeat sequence. The last event in the repeat sequence is the Repeat event. Event F must precede the Repeat event in the part program. On execution the control will command any preparatory[1] step or rotation, then execute all events from F to the "calling" Repeat event. F may identify a range of one or more events immediately prior to the Repeat event. All Repeat events must complete this entry.

T - NUMBER OF REPETITIONS

T specifies the number of times that a range of events will be repeated. The range of events always executes once in the normal program sequence and then T times under control of the Repeat event for a total of T+1 executions. From 1 to 999 repetitions may be programmed. All Repeat events must complete this entry.

[1] Any axis motion commanded by the preparatory step will occur at the machine supplier's rapid rate.

PREPARATORY STEP

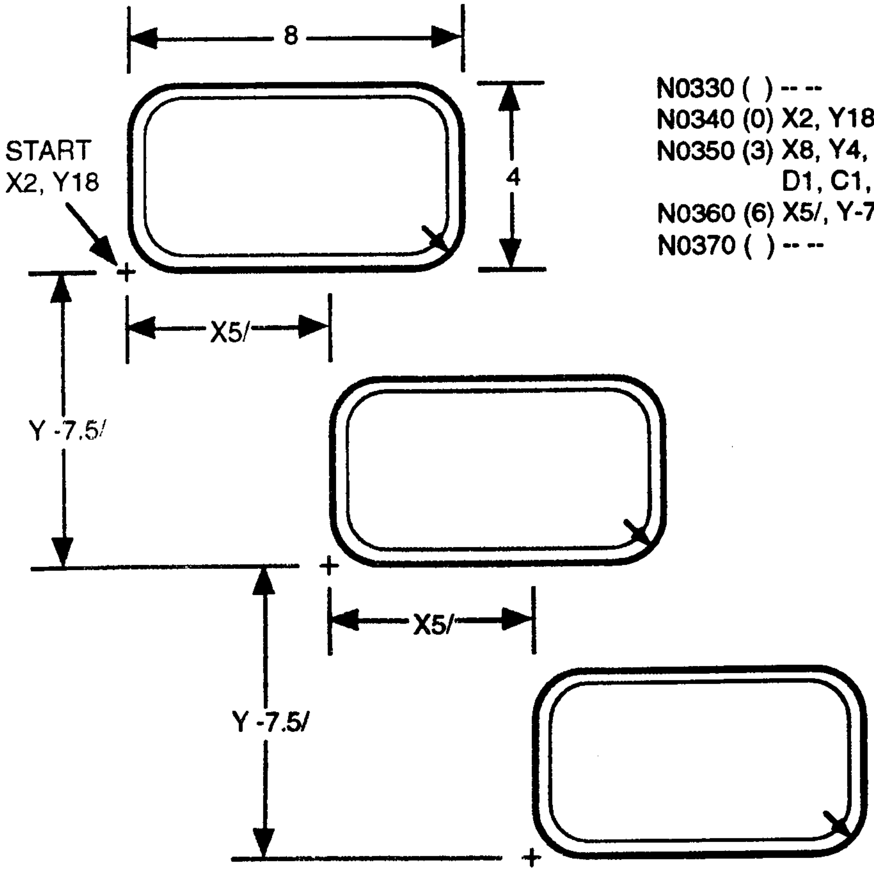
You can program a positioning step with Cartesian or polar coordinates, or axis rotation that will occur before each repetition. The preparatory step is commanded by the control as a position event with G cycles turned off. Refer to the Position event section for further information. Rotary increment C **may not** be programmed when a Cartesian or polar step has been entered.

All events within the Repeat event's F range should program relocatable incremental entries. An absolute XYZ start point should be programmed prior to the sequence.

X, Y, Z - CARTESIAN PREPARATORY STEP

Enter dimensions for each axis which must preposition before each repeat sequence begins. Signed incremental or signed absolute entries are permitted, but absolute entries are usually illogical when T exceeds one repetition. The example **below**, uses incremental Cartesian (XY) coordinates to step between three Frame Mill events. Recall that the tool's position at the start of the Frame Mill event determines the location of the frame's lower left corner.

REPEAT WITH CARTESIAN STEP



```
N0330 ( ) -- --
N0340 (0) X2, Y18, Z.1,
N0350 (3) X8, Y4, Z-1, R.75, W.1,
          D1, C1, G0, F10
N0360 (6) X5/, Y-7.5/, F350, T2
N0370 ( ) -- --
```

P, V, D - POLAR PREPARATORY STEP

P - PLANE SELECT - POLAR EVENT

Plane select establishes the plane of motion for a preparatory polar move. When not programmed, P defaults to P0, which programs motion in the XY plane. You must program P1 or P2 to program a polar move in the respective YZ or ZX plane.

This entry also specifies the plane to rotate when a preparatory axis rotation is programmed in this event.

V, D - PREPARATORY STEP - POLAR EVENT

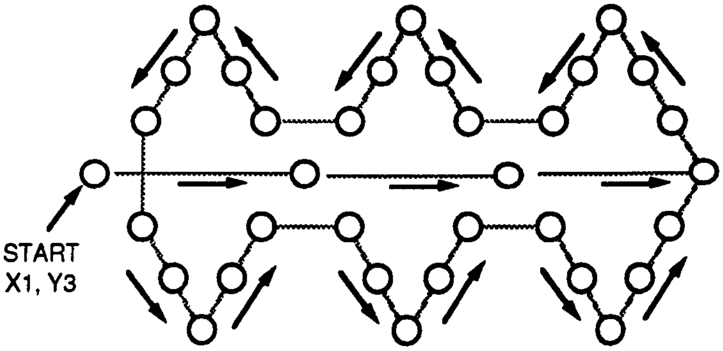
The tool moves distance D in the direction of angle V from its last position in the selected plane to complete the step before the F sequence is repeated.

Note: The letter "V" is programmed with the letter "J" if your control is equipped with an auxiliary axis that is programmed with the letter "V". Refer to Part 1-Section 2.

The example **below**, uses several polar steps between repetitions to program linear drill arrays in a diamond pattern.

REPEAT WITH POLAR STEP

```
N0030 ( ) -- --
N0040 (0) X1, Y3, Z.1, G0           - position to start point, G cycles off
N0050 (0) X0/, G1, W.1, F5          - drill a hole at current XY position
N0060 (6) V0, D3, F50, T3           - polar move and repeat N50 3 times
N0070 (0) V120, D1,G0               - position in XY, G cycles off
N0080 (0) X0/, G1                   - drill a hole at current XY position
N0090 (6) V120, D1, F80, T2         - polar move and repeat N80 2 times
N0100 (0) G0                        - turn off G cycles, hole already drilled
N0110 (0) X0/                       - if G cycle is not off, drill at present XY position
N0120 (0) G1                        - turn on drill G cycle, (no motion)
N0130 (6) V240, D1, F110, T2        - polar move and repeat N110 - N120 2 times
N0140 (6) V180,D1, F80, T1          - polar move and repeat N80 - N130 2 times
N0150 (0) V270, D1.732, G0          - position in XY, G cycles off
N0160 (0) X0/, G1                   - drill a hole at current XY position
N0170 (6) V300, D1, F160, T2        - polar move and repeat N160 2 times
N0180 (0) G0                        - Turn off G cycles, hole already drilled
N0190 (0) X0/                       - if G cycle is not off, drill at present XY position
N0200 (0) G1                        - Turn of drill cycle (no motion)
N0210 (6) V60, D1, F190, T2         - polar move and repeat  N190 - N200 2 times
N0220 (6) V0, D1, F160, T2          - polar move and repeat N160 - N210 2 times
N0230 ( ) -- --
```



ROTATE PREPARATORY STEP

When a Repeat event programs axis rotation in plane P, the range of events to be repeated will usually describe a machining sequence that is radially oriented about a predetermined center of rotation. The sequence will characteristically repeat itself after a uniform angular interval.

A Rotate event must initially designate a center of rotation for the machining sequence. This Rotate event will establish a C absolute orientation for an initial run through the sequence prior to execution of the Repeat event. This initial Rotate event must also **precede** the range of events to be repeated, otherwise it would reset rotation each time it is repeated. The Rotate event's C orientation will also serve as an initial reference for the Repeat event's C incremental entry. The Repeat event's C incremental entry makes an adjustment to the last rotation angle prior to each repetition.

The rotary increments are cumulative and the last axis orientation will remain active following the last repetition T of the Repeat event. A Rotate event **must program C0** to reset the normal axis orientation after the Repeat event.

C - ROTATE INCREMENT

C assigns the rotation angle before each repetition occurs. When Repeat event rotation occurs, the tool immediately rapids to its last assigned coordinates, but relative to the current axes orientation. C may be programmed as a signed incremental entry that is referenced from the last active axis orientation, or as a signed absolute entry, referenced from the zero degree reference line. An absolute entry is usually illogical when T exceeds one repetition.

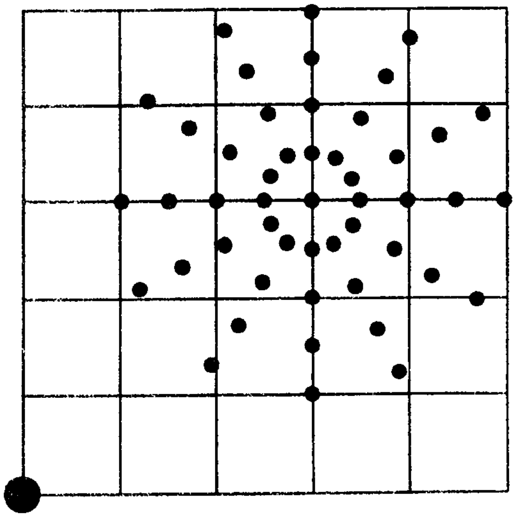
Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

Refer to the Rotate event for further discussion of axis rotation.

RADIAL DRILL PATTERN

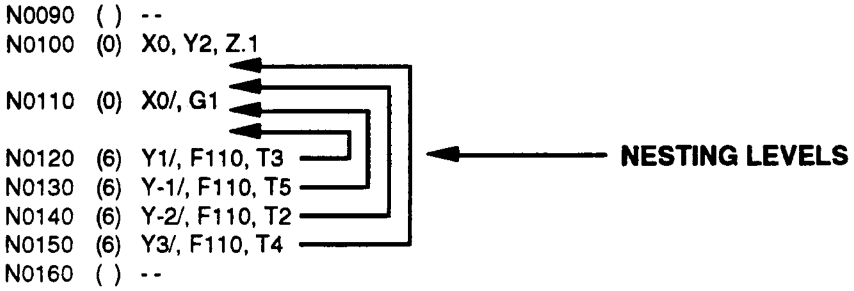
The example **below**, shows a radial drill pattern. N350 sets the initial rotation center to X3, Y4. N360-380 drills the 5 hole array. N390 rotates 30° incremental (about X3, Y4) and repeat the 5 hole array 11 times. The drill moves to the center of rotation prior to each repeat to avoid tool motion as a result of the rotation.

```
N330  ( )  - -
N340  (0)  X3, Y4, Z.1, G0
N350  (R)  X3, Y4, C0
N360  (0)  X0/, Z-1, G1, W.1, F5
N370  (6)  X.5/, F360, T4
N380  (0)  X3, G0
N390  (6)  C30/, F370, T11
N400  (R)  X3, Y4, C0
N410  ( )  - -
```



NESTING

4 levels of Repeat nesting is allowed. This means that 3 Repeat loops may be programmed inside of a main Repeat event. In the example **below**, Repeat event N0120 is nested (contained within) Repeat events N0130, N0140 and N0150. The sequence below will execute N110 360 times.

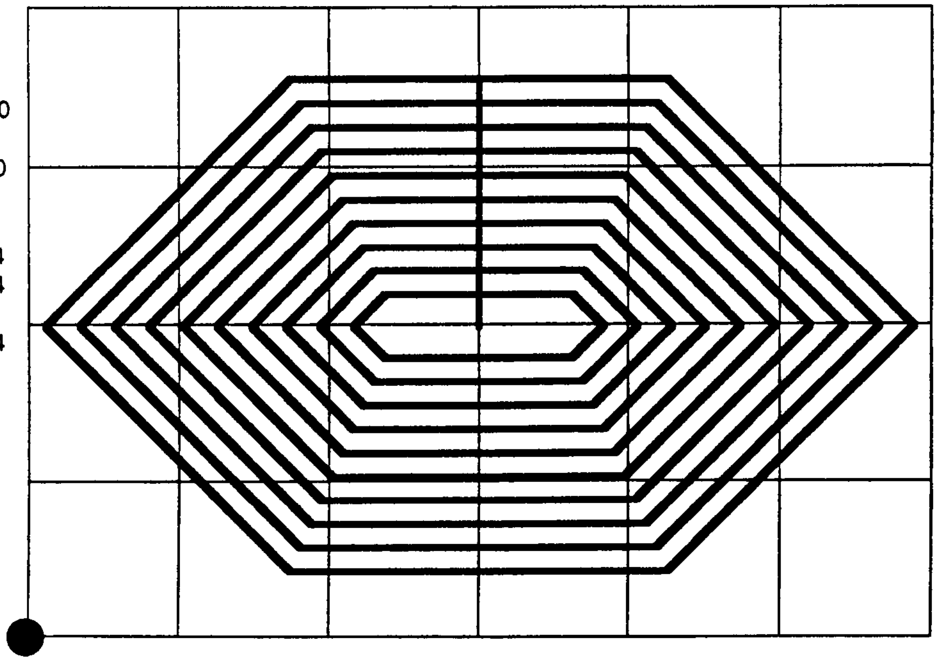


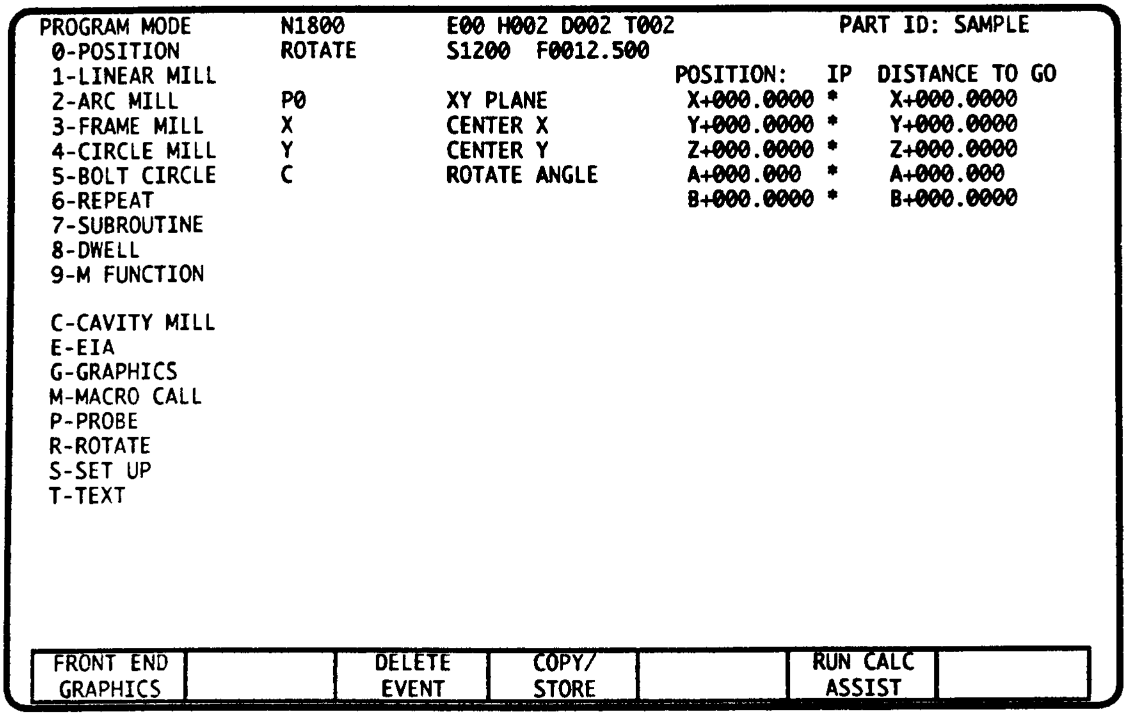
```
N0090  ( )  --
N0100  (0)  X0, Y2, Z.1

N0110  (0)  X0/, G1

N0120  (6)  Y1/, F110, T3
N0130  (6)  Y-1/, F110, T5
N0140  (6)  Y-2/, F110, T2
N0150  (6)  Y3/, F110, T4
N0160  ( )  --
```

HEXAGONAL POCKET

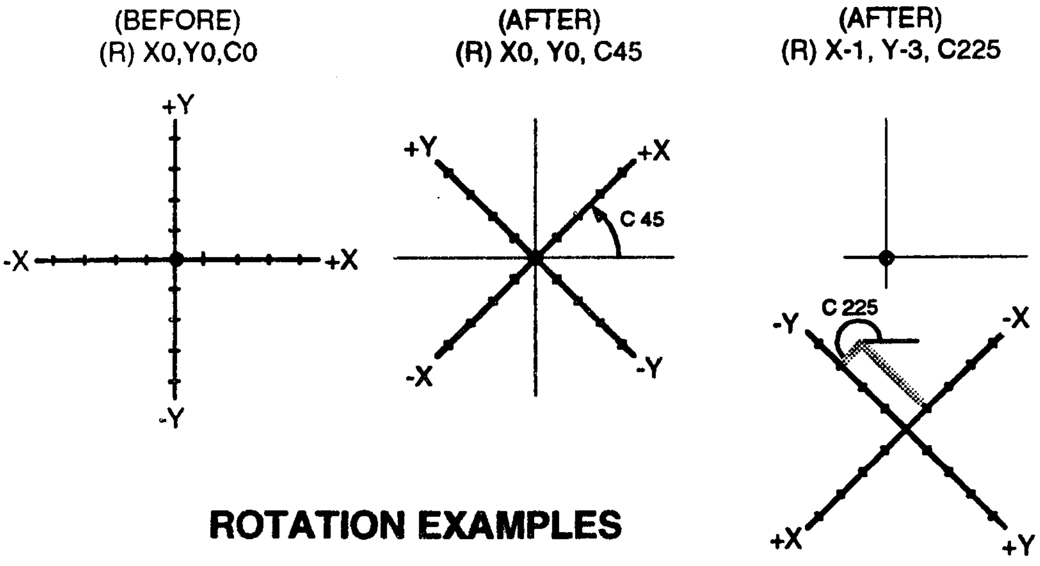
The example **below**, shows a hexagonal pocket created by an incremental scale and repeat. An XY scale factor of .1 is set, then incrementally enlarged until the last repeat pass is milled at the scale of 1.

```
N200  ( )  --
N220  (0)  X3, Y2, Z.1, G0
N230  (S)  X.1
N240  (1)  V90, D1.5, F10
N250  (1)  Z-1
N260  (1)  V0, D1.5
N270  (1)  V315, D2.1234
N280  (1)  V225, D2.1234
N290  (1)  V180, D3
N300  (1)  V135, D2.1234
N310  (1)  V45, D2.1234
N320  (1)  V0, D1.5
N330  (0)  Z.1
N340  (1)  V270, D1.5
N350  (S)  X.1/
N360  (6)  F240, T9
N370  (S)  X1
```





```
PROGRAM MODE       N1800        E00 H002 D002 T002             PART ID: SAMPLE
 0-POSITION        ROTATE       S1200  F0012.500
 1-LINEAR MILL                                   POSITION:    IP  DISTANCE TO GO
 2-ARC MILL        P0           XY PLANE          X+000.0000 *     X+000.0000
 3-FRAME MILL      X            CENTER X          Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL     Y            CENTER Y          Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE     C            ROTATE ANGLE      A+000.000  *     A+000.000
 6-REPEAT                                         B+000.0000 *     B+000.0000
 7-SUBROUTINE
 8-DWELL
 9-M FUNCTION

 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```



ROTATION EXAMPLES

ROTATE EVENT - INTRODUCTION

The Rotate event can reduce programming effort and memory storage by rotating the axes and programming non-orthogonal[1] part dimensions directly from a part blueprint. Complex radial patterns can be programmed by rotating the axes before each repetition. A Rotate event must be programmed to establish a rotation center before any Repeat event that programs a rotary step.

The Rotate event generates no motion itself, but following events in the part program will follow the axis orientation programmed in this event. Although its execution reorients programmed axis commands, the Position Display registers will track unrotated axis coordinates. Manual axis jogs are not affected by axis rotation. Rotation may be active in only one plane at a time.

The first part of this section describes rotation in the XY plane. Vertical (YZ or ZX) plane rotation is described at the end of this section.

ROTATE EVENT DATA

Parameter	Description	Entry
N	Sequence number	Required
P	Plane Select	Defaults to P0
X	Rotation Center, P0, P2	Modal (see text)
Y	Rotation Center, P0, P1	Modal (see text)
Z	Rotation Center, P1, P2	Modal (see text)
C	Rotation Angle	Modal (see text)

P - PLANE SELECT

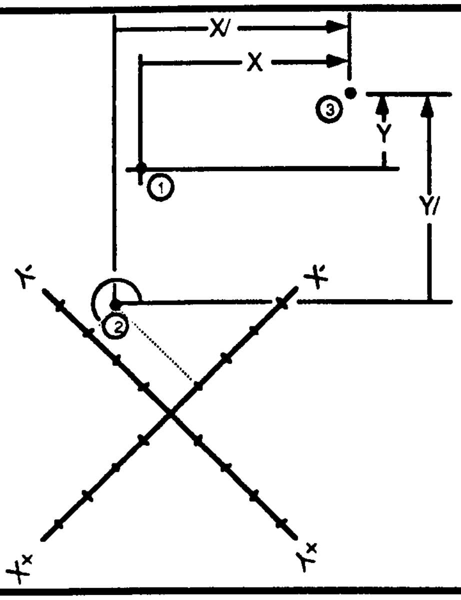
P selects the axes pair (XY, YZ, ZX) to rotate. The default is to P0, the XY plane. Rotation can be active in only one plane at a time. (If you rotate one plane, then choose to rotate a different plane, the prior plane's rotation, center and angle are canceled.) Changing the plane select within a part program is described later in this section.

X, Y, Z - ROTATION CENTER

The coordinate system is rotated about a point designated by the rotation center entries. This point is defined with a dimension for each axis in the selected plane P.

The rotation center is initialized at 0, 0, 0 when the control is first turned On. The control retains the rotation center independent of the Position Display registers. Rotation center coordinates may only be changed by execution of a Rotate event or by turning the control Off. Execution of the M02 or M30 End of Program or pressing RESET will reset the rotation angle to zero degrees (as described below), but does not alter the rotation center coordinates.
Rotation center coordinates are programmed as signed absolute entries which are dimensioned orthogonal to the unrotated coordinate system's part zero, or as signed incremental entries which are dimensioned orthogonal to the unrotated coordinate system from the last center of rotation. The illustration below shows absolute and incremental rotation centers in the P0 plane. Absolute and incremental entries may appear in the same event.

[1] Dimension that are parallel to the unrotated X, Y, or Z axis are "orthogonal". Dimensions that are not parallel to the unrotated axes are "non-orthogonal".



① X,Y PROGRAM ZERO (UNROTATED SYSTEM)

② CURRENT ROTATION CENTER

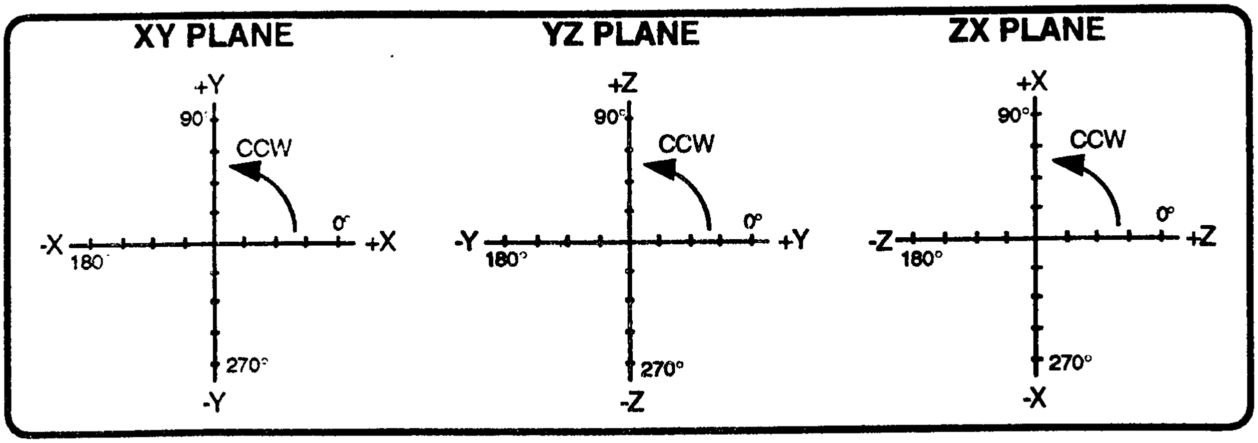
③ ROTATION CENTER FOR NEXT ROTATE EVENT

ABSOLUTE ROTATION CENTER	INCREMENTAL ROTATION CENTER
DIMENSION ORTHOGONAL TO UNROTATED COORDINATE SYSTEM	
DIMENSION FROM PART ZERO	DIMENSION FROM LAST PROGRAMMED CENTER

C - ROTATION ANGLE

The part coordinate system's angular orientation is expressed as degrees of rotation from zero degree reference. Zero degree reference is the unrotated +X, +Y and +Z axis in the respective XY, YZ and ZX plane. Positive angles define counterclockwise rotation, negative angles define clockwise rotation. The illustration **below** shows zero degree reference and the counterclockwise direction in each plane.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 2.

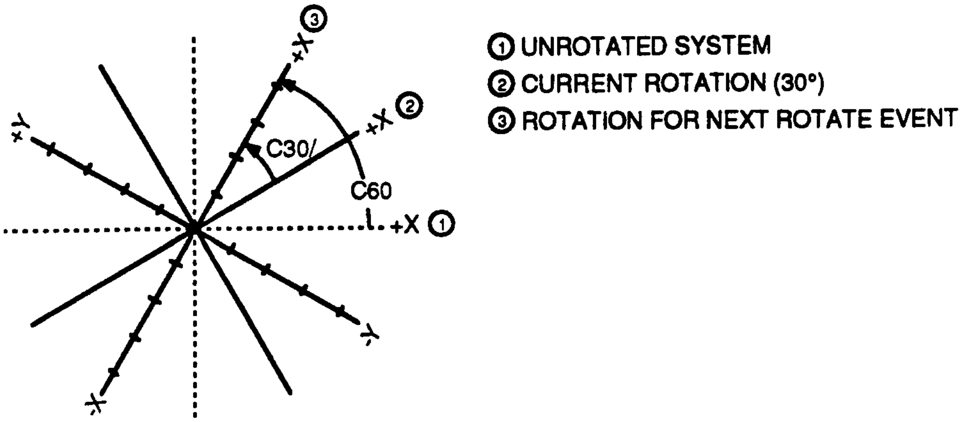


C enters the coordinate system's angular orientation about the rotation center. C is reset to zero degrees when the control is first turned On, by an M02 or M30 command, or by pressing RESET when the program is halted.

C may be programmed as an absolute or incremental entry. Absolute C is measured from zero degrees reference in the selected plane. The absolute angle you enter becomes the active rotation angle.

Incremental angles are measured from the active rotation angle. The rotation angle becomes the sum of the active rotation and the incremental value you program. The illustration **below** shows absolute and incremental angles in the P0 plane.

ABSOLUTE AND INCREMENTAL ROTATION ANGLES



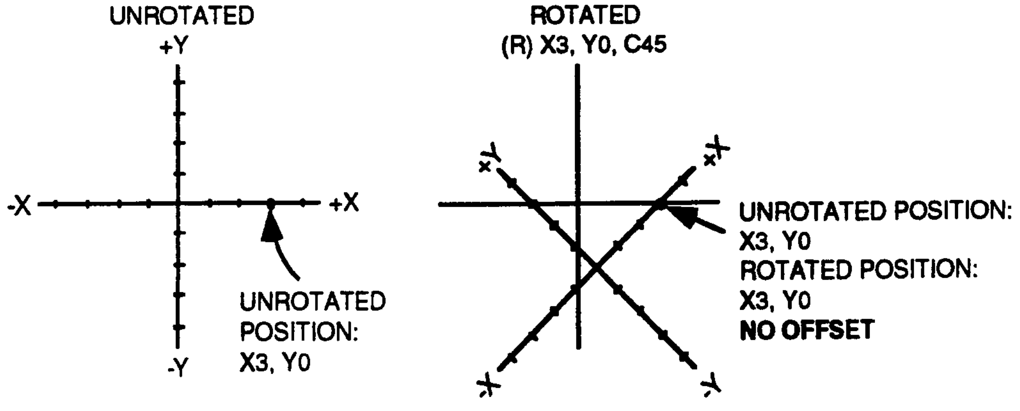
ROTATION AND TOOL CENTER POSITION

Executing a Rotate event alters axes orientation, but does not cause any tool motion. When you change the axes orientation, your next motion command will establish the tool's position in the new coordinate system. The effect of the following motion command is determined by the tool center position in relation to the programmed rotation center.

TOOL POSITION AT ROTATION CENTER

The rotation center is the only point with the same location in the unrotated and rotated coordinate systems. Therefore, positioning the tool to the rotation center, then rotating the axes leaves no offset between command position and the actual tool position. Refer to the illustration **below**. The tool position prior to the Rotate event is X3, Y0. The following event rotates the XY axes 45° about X3, Y0. Note that the tool is at the same position (X3, Y0) after the coordinate system has rotated.

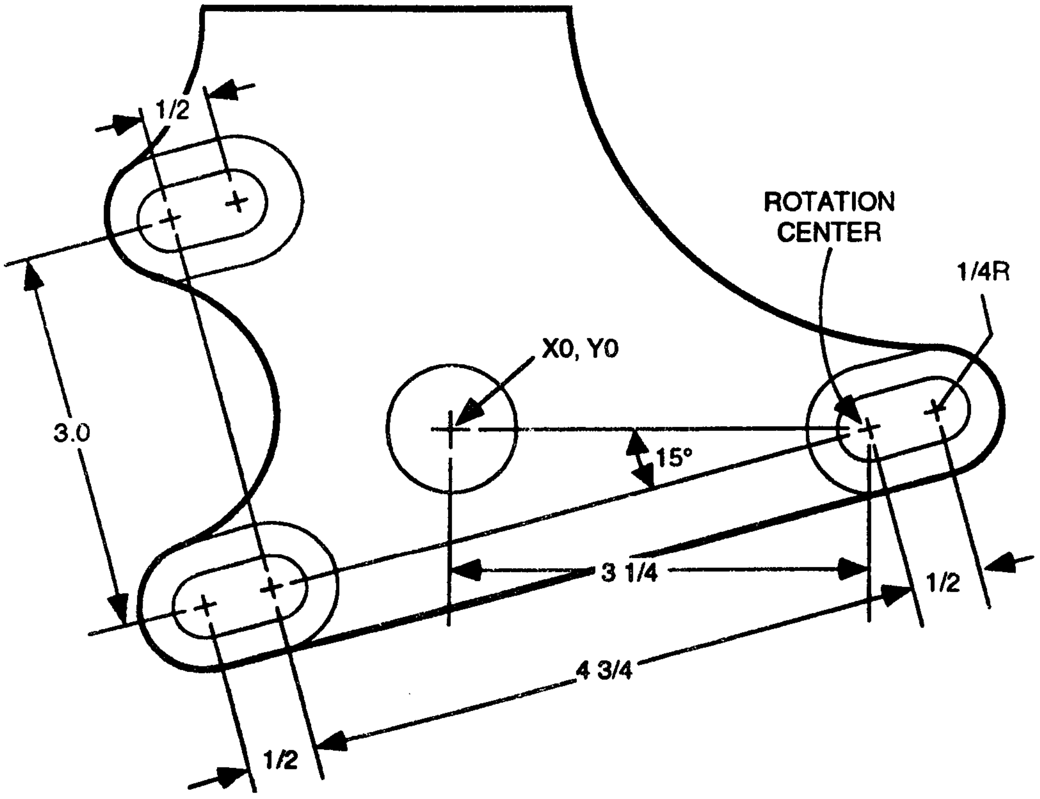
TOOL LOCATED AT ROTATION CENTER



EXAMPLE - TOOL POSITIONED AT ROTATION CENTER

The part program and illustration **below,** shows how the Rotate event can be used to program non-orthogonal dimensions. The sample part contains three 1/2 inch slots which are dimensioned 15° from zero degrees reference. The part program positions the tool over the first slot (X3.25, Y0) and rotates the axes 15° about this point. With the coordinate system rotated, all slot dimensions can be read directly from the part print. After the three slots are milled, the program returns the tool to the rotation center and rotates the axis back to 0°. Extraneous dimensions have been removed to draw attention to the rotated dimensions (slots to be milled), and their relationship to X0, Y0 and the Rotate event's rotation center.

```
N410  (0)  X3.25, Y0           - move to rotation center
N420  (R)  X3.25, Y0, C15      - rotate 15°
N430  (1)  Z-1.75, F5          - plunge
N440  (1)  X.5/, F10           - mill first slot
N450  (1)  Z.1                 - retreat to clearance
N460  (0)  X-5.75/             - position
N470  (7)  F430, L450          - mill second slot
N480  (0)  Y3/                 - position
N490  (7)  F430, L450          - mill second slot
N500  (0)  X4.25/, Y-3/        - return to rotation center
N510  (R)  X3.25, Y0, C0       - reset rotation angle to 0°
```

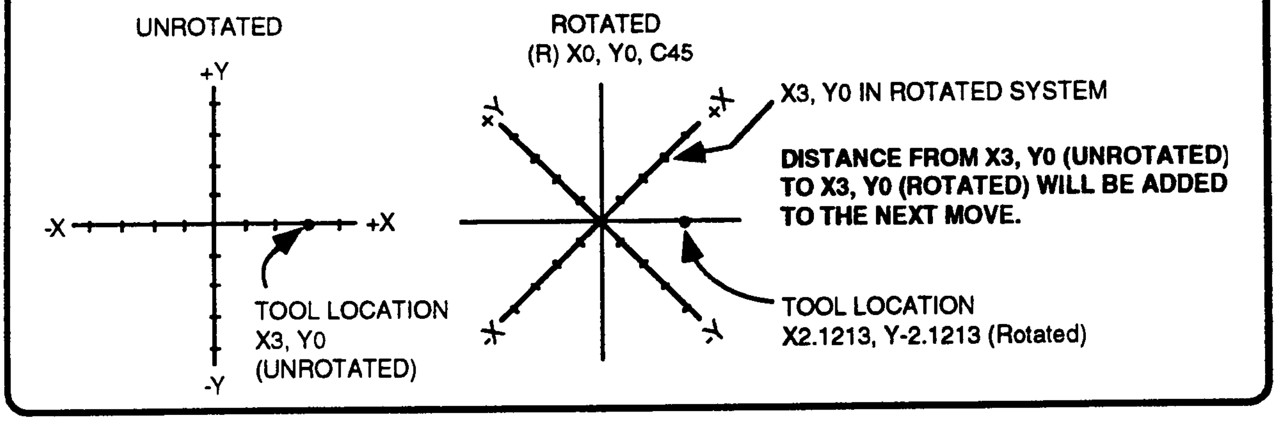


TOOL POSITION NOT AT ROTATION CENTER

Rotating the coordinate system with the tool center not located at the rotation center requires the ability to predict the response to a motion command immediately after a Rotate event. This is because the Rotate event causes an offset equal to the distance that the tool's start coordinate moves. The control will resolve this offset when the next motion command executes.

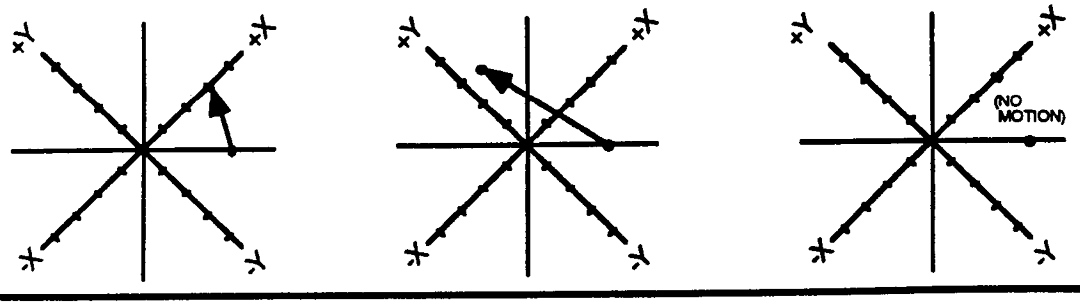
The illustrations and example **below** shows how the Rotate event causes an offset. The tool starts at X3, Y0. A Rotate event then rotates the coordinate system 45° about the point 0,0. This causes an offset equal to the the distance between 3,0 in the unrotated system, and 3,0 in the rotated system.

TOOL NOT LOCATED AT ROTATION CENTER


UNROTATED
+Y
-X
+X
-Y
TOOL LOCATION
X3, Y0
(UNROTATED)
ROTATED
(R) X0, Y0, C45
X3, Y0 IN ROTATED SYSTEM
DISTANCE FROM X3, Y0 (UNROTATED) TO X3, Y0 (ROTATED) WILL BE ADDED TO THE NEXT MOVE.
TOOL LOCATION
X2.1213, Y-2.1213 (Rotated)

The motion command that follows a Rotate event can be dimensioned as absolute or incremental. The illustrations **below**, show three motion commands which might follow the Rotate event shown **above**. In the three cases, note that incremental moves are dimensioned from the control's last command position (X3, Y0 in our example) and absolute moves are dimensioned from the rotated Part Zero.

RETURN TOOL TO ITS LAST POSITION, BUT IN THE ROTATED SYSTEM	MOVE TOOL TO A NEW POSITION, BUT IN THE ROTATED SYSTEM	KEEP TOOL IN ITS SAME POSITION, BUT IN THE ROTATED SYSTEM
X0/, Y0/ OR X3, Y0	X-2/, Y3/ OR X1, Y3	X2.1213, Y-2.1213 OR X-.8787/, Y-2.1213/

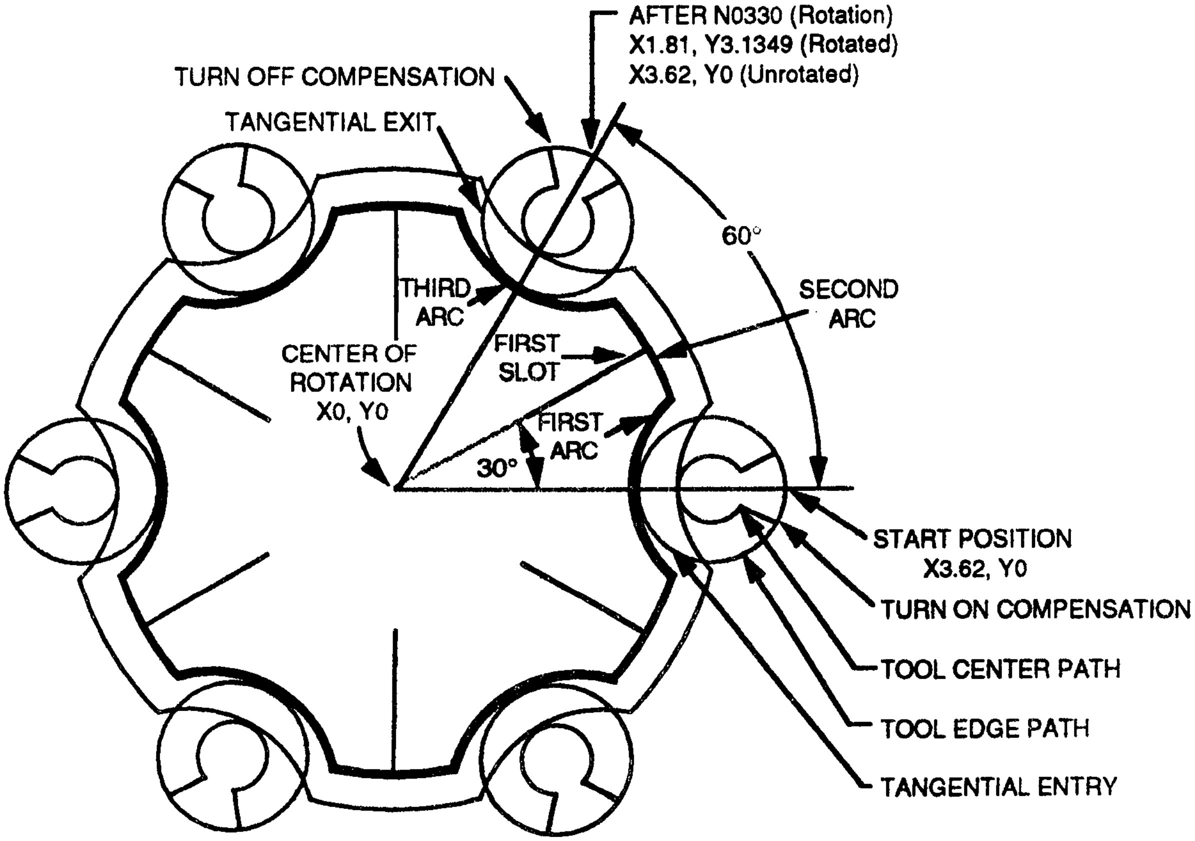


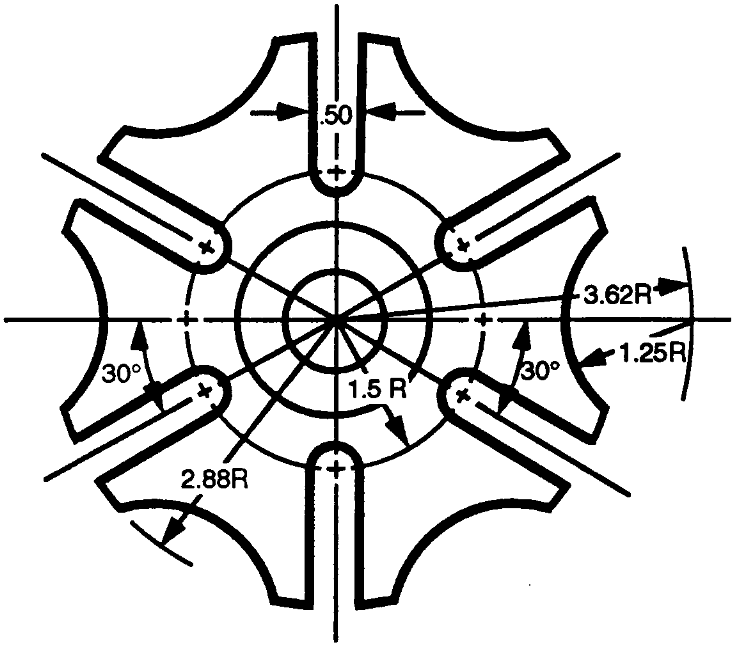
EXAMPLE - TOOL NOT LOCATED AT ROTATION CENTER

The following program segment illustrates the programming of axes rotation when the tool is not located at the rotation center. The sample program is divided into two sections. Events N220 - N340 mill the profile. Events N350 - N420 cut the slots to complete the Geneva cam. The entire part will be milled with a .5 inch diameter cutter. Both the profile and the slots can be cut by programming a 60° segment of the part, then rotating the axes 60° and repeating the 60° segment 5 times.

To start the profile, the tool is programmed to X3.62, Y0. Next, the tool tangentially enters the workpiece while turning on cutter compensation at events N250 and N260. Events N270 through N310 cut the 60° profile and turn off cutter compensation. The tool's position at the end of the first 60° segment is X1.81, Y3.1349. Now the coordinate system is rotated 60°. In the rotated system, the tool is located at the point X3.62, Y0, the start point of the sequence. N230, the first event the Repeat event calls, **does not cause tool motion**, but cancels the offset between the command position (X1.81, Y3.1349) and rotated position (X3.62, Y0). This entire sequence is repeated 5 times.

Events N360 - N420 then cut the 6 slots shown in the **upper** illustration. Events N370 - N380 position the tool to depth, outside of the first slot. Events N390 - N410 mill a slot and return the tool's start position. N420 incrementally rotates the axes 60° and repeats N390-N410 five times. Repeat event rotation causes the tool to rapid in a straight line to its programmed position in the new orientation. There is no position offset when a Repeat event commands axes rotation.





```
N220  (T)  GENEVA CAM                         -
N230  (0)  X3.62, Y0, Z.1                     - start 60° segment
N240  (1)  Z-2, F5                            - plunge in Z
N250  (2)  A0, B330, R.625, F7.5, C0          - compensation turn on
N260  (2)  A330, B180, R.625, C2              - enter profile next event
N270  (2)  A180, B134.76, R1.25               - mill first arc
N280  (2)  A17.95, B42.05, R2.88, D1          - mill second arc
N290  (2)  A285.24, B240, R1.25               - mill third arc
N300  (2)  A240, B90, R.625                   - exit profile turn off
N310  (2)  A90, B60, R.625, C0                - comp, end 60° segment
N320  (1)  Z.1                                - retreat in Z
N330  (R)  P0, X0, Y0, C60/                   - incremental rotate
N340  (6)  F230, T5                           - repeat N230-330 5 times

N345  (T)  MILL SLOTS                         - Text event
N350  (R)  P0, X0, Y0, C0                     - reset rotation to 0°
N360  (0)  X0, Y0, Z1                         - position for next polar
N370  (0)  V30, D3.14                         - move to start of 30° slot
N380  (1)  Z-2, F5                            - plunge in Z
N390  (1)  V210, D1.64, F7.5, C0              - mill slot
N400  (1)  Z.1                                - retreat in Z
N410  (0)  V30, D1.64                         - return to start point
N420  (6)  C60/, F380, T5                     - incr. rotate repeat N380-410 5 times
```

ROTATION AND CUTTER RADIUS COMPENSATION

Axis rotation impacts the use of cutter radius compensation. The P0 (XY plane) must be selected when milling with cutter compensation in a rotated coordinate system. Compensation must not be active while executing the Rotate event. If it is desirable to use cutter compensation, turn compensation off, rotate the coordinate system, then turn cutter compensation on. Turn compensation off before you execute another Rotate event.

If you program axis rotation in a vertical (YZ, or ZX) plane, you must program the tool center path. You **cannot** use cutter compensation.

ROTATION IN A VERTICAL PLANE

Rotation in a vertical (YZ, or ZX) plane can greatly assist in programming cavities with angled walls, three dimensional arcs, and other custom features. To program a complex cavity, you must complete the following steps.

1) Locate the cavity's rotation plane. The cavity's centerline must be parallel to the X or Y axis. If the cavity is parallel to the X axis, YZ is the rotation plane; if the cavity is parallel to the Y axis, ZX is the rotation plane. Also locate the cavity's start point. The start point must be located on the cavity's centerline and will be the rotation center for all following Rotate events.

2) Program a Rotate event that sets the rotation plane (YZ or ZX), the rotation center to the start point coordinates, and the initial rotation angle to 0°.

3) Program the center of the ball nose tool to move to the cavity's start point. This move may cause the tool to feed into the work.

4) Program a tool center path that defines one half of the cavity's XY profile (the tool should remain at the same Z depth). The endpoint of this sequence must have the same coordinates in the rotation plane as the rotation center.

5) Program a Rotate event with an incremental rotate command.

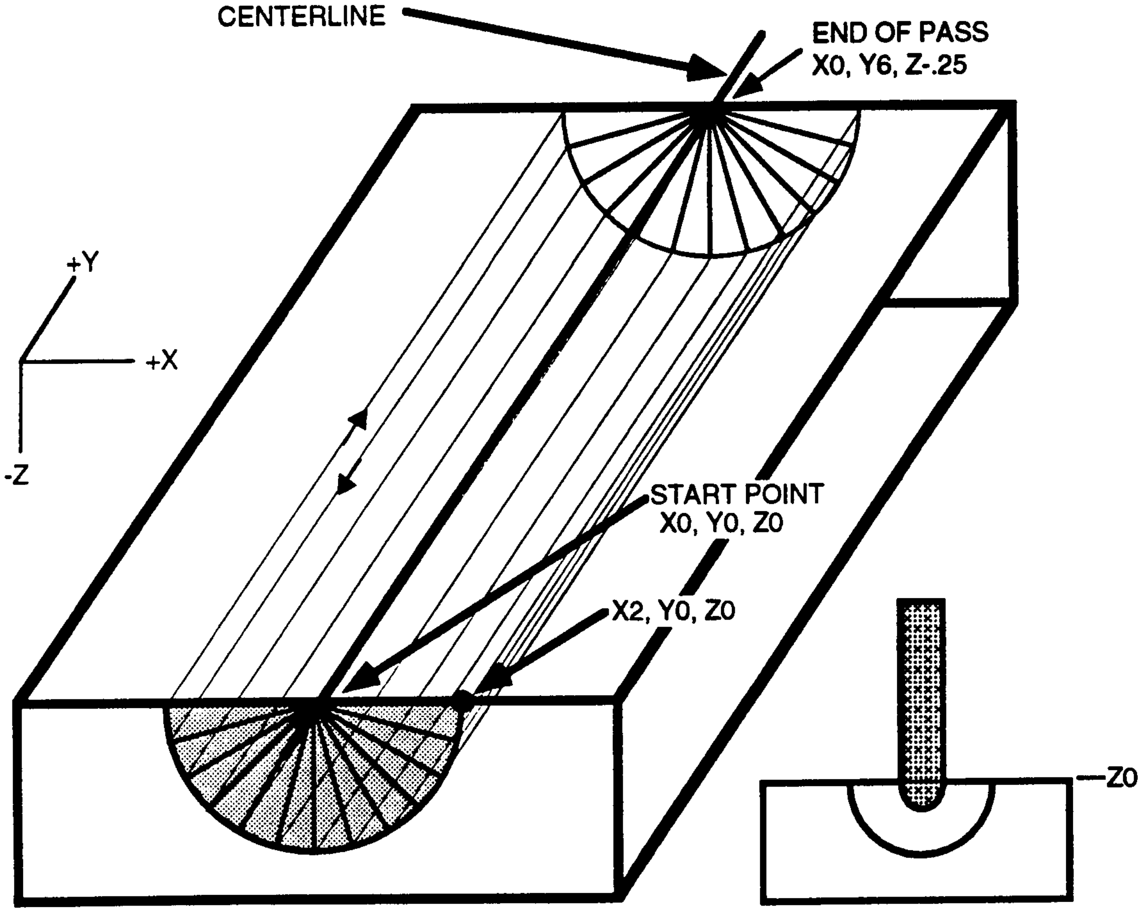
6) Program an XY tool center path that will retrace the path defined in step 4.

7) Program a Repeat event, with an incremental rotate command, that recalls the events programmed in steps 4-6. The number of repeats required to cut the cavity will depend on the incremental angle that you program. The rotation angles and number of repeats should be selected to sweep the profile through 180°

ELEMENTARY EXAMPLE

Although the cavity below can be easily programmed with a Cavity Mill event, it is used as a simple example to illustrate how to construct a cavity using the procedure above. The cavity will be cut with a .5" ball nose cutter. The tool length should be calibrated to the center of the ball nose cutter. You will be required to adjust the tool length by a distance equal to the tool radius if you originally calibrated to the tip of the cutter. Refer to the Z0 position of the tool shown on the following page. The illustration shows the tool center path.

This example can be extended to mill highly complex cavities by modifying the defined profile.



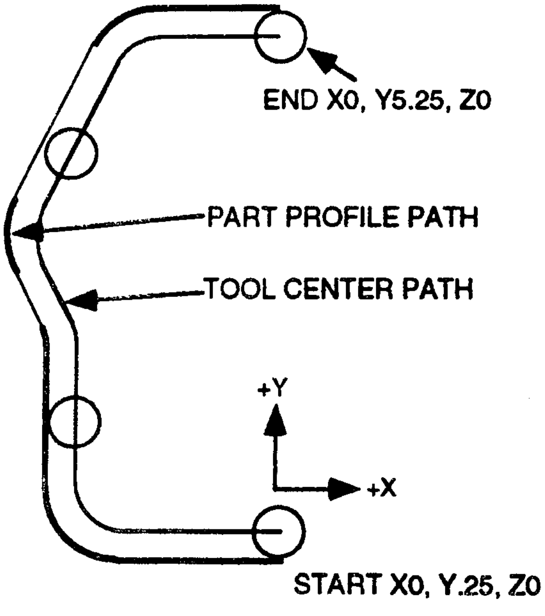
Since the cavity centerline is parallel to the Y axis, the ZX plane (P2) is the rotation plane. The rotation center is X0, Z0. The program that cuts this cavity is shown below.

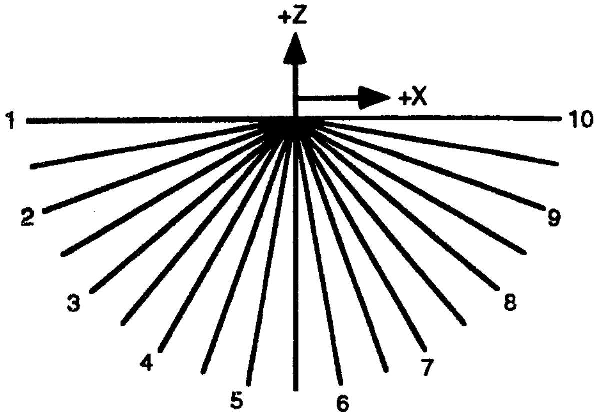
```
N10   (0)  X0 Y0 Z.5           - position above start point.
N20   (1)  X0 Y0 Z0 F20        - feed center of tool to start point.
N30   (R)  P2 X0 Z0 C0         - set rotation center in ZX plane.
N40   (1)  X-2                 - start profile.
N50   (1)  Y6                  -
N60   (1)  X0                  - end profile.
N70   (R)  P2 C-10/            - Rotate coordinate system -10°/ in ZX plane.
N80   (1)  X-2                 - start return profile.
N90   (1)  Y0                  -
N100  (1)  X0                  - end return profile.
N110  (6)  P2 C-10/ F40 T9     - rotate -10°/ in ZX plane and repeat from N40 9 times.
N120  (R)  P0 X0 Y0 C0         - cancel axis rotation.
N130  (9)  M30                 - end of program.
```

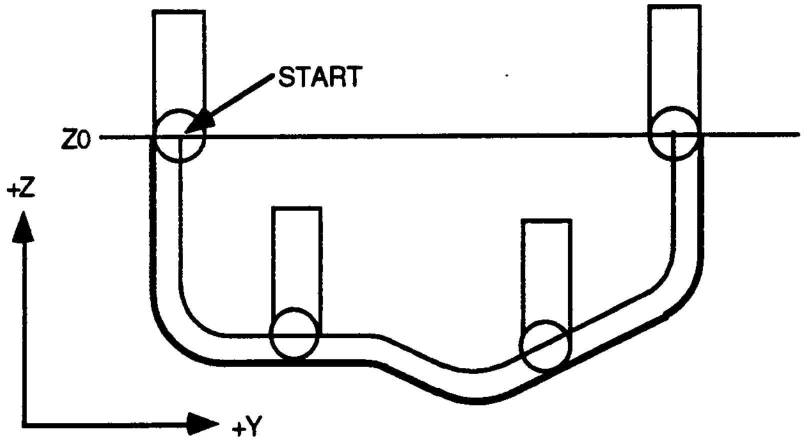
COMPLEX CAVITY EXAMPLE

The following example shows how to use vertical plane rotation to mill a custom cavity. Since cutter compensation is not allowed, a tool center path for a .5" cutter must be programmed. In this example, the axis will rotate about the point X0, Z-.25 in the ZX plane. N5 - N9 define the cavity's profile in the XY plane. Notice that the tool position at the end of the profile has the same coordinates (X0, Z-.25) as the center of rotation. This means that axis rotation can occur here without generating an offset. After the Rotate event in N10 rotates the axes, N11 - N15 define the cavity's return path in the XY plane. To complete the cavity, N16 rotates the axes, then recalls the entire sequence until the profile has been "swept" 180°. The illustration, **below**, shows the ZX cross section of the cavity.

```
N01  (9)  T1, H1, D1 M3, S1800    - tool #1, turn spindle on
N02  (0)  X0, Y.25, Z.5, F50      - start point
N03  (1)  Z0 F10                  -
N04  (R)  P2, X0, Z0, C0          - set rotation plane, center and initial angle
N05  (1)  X-2, R.5                - begin profile
N06  (1)  Y2/, R.5                -
N07  (1)  X-.5/, Y1/, R.75        -
N08  (1)  X1/, Y2/, R.5           -
N09  (1)  X0                      - end of profile
N10  (R)  P2, C-10/               - rotate 10°/. Tool is at rotation center - no offset
N11  (1)  X-1.5, R.5              - start return profile
N12  (1)  X-1/, Y-2/, R.75        -
N13  (1)  X.5/, Y-1/, R.5         -
N14  (1)  Y-2/, R.5               -
N15  (1)  X0                      - end return profile
N16  (6)  P2, C-10/, F5, T9       - rotate 10°/ and repeat entire sequence
N17  (R)  P0, X0, Y0, C0          - cancel rotation
N999 (9)  M30                     - end of program
```







CHANGING PLANES OF ROTATION

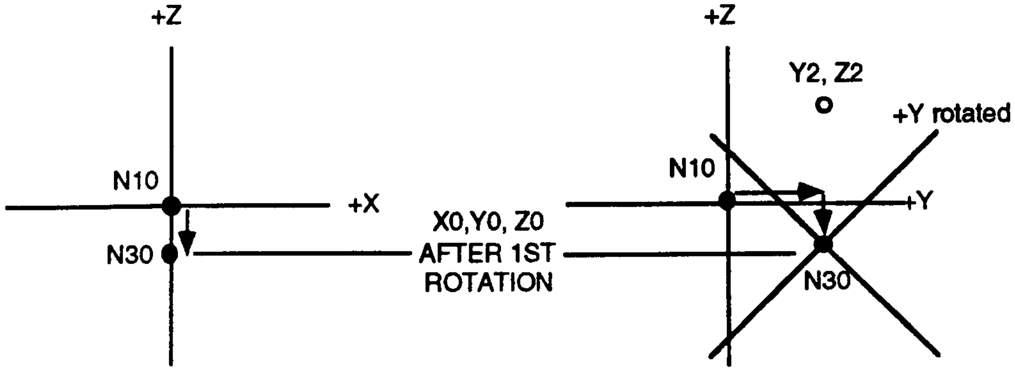
You may only rotate one plane at a time. As described earlier in the text, axis rotation does not cause axes motion, but the disparity resulting from the rotation is added to the next motion command. The first programmed Rotate event causes a disparity for both axes in the selected plane. When you switch the plane select, a disparity for all three axes will exist, because the control cancels the rotation for the out of plane axis. The program fragment **below** shows tool motion resulting from changing the plane select within a part program.

```
N10  (0)  X0 Y0 Z0         - position tool center to  X0, Y0, Z0
N20  (R)  P1 Y2 Z2 C45     - rotate 45° about Y2, Z2, no tool motion occurs but the point
                             X0,Y0, Z0 has moved.
N30  (0)  X0 Y0 Z0         - update position in YZ.  Endpoint (unrotated) is
                             X0.0000, Y2.0000, Z-.8286
N40  (R)  P0 X2 Y2 C45     - rotate 45° in XY plane, no tool motion occurs, but
                             X0,Y0,Z0 has moved. Previous Z axis rotation is canceled
N50  (0)  X0 Y0 Z0         - update position in XY, cancel Z axis rotation.  Endpoint
                             (unrotated) is X2.0000, Y-.8286, Z0000
```

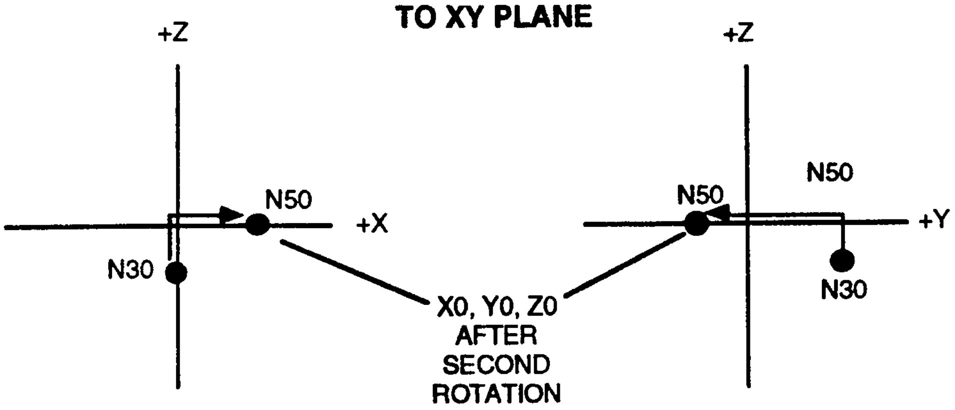
In the above program segment, N10 positions the tool to part zero. Next the YZ axes are rotated 45° about the center Y2, Z2. N30, positions the tool tip to X0, Y0, Z0 in the new orientation. Notice, **following page, upper illustration**, that only the X and Y axes move on execution of N30.

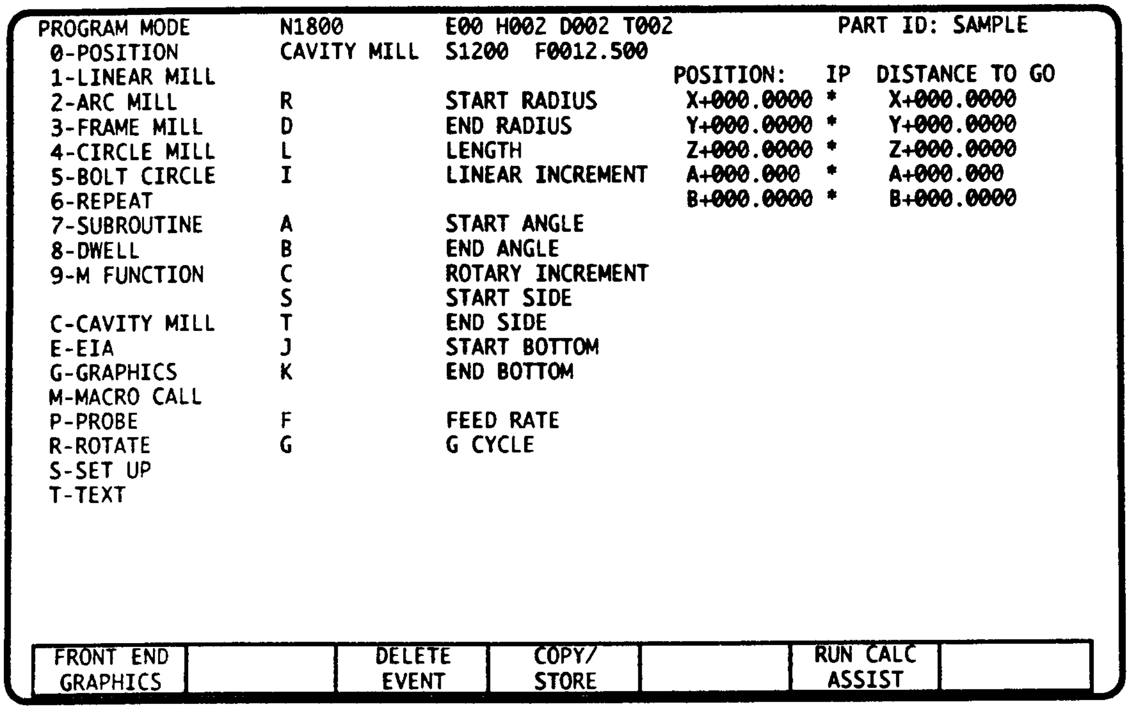
Next, the XY axes are rotated about the point X2, Y2. Since rotation is allowed in only one plane select at a time, the previous YZ rotation is canceled, and the XY rotation is executed. Finally, N50 positions the tool to the new X0, Y0, Z0 location. Note, **following page, lower illustration**, that all three axes must move because of the cancellation of the YZ rotation.

AXES ROTATION IN YZ PLANE



CHANGE AXES ROTATION TO XY PLANE





```
PROGRAM MODE      N1800         E00 H002 D002 T002            PART ID: SAMPLE
 0-POSITION       CAVITY MILL   S1200  F0012.500
 1-LINEAR MILL                                        POSITION:    IP  DISTANCE TO GO
 2-ARC MILL       R             START RADIUS           X+000.0000 *     X+000.0000
 3-FRAME MILL     D             END RADIUS             Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL    L             LENGTH                 Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE    I             LINEAR INCREMENT       A+000.000  *     A+000.000
 6-REPEAT                                              B+000.0000 *     B+000.0000
 7-SUBROUTINE     A             START ANGLE
 8-DWELL          B             END ANGLE
 9-M FUNCTION     C             ROTARY INCREMENT
                  S             START SIDE
 C-CAVITY MILL    T             END SIDE
 E-EIA            J             START BOTTOM
 G-GRAPHICS       K             END BOTTOM
 M-MACRO CALL
 P-PROBE          F             FEED RATE
 R-ROTATE         G             G CYCLE
 S-SET UP
 T-TEXT
```

INTRODUCTION

This event mills a concave or convex cavity. A variety of cavity shapes can be created by completing entries that define:

- The cavity's start cross section profile. (G, R, S, J)
- The cavity's end cross section profile. (D, T, K)
- The cavity's shape. (linear, hemisphere, or toroid) (see illustration, **facing page**)
- The cavity's surface finish by defining the step increment (linear cavity) or rotate increment (hemisphere and toroid cavities) that occurs between each cutting cycle.

You can program a sequence of Cavity Mill events to mill a complex cavity.

REQUIREMENTS

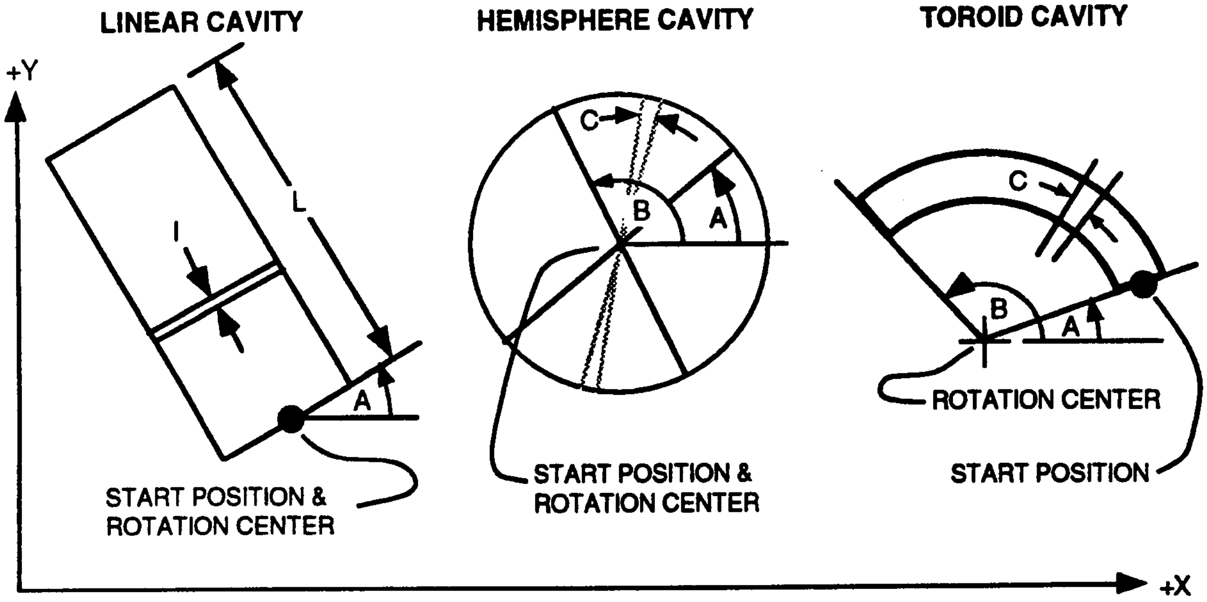
- You must use a ball end cutter whose length and diameter offsets are stored in the Tool Tables, and have been activated by a prior M Function event. You are not required to do any tool offset calculations.
- A prior Position event must move the tool to the cavity's XY start point, and .1" (2 mm) above the cavity's surface.
- A prior Rotate event must set the rotation center for the appropriate cavity shape. The rotation center is the start point of linear and hemisphere cavities, and the center of toroid cavities. The rotation angle must be 0°.

CAVITY MILL EVENT DATA

Parameter	Description	Entry
N	Sequence Number	Required
R	Start Radius	Required
D	End Radius	Optional
L	Length	Req. for linear cavity
I	Linear Increment	Req. for linear cavity
A	Start Angle	Required
B	End Angle	Req. for rotary cavity
C	Rotary Increment	Req. for rotary cavity
S	Start Side Dimension	Optional
T	End Side Dimension	Optional
J	Start Horizontal Dimension (base, G8 or ceiling, G9)	Optional
K	End Horizontal Dimension (base, G8 or ceiling, G9)	Optional
F	Feedrate	Modal
G	G Cycle (G8 = concave, G9 = convex)	Modal

REQUIRED ENTRIES FOR EACH CAVITY SHAPE

Linear	L, I, A, R
Hemisphere	A, B, C, R
Toroid	A, B, C, R

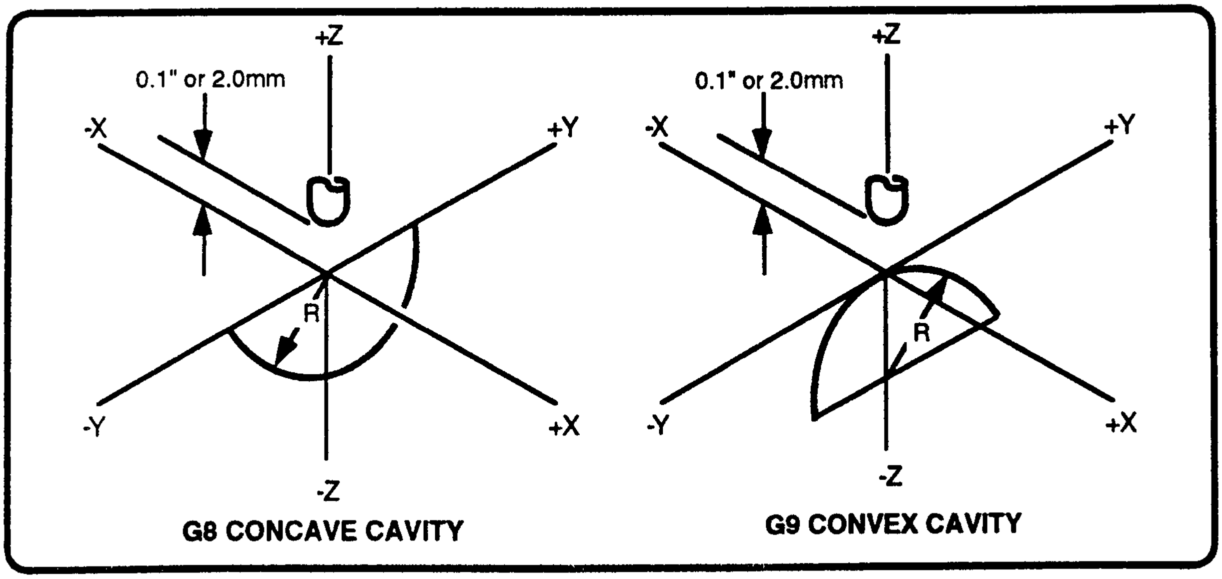


START CROSS SECTION PROFILE ENTRIES

The following entries define the cavity's shape for the first cutting pass. If any "end pass" entries are programmed, the control will incrementally increase or decrease each dimension prior to each cutting pass. These entries can be programmed for all three cavity shapes. Tool motion illustrations for example start and end profile passes are shown on the **following pages**.

G8, G9 - CONCAVE, CONVEX CAVITY CUTTING CYCLES

G chooses the cycle that will mill the cavity cross-section. G8 chooses a concave cavity, **below left**; G9 chooses a convex cavity, **below right**. G8 and G9 are modal with following Cavity Mill events. Tool motion for basic concave and convex cavities is shown on the **following** page.



R - START RADIUS

R programs the arc component of the cavity's cross section. R is a signed absolute value and is required in all Cavity Mill events. The sign of R denotes the tool direction while milling the cavity cross section. Positive R directions for G8 and G9 are shown on the **following page**. To reverse the direction, enter R as a negative value.

J - START HORIZONTAL DIMENSION

J adds an optional horizontal dimension to the cavity. For G8 concave cavities, the horizontal dimension is a linear base separating the two arc motions. For G9 convex cavities, the J is a ceiling dimension between the arcs. If J is not programmed, the two joined arc motions would describe a semicircle. J is an optional unsigned absolute entry.

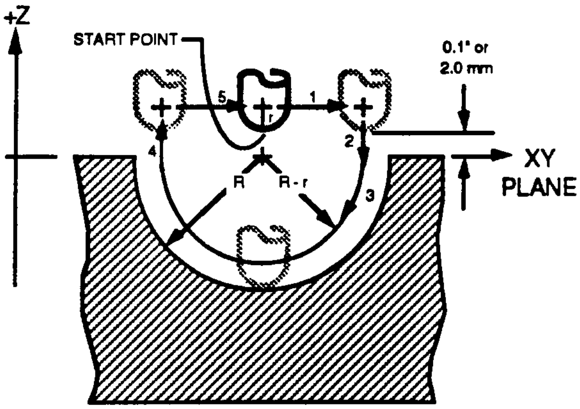
S - START SIDE DIMENSION - CAVITY CUTTING CYCLE

S adds an optional linear side dimension to the cavity. S extends the concave cavity's arc start point S units in -Z. For the convex cavity, S extends the initial -Z plunge S units beyond the arc's start point. S is an optional unsigned absolute entry.

F - FEEDRATE

F programs the milling feedrate. F is modally recognized by all following event types that program feedrate F for their respective operations.

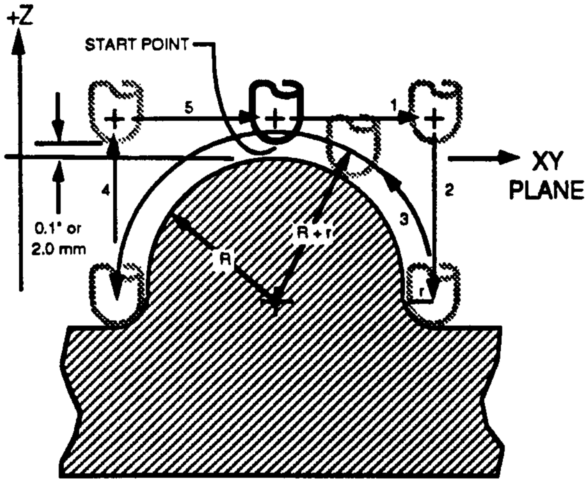
G8 - BASIC CONCAVE CUTTING CYCLE



CONCAVE (G8)
POSITIVE R

1. From the start point, the tool moves a distance equal to R-r at feedrate F in the XY plane.
2. The tool plunges a distance equal to tool radius r plus (0.1" or 2.0 mm) at feedrate F.
3. A 180° arc feed at feedrate F is described with tool center moving on radius R-r.
4. The tool rapids (0.1" or 2.0 mm) plus a distance equal to tool radius r to return to the starting Z dimension.
5. A rapid XY motion returns the tool to the start point.
6. (Not Shown) A half cylinder or toroidal cavity event will complete a short XY feed to move to the next cross-section's start point.

G9 - BASIC CONVEX CUTTING CYCLE



CONVEX (G9)
POSITIVE R

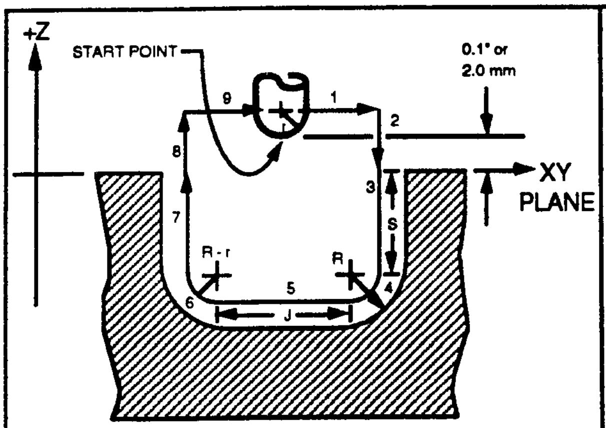
1. From the start point, the tool moves a distance equal to R+r at feedrate F in the XY plane.
2. The tool plunges a distance equal to tool radius r plus (0.1" or 2.0 mm) plus radius R at feedrate F.
3. A 180° arc feed at feedrate F is described with tool center moving on radius R+r.
4. The tool rapids a distance equal to R plus (0.1" or 2.0 mm) plus the tool radius in +Z to return to the starting Z dimension.
5. A rapid XY motion returns the tool to the start point.
6. (Not Shown) A half cylinder or toroidal cavity event will complete a short XY feed to move to the next cross-section's start point.

END CROSS SECTION PROFILE ENTRIES

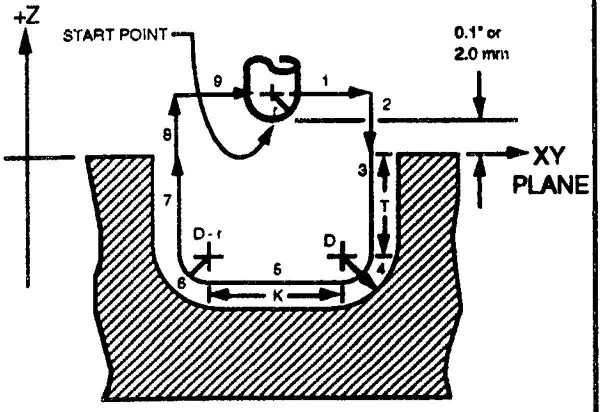
The following entries define the cavity's shape for the last cutting pass. If any "end pass" entries are programmed, the control will incrementally increase or decrease each dimension prior to each cutting cycle. These entries can be programmed for all three cavity shapes.

D - END RADIUS - CAVITY CUTTING CYCLE

D is an optional entry that assigns an arc radius to the last cavity cutting cycle. If D is not programmed, the entire cavity will be cut with radius R. When D is programmed, start radius R will uniformly change in length over each repetition of the cavity cutting cycle until the final cycle is cut with radius D. D may be larger or smaller than R, but may not be entered unless R has also been entered. D is an optional unsigned absolute entry.



START CAVITY CONCAVE (G8)
POSITIVE RADIUS WITH S, J, R VARIABLES



END CAVITY CONCAVE (G8)
POSITIVE RADIUS WITH T, K, D VARIABLES

G8 - CONCAVE CYCLE OPTIONS

Prior to the first cutting cycle the cutter is to be positioned on center and 0.1" or 2.0 mm above the cavity ceiling.

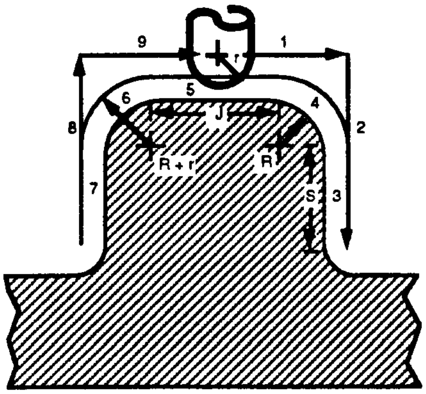
1. I f J has been programmed, the cutter moves 1/2J units at feedrate F in the XY plane. The cutter then feeds a distance equal to R-r in the XY plane.
2. The cutter then plunges a distance equal to tool radius r plus (0.1" or 2.0 mm).
3. If S has been programmed, the cutter will continue to plunge S additional units.
4. A 90° arc feed is described with -Z motion, the tool center moving on radius R-r.
5. If J has been programmed, the cutter feeds J units in XY.
6. A 90° arc feed is described with +Z motion, the tool center moving on radius R-r.
7. If S has been programmed, the cutter feeds S units in +Z.
8. The cutter moves (0.1" or 2.0 mm) plus a distance equal to tool radius r to return to the starting Z dimension.
9. The cutter rapids in XY back to the start point.
10. (Not Shown) A half cylinder or toroidal cavity event will complete a short XY feed to move to the next cross-section's start point.

K - END HORIZONTAL DIMENSION - CAVITY CUTTING CYCLE

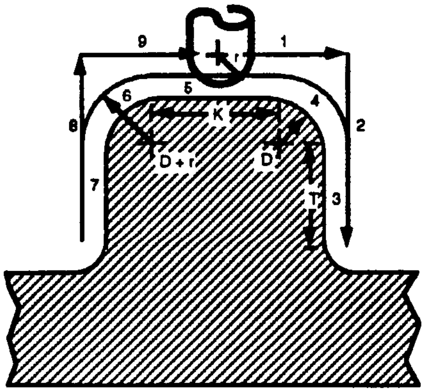
K is an elective entry that assigns a linear horizontal dimension to the last cavity cutting cycle. If J is programmed but K is not, the entire cavity will be cut with horizontal dimension J. When K is programmed, start dimension J will uniformly change in length over each repetition of the cycle until the final cavity cycle is cut with dimension K. K may be larger or smaller than J, but K may not be programmed without J or a fault will occur. K is an optional unsigned absolute entry.

T - END SIDE DIMENSION - CAVITY CUTTING CYCLE

T is an elective entry that assigns a linear side dimension to the last cavity cutting cycle. If S is programmed but T is not, the entire cavity will be cut with side S. When T is programmed, Start side dimension S will uniformly change in length over each repetition of the cycle until the final cycle is cut with side dimension T. T may be larger or smaller than S, but T may not be programmed without S or a fault will occur. T is an optional unsigned absolute entry.



START CAVITY CONVEX (G9)
POSITIVE RADIUS WITH S, J, R VARIABLES



END CAVITY CONVEX (G9)
POSITIVE RADIUS WITH T, K, D VARIABLES

G9 - CONVEX CYCLE OPTIONS

Prior to the first cutting cycle the cutter is to be positioned on center and 0.1" or 2.0 mm above the cavity ceiling.

1. I f J has been programmed, the cutter moves 1/2J units at feedrate F in the XY plane. The cutter then feeds a distance equal to R+r in the XY plane.
2. The cutter then plunges a distance equal to tool radius r plus (0.1" or 2.0 mm) plus R, and if S has been programmed, S additional units.
3. If S was programmed, the cutter feeds S units in +Z.
4. A 90° arc feed is described with +Z motion, the tool center moving on radius R+r.
5. If J has been programmed, the cutter feeds J units in XY.
6. A 90° arc feed is described with -Z motion, the tool center moving on radius R+r.
7. If S was programmed, the cutter feeds S units in +Z.
8. If S was programmed, the cutter retreats, at rapid, S units in +Z. The cutter then retreats, at rapid, R units plus (0.1" or 2.0 mm) plus a distance equal to cutter radius r.
9. The cutter rapids in XY back to the start point.
10. (Not Shown) A half cylinder or toroidal cavity event will complete a short XY feed to move to the next cross-section's start point.

CAVITY SHAPE DEFINITION

After you complete entries for the cavity's start and end pass, you must then program entries to describe the cavity's shape and size. You can specify a linear, hemisphere or toroid cavity. The combination of L, I, A, B, C and the relationship between the tool's start position and the center of rotation determines the cavity's shape and size.

LINEAR CAVITY

A linear cavity mills a series of cavity cross-sections at right angles to the cavity's center line. Standard and tapered half cylinders are representative of this class of cavity shapes. The following L, I, and A entries assign a linear cavity construction and define the cavity's dimensions. Prior to programming a linear cavity, you must first move the tool to the cavity's XY start point, 0.1" (2 mm) above the surface, and set the XY rotation center to the cavity's start point.

A - START ANGLE

Start angle A sets the linear cavity's orientation in the XY plane by rotating the cavity's cross section A degrees. Angle A opens clockwise from the zero degree reference line that starts from the cavity's rotation center. A is an unsigned absolute entry and is required in all Cavity Mill events. As describe in following topics, start angle A opens from the center of hemisphere and toroid cavities.

L - LENGTH - LINEAR CAVITY

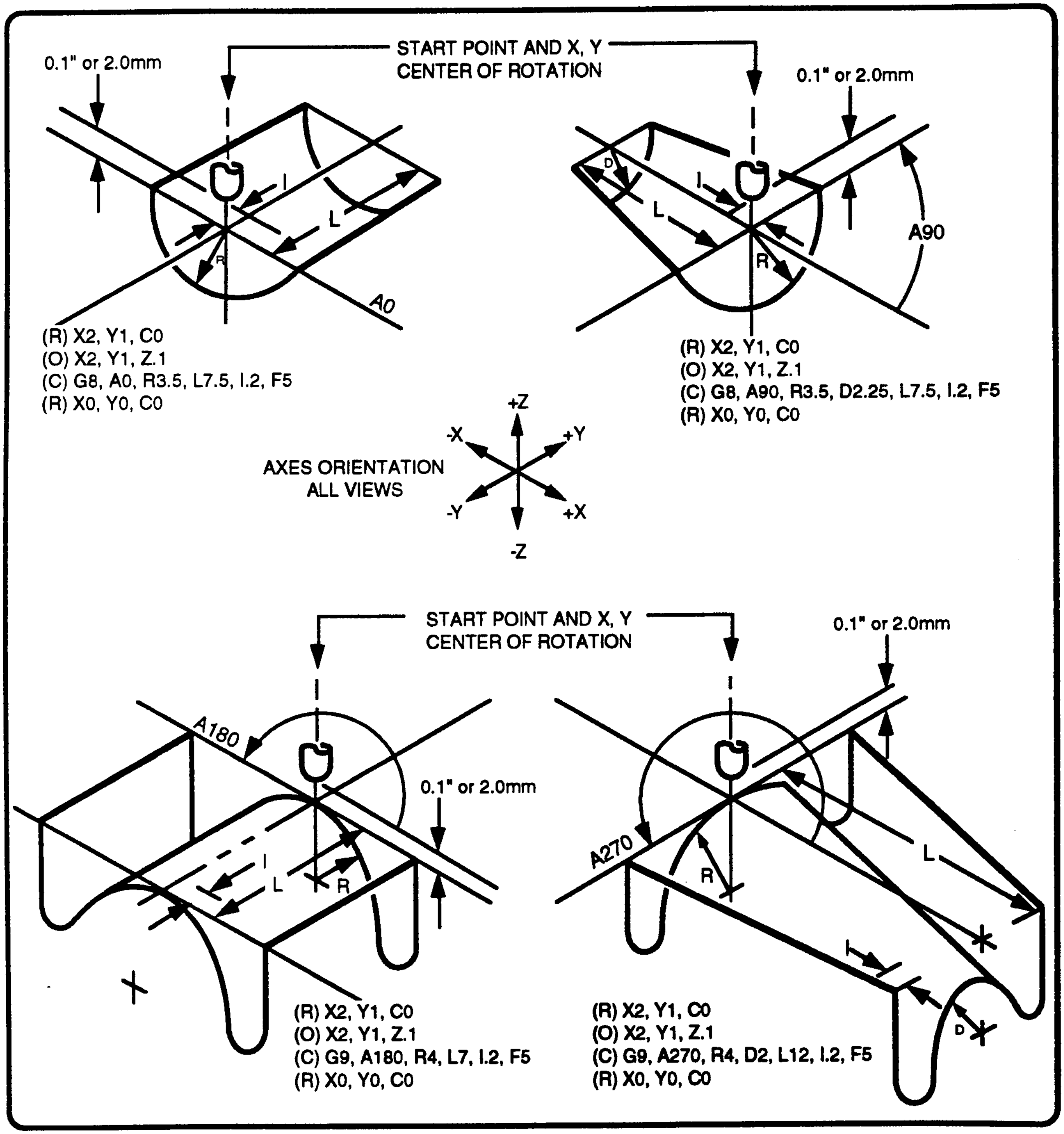
L enters the length along the linear cavity's centerline. The length is defined from the cavity's start point and is directed counterclockwise and perpendicular from Start Angle A in the XY plane. L is an unsigned absolute entry and is required in all linear Cavity Mill events.

I - LINEAR INCREMENT - LINEAR CAVITY

Linear Increment I lets you specify the finish quality of the linear cavity by specifying the incremental distance of feed along the linear cavity's centerline between each repetition of the cavity cutting cycle. I is an unsigned absolute entry and is required in all linear Cavity Mill events.

LINEAR CAVITY ILLUSTRATIONS AND PROGRAM EXAMPLES

The illustrations and program segments **below**, demonstrate standard and tapered half cylinder concave and convex linear cavities. The cavities are shown with various orientations in the XY plane, as determined by start angle A. In the following examples, note preparatory Position and Rotate events and their location with respect to the cavity's start point. Further note that the tool is pre-positioned 0.1" or 2 mm above the cavity's surface. A Rotate event follows each cavity. Although it is not required with each sequence, it restores the center of rotation to default X0, Y0 values.



```
(R) X2, Y1, C0
(O) X2, Y1, Z.1
(C) G8, A0, R3.5, L7.5, I.2, F5
(R) X0, Y0, C0
```

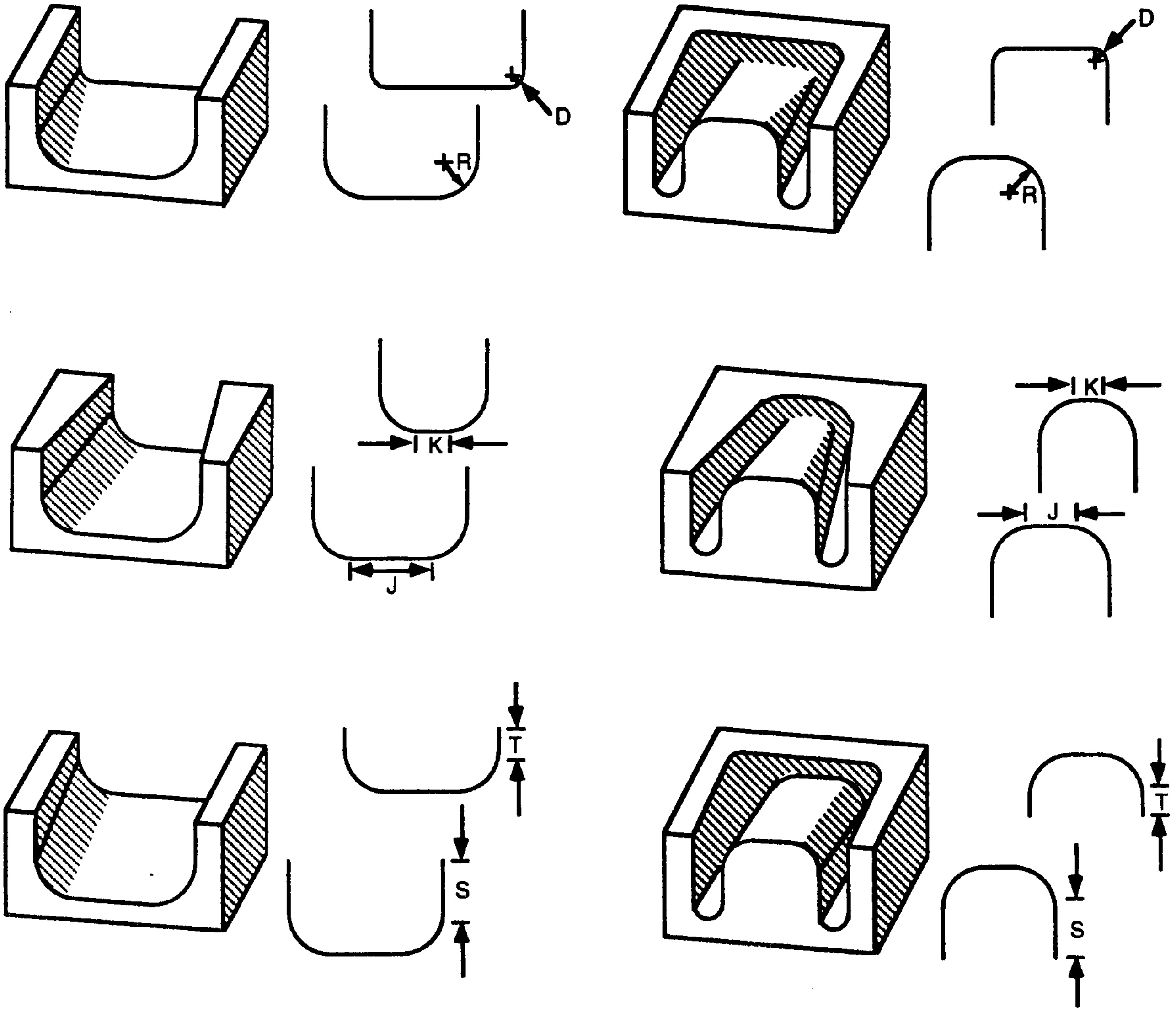
```
(R) X2, Y1, C0
(O) X2, Y1, Z.1
(C) G8, A90, R3.5, D2.25, L7.5, I.2, F5
(R) X0, Y0, C0
```

```
(R) X2, Y1, C0
(O) X2, Y1, Z.1
(C) G9, A180, R4, L7, I.2, F5
(R) X0, Y0, C0
```

```
(R) X2, Y1, C0
(O) X2, Y1, Z.1
(C) G9, A270, R4, D2, L12, I.2, F5
(R) X0, Y0, C0
```

ADVANCED LINEAR CAVITIES

The cavities **below** further adjust the profiles by programming optional J, K, S and T entries. Each cutting cycle has added linear dimensions J and S. The upper illustrations program end radius D. The center illustrations program end horizontal dimension K. The lower illustrations program end side dimension T.



HEMISPHERE and TOROID CAVITIES

To cut hemisphere and toroid cavities, the control rotates the cavity's cross section path in the XY plane before each pass. Hemisphere shapes are rotated about the cavity's center; toroid shape are rotated about a point external to the cavity. The following A, B and C entries must be programmed for every hemisphere and toroid cavity shape.

Note that the difference between the hemisphere and toroid shape is the relationship between the tool's start position and the programmed rotation center. For hemisphere cavities, the start position and rotation center are at the same XY point. For toroid cavities, the start position and rotation center must be different XY points. The cavity's centerline will be rotated about the point programmed in the preparatory Rotated event.

A - START ANGLE - HEMISPHERE and TOROID CAVITIES

A is an unsigned absolute entry which is required in all hemisphere and toroid Cavity Mill events. A programs an angle which sets the cavity's initial orientation in the XY plane. A opens counterclockwise from the rotation center that was programmed in a prior Rotate event. For a toroid shape, A is the angle from the rotation center to the tool's start position.

B - END ANGLE - HEMISPHERE and TOROID CAVITIES

B is an unsigned absolute entry which is required in all hemisphere and toroid Cavity Mill events. B opens counterclockwise from the rotation center that was programmed in a prior Rotate event. B may be larger or smaller than start angle A.

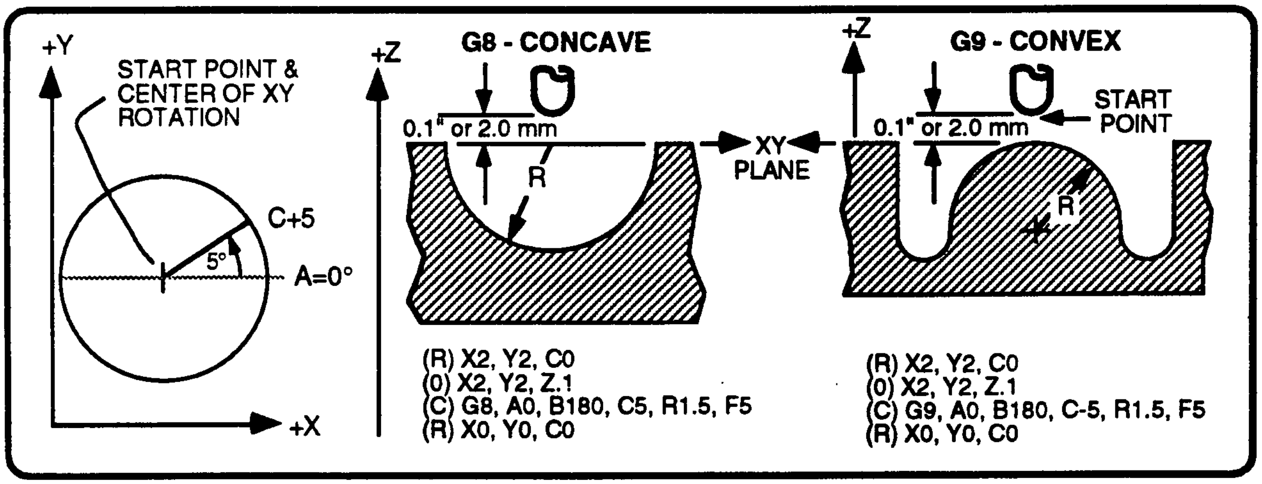
C - ROTARY INCREMENT - HEMISPHERE and TOROID CAVITIES

C programs the rotary increment about the XY center of rotation between repetitions of the cavity cutting cycle. Smaller increments produce a smoother surface finish but require additional machining time. C is a signed absolute entry. Positive C programs counterclockwise incremental rotation while negative C sets clockwise incremental rotation between Start Angle A and End Angle B.

Note: This entry is programmed with the letter "H" if your control is equipped with an auxiliary axis that is programmed with the letter "C". Refer to Part 1-Section 1.

HEMISPHERE

Hemisphere cavities must have the same start point and rotation center. The cavity's profile is rotated C° before each cutting pass to produce a hemispherical shape. The illustrations **below** show XY and cross-section views of the G8 and G9 cavities. Note that the angle between A and B need only be 180° to cut the entire hemisphere.



```
(R) X2, Y2, C0
(0) X2, Y2, Z.1
(C) G8, A0, B180, C5, R1.5, F5
(R) X0, Y0, C0
```

```
(R) X2, Y2, C0
(0) X2, Y2, Z.1
(C) G9, A0, B180, C-5, R1.5, F5
(R) X0, Y0, C0
```

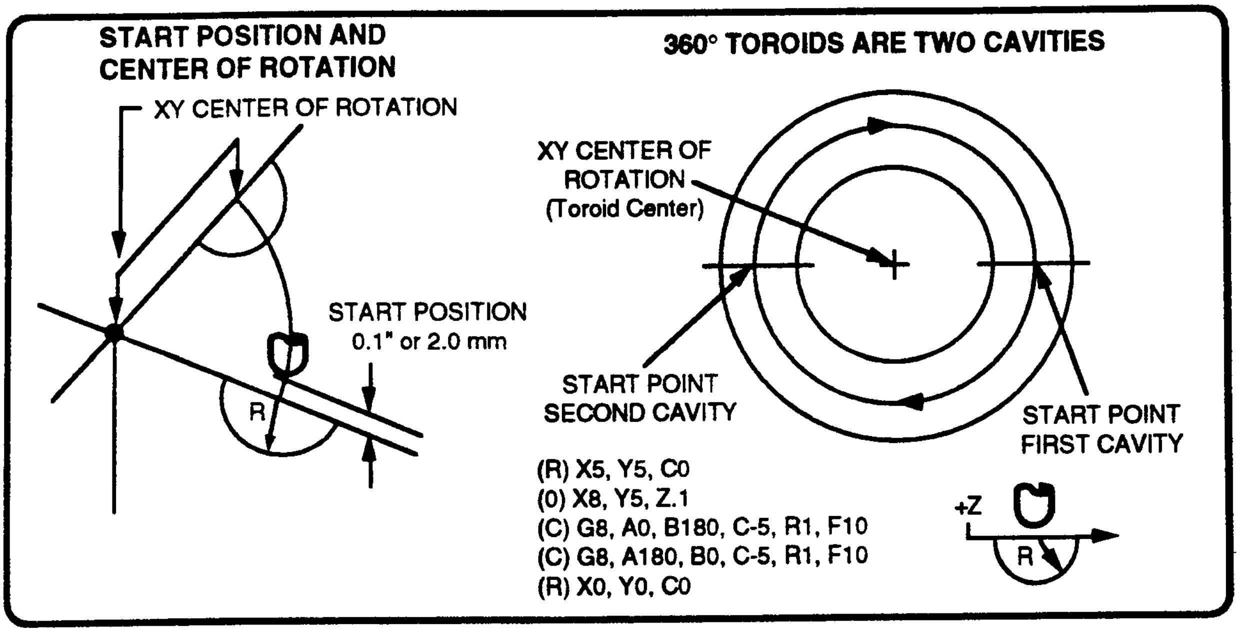
TOROID

Toroid cavities, **following page**, must have the tool's start position at a different point than the rotation center. The distance between these two points determines the toroid's radius (the toroid's radius is the distance from the rotation center to the cavity's centerline).

As with preceding cavity shapes, the preparatory Rotate and Position events share constructive data with the Cavity Mill event that will mill the toroid. In the case of the toroid, however, the programmer must take sufficient steps to insure that start angle A and end angle B aligns the initial and final cavity pass with the XY rotation center.

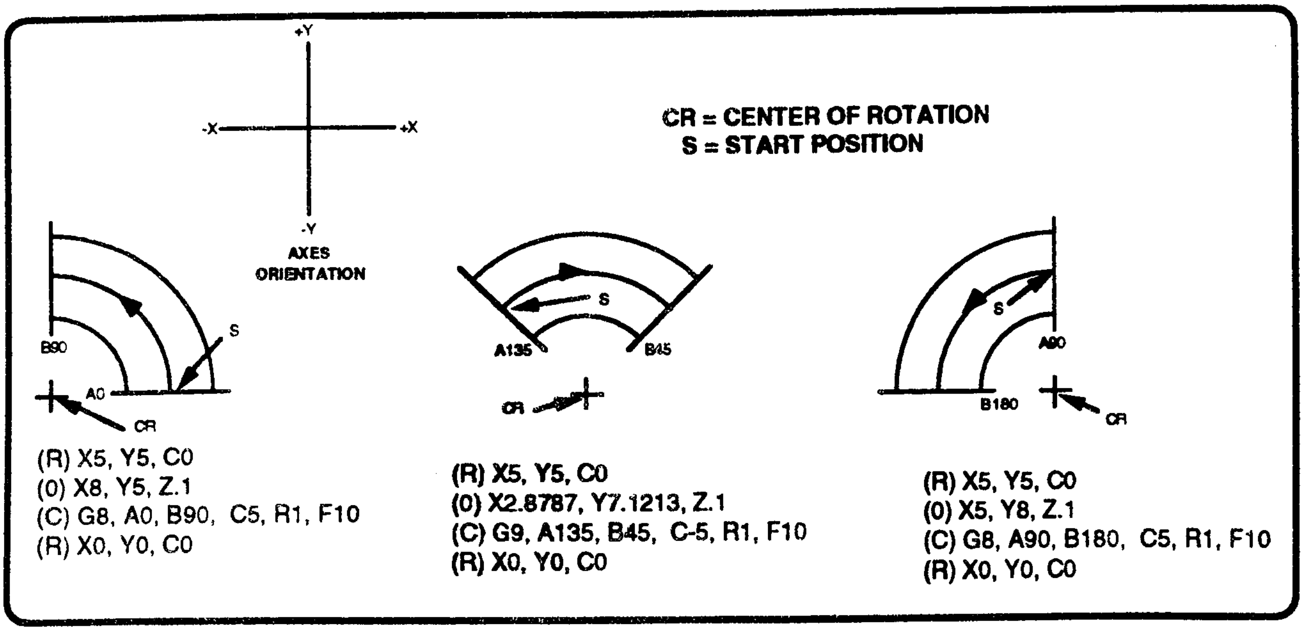
The toroid's center of rotation is entered by a preparatory Rotate event. The Rotate event's X and Y coordinates locate the toroid's center. The initial cavity cutting cycle must be radially aligned with the toroid's center and start point through careful determination of Start Angle A. Similarly, the final cavity cutting cycle must be aligned with the toroid's center and end point with end angle B.

The included angle between angles A and B must be less than 360°. The **lower right** illustration shows a 360° toroid milled by two successive Cavity Mill events. In this case both cavities share the same XY center of rotation and the endpoint of cavity 1 is the start point of cavity 2, so intermediate Rotate and Position events are not required.


START POSITION AND CENTER OF ROTATION
XY CENTER OF ROTATION
R
START POSITION
0.1" or 2.0 mm
360° TOROIDS ARE TWO CAVITIES
XY CENTER OF ROTATION (Toroid Center)
START POINT SECOND CAVITY
START POINT FIRST CAVITY
+Z
R

```
(R) X5, Y5, C0
(0) X8, Y5, Z.1
(C) G8, A0, B180, C-5, R1, F10
(C) G8, A180, B0, C-5, R1, F10
(R) X0, Y0, C0
```

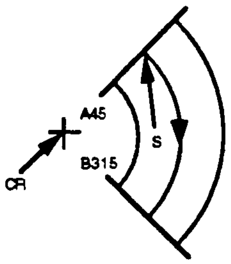
Various orientations are shown for G8 and G9 segments of a toroid with a 3 inch radius and center at X5, Y5, **below, and following page**. Note the preparatory Position event's X and Y coordinates and the assignments for angles A and B.



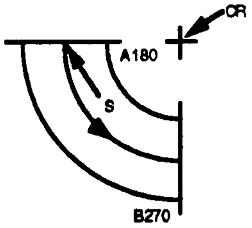
```
(R) X5, Y5, C0
(0) X8, Y5, Z.1
(C) G8, A0, B90, C5, R1, F10
(R) X0, Y0, C0
```

```
(R) X5, Y5, C0
(0) X2.8787, Y7.1213, Z.1
(C) G9, A135, B45, C-5, R1, F10
(R) X0, Y0, C0
```

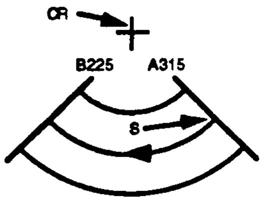
```
(R) X5, Y5, C0
(0) X5, Y8, Z.1
(C) G8, A90, B180, C5, R1, F10
(R) X0, Y0, C0
```



```
(R) X5, Y5, C0
(0) X7.1213, Y7.1213, Z.1
(C) G9, A45, B315, C-5, R1
(R) X0, Y0, C0
```

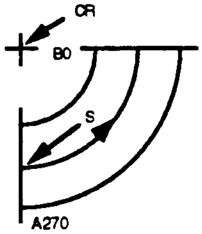


```
(R) X5, Y5, C0
(0) X2, Y5, Z.1
(C) G8, A180, B270, C5, R1
(R) X0, Y0, C0
```

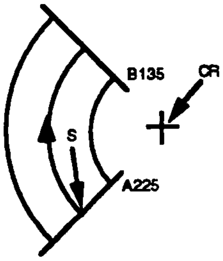


```
(R) X5, Y5, C0
(0) X7.1213, Y2.8787, Z.1
(C) G9, A315, B225, C-5, R1
(R) X0, Y0, C0
```

CR = CENTER OF ROTATION
S = START POSITION



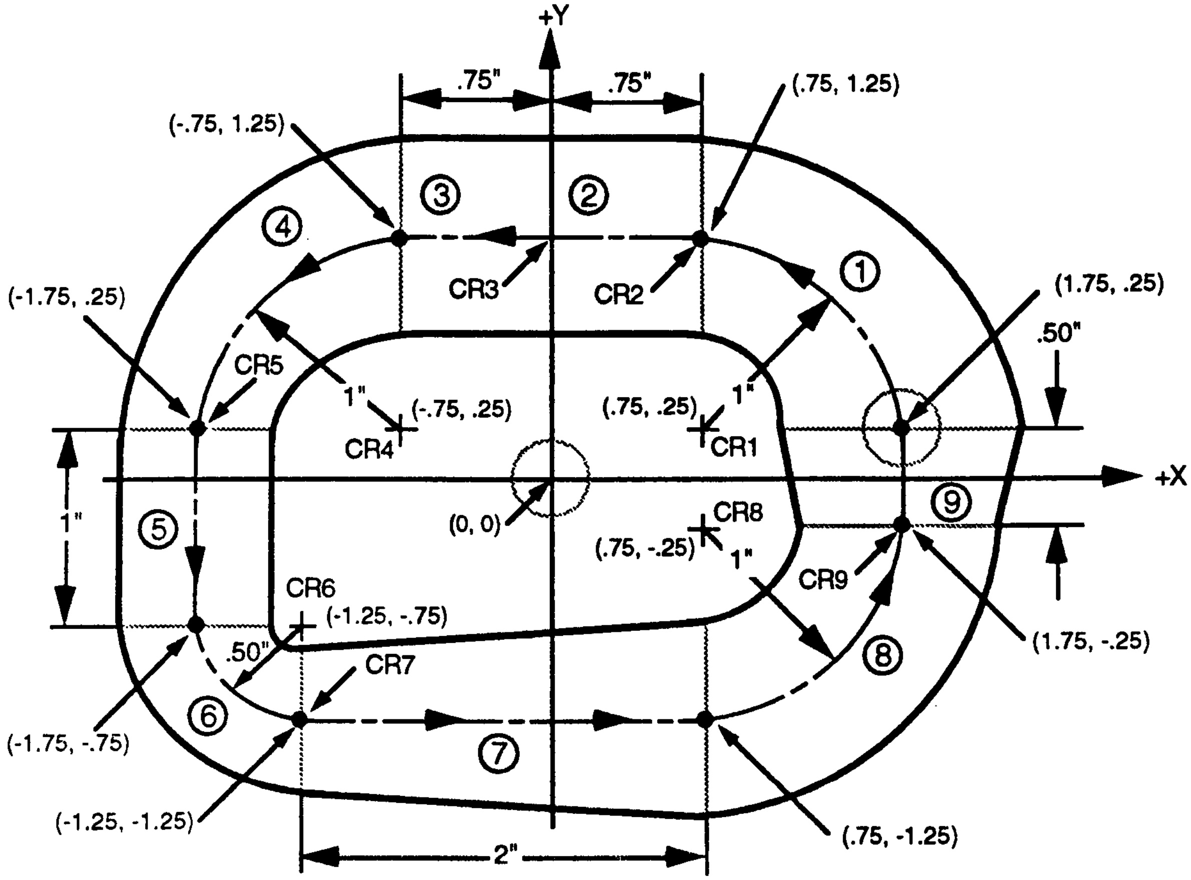
```
(R) X5, Y5, C0
(0) X5, Y2, Z.1
(C) G8, A270, B0, C5, R1, F10
(R) X0, Y0, C0
```



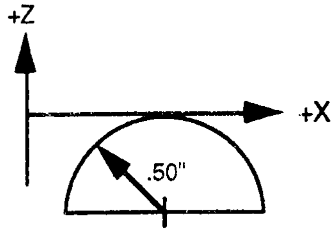
```
(R) X5, Y5, C0
(0) X2.8787, Y2.8787, Z.1
(C) G9, A225, B135, C-5, R1, F10
(R) X0, Y0, C0
```

COMPLEX CAVITY

The complex cavity, **following page**, is a closed construction of convex quarter toroids and half cylinders. Optional D, S, T, J and K entries have also been programmed to suggest various approaches to cavity joining. A 0.25 inch diameter ball end cutter was used to mill the complex cavity. Note from the adjoining program sequence that each preparatory (R) Rotate event assigns a C0 (unrotated) orientation and unique center of rotation to its respective cavity. When successive Cavity Mill events share the same end/start points and the Rotate event programs C0, intermediate preparatory Position events are not required. Since the entire construction relies on a modal G cycle entry, this program sequence can also mill a concave construction by replacing the G9 modal entry with G8 in event N0040. The illustration's circled numbers key each elementary cavity in the construction to the following description.



CR DESIGNATES CENTER OF ROTATION



```
N0000  (T)  COMPLEX CAVITY
N0010  (9)  M03, T1, H1, D1, S1000
N0020  (R)  X.75, Y.25, C0
N0030  (0)  X1.75, Y.25, Z.1
N0040  (C)  G9, A0, B90, C2.5, R.5, J.25, K0, F60
N0050  (R)  X.75, Y1.25, C0
N0060  (C)  A90, L.75, I.05, R.5, S0, T.25
N0070  (R)  X0, Y1.25, C0
N0080  (C)  A90, L.75, I.05, R.5, S.25, T0
N0090  (R)  X-.75, Y.25, C0
N0100  (C)  A90, B180, C2.5, R.5, D.25, J0, K.25
N0110  (R)  X-1.75, Y.25, C0
N0120  (C)  A180, L1, I.05, R.25, J.25
N0130  (R)  X-1.25, Y-.75, C0
N0140  (C)  A180, B270, C2.5, R.25, J.25
N0150  (R)  X-1.25, Y-1.25, C0
N0160  (C)  A270, L2, I.05, R.25, D.5, J.25, K0
N0170  (R)  X.75, Y-.25, C0
N0180  (C)  A270, B0, C2.5, R.5
N0190  (R)  X1.75, Y-.25, C0
N0200  (C)  A0, L.5, I.05, R.5, J0, K.25
N0210  (R)  X0, Y0, C0
N0220  (9)  M30
```

CAVITY 1 is a quarter convex toroid with a radius of 1".
This cavity's ceiling dimension changes from J.25 to K0 inches.

N0020 sets the center of rotation at X.75, Y.25.
N0030 positions to the start point X1.75, Y.25, Z.1 (unrotated).
N0040 mills the cavity. The cutter is now at the CAVITY 2 start point.

CAVITY 2 is a convex half cylinder .75" long.
This cavity's side dimension changes from S0 to T.25 inches.

N0050 sets the center of rotation at X.75, Y1.25, C0.
N0060 mills the cavity. The cutter is now at the CAVITY 3 start point.

CAVITY 3 is another convex half cylinder .75" long.
This cavity's side dimension changes from S.25 to T0.

N0070 sets the center of rotation at X0, Y1.25, C0.
N0080 mills the cavity. The cutter is now at the CAVITY 4 start point.

CAVITY 4 is a quarter convex tapered toroid with a 1" radius.
This cavity's cross-section radius changes from R.5 to D.25 inches.
Its ceiling dimension also changes from J0 to K.25 inches.

N0090 sets the center of rotation at X-.75, Y.25, C0.
N0100 mills the cavity. The cutter is now at the CAVITY 5 start point.

CAVITY 5 is a convex half cylinder 1" long.
This cavity's ceiling dimension is uniformly maintained at J.25 inches.

N0110 sets the center of rotation at X-1.75, Y.25, C0.
N0120 mills the cavity. The cutter is now at the CAVITY 6 start point.

CAVITY 6 is a quarter convex toroid with a .5" radius.
This cavity's ceiling dimension is uniformly maintained at J.25 inches.

N0130 sets the center of rotation at X-1.25, Y-.75, C0.
N0140 mills the cavity. The cutter is now at the CAVITY 7 start point.

CAVITY 7 is a convex tapered half cylinder 2" long.
This cavity's cross-section radius changes from R.25 to D.5 inches.
Its ceiling dimension also changes from J.25 to K0 inches.

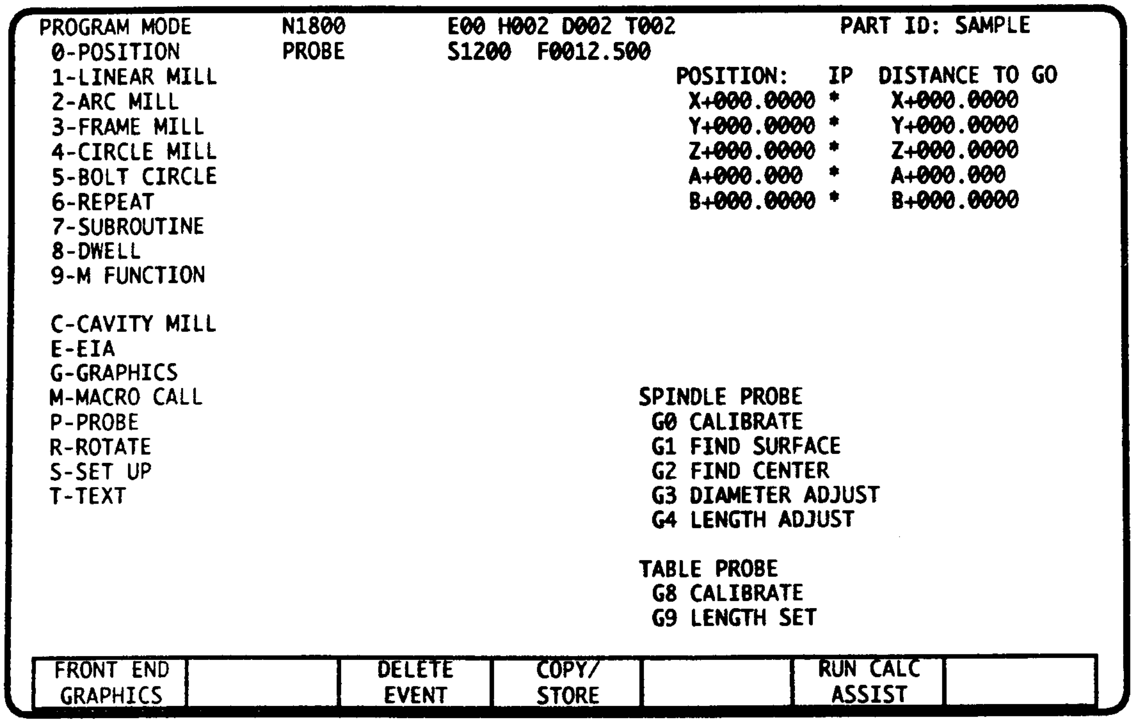
N0150 sets the center of rotation at X-1.25, Y-1.25, C0.
N0160 mills the cavity. The cutter is now at the CAVITY 8 start point.

CAVITY 8 is a quarter convex toroid with a 1" radius.

N0170 sets the center of rotation at X1.75, Y-.25, C0.
N0180 mills the cavity. The cutter is now at the CAVITY 9 start point.

CAVITY 9 is a convex squat tapered half cylinder .5" long.
This cavity's ceiling dimension changes from J0 to K.25 inches.

N0190 sets the center of rotation at X1.75, Y-.25, C0.
N0200 mills the cavity. The cutter has returned to the CAVITY 1 start point.
N0210 resets the center of rotation to the default X0, Y0 values.
N0220 returns the cutter in Z, then XY, to Home position.



```
PROGRAM MODE        N1800       E00 H002 D002 T002              PART ID: SAMPLE
 0-POSITION         PROBE       S1200  F0012.500
 1-LINEAR MILL                                   POSITION:   IP  DISTANCE TO GO
 2-ARC MILL                                       X+000.0000 *    X+000.0000
 3-FRAME MILL                                     Y+000.0000 *    Y+000.0000
 4-CIRCLE MILL                                    Z+000.0000 *    Z+000.0000
 5-BOLT CIRCLE                                    A+000.000  *    A+000.000
 6-REPEAT                                         B+000.0000 *    B+000.0000
 7-SUBROUTINE
 8-DWELL
 9-M FUNCTION

 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL                                  SPINDLE PROBE
 P-PROBE                                        G0 CALIBRATE
 R-ROTATE                                       G1 FIND SURFACE
 S-SET UP                                       G2 FIND CENTER
 T-TEXT                                         G3 DIAMETER ADJUST
                                                G4 LENGTH ADJUST

                                               TABLE PROBE
                                                G8 CALIBRATE
                                                G9 LENGTH SET
```

PROBE EVENT - INTRODUCTION

Delta milling controls offer an optional Probing event with specialized cycles for both spindle and table probes. A table probe permits fast and accurate tool length calibration. Spindle probe cycles locate parts or fixtures by surface or hole center detection. Further cycles allow adjustments of stored tool lengths and diameters when the probe senses error between programmed and part dimensions.

The first time you install a probe, change probes, insert a new stylus, or move the calibration fixture, you must complete the Probe Setup Routine described in this section.

MACHINE ACCURACY

Probe accuracy is a function of machine accuracy. The machine's axes must be properly aligned and able to position without play or binding. For machines that sense position by lead screw rotation, it may be necessary to compensate for pitch error to obtain sufficient positioning accuracy for probing.

DATA RETENTION - CALIBRATION

The probe must be calibrated before use. Probe calibration values will be retained only until the control is turned off. Probe calibration determines a "probe offset" that represents the probe's deflection angle when in hits a surface along each axis. Probe calibration procedures are described later in this section. These cycles must be run at least once a day, and with greater frequency depending on temperature fluctuation and other environmental variations in your shop.

PROBE PROTECT

Probe Protect is designed to prevent probe damage due to accidental probe contact.

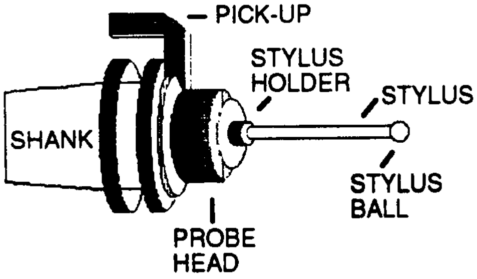
When Probe Protect is ON, any probe contact that occurs outside of a Probe event's target range, the CNC halts all machine motion and displays a PROBE CONTACT fault. Probe Protect can be turned on by your supplier's machine interface, optional switches or pushbuttons, or optional M Codes. Removing the probe from the spindle appears to the control as a probe contact. Therefore, you must turn Probe Protect OFF before removing the spindle probe.

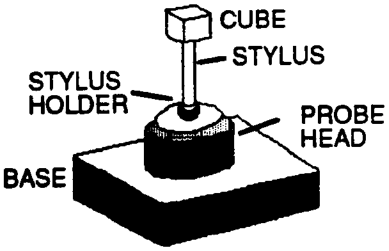
PROBES

The probe that is used with the Autocon system must be the closing contact type. The contact opens when a surface is detected along the X, Y, or Z axes. A list of approved probe manufacturers can be obtained from our Sales Engineering department.

The spindle probe, **below left**, requires that a pick-up device be used, rather than wires running directly to the probe, so that the probe can be easily dismounted, or handled by a tool changer. The table probe, **below right**, is mounted at a known location on the machine table, and directly connected by a cable. It is typically used to automatically calibrate tools.

Dirt or deposits on the shank can cause probe misalignment and inaccuracy. The stylus, shank, and tool holder should be kept meticulously clean.





SPINDLE PROBE - INITIAL SETUP

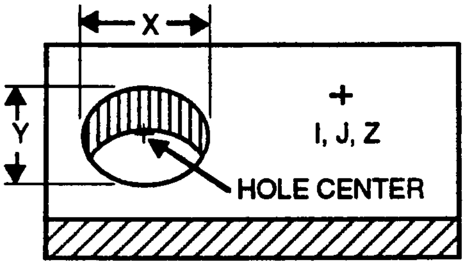
You must compete this procedure the first time you install a new probe, change the probe's stylus, or change the location of the probe calibration fixture. Probe accuracy is dependent on the accuracy of the data found during this setup. The dimensions that you must find during this setup routine are:

- Width, length, and center coordinates for a slot or hole on the probe calibration fixture.
- Z surface coordinate of the probe calibration fixture.
- The spindle probe's tool length offset.
- XYZ machine coordinates of the table probe.

WHAT YOU NEED

Reference Tool - The reference tool is used as a gauge for measuring Z dimensions. Since the reference tool is usually not used for machining, it can be as simple as a bar seated in a shank. The exact length of the reference tool, the distance between the spindle nose and the tip of the tool, must be known. The reference tool's length will also be used in the daily calibration of the table probe. A 10" reference tool is used in our following examples.

Test Fixture - The test fixture, **below**, provides a very accurate and repeatable datum. Because the test fixture is used to calibrate the probe, it's dimensions and coordinate placement on the machine table are critical. The fixture should be permanently mounted to the table, or pinned to provide accurate replacement. The slot or hole that will be machined in the test fixture is typically 2 inches or larger.



FIND HOLE or SLOT DIMENSIONS

After you mount this fixture on the machine, you must determine the width, length and center coordinates of the hole or slot. It is easiest if you make the hole or slot when the fixture is at its permanent position. If boring a hole, record the XY coordinates of the hole. If you mill a slot with a G7 Frame Mill event (with 2R=X), the center of the slot is the point where the initial plunge occurs. Next measure and record the width X and height Y of the hole or slot. If the hole or slot was milled prior to mounting, you will need to mount a dial indicator in the spindle and adjust for run-out less than .001", to locate the center of the hole.

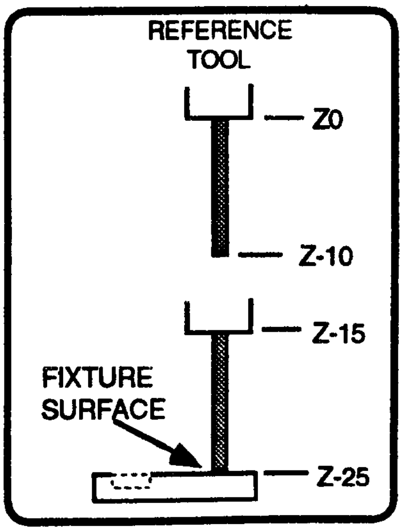
FIND SURFACE DIMENSION

Next you must find the Z coordinate of the fixture's surface. First, reference the machine, then insert the reference tool. Now jog the Z axis until the tip of the reference tool touches the fixture's surface. The machine coordinate of the fixture's surface is determined by subtracting the length of the reference tool from the Z axis Position Display register. In the example **below**, a 10.0" reference tool is used to determine this coordinate. The X, Y, Z coordinates of the point on the fixture will be referred to as I, J, K in the G0 probing calibrate cycle.

In this example, the spindle nose position at reference is Z000.0000, and the spindle nose position when the reference tool touches the fixture is Z -15.0000.

Calculate the Z coordinate of the fixture's surface as follows:

Z -15.0000 Spindle nose position
-Z +10.0000 Reference tool length
Z -25.0000 Machine coordinate of fixture's surface



FIND PROBE'S LENGTH OFFSET

The next step is to find the probe's tool length offset.

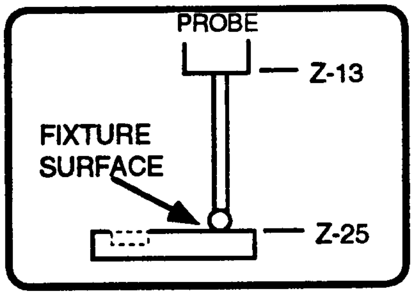
First reference the machine and put the probe in the spindle. For accurate probing the center of the probe's stylus ball should be aligned with the spindle centerline. Use a dial indicator to check run-out. Run-out must be within .005" for probing accuracy. Values as low as .001" can typically be achieved.[1]

To avoid damaging the probe during this operation, make sure that Probe Protect is on by triggering the spindle probe with your hand. If a PROBE PROTECT fault occurs, press RESET, and continue. If no error occurs, turn Probe Protect on and repeat this step.

Slowly jog the probe onto the fixture's surface. A PROBE CONTACT fault will halt all motion when the probe contacts the fixture. Clear the fault condition by pressing RESET. You can now calculate the probe's length by subtracting the fixture's Z surface dimension from the Z axis Position display register coordinate.

Z -13.0000 Position Display
-Z -25.0000 Surface dimension

Z +12.0000 Length of Probe



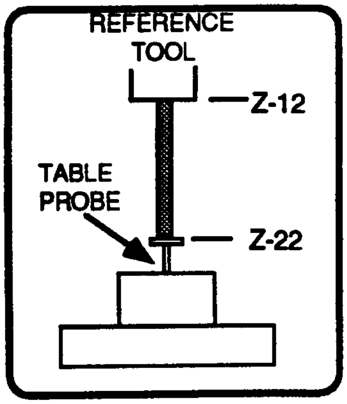
Store the probe's length offset in the Tool Tables. Always enter zero for the probe's diameter. This offset must be active whenever you are using the spindle probe to locate machine coordinates. As an exception, **T00[2] and E00 must be the active codes during the G0-Probe Calibrations cycle.**

TABLE PROBE - INITIAL SETUP

The table probe is used to automatically find and store tool length offset values. The X,Y center and Z axis machine coordinates of the datum cube are entered in the Set Up mode.

The table probe is typically mounted at a location on the table where removal will not be necessary. If it will be necessary to periodically remove and replace the table probe, locator pins should be used to ensure exact replacement.

To determine the table probe's machine coordinates, reference the machine and put the reference tool in the spindle. Now jog the reference tool onto the probe until a PROBE CONTACT fault occurs. Subtract the length of the reference tool from the Z axis Position Display to find the Z machine coordinate. Record the XY Position Display coordinates. Finally move to the Set Up mode and enter the XYZ machine coordinates for the table probe.



Z -12.0000 Spindle nose position
-Z +10.0000 Reference tool length

Z -22.0000 Machine coordinate of table probe cube

[1] If stylus centering is necessary, see probe manufacturers literature for adjustments.

[2] If your control uses H and D codes (refer to Part 1-Section 2) you must activate H00 and D00 when performing this operation. Since the T code identifies the tool in the spindle, it may remain non-zero.

PROBE EVENT DATA

To program a Probe cycle, key in G followed by the appropriate number that identifies the cycle and press ENTER. The options that program the selected cycle will appear on the screen. As described later, all Probing cycles must program A, L, F, and V.

To ensure highest accuracy and repeatability, the control feeds along only one axis (X, Y, or Z) when a probe cycle is looking for a probe contact. Position is stored when the active probe makes contact. Since position data can be recorded virtually instantaneously, feedrate is not important. However, it is usually wise to use the same feedrate (50 ipm for example) for all probing cycles (including calibration cycles).

G CYCLE ENTRIES

Parameter	Description	G Cycle Entries	
G0	Calibrate Spindle Probe	X,Y,Z,	I,J,K
G1	Surface Probing	X,Y,Z,	P,E
G2	Center Probing	X,Y,Z,	I,J,E
G3	Cutter diameter Adjust	X,Y,Z,	T[3]
G4	Tool Length Adjust	X,Y,Z,	T[3]
G8	Calibrate Table Probe		R
G9	Tool Length Set	X,Y,	R,T[3]

Parameter	Description	Entry
N	Sequence Number	Required
G	G cycle (G0-G4, G8,G9)	Required
X,Y,Z	Axis Datum or Distance	See G cycle description
I,J,K	Axis Secondary Datum	See G cycle description
P	Surface Program coordinate	See G cycle description
E	Fixture offset Number	See G cycle description
H	H Table length offset number	See G cycle description
D	D Table diameter offset number	See G cycle description
T	Tool Number	See G cycle description
R	Tool Length	See G cycle description

ENTRIES COMMON TO ALL PROBING CYCLES

A	Approach distance[4]	Near target distance from contact point
L	Limit distance[4]	Far target distance from contact point
F	Initial feedrate[4]	Probe feedrate until reaching point A
V	Final feedrate[4]	Probe feedrate after reaching point A

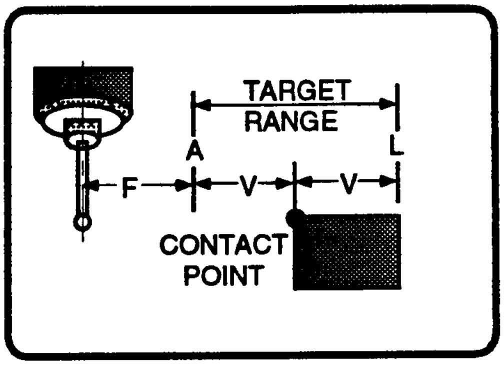
[3] If your control uses H and D codes (refer to Part 1-Section 2) this letter will be replaced with the letter D when a diameter value is stored; it will be replaced by the letter H when a tool length is stored.

[4] A, L, F, and/or V values from the Calibration cycle will be used in any Probing event that does not program these values.

EXPLANATION OF APPROACH SEQUENCE and TARGET RANGE

A[5], L, F and V are used in all probing cycles. If not entered, the values that were programmed in the calibration cycles will be used. These entries define the target range where you expect a probe contact to occur. A is the "near" distance prior to the expected contact point; L is the "far" distance after the expected contact point. Both are entered as unsigned, absolute values. The approach sequence for all probing cycles is defined below.

The diagram, **right**, shows an example of the spindle probe executing a probing cycle[6]. The spindle slows from feed F to feed V when the probe center passes approach point A. Any contact that occurs before approach point A causes a PROBE CONTACT fault and halts all motion. If probe contact is made within the target range, the contact position is stored by the active probing cycle. The probe then retreats to A at feedrate V. If no contact is made by the time the probe center reaches limit L, a Feed Hold occurs and a NO PROBE CONTACT fault is displayed.



RECOVERY

If a fault occurs, go to the FAULT/STATUS display and check the fault message[7].
Press RESET to remove the fault message, edit the cycle, and repeat the operation.

SPINDLE PROBE CYCLES

G0 - CALIBRATE SPINDLE PROBE

This cycle brings the probe into contact with the known surface positions of a test fixture. The test fixture is described earlier in this section. The slot or hole is a ring gauge. Its dimensions or diameter should be known as precisely as possible. During the following XY Datuming motions, the control accurately computes and stores data to compensate for probe deflection and diameter for the ±X, ±Y and +Z axes. This data is used during all following probing cycles. The probe needs to be calibrated at least once a day. The interval that serves your accuracy requirement is best determined with continued use. **T00[2] and E00 must be the active codes when this cycle executes.**

ENTRIES

X - Width of slot or diameter of hole.
Y - Length of slot or diameter of hole.
Z - Depth for XY datuming. (Machine coordinates)

I - X Coordinate for Z datuming. (Machine coordinates)
J - Y Coordinate for Z datuming. (Machine coordinates)
K - Z Contact point. (Machine coordinates)

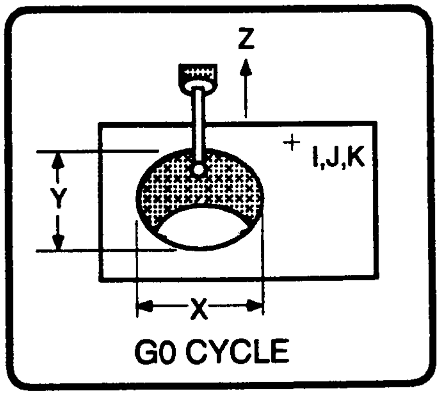
[5] The A entry must greater than the radius of the probe's stylus ball.

[6] Probing cycles cannot be executed with Rotation, Scaling, or Mirror Imaging active.

[7] If the message is PROBE CONTACT, you will need to jog the probe away from the surface before proceeding.

PROGRAMMING

- The probe calibration fixture must be mounted on the machine and its coordinates must be known prior to executing this event. See the Initial Setup topic.
- Program an M Function event to activate the T00 Code, and deactivate any E Codes with E00.
- Jog or execute a Position event to move the probe over the exact hole center.
- Program and execute the G0 probe event.



G0 CYCLE

CYCLE OPERATION

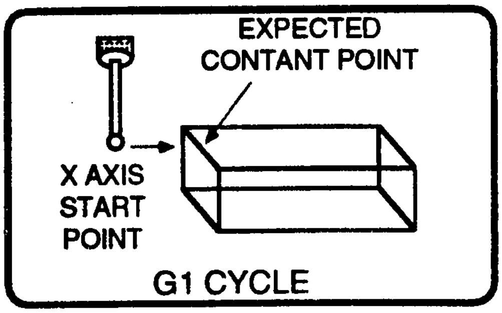
The probe feeds to Z and approaches the +X target range to determine the +X probe compensation value. The probe then approaches the -X target range to determine the -X probe compensation value. After feeding to the X starting position, the probe repeats the sequence for the Y axis. With the XY axis compensation values computed and stored, the probe feeds in +Z to the Z starting position. The probe then feeds to I, J and approaches the -Z target range to determine the -Z compensation value. To complete the cycle, the probe feeds to the start Z then XY position.

G1 - SURFACE PROBING

This cycle uses the spindle probe to accurately find the position of a fixture or part. Once the position is found, the control automatically stores its coordinate in the specific fixture offset (E Code). If you want the probed point to be part coordinate zero when the E Code executes, enter P0 in this cycle. You may also assign a different part coordinate to the probed point point by entering a nonzero P value. Later execution of the E Code will eliminate the need for a ZERO SET operation.

ENTRIES

X -	Expected contact point. (Machine coordinates)
Y -	Expected contact point. (Machine coordinates)
Z -	Expected contact point. (Machine coordinates)
P -	Desired part coordinate of programmed axis.
E -	Fixture offset to use.



G1 CYCLE

PROGRAMMING

- Program an M Function event to activate the T Code for the probe, and deactivate any E Codes with E00.
- Jog or execute a Position event to move the probe to the start point. This start point locates the coordinates for the two axis which are not being probed.
- Program and execute the G1[8] probe event.
- Repeat steps 2 and 3 for other axis desired.

CYCLE OPERATION

From the start point, the probe moves towards the expected contact point. When the surface is detected, the control automatically stores its coordinate in the specific fixture offset (E Code). After making contact, the probe returns to the start point.

[8] The G1 cycle will calculate the fixture offset values for all three axis, one axis per event. If more than one axis is programmed in an event, a "PROGRAMMING ERR" fault will be displayed.

G2 - CENTER PROBING

This cycle finds the XY axis position of a fixture or part by locating the center of a hole or slot on the part. After the probe locates the center of the hole or slot, you can choose to store the coordinates in the Fixture Offset table. If you want this point to be X0, Y0 when the E Code executes, enter I0, J0 in this cycle. You may also assign a different part coordinate to this point by entering nonzero I and J values. Later execution of the E Code will eliminate the need for a Zero Set operation.

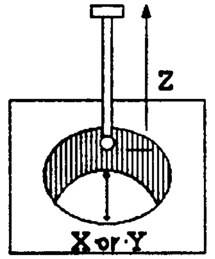
If you only want to find the center without storing the values, leave I, J, and E non-programmed; the probe's XY position at the end of this event will be the center of the hole or slot.

ENTRIES

X - Radius of the hole, or half the width of the slot.
Y - Radius of the hole, or half the length of the slot.
Z - Depth for probing.

I - Desired X axis part coordinate of the center.
J - Desired Y axis part coordinate of the center.

E - Fixture offset to use.[9]



G2 CYCLE

PROGRAMMING

- Program an M Function event with a T Code to activate offsets for the probe, and deactivate any E Codes with E00.
- Jog or execute a Position event to move the probe to a start point above the expected center of the hole.
- Program and execute the G2 probe event.

CYCLE OPERATION

The probe feeds to Z depth, and approaches first the +X and then the -X target range, to find the center of the hole or slot. The control uses the actual hole center, and the part coordinate you programmed as I, to calculate and store the X axis fixture offset value. The probe then feeds to the center of the hole and repeats the sequence for the Y axis. After a return to the hole center, the probe rapids back to the Z axis starting point.[10]

G3 - CUTTER DIAMETER ADJUST

This cycle measures the X or Y surface of a machined part. If a difference exists between the actual X or Y dimension and the programmed dimension, the control automatically adjusts the stored diameter (the diameter is stored in the Tool Tables mode) for the tool you identify. Proper dimensions will be machined the next time you run the part.

ENTRIES

X - Correct dimension of part. Program X or Y (not both) in part coordinates.
Y - Correct dimension of part. Program X or Y (not both) in part coordinates.
Z - Depth for probing cycle. (part coordinates)

T^3 - Tool number to be compensated.

[9] If E is programmed, I and/or J must be programmed.
[10] The Z axis position of the part or fixture can be found using a G1 cycle.

PROGRAMMING

- Program an M Function event with a T^3 Code to activate offsets for the probe.
- Jog or execute a Position event to move the probe to the start point.[11]
- Program and execute the G3 cycle.

CYCLE OPERATION

The probe feeds to Z, and approaches the programmed target range. When the part surface is detected, the control computes the difference between the actual part surface and the programmed dimension. The control then uses this value to correct the tool diameter for the T Code that you entered in this event. The probe feeds back to the start point, first in X or Y and then in Z, to complete the cycle.

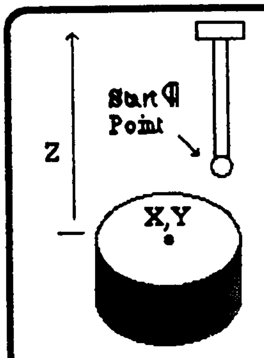
G4 - TOOL LENGTH ADJUST

This cycle uses the Z surface measurement of a machined part to adjust length offset for the tool that was used to mill the part. If a difference exists between the actual Z dimension and your programmed Z part coordinate, the control automatically adjusts the stored length offset for the tool you identify with T.

ENTRIES

X - X coordinate of surface to be measured. (Part coordinates)
Y - Y coordinate of surface to be measured. (Part coordinates)
Z - Correct Z dimension of part. (Part coordinates)

T^3 - Tool number to be compensated.



G4 CYCLE

PROGRAMMING

- Program an M Function event to activate the probe's T Code.
- Jog or execute a Position event to move the probe to the start point.
- Program and execute the G4 cycle.

CYCLE OPERATION

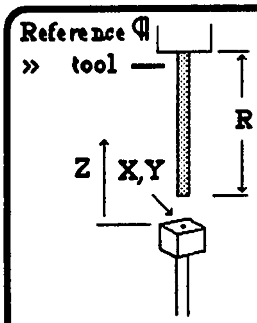
The probe feeds to X,Y, and approaches the -Z target range. When probe contact is made, the control computes the difference between the actual part surface and the programmed dimension. The control uses this value to correct the tool length offset for the T^3 code that you program. The probe then retreats in Z and rapids back to the X,Y start point.

[11] The start point must be chosen to provide an unobstructed path for the probe.

TABLE PROBE CYCLES

G8 - CALIBRATE TABLE PROBE

This cycle uses a reference tool of known length to establish and store the exact Z dimension of the table probe. This Z dimension will be used in the G9 cycle to set tool lengths. The probe needs to be calibrated at least once a day. The interval that serves your accuracy requirement is best determined with continued use.



G8 CYCLE

ENTRIES

R - Length of reference tool.

PROGRAMMING

- Mount the table probe on the machine. Complete the Table Probe Setup procedure described earlier in this section.
- Program an M Function event to activate the T^3 Code for the reference tool, and deactivate any E Codes with E00.
- Jog or execute a Position event to move the reference tool to the start point.[12]
- Program and execute the G8 probe cycle.

CYCLE OPERATION

The reference tool positions to the X,Y center of the table probe and approaches the -Z target range. When probe contact is made, the control will determine and store the -Z table probe compensation value. The reference tool then retreats to A and returns to the start point.

G9 - TOOL LENGTH SET

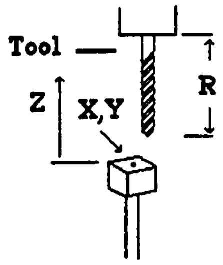
This cycle uses the table probe to establish and store the tool length offsets for the selected tool.

ENTRIES

R - Approximate length of tool to be calibrated.

X - X coordinate of the table probe.[13] (Machine coordinate)
Y - Y coordinate of the table probe.[13](Machine coordinate)

T^3 - Tool to be compensated.



G9 CYCLE

PROGRAMMING

- Jog or execute a Position event to move the tool to the start point.[12]
- Program and execute the G9 cycle.

CYCLE OPERATION

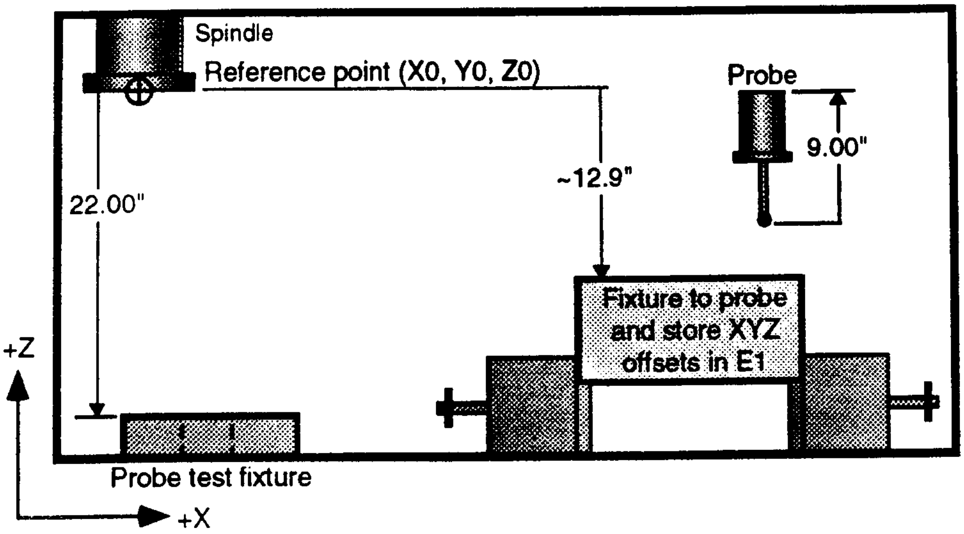
The tool positions to the X,Y center of the table probe and approaches the -Z target range. When probe contact is made, the control will determine and store the correct tool length offset in the Tool Tables mode. The tool will then position back to the start point.

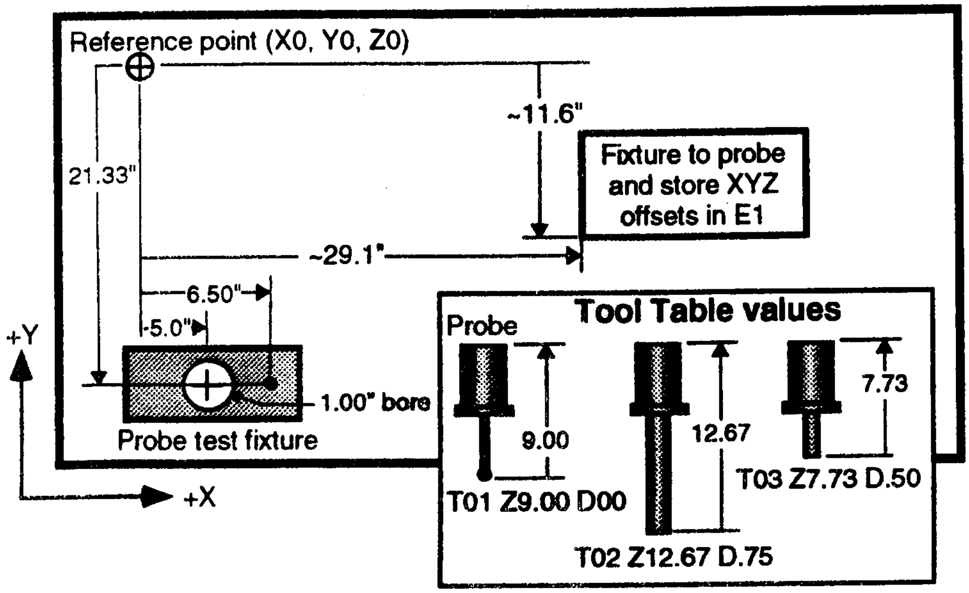
[12] A start point should be chosen to ensure that the machine coordinate of the tool tip is +Z in respect to the probe to provide clearance.

[13] If an E Code is not active, X,Y should not be programmed. The Set Up mode values will be used. If an E Code is active, then X,Y must be programmed to locate probe coordinates with respect to the E Code position.

EXAMPLE

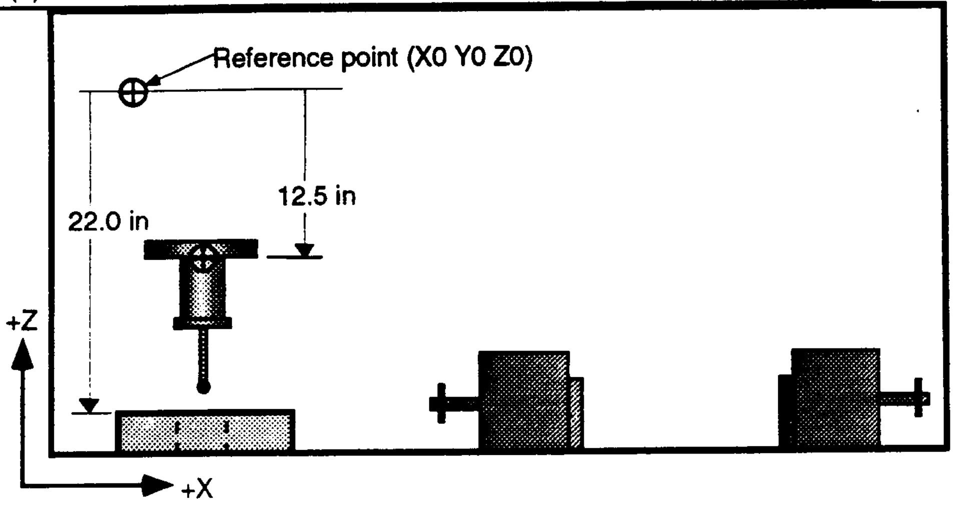
The following part programs use the machine envelope shown **below**. The important dimensions are distance from machine zero (X0, Y0, Z0) to the test fixture's center, and the distance from machine zero to the point on the fixture where the Z calibration occurs. The distance from machine zero to the fixture (E1) can be approximated since probe events will locate the exact coordinates and store them in the Fixture Offset table. The stored tool offsets for the probe and cutting tools are shown in the sidebar. Some minor adjustments will be required if you wish to run these examples on a control that uses H and D codes.


Spindle
Reference point (X0, Y0, Z0)
Probe
9.00"
~12.9"
22.00"
Fixture to probe and store XYZ offsets in E1
+Z
Probe test fixture
+X



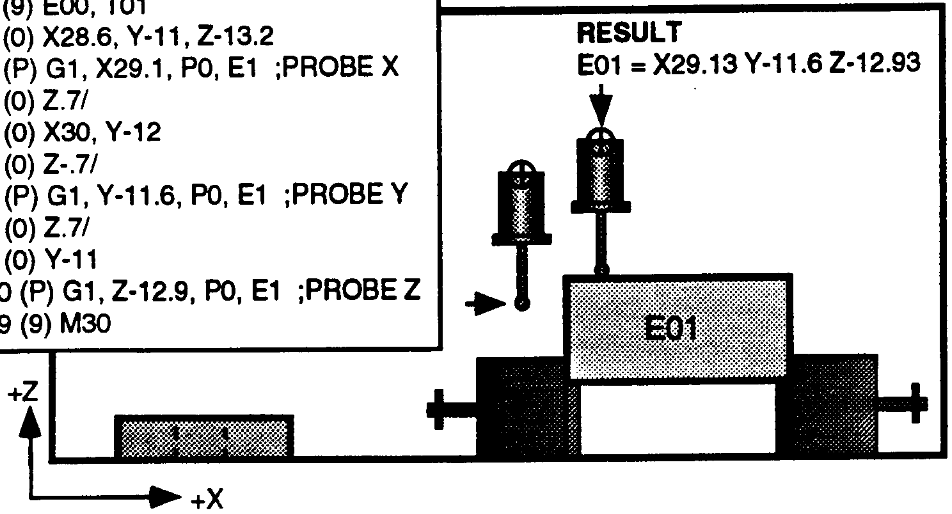
The part program below calibrates the spindle probe. The offsets found during N3 are stored by the control and used in all following Probe events to insure high accuracy.

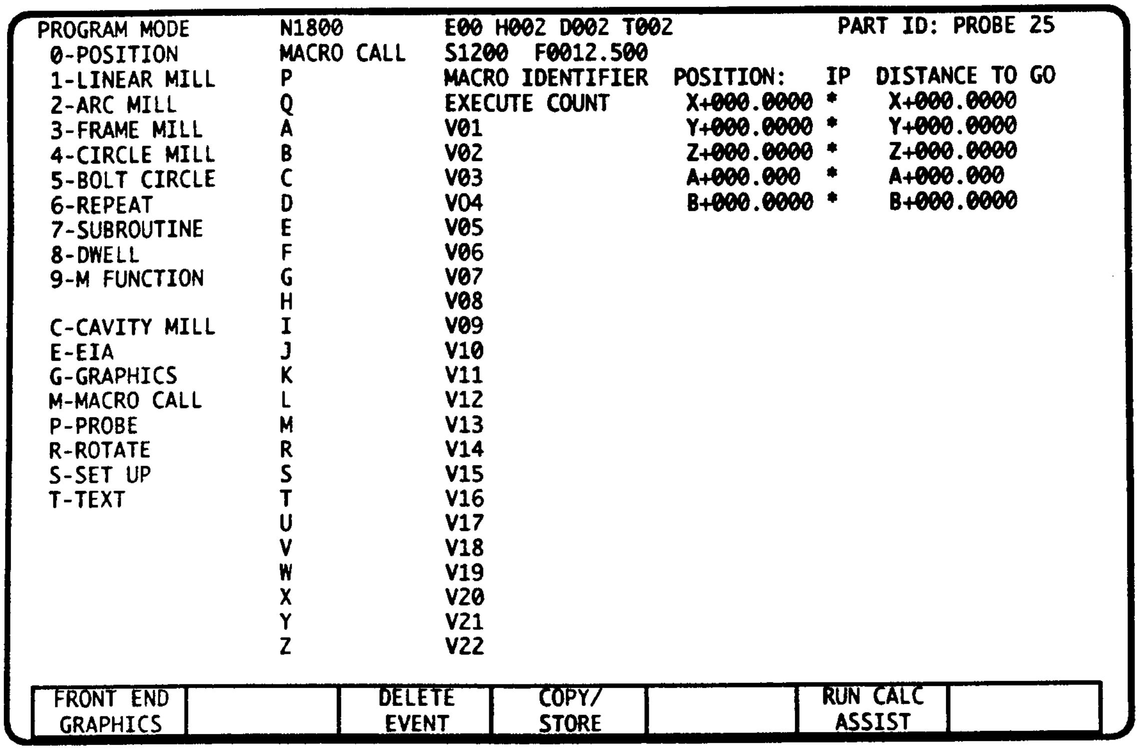
```
(PROBECAL)
N1 (9) E00, T00 ; T00, E00 must be the active offsets during calibration.
N2 (0) X5 Y-21.33 Z-12.5  ; position 1'"above test fixture.
N3 (P) G0 X1.0 Y1.0  Z-13.5 I6.5 J-21.33 K-13.0 L.2 F40 V20 A.2
N4 (9) M30
```



The program below will locate the X, Y, Z position of the fixture and its machine coordinates in E1 in the Fixture Offset table. Notice that the probe's length offset (T1) is active prior to executing any probing cycle.

```
FIND FIXTURE CYCLE
(FIND FIX)
N1 (9) E00, T01
N2 (0) X28.6, Y-11, Z-13.2
N3 (P) G1, X29.1, P0, E1  ;PROBE X
N4 (0) Z.7/
N5 (0) X30, Y-12
N6 (0) Z-.7/
N7 (P) G1, Y-11.6, P0, E1  ;PROBE Y
N8 (0) Z.7/
N9 (0) Y-11
N10 (P) G1, Z-12.9, P0, E1  ;PROBE Z
N99 (9) M30
```





```
PROGRAM MODE       N1800       E00 H002 D002 T002               PART ID: PROBE 25
 0-POSITION        MACRO CALL  S1200  F0012.500
 1-LINEAR MILL     P           MACRO IDENTIFIER  POSITION:    IP  DISTANCE TO GO
 2-ARC MILL        Q           EXECUTE COUNT      X+000.0000  *    X+000.0000
 3-FRAME MILL      A           V01                Y+000.0000  *    Y+000.0000
 4-CIRCLE MILL     B           V02                Z+000.0000  *    Z+000.0000
 5-BOLT CIRCLE     C           V03                A+000.000   *    A+000.000
 6-REPEAT          D           V04                B+000.0000  *    B+000.0000
 7-SUBROUTINE      E           V05
 8-DWELL           F           V06
 9-M FUNCTION      G           V07
                   H           V08
 C-CAVITY MILL     I           V09
 E-EIA             J           V10
 G-GRAPHICS        K           V11
 M-MACRO CALL      L           V12
 P-PROBE           M           V13
 R-ROTATE          R           V14
 S-SET UP          S           V15
 T-TEXT            T           V16
                   U           V17
                   V           V18
                   W           V19
                   X           V20
                   Y           V21
                   Z           V22
```

MACRO CALL EVENT INTRODUCTION

This event helps you avoids repetitive programming tasks by calling a program that resides in the Macro Catalog. These programs usually define cutting routines that are used in multiple part programs. Once called, the identified macro program will execute from start to finish, then return control back to the calling event. If you or your machine supplier have developed intelligent Customer Prepared Cycles (CPCs), you can enter up to 22 variables that are passed to the CPC. The CPC developer will identify which variables, if any, must be programmed.

For information on how to program macros, read the Macro Mode section of this manual. For information on how to design flexible Customer Prepared Cycles, read the Developing a CPC part of this manual.

P - MACRO IDENTIFIER

Key in the letter P, then up to eight characters to identify the Macro program that this event will command to begin execution. Once the Macro[1] and and other Macros it has called complete execution, the main program will resume operation with the event following this "calling" event.

Q - MACRO EXECUTION TIMES COUNTER

Key in Q, then the number of times (0 to 99) that you would like the called Macro to be executed before returning control to the calling program. If not entered, the default is to Q1. The Q0 entry is permitted, but results in zero executions of the cycle.

[1] Nesting permits up to four Macro calls to be active at any one time. The calls are operational in conjunction with the normal nesting limits for Subroutine and Repeat events. There is also "recursive" call checking, as a Macro is not allowed to call itself.

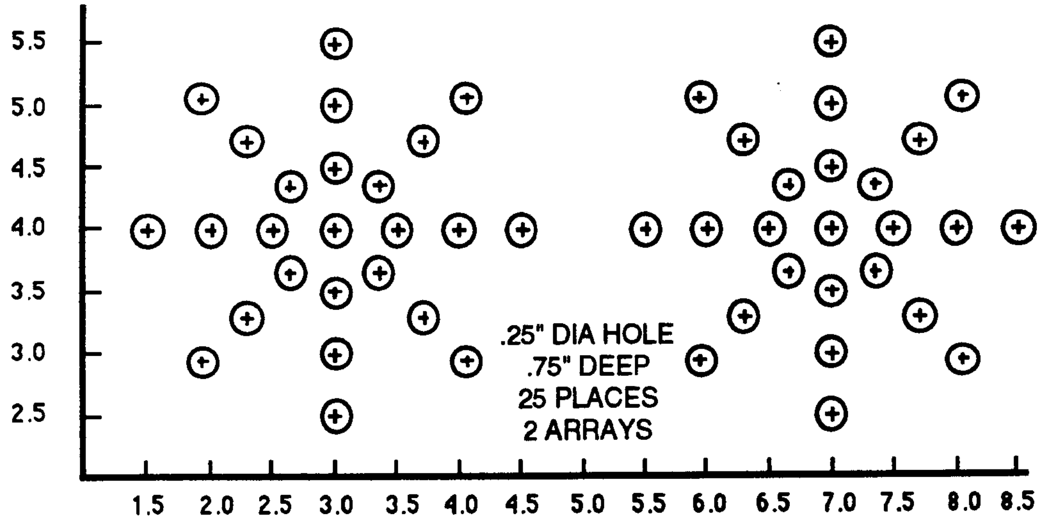
A through M and R through Z - CPC VARIABLES (in Macro Call)

You should not attempt to fill in these CPC variables unless you have specific CPCs that require variable information. The CPC developer will identify which variables, if any, must be programmed.

MACRO SAMPLE PART

The Macro Sample Part appears **below**, along with the code necessary to program both the main program and the macro. The macro RADIHOLE is similar to the Repeat event example program in the Repeat event section of this manual.

THE PART:



.25" DIA HOLE
.75" DEEP
25 PLACES
2 ARRAYS

MAIN PROGRAM:

Code	Comments
N10 (R) X3,Y4,C0	Set center for rotation
N20 (0) X3,Y4,Z.1,G0	Position to start
N30 (M) Q1,PRADIHOLE	Call Macro "RADIHOLE" (PRADIHOLE) and execute once (Q1).
N40 (R) X8,Y4,C0	Set center for rotation
N50 (0) X8,Y4,Z.1,G0	Position for second hole array
N60 (M) Q1,PRADIHOLE	Call Macro again for one execution.
N70 (9) M30	End of Program.

MACRO RADIHOLE:

Code	Comments
N36 (0) X0/,Z-1,G1,W.1,F5	Drill center hole.
N37 (6) X.5/,F360,T3	Move and repeat 3 times.
N38 (0) X-1.5/,G0	Return to center.
N39 (6) C45/,F370,T7	Rotate and repeat 7 times.
N40 (9) M30	End of Macro. Return to calling program.

CALC-ASSIST

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON

BASIC CALC-ASSIST INTRODUCTION

Calc-Assist is a feature that you can employ as an engine to calculate missing data in many Position, Linear Mill, and Arc Mill events. Any part program, or program segment, that contains lines and arcs which are not defined with complete Cartesian[1], or complete polar[2] coordinates can be run through the Calc-Assist engine. During processing, Calc-Assist will fill in all missing Cartesian and Polar data within the constraints of the programmed information. When two geometric solutions are possible, you can examine the developing part shape and numeric data, register your choice, then continue processing.

The Calc-Assist feature is divided into the following two sections:

Basic Calc Assist - This is referred to as one block Calc-Assist since each event must provide enough information (combination of Cartesian and polar data) for the control to be able to calculate a geometric solution (i.e. the event's endpoint). When Basic Calc-Assist cannot solve the active[3] event, the control will automatically begin using Advanced Calc-Assist logic.

Advanced Calc-Assist - In order to process an event that Basic Calc-Assist cannot solve, Advanced Calc-Assist will look to the following one or two events to obtain profile information that allows the feature to calculate the endpoint of the active[3] event. Once the active event is solved, Advanced Calc-Assist steps to the next event then passes control back to Basic Calc-Assist.

After processing a part in Calc-Assist, you should cycle the part in Graphics mode to verify the profile and tool offset path. It is important to note that the Front End graphic utility is provided to run Calc-Assist. It is not intended as a replacement for the Graphics mode. Since Front End Graphics does not recognize cutter compensation, Fixture Offsets, Part Offsets, axes rotation, etc., all of your final proofing operations should occur in the Graphics mode.

TOPICS

Section 1 describes how to set up and run Calc-Assist. This section also describes how Basic Calc-Assist works. Section 2 describes how Advanced Calc-Assist works. You should be familiar with the terms and operations that are described in the Basic Calc-Assist section before you read about Advanced Calc-Assist. Each section also provides a summary sheet that outlines the specific cases that Calc-Assist can solve. For BCA, each case lists the required entries for the line or arc. For ACA, each case lists the required entries for the two or three event sequence.

WHAT YOU NEED TO KNOW

CONVERSATIONAL PROGRAMMING

You should be familiar with Conversational modes and events. If not, you should review the sections that describe Catalog, Program, and Graphics modes, and the M Function, Position, Linear Mill, and Arc Mill events.

[1] Cartesian data for a line includes its endpoint coordinates (X,Y,Z). Cartesian data for an arc includes its center coordinates (I,J,K) and endpoint coordinates (X,Y,Z).

[2] Polar data for a line includes its angle V and length D. Polar data for a arc includes its start angle A, end angle B, and radius R.

[3] The Active event is a Position, Linear or Arc Mill event in your designated range that Basic or Advanced Calc-Assist is attempting to solve.

TERMS

CA, BCA, ACA - abbreviated references to Calc-Assist, Basic Calc-Assist, and Advanced Calc-Assist, respectively.

Element, Line, Arc - An "Element" is a general reference to a line or arc. A "Line" refers to a Position or Linear Mill event. An "Arc" refers to an Arc Mill event.

OPERATING CALC-ASSIST

The 11 steps you will follow to operate Calc-Assist are listed here and then repeated as you step through a sample part shape. For learning purposes, we'll ask you to run in All-Axis inhibit so an accidental CYCLE START won't cause any motion. For this trial, you should work on a copy of your program so you can compare the CA output to your initial part data.

OVERVIEW

This is the general sequence you would follow to use Calc-Assist:

1) **Move to Set Up mode. Choose TM3 (All-Axis Inhibit). Later you can treat this as an optional step.**

2) **Move to Catalog mode and enter your new part title.**

3) **Move to the Tool Table mode and store your tool information.**

4) **Move to Program mode, but DO NOT turn Calc-Assist on. Begin programming by filling in as much data as you can, including descriptive data for lines or arcs that you would normally need to triangulate.**

5) **Return to Catalog mode and make a copy of your program. You will run the copy. Later, you can also treat this as an optional step.**

6) **Set Up the Calc-Assist information in Program mode. This includes the plotting Plane Select, Origins, Size, First & Last event sequence and the A2 entry that turns Calc-Assist ON.**

7) **Move to the Program mode Front End Graphics Display.**

8) **Press the RUN CALC ASSIST key to start Calc-Assist.**

9) **Complete any Alternate Choice selections or program edits necessary to process the run.**

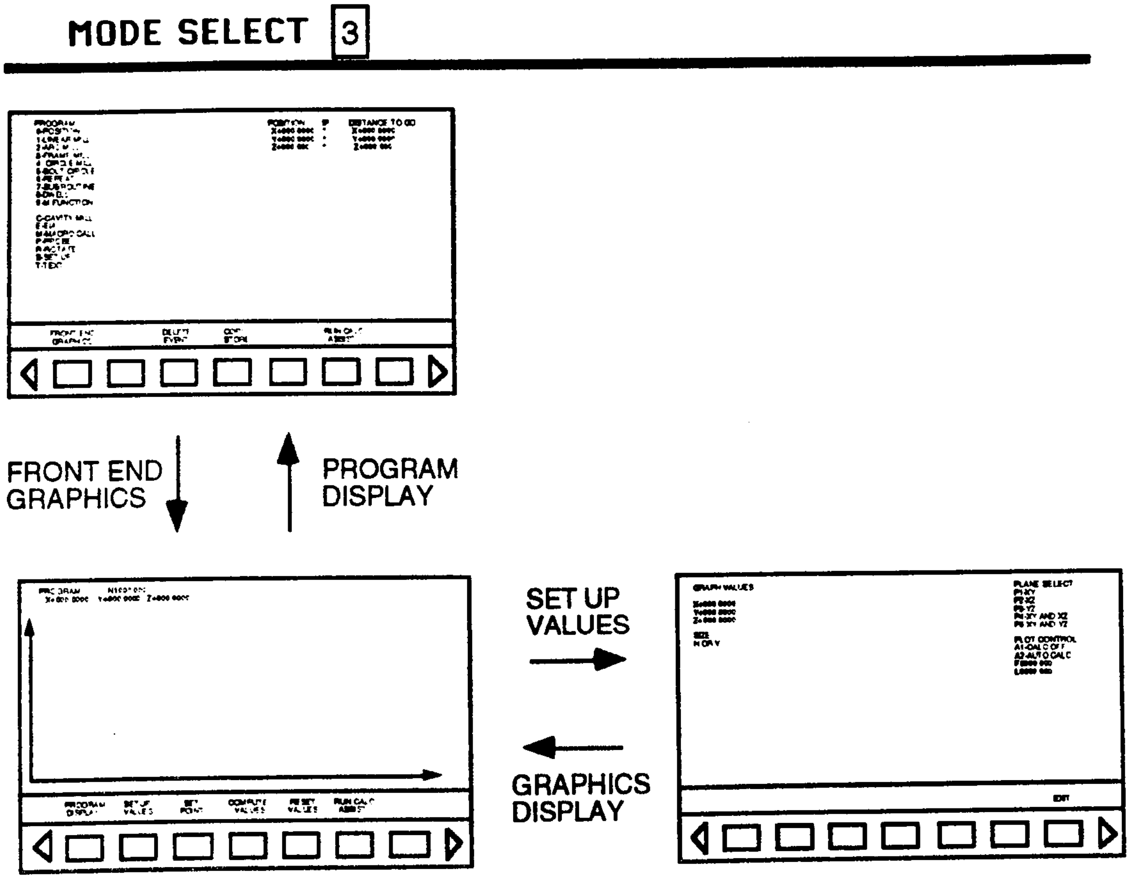
   You may want to find points that Calc-Assist triangulated after the run.

10) **After Calc-Assist has completed its run through your F through L range make a run in Graphics mode for profile verification.**

    Remember, Front End Graphics does not process cutter compensation, Fixture Offsets, Part Offsets, axes rotation, etc.

11) **If you chose TM3 in Set Up mode, go back and set TM1. Your program should now be ready for you to set the datum and then cycle in Auto mode.**

The following illustration shows how soft keys are used to move between the Program mode display screens.



To use Calc-Assist you must learn to operate the feature from its 3 primary screens.

The first is the Set Up Values screen. To view this screen, press the FRONT END GRAPHICS soft key, then the SET UP VALUES soft key. On this screen, select the Plane, Plot Origins, Size (H or V), and set your F, L Calc-Assist range[4] as you normally would in Graphics mode. Note that the A2 option here turns CA on. Leave it off for now.

Press GRAPHICS DISPLAY to return to the Front End Graphics display. The Front End Graphics screen will enable you to watch as CA plots your part while it computes and deposits data in the program. The SET POINT and COMPUTE VALUES soft keys permit you to visually size an existing plot with the jog switches[5].

The PROGRAM DISPLAY key moves you to the main Program mode screen for you to inspect and edit data when CA is halted.

[4] F,L identifies the events CA will run on. The first event (F) must be defined with absolute coordinates.

[5] You can SET POINTS and COMPUTE a new screen size for plotting in the same way as you would in Graphics mode.

A TRIAL CALC-ASSIST RUN

Follow this sequence to program the Basic Calc-Assist part shown **below**. If you are unsure of how to carry out any of the following operations (i.e. how to enter program data), see the appropriate topic in this manual. Begin as follows:

1) **Move to Set Up mode. Choose TM3 (All-Axis Inhibit). Later you can treat this as an optional step.**

   This will inhibit axis motion if you accidentally press the CYCLE START button.

2) **Move to Catalog mode and enter a name for the part.**

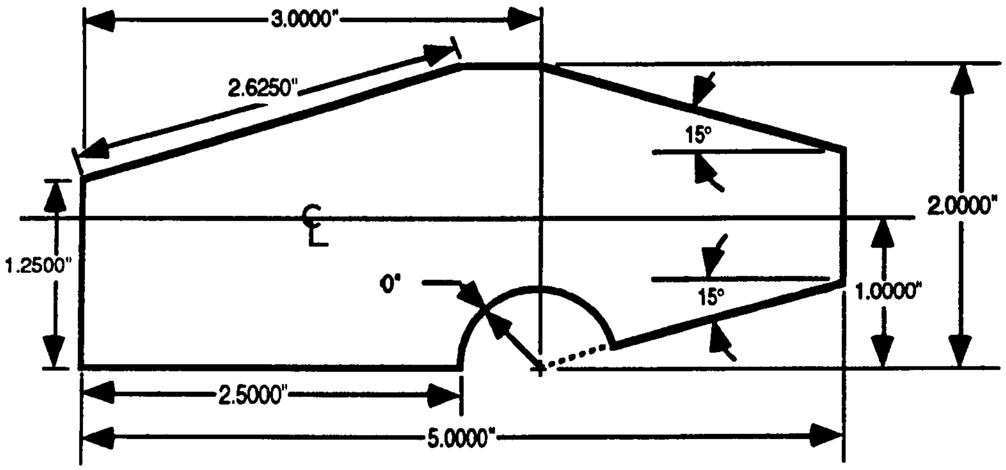
3) **Move to the Tool Table and enter your tool numbers and dimensions.**

   For the sample part select T03 and enter a .500" cutter diameter. If your control uses H and D codes (refer to Part 1-Section 2) you must enter this diameter into the D table.

4) **Move to Program mode, but DO NOT turn Calc-Assist on. Enter the following program data:**

PROGRAM DATA	COMMENT
N005 (G) X-1 Y-1 Z-1 H7 P1	Graphic setup values
N010 (9) T3 H3 D3 M03 S1800	
N020 (0) X0 Y-.75 Z-.6 G0	Position for entry
N030 (1) Y0/ C0 F10	Dummy move; comp off.
N040 (1) Y1.25 C1	Feed into part; comp on.
N050 (1) D2.625 Y2	endpoint X not known or needed: CA will use line length D and endpoint Y to triangulate.
N060 (1) X3	No motion in Y; no Y endpoint needed.
N070 (1) V345 X5	Line by angle and endpoint.
N080 (1) V270 Y1	Endpoint not known; use Y1 now and edit the event later.
N090 (1) V195 D__	Distance not known; edit the event later.
N100 (2) I3 J0 B180 D1	Program arc center and end angle.
N110 (1) X-.75	Final feed will exit stock.
N120 (1) X0/ C0	Dummy move; comp off.
N130 (9) M30	

Notice that at many points we did not program all the information that was available on the part print. It is best to leave out redundant information and let Calc-Assist fill it in to avoid errors.



5) **Return to Catalog mode and make a copy of your program to run.**

While this may later be treated as an optional step, it is a good practice to make a copy of the program before running it through CA. CA will store many values in the program and your original can later be reviewed for comparison.

6) **Set up the Calc-Assist information in Program mode.**

Move to the Program mode, press the FRONT END GRAPHICS soft key, then the SET UP VALUES soft key to move to the Set Up screen. You do not need to enter a graphic size and origin because the control will update these parameters when the Graphics Setup event executes. Enter the following:

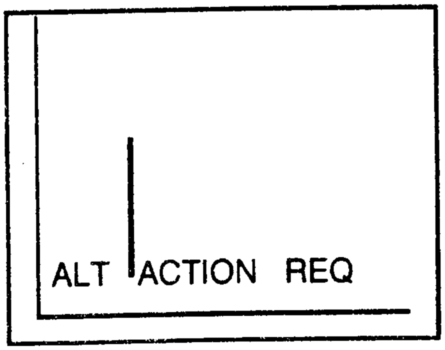
PLOT CONTROL:	F: set to 0 to start at N0. L: set to 9999.999 to end at N9999.999.
AUTO CALC:	A2 - Calc On

7) **Press** GRAPHICS DISPLAY **to return to the Program mode Front End Graphics Display.**

8) **Press the** RUN CALC ASSIST **soft key to start Calc-Assist.**

The program will run, then stop at N50 with an Alternate Choice message.

The graph will look like:

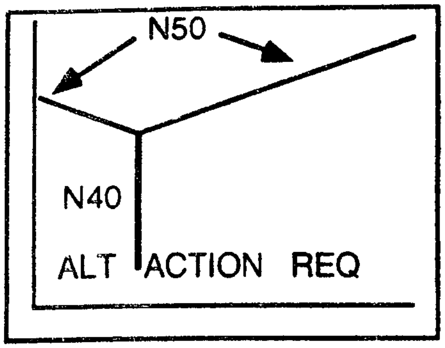


The code will look like:

```
N10  (6)  T3 H3 D3 M3 S1800
N20  (0)  X0 Y-.75 Z-.6 V270* D.75*
          [V and D will vary with the
           home position]
N30  (1)  X0* Y0/ V0* D0* C0 F10
N40  (1)  X0* Y1.25 V90* D2* C1
N50  (1)  Y2 D2.625
```

At N50 plotting halts and you are asked to choose between two alternate paths.

9) **Complete any Alternate Choice selections or program editing necessary to process the run.**



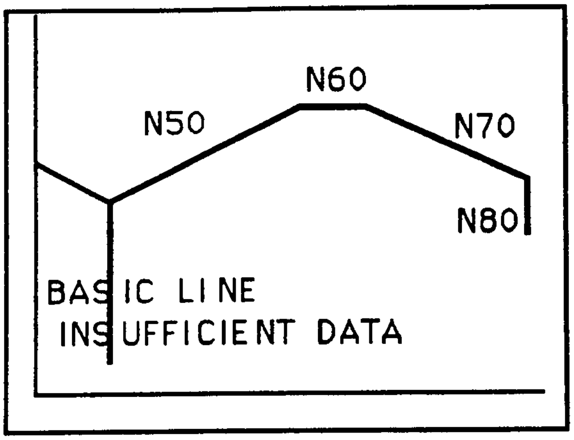
Pressing the ALTERNATE CHOICE soft key once will draw a half-intensity line to the left of the vertical line. Press ALTERNATE CHOICE again and the control will draw a similar line to the right. At this point, N50 contains:

X 2.5156*
Y 2.0000
V 16.601*
D 2.6500

This data describes the right line (the correct line), so press the STORE CHOICE soft key. When you do, the data that Calc-Assist triangulated (X and V) will be stored as hard data with the asterisks removed.

Now press the RUN soft key. Calc-Assist will plot N60 through N80, where it will stop and give an error.

The graph will look like:



BASIC LINE
INSUFFICIENT DATA

The code will look like:

```
N60  (1)  X3, Y2*, V0*, D.4844*
N70  (1)  X5, Y1.4641*, V345, D2.0706*
N80  (1)  X5*, Y1, V270, D.4641*
N90  (1)  V20
```

This is exactly what we wanted to happen. Notice in N80 that Calc-Assist has computed a D value of .4641 for a feed down to Y1, the centerline of the part. if we double this we'll get the D value for a feed down to the following line at N90. Doubling .4641 gives us .9282, which is the proper D length at N80 for this part. Now press CLEAR, move to the Program Display screen and program D.9282 in N80. Also enter Y CLEAR.

This will allow the control to execute N80, but it still doesn't have enough information to complete the line at N90-N100. It needs the line length D. From the part print we can see that the lines at N70 and N90 are identical except the N90 line is .5" shorter because of the intersecting arc. CA computed a 2.0705" D length at N70 for us. Therefore 2.0706 - .5 = 1.5706". Enter D1.5706 at N90.

This is all the information Calc-Assist needs to plot the rest of the part. Move to the Graphics Display, press CLEAR and RESET, and press the RUN CALC ASSIST soft key to plot the completed part.

10) Make a run in Graphics mode to verify part profile.

Enter Graphics mode. Here you have the option of viewing either the tool center path or the part profile. Select T1 - TOOL CENTER and return to the Graphics screen. CLEAR, RESET, and press CYCLE START. When the graph is done select T2 - PART PROFILE. Move back to the Graphics display and press RESET but **DO NOT PRESS CLEAR.** Press CYCLE START. The resulting graph will overlay a part profile plot onto the tool center plot.

11) If you chose TM3 from Set Up mode, go back and set TM1. Your program should now be ready to cycle in Auto Mode.

This completes the sequence.

PROGRAMMING WITH BASIC CALC-ASSIST

Here are some rules for operating Basic (and Advanced) Calc-Assist.

1) **CA runs only in Program mode**

Calc-Assist is enabled and run only in Program mode. Once A2 - AUTO CALC is chosen from the Program mode Graphics Setup menu, Calc-Assist is enabled and will remain enabled until A1 - CALC OFF is chosen from the same menu. However, enabling Calc-Assist isn't the same as running it. Calc-Assist doesn't run until you press the RUN CALC ASSIST soft key in Program mode. The Calc-Assist enable does not alter a program that cycles in Auto, Single Event or Graphics mode.

2) **Skip-Through Events and Calc-Assist**

Calc-Assist only triangulates data for Conversational event types 0, 1 and 2, as detailed in the following "Basic Calc-Assist Cases" topic. There are many events, however, that Calc-Assist will interrogate without altering data. These are called skip-through events. CA uses data from them to determine the starting point for following position, Linear or Arc Mill events. If you adjust an entry in any of these events, run Calc-Assist to recompute entries in all the following events. Skip-through events are listed in the following table.

Event	**Calc-Assist use**
Cam Mill	This is an Arc Mill event with an end radius (E) programmed. When E is programmed, Calc-Assist will not triangulate for missing data in the event. However, CA will use the endpoint of a completely defined cam mill as the start point for a following event.
Frame Mill, Circle Mill, Bolt Circle, Cavity Mill	Calc-Assist will find the endpoint of any of these events for use as the start point of the next event.
Repeat	The endpoint will be adjusted for a Repeat event in which a step or rotate value is programmed. This endpoint is the start point of the next event.
EIA/ISO	Even though an EIA/ISO event may define a position, linear or arc motion, Calc-Assist will not triangulate data in EIA/ISO format. CA will treat a completely defined EIA block as a skip-through event and use its endpoint as the start point for the next event.
G6 Position	Calc-Assist recognizes the change in coordinates and uses the G6 position as the start point for the next event.

3) Null Events and Calc-Assist

The remaining Conversational event types are considered as "null" events because Calc-Assist will not adjust a following start point for any these events. While your plot will reflect rotation, for example, rotated coordinates will not be stored or influence the next start point, as rotation will occur at run time.

Event	Description
Subroutine	Calc-Assist does not act on the Subroutine event itself, but it will compute and store values for Position, Linear and Arc events contained within the subroutine's F, L range. The endpoint of the last event in the F, L range is the next start point.
Dwell	Calc-Assist ignores dwell.
M Function	Calc-Assist ignores all but M30. Calc-Assist will not compute beyond M30 in an F, L range.
Rotate	Rotation will be plotted. However, CA never stores rotated coordinates in your program.
Text	Calc-Assist ignores text.
Set Up (scale)	The endpoint of events following the Set Up event are scaled as specified. However, CA never stores scaled data in your program, or adjusts a following start point for scaling.
Set Up (part offset), E table fixture offset	Calc-Assist ignores both. Offsets do not take effect until executed at run time or during Graphics mode simulation. This avoids the need to recompute when you edit an offset.

4) Redundant Data

When programming a part with Calc-Assist you can often figure several ways to describe a feed. For instance, you may know that a feed is to X3, Y3, that it is 2 inches in length and is directed at a 90° angle in the programmed plane. In these cases it is best to enter the Cartesian endpoint data and avoid entering redundant data.

When you cycle a part in Graphics, Auto, or Single Event modes, the control only uses polar data when the Cartesian data for an event is incomplete. This means that you should always favor Cartesian coordinates when both Cartesian and polar coordinates are available. You should avoid redundant information because minor variances between Cartesian and Polar entries (as, for example, an angular variation of .001 degree) can flag an INCONSISTENT DATA message that would not occur if you did not enter the redundant data.

Calc-Assist will compute and enter a value for every blank dimension in the event. It will append an asterisk (*) to computed values unless they are stored as the result of an ALTERNATE CHOICE entry.

Alternate choice entries are stored as hard values (i.e. stored without an asterisk) so that you are not forced to repeat the same alternate choice selection when you re-execute the F, L sequence). Values with appended asterisks can be recalculated by Calc-Assist, as might occur when you adjust endpoint data in an earlier event, but Calc-Assist will **never** attempt to change keyboard or Alternate Choice stored values.

5) **Planes**

All examples presented in this appendix are programmed in the XY (P0) plane where X is the primary axis. CA will also operate in the YZ (P1) plane with Y as the primary axis or the ZX (P2) plane with Z as the primary axis.

The SETUP VALUES screen Plane Select entry does not limit Calc-Assist operation but merely chooses the drawing plane. Instead, the CA plane is determined by the P entry in each linear or arc event. The data to be triangulated must be consistent with the event's plane selection.

```
N212 (0) P0, X0, Y0, Z0            N212 (0) P0, X0, Y0, Z0 V0*, D0*
N214 (1) P0, V135, Y1, Z-1         N214 (1) P0, V135, Y1, Z-1,  X-1*, D1.4142*
```

For instance, Calc-Assist would take the program segment **above left**, which describes a diagonal feed in both planes, and triangulate the values marked with asterisks in the segment **above right** because triangulation occurred in, and was limited to the XY plane.

When programming multiple planes in a program that will be run through Calc-Assist, it is possible for a Calc-Assist contour of the part to be accurate, but a Cutter Radius compensated offset milling path that is plotted in Graphics mode may reveal improper look-ahead joining or may even result in a run-time error. For this reason, do not change planes in a compensated milling sequence.

6) **F/L Range**

You can control the portion of your program that runs through Calc-Assist by going to the Program mode's Front End Graphics Display and pressing the SETUP VALUES soft key. The entries F and L under PLOT CONTROL specify the First through Last N numbers that will be run through Calc-Assist. F and L need not be actual events, but the first event CA encounters must be an event describing a fully-defined position. In other words, there should be no need for Calc-Assist to triangulate any data for the first event it encounters; it has to have a concrete place to start. All events within the F through L range will be executed. To conserve memory and time it is usually a good idea to restrict the F through L range so that only the program segment that requires Calc-Assist computation is run.

7) **Removal of Polar key**

Calc-Assist's event formats eliminate the need for a POLAR key, a feature found on Conversational DynaPath controls predating the Delta series. Programs that were developed using earlier systems with the POLAR key can be loaded into Delta controls for keyboard edit and execution, but cannot be run through Calc-Assist

8) **Basic Calc-Assist Geometry**

Section 1 continues with a geometric case summary and graphic descriptions of the line and arc cases with their respective entries.

BASIC CALC-ASSIST CASES

(XY EXAMPLES SHOWN HERE, YZ OR ZX PLANE MAY BE SUBSTITUTED.)

POSITION OR LINEAR MILL

		Calc-Assist Finds :		
	Given	**Cartesian**	**Polar**	
Routine Solutions				
Case 1	X,Y	-	V,D	
Case 2	X	-	V,D	
Case 3	Y	-	V,D	
Case 4	V,D	X,Y	-	
Case 5	Z	-	-	(Out of Plane)
Convenient Solutions				
Case 6	X,V	Y,	D	
Case 7	Y,V	X,	D	
Case 8	X,D	Y,	V	Alt. Choice
Case 9	Y,D	X,	V	Alt. Choice

ARC MILL

Routine Solutions				
Case 1	X,Y,I,J	-	A,B,R	
Case 2	A,B,R	X,Y,I,J	-	
Convenient Solutions				
Case 3	X,Y,I	J,	A,B,R	
Case 4	X,Y,J	I,	A,B,R	
Case 5	X,Y, A,R	I,J,	B	
Case 6	X,Y, A	I,J,	B,R	
Case 7	X,Y, B	I,J,	A,R	
Case 8	X,Y, R	I,J,	A,B	Alt. Choice
Case 9	X,I,J	Y,	A,B,R	Alt. Choice
Case 10	X,A,B	Y,I,J,	R	
Case 11	X,A,R	Y,I,J,	B	Alt. Choice
Case 12	Y,I,J	X,	A,B,R	Alt. Choice
Case 13	Y,A,B	X,I,J		
Case 14	Y,A,R	X,I,J,	B	Alt. Choice
Case 15	I,J,B	X,Y,	A,R	

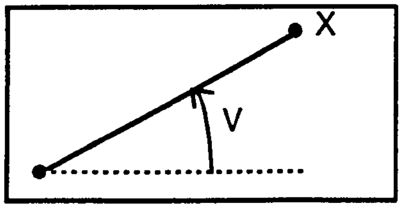
Note: The angle V entry in Linear Mill events is programmed with the letter "J" if your control is equipped with an auxiliary axis that is programmed with the letter "V". Refer to Part 1-Section 2.

ROUTINE LINE SOLUTIONS.

In cases 1 - 5, your data completely defines the event and the CA generated error checking data may seem redundant. In Cases 6-9, useful data can be obtained.

CONVENIENT LINE SOLUTIONS

CASE 6



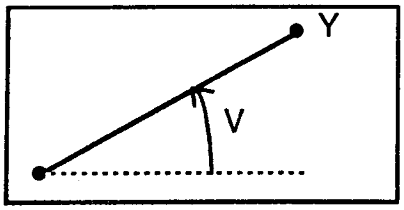
Given X,V

Solves For Y and D

Possible error codes:
Angle Does not agree with Axis Direction.

Suspected entries: X,V
The direction to X disagrees with the angle.

CASE 7



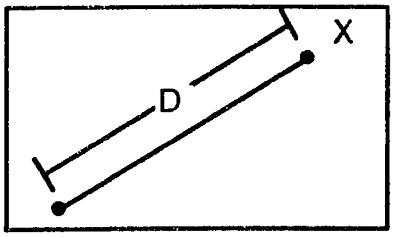
Given Y,V

Solves For X and D

Possible error codes:
Angle does not agree with Axis Direction

Suspected entries: Y,V
The direction to Y disagrees with the angle.

CASE 8



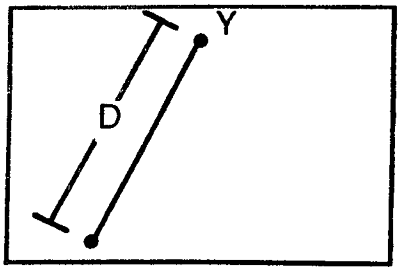
Given X,D

Solves For Y and V Alt. Choice

Possible error codes:
Axis move greater than Dimension

Suspected entries: X,D
The distance to X is greater than D.

CASE 9



Given Y,D

Solves For X, and V

Possible error codes:
Axis move greater than Dimension

Suspected entries: Y,D
The distance to Y is greater than D.

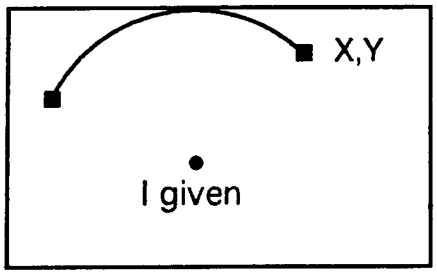
ROUTINE ARC SOLUTIONS

In cases 1 & 2, your entered data completely defines an event and the CA generated error checking data may seem redundant.

CONVENIENT ARC SOLUTIONS

In all Arc Mill events you must program counterclockwise or clockwise direction. If you do not program either direction the control will choose clockwise direction for you. In the following cases CA will provide useful data.

CASE 3

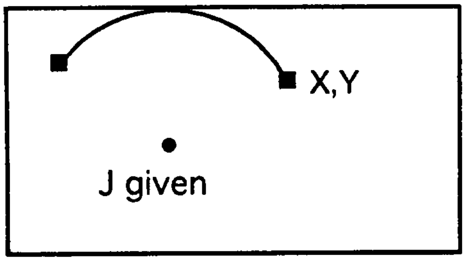


Given	Solves For
X,Y,I	J, A,B,R

Possible error codes:
Insufficient Data

Occurs with the same Y value at start and endpoint. Program R if Y start point = Y endpoint.

CASE 4

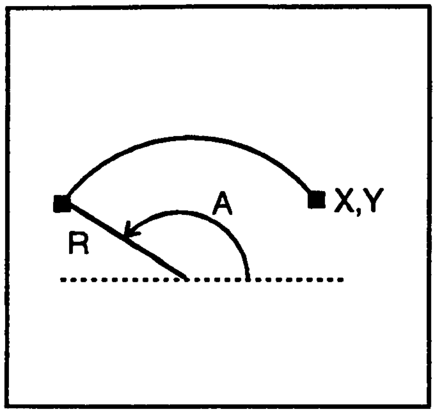


Given	Solves For
X,Y,J	I, A,B,R

Possible error codes:
Insufficient Data

Occurs with the same X value at start and endpoint. Program R if X start point = X endpoint.

CASE 5

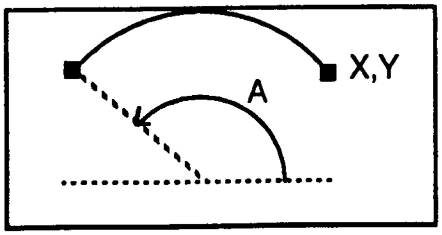


Given	Solves For
X,Y, A,R	I,J, B

Possible error codes:
Prog. and End Radii not Same

Suspected entries: X,Y,R,A or the start point. Calc-Assist found the center by working over distance R along an angle opposite to A from the starting position. A different R distance was found from this derived center to the programmed X,Y endpoint. X,Y,R or A could be wrong, but also check too see if the start point is where you thought it should be by looking at the last event's endpoint coordinates.

CASE 6

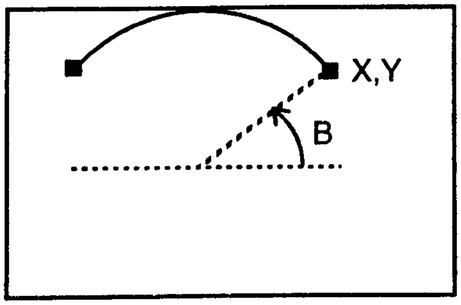


Given	Solves For
X,Y, A	I,J, B,R

Possible error codes:
Insufficient Data

Suspected entries: X,Y
Insufficient Data

CASE 7



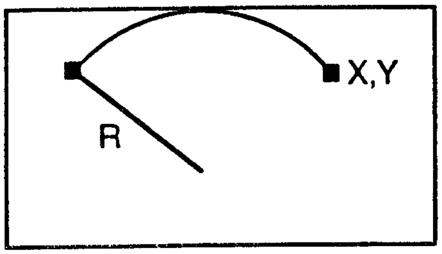
Given	Solves For
X,Y, B	I,J, A,R

Possible error codes:
Insufficient Data

Suspected entries: X,Y

If the arc is a 360 degree arc, as in case 6, you must also program radius R.

CASE 8

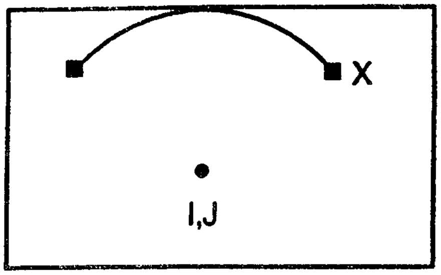


Given	Solves For	
X,Y, R	I,J, A,B	Alt. Choice

Possible error codes:
Radius does not Reach End

Suspected entries: X,Y,R
The straight line distance from the start point to X,Y is greater than twice the R radius value.

CASE 9



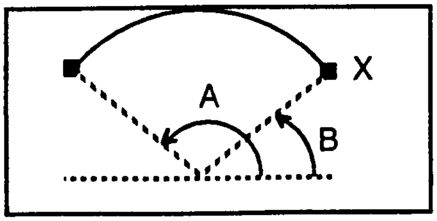
Given	Solves For	
X,I,J	Y, A,B,R	Alt. Choice

Possible error codes:
Start Radius < > End Radius

Suspected entries: X (or the starting position)

Using the starting position's distance from I,J, an arc could not be generated that strikes the programmed X endpoint.

CASE 10



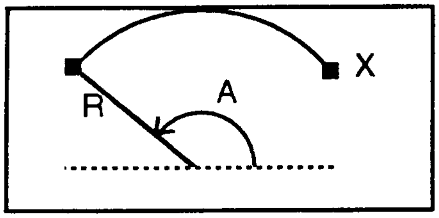
Given X,A,B

Solves For Y,I,J ,R

Possible error codes:
Insufficient Data

Suspected entries: A,B
When A = B (360 degree arc) add R radius.

CASE 11



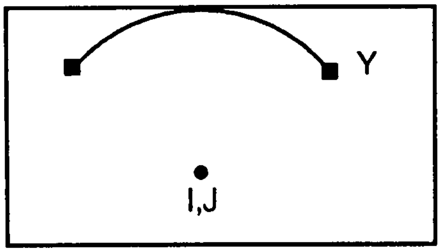
Given X,A,R

Solves For Y,I,J, B Alt. Choice

Possible error codes:
Radius does not Reach End

Suspected entries: X,A,R
The distance from the start point around the A,R center to X is further than R radius would reach.

CASE 12



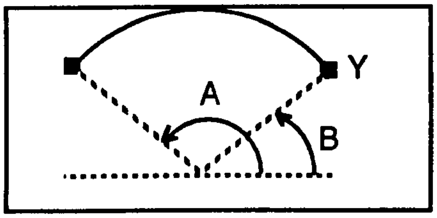
Given Y,I,J

Solves For X, A,B,R Alt. Choice

Possible error codes:
Start Radius < > End Radius

Suspected entries: Y
Using the starting position's distance from I,J, an arc could not be generated that strikes the programmed Y endpoint.

CASE 13



Given Y,A,B

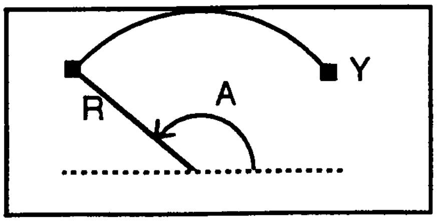
Solves For X,I,J, and R

Possible error codes:
Insufficient Data

Suspected entries: Y
If you will have the same Y coordinate, start and endpoint, also program radius R.

CASE 14

Given	Solves For
Y,A, R	X,I,J, B Alt. Choice

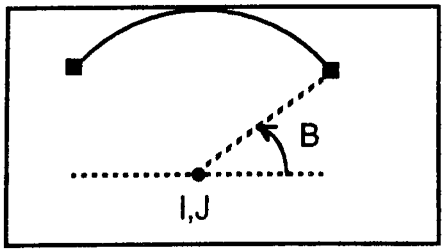


Possible error codes:
Radius does not reach End

Suspected entries: Y,R
The distance from the start point around the A,R center to Y is further than R radius would reach

CASE 15

Given	Solves For
I,J, B	X,Y, A,R



Possible error codes:
None

This event will always produce an arc motion unless I,J is also the start point.

BASIC CALC-ASSIST ERROR MESSAGES

1. **Angle Does Not agree with Axis Direction of motion.**
   See line cases 6,7

2. **Axis motion would be greater than D length.**
   See line cases 8,9

3. **Inconsistent Axis**
   X or Y Endpoint, from presumed start point, does not agree with angle/distance pair.
   Improper data/too much data entered, see line cases for required entries.

4. **Inconsistent Dimension**
   Length D is too long or short (from actual start point) for X, Y
   Improper data/too much data entered, see line cases for required entries.

5 **Inconsistent Angle**
The angle from actual start point, does not agree with X, Y
Improper data/too much data entered, see line cases for required entries.

6 **Insufficient Data**
At least one specified entry is missing. This is applicable to all cases.

7. **Prog and end Radii not same**
   See arc case #5

8. **Radius does not Reach End**
   See cases 8, 11, 14

9. **Start Radius<> End Radius**
   See arc cases 9, 12

10. **Plane Change**
    Arc mill programmed without out of plane motion.

11. **Inconsistent Center**
    Improper data/too much data entered, see your arc case for required entries.

12. **Prog and Start Radius not same**
    Improper data/too much data entered, see applicable arc case for required entries.

13. **Inconsistent Data**
    Improper data/too much data entered, but cannot be isolated at an angle or dimension. See applicable case for required entries.

ADVANCED CALC-ASSIST INTRODUCTION

It is important that you review Section 1 before proceeding because Basic and Advanced CA share many operating principles.

Advanced Calc-Assist is called into operation when Basic Calc-Assist cannot find a solution with information contained within the event. It does this by using data from the following event to construct an endpoint or arc center for the active event, then solving for the missing data in the active event. If unsuccessful, ACA looks to the second following event and uses it to construct an endpoint for the first event.

Once an event has been solved by either two or three event ACA, ACA steps to the next event and passes control back to BCA.

ADVANCED CALC-ASSIST SPECIFICS

While BCA and ACA work on the same general principles, Advanced CA adds the following rules.

EMPTY EVENTS

ACA allows you to leave some events incomplete or blank. It is then up to Calc-Assist to fill them in, using information that it finds in the next event or two. However, CA only fills in information according to what is available elsewhere in the program; it can't arbitrarily place a line or an arc. This requires you to understand the different ways in which Calc-Assist views lines and arcs.

FIXED LINES AND ARCS

These are events to be joined to; they give ACA a reference point on which it can anchor other elements.

Fixed lines must have a fixed start or end point and some direction.

Fixed arcs must have a direction, fixed center and a specific radius.

FLOATING LINES and ARCS

These are joining events; they are minimally defined, if at all. Calc-Assist can size or orient a joining event (based on information given in the surrounding events) to make it fit. Usually a floating event is the first event in a two event sequence or the second in a three event sequence.

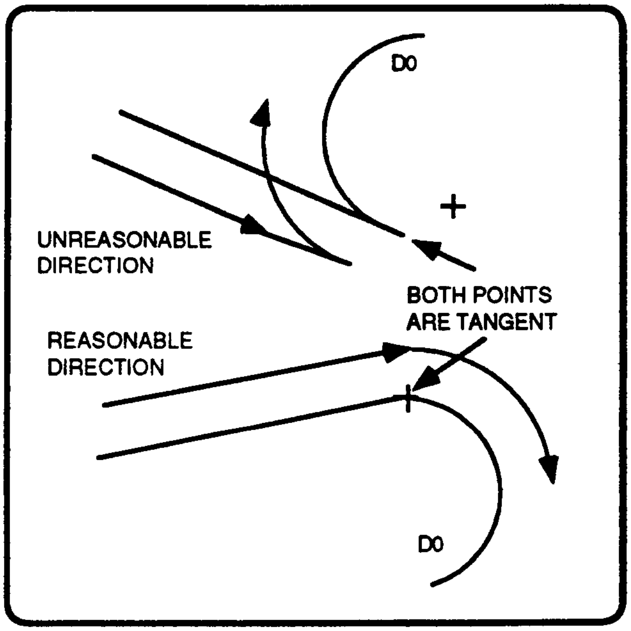
Floating lines are empty events. To create an empty event, select EVENT TYPE 1 (for an empty Linear Mill) and press the COPY/STORE soft key without entering data. Empty Linear events are typically used in Line to Line, Line to Arc, and Arc to Line to Arc joins. Lines can also be defined as empty Position events.

Floating arcs require that radius and direction be programmed. They appear as the second event in most three-event sequences. A floating arc is programmed with its radius R and direction D.

REASONABLE DIRECTION

When a **fixed** arc is joined to a line, it can join at one of two tangent points (see illustration). ACA uses **Reasonable Direction** logic to choose the point of tangency that will give tool motion without abrupt axis reversal. This intelligent selection relieves you from the burden of manually processing multiple Alternate Choices.

If you need a join like the upper one, program D1 for the arc (in this case) to find out the coordinates of the tangency. Then rekey the points as stored (rather than CA-computed) entries and reverse direction to D0.



REDUNDANT DATA

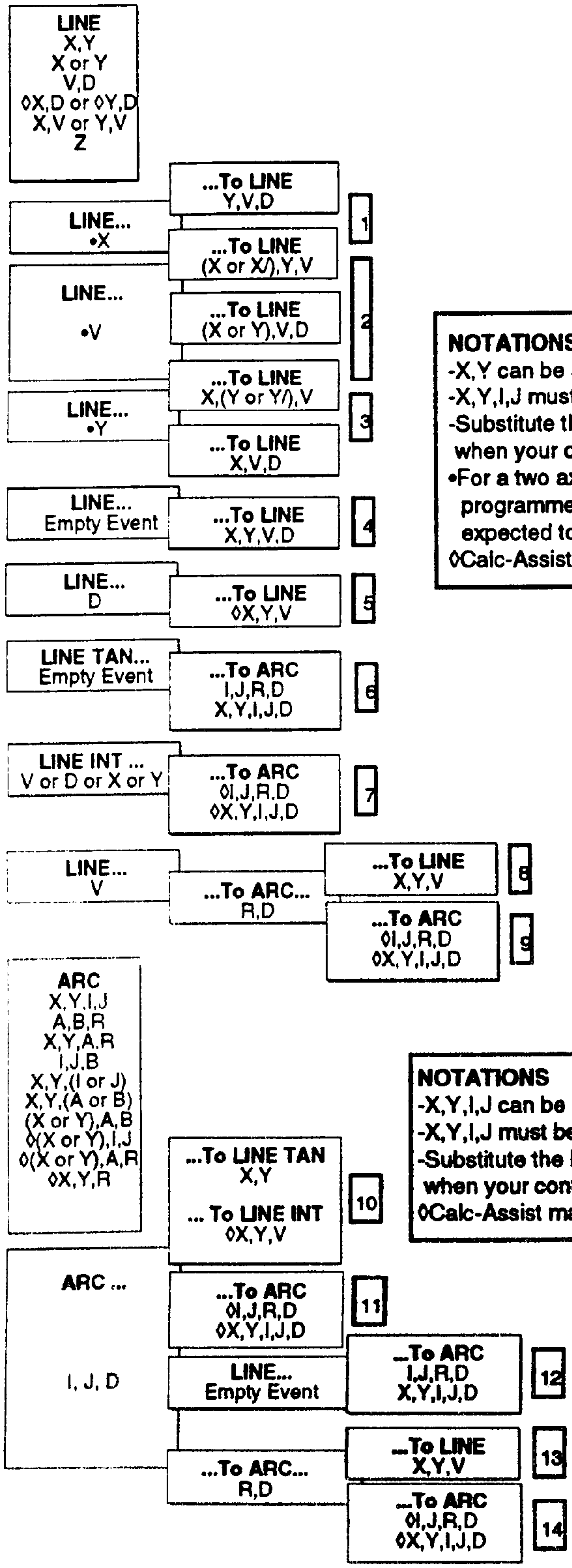
As with BCA, keyboard entry of redundant data should be avoided.

However, remember that ACA often uses information from one event to solve for data in another. This means that occasionally you will be required to program an event in a two or three event sequence more thoroughly than a normal or BCA defined event, so CA will have access to the information it needs to complete the sequence.

For example, a simple line to line join can be programmed by storing the first element as a linear motion without additional data. In this case you would program endpoints, angle and length for the second line. On encountering the empty event, Advanced Calc-Assist would take over and look to the next event. There it would find an absolute endpoint; using this fixed endpoint and working back, by reversing angle V it could find the second line's start point, which is also the endpoint of the first line. These coordinates are stored in the empty event. It is important that you understand why data that seems "redundant" (angle, length) in the second event is necessary for CA to solve the sequence. Provide only the required entries.

CALC-ASSIST REFERENCE CHART

Use this chart as a quick reference guide to Calc-Assist cases. Examples are shown in P0.



NOTATIONS

-X,Y can be absolute or incremental in the first event.
-X,Y,I,J must be entered as shown in the 2nd and 3rd events.
-Substitute the letter "J" with the letter "V" in a Linear Mill event when your control has an auxiliary axis that is labelled "V".
•For a two axis linear move, where only one endpoint is programmed, enter D0 to identify that the other axis is expected to move.
◊Calc-Assist may provide two solutions (Alternate Choice).

NOTATIONS

-X,Y,I,J can be absolute or incremental in the first event.
-X,Y,I,J must be absolute in the second and third events.
-Substitute the letter "J" with the letter "V" in a Linear Mill event when your control has an auxiliary axis that is labelled "V".
◊Calc-Assist may provide two solutions (Alternate Choice).

TWO EVENT CALC-ASSIST CASES

Remember, by looking ahead one event, Calc-Assist is trying to solve for missing information in the first event. The second event must not only contain data to describe its own endpoint, it must contain additional data to solve for missing information in the previous event.

LINE TO LINE

Since the first event in a line to line join will be minimally defined, the second will require V and/or D entries in addition to endpoints so that CA can work back from its endpoint and triangulate the endpoint of the first line.

General Rules:

1) ..In a linear event where both axes must move but only one axis endpoint is known, program D0 with X or Y. Without D0 the control would interpret the event as a move in one axis only.

2) ..If you program the angle in the first event, leave all other first event entries blank.

Examples:

Use this table as a key for the following examples.

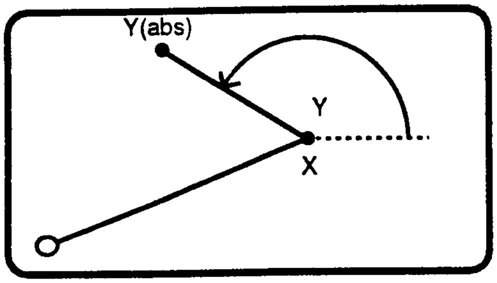
You Enter	must be programmed
CA Finds	will be triangulated and filled in by Calc-Assist
[ ] (brackets)	indicate a group of entries which must be programmed together
(abs)	indicates absolute coordinates are required
no (abs)	indicates either absolute or incremental are acceptable
*	indicates Alt Choice will be given
hollow circle	indicates start point of sequence of sequence
black dot	indicates joining point or end of part

Known values are labeled in the illustration. Other values that must be programmed can be found within bracketed groups along with the illustrated values.

The **CA Finds** column lists the vital Cartesian coordinates that will be calculated and inserted into your part program. Polar coordinates will also be inserted into the program, but are considered non-vital since the control will always use Cartesian coordinates (when available) when it is cycling a part in Auto, Single Event, or Graphics mode.

Note that under the **CA Finds** column, values are generally stored as absolute. If you prefer, your machine supplier can configure your control so that CA stores incremental entries. This would allow events containing CA-generated values to be called by a Subroutine or Repeat event.

LINE TO LINE



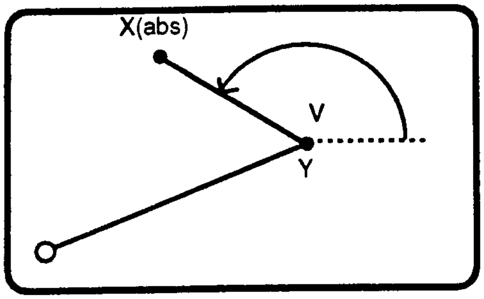
You Enter:

1st event:
X, D0

2nd event:
[Y(abs), V, X] or
[Y(abs), V, D]

CA Finds:

Y(abs)



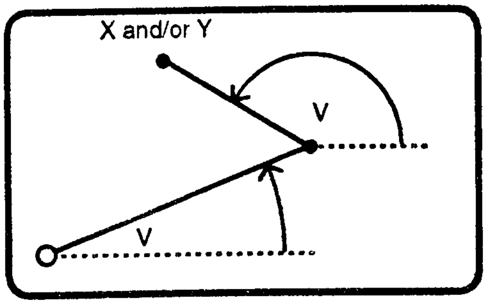
You Enter:

1st event:
Y, D0

2nd event:
[X(abs), V, Y] or
[X(abs), V, D]

CA Finds:

X(abs)



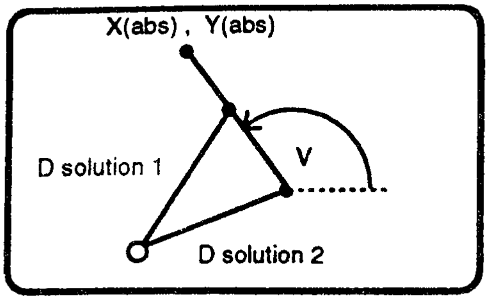
You Enter:

1st event:
V

2nd event:
[V, X(abs), Y] or
[V, Y(abs), X] or
[V, D, X] or
[V, D, Y]

CA Finds:

X(abs), Y(abs)



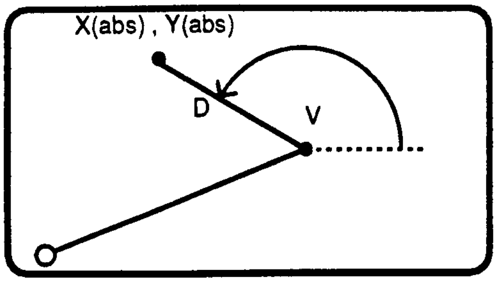
You Enter:

1st event:
D

2nd event:
X(abs), Y(abs), V

CA Finds:

*X(abs), Y(abs)



You Enter:	CA Finds:
1st event:	
Empty: event type 1; Copy/Store	X(abs), Y(abs)
2nd event:	
[X(abs), Y(abs), V, D]	

LINE TO ARC

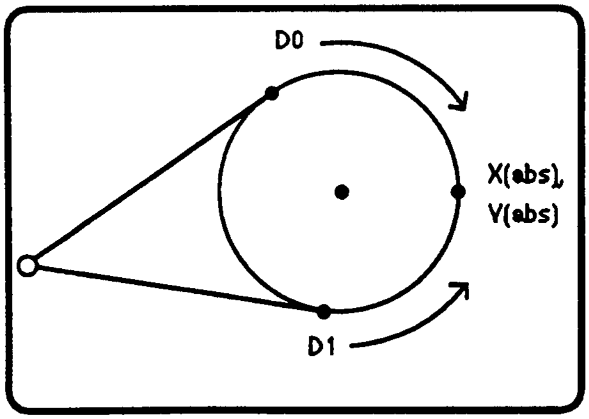
The line in a line to arc join may or may not be partially defined, depending on the type of join. The arc is a fixed arc, with endpoints and arc center or radius and arc center known. Arc Direction D should be programmed.

General Rules:

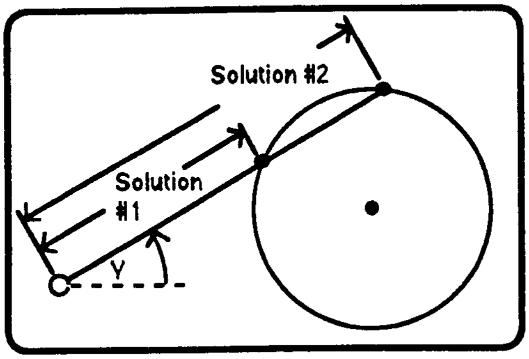
1) When you want a tangential join, use an empty line. ACA will use reasonable direction to choose the reasonable point of tangency.

2) When you want an intersection, program the line's length, angle or one endpoint. ACA will find two possible intersection points and give an Alternate Choice message.

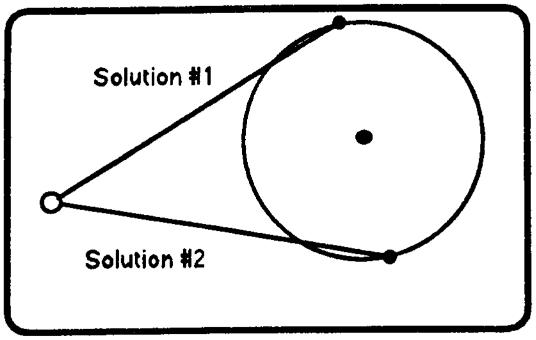
LINE TO ARC (Tangency)



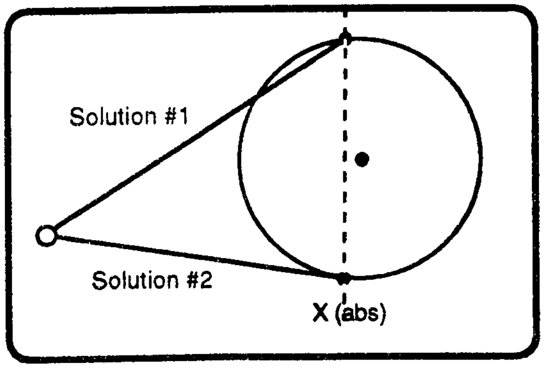
You Enter:	CA Finds:
1st event:	
Empty: event type 1; Copy/Store	X(abs), Y(abs)
2nd event:	
[I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	



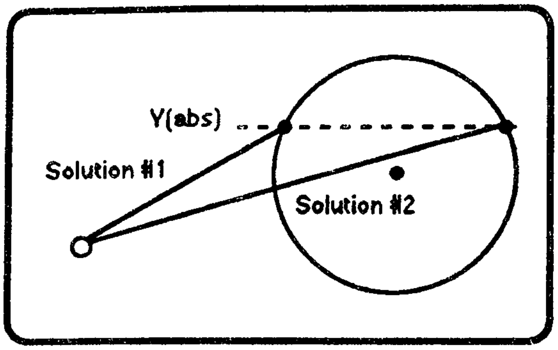
You Enter:	CA Finds:
1st event: V	*X(abs), Y(abs)
2nd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	



You Enter:	CA Finds:
1st event: D	*X(abs), Y(abs)
2nd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	



You Enter:	CA Finds:
1st event: X(abs)	*Y(abs)
2nd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	



You Enter:	CA Finds:
1st event: Y(abs)	*X(abs)
2nd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	

ARC TO LINE

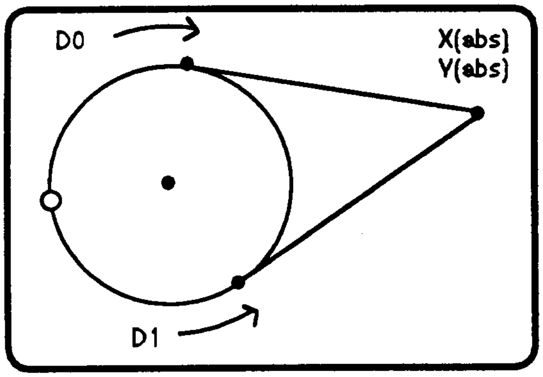
In an Arc to Line join,the arc's center and direction are fixed. ACA can determine its radius using the endpoint of the previous event. The line endpoint must be programmed so ACA can work back to find where the line joins the arc.

General Rules:

1) If the line is defined by its endpoint only, ACA will give a tangent join, using reasonable direction to choose the proper point of tangency.

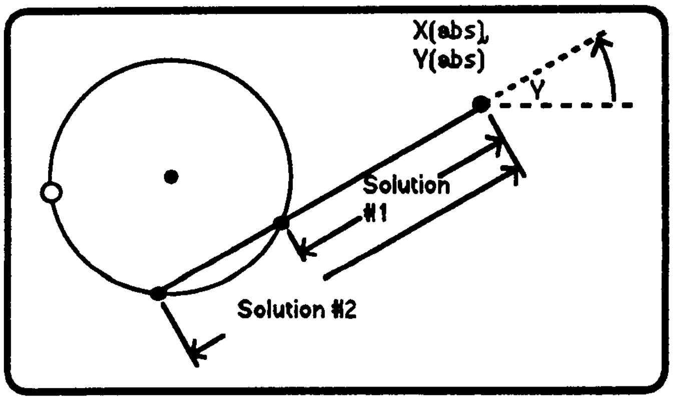
2) If the line is defined by its endpoint and angle V, ACA will give an intersection join and, if two intersection points are possible, an Alt Choice.

ARC-LINE (Tangency)



	You Enter:	**CA Finds:**
1st event:	[I, J, D]	X(abs), Y(abs)
2nd event:	[X(abs), Y(abs)]	

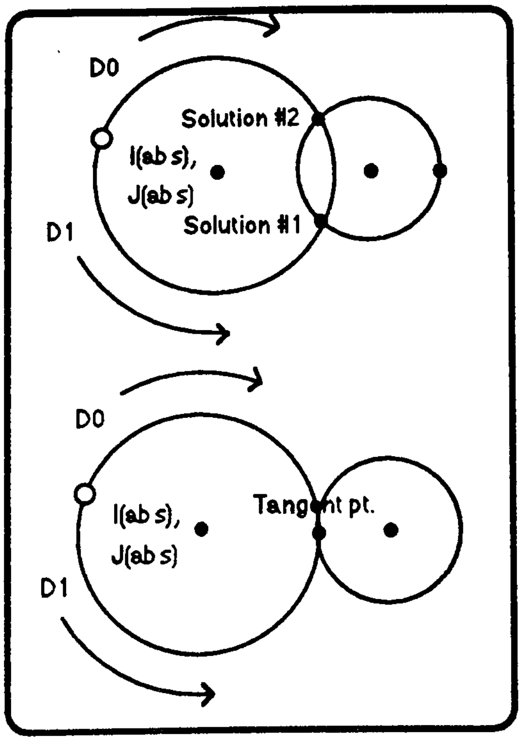
ARC-LINE (Intersect)



	You Enter:	**CA Finds:**
1st event:	[I, J, D] Y(abs)	*X(abs),
2nd event:	[X(abs), Y(abs), V]	

ARC TO ARC (Tangency and Intersection)

In an Arc to Arc join, the first arc's center and direction are fixed. ACA will use the start point to find its radius. In the second event program center, direction and radius so ACA can work back and find the two points where the arcs intersect. It then gives an Alt Choice. Since tangent arcs have only one possible point of intersection, no Alt Choice will be given.



You Enter:	**CA Finds:**
1st event: [I, J, D]	*X(abs), Y(abs)
2nd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	

THREE EVENT CALC-ASSIST

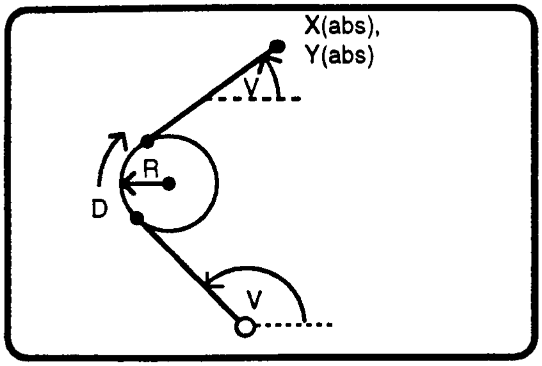
When an active event cannot be solved by examining the next event, Calc-Assist will look ahead two events. Here the third event must contain enough information (which again may be redundant within the event itself) so that ACA can work back, first computing an endpoint for the second event, then adjusting this event's start point to allow a smooth join with the first event.

Notice that in four of the five following cases the middle event is a floating arc defined only by its Radius and Direction. ACA can also locate endpoints for a line tangent to two fixed arcs.

LINE TO ARC TO LINE

In a line-arc-line join, program an angle for the first and third events. For the third event, also program the endpoints, and in the second event (which is a **floating** arc) enter the radius and direction.

When solving a line-arc-line join, ACA knows the angle of the first event and its start point (from the endpoint of the previous event). Similarly, it knows the endpoint of the third event and its angle, which it can reverse to find the direction from which the line came. ACA can vary the length of each line until it is tangent to the joining arc.



You Enter:	CA Finds:
1st event: V	X(abs), Y(abs)
2nd event: [R, D]	
3rd event: [X(abs), Y(abs), V]	

LINE TO ARC TO ARC

In a line-arc-arc join, the third event is a fixed arc and the second event is a floating arc, which ACA will locate. The first event is defined only by its vector angle.

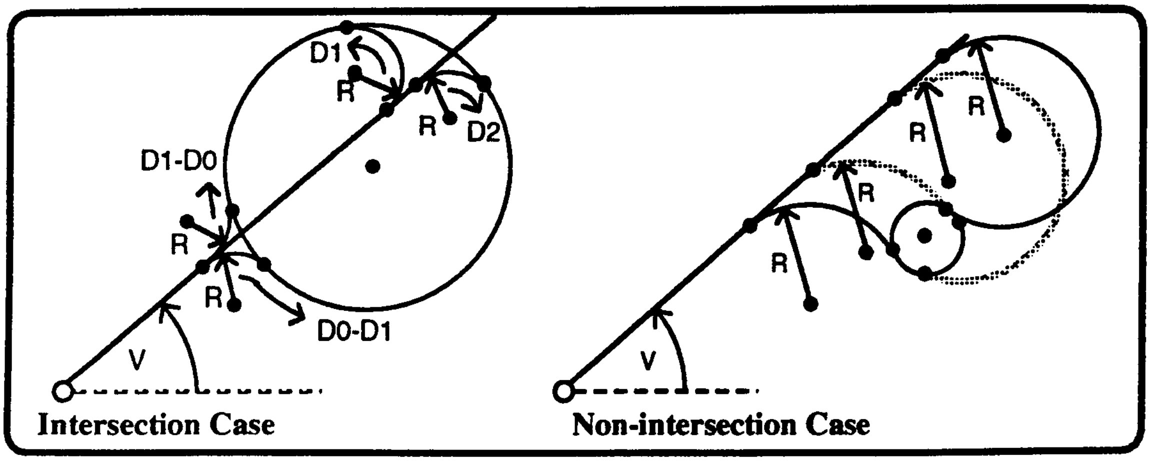
General Rules:

1) Arc direction must be programmed for both arcs. If the line in the first event intersects the arc in the third, ACA uses reasonable direction and gives no Alt Choice (see upper illustration). If the first and third events do not intersect, one of two possible Alternate Choice sets will be given (see lower illustration).

2) Opposite arc directions cause the floating arc to join tangent to the outside of the fixed arc. Similar directions cause an inside join.

You Enter:	CA Finds:
1st event: V	*X(abs), Y(abs)
2nd event: [R, D]	X,Y,I,J, A,B
3rd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	

Note:
The lower illustration shows the non-intersection case for a clockwise (D0) joining arc. In this case there are two sets of Alternate Choices: the gray arcs show the set given if the fixed arc is clockwise; the solid arcs show the set given if the fixed arc is counter-clockwise.



ARC TO ARC TO LINE

In an arc-arc-line join, ACA works back from a fixed line endpoint through a floating middle arc to find the endpoint of the first arc, which is defined only by center and direction. The middle (floating) arc is tangent to both the line and the first arc.

General Rules:

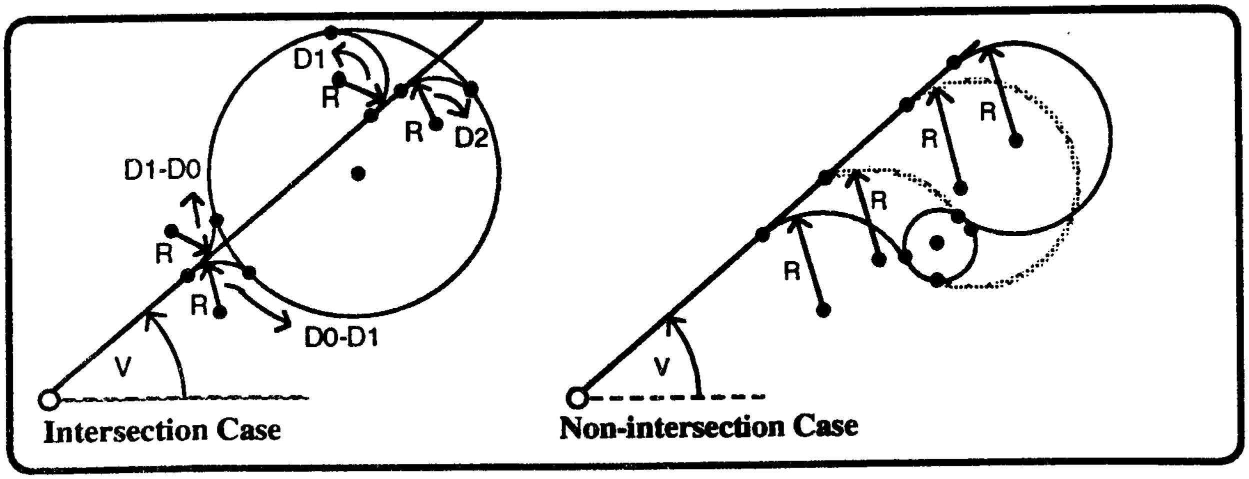
1) If the arc in the first event intersects the line in the third, ACA uses reasonable direction and no Alternate Choice is given. If the first and third events do not intersect, one of two possible Alternate Choice sets will be given (see lower illustration).

2) When the first arc and line intersect, opposite direction arcs give outside joins and similar direction arcs give inside joins.

You Enter:	**CA Finds:**
1st event: V	*X(abs), Y(abs)
2nd event: [R, D]	X,Y,I,J, A,B
3rd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	

Note:

The lower illustration shows the non-intersection case for a clockwise (D0) joining arc. In this case there are two sets of Alternate Choices: the gray arcs show the set given if the fixed arc is clockwise; the solid arcs show the set given if the fixed arc is counter-clockwise.



Intersection Case

Non-intersection Case

ARC TO ARC TO ARC

In an arc-arc-arc join, the first event is defined by center and direction, the second arc floats and the third is fixed by its center. Calc-Assist adjusts a floating middle arc of fixed radius to find where it is tangent to both other arcs, then uses reasonable direction to limit the possibilities to two, which it gives as Alt Choice selections.

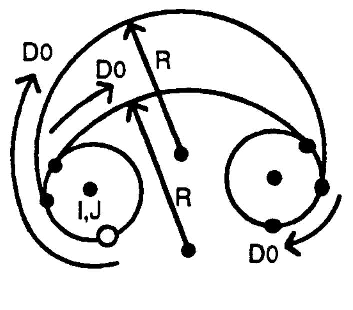
The illustrations below show all the possible cases for a three arc join in which the first arc is programmed with clockwise direction (D0). Joining arcs are gray when programmed D0, solid when programmed D1.

General Rules:

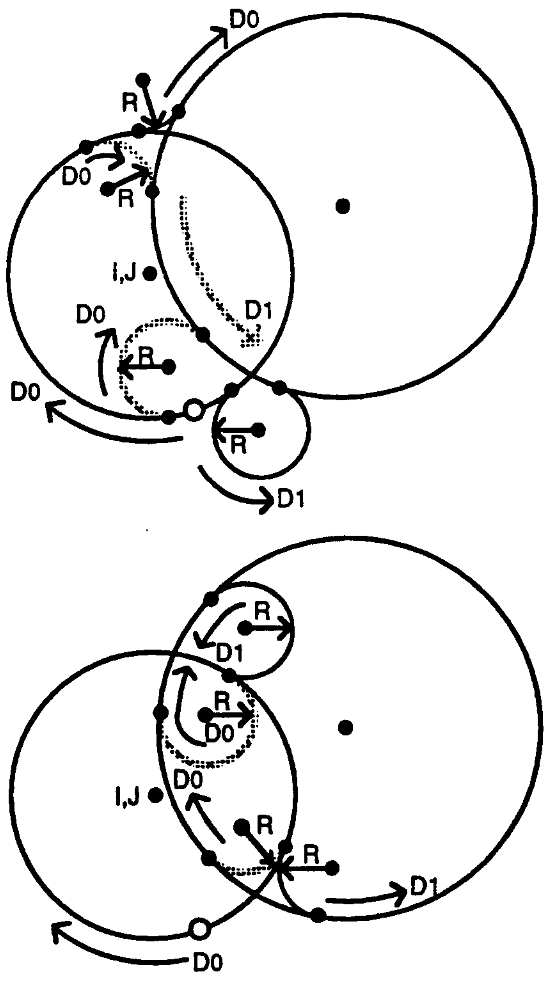
1) As in the line-arc-arc case, opposite direction arcs give outside joins and similar direction arcs give inside joins.

2) If only one joining point is possible, Calc-Assist will not give an Alternate Choice.

You Enter:	CA Finds:
1st event: [I,J,D]	X(abs), Y(abs)
2nd event: [R, D]	X,Y,I,J, A,B
3rd event: [I(abs}, J{abs}, R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	



Non-Intersecting first and last arcs



Intersecting first and last arcs

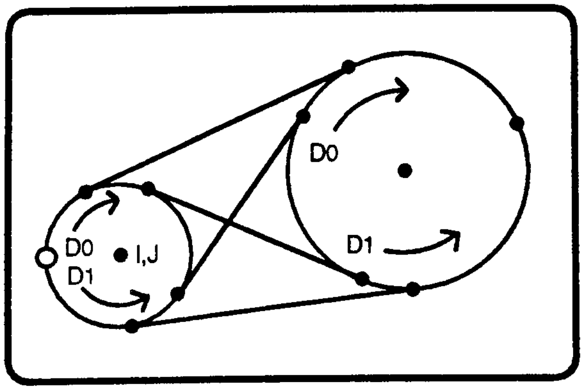
ARC TO LINE TO ARC

In an arc-line-arc join, the first event is defined by center and direction only. CA can use its center and the endpoint of the last event to find its radius. ACA then finds its endpoint by working back from the third event (which is fixed by its center) through a floating middle line which it adjusts to be tangent to both arcs.

General Rules:

1) The line in the second event must be empty (enter event type 1; COPY/STORE).

2) So that Calc-Assist can determine which of the several possible tangent lines to use, direction must be programmed for both arcs.



You Enter:	CA Finds:
1st event: [I, J, D]	X(abs), Y(abs)
2nd event: Event type 1; Copy/Store	
3rd event: [I(abs), J(abs), R, D] or [X(abs), Y(abs), I(abs), J(abs), D]	

A TRIAL ADVANCED CALC-ASSIST RUN

Follow this sequence to program the ACA part shown on the next page.

1) **Move to Set Up mode. Choose TM3 (All-Axis Inhibit). Later you can treat this as an optional step.**

   This will inhibit axis motion if you accidentally press the CYCLE START button.

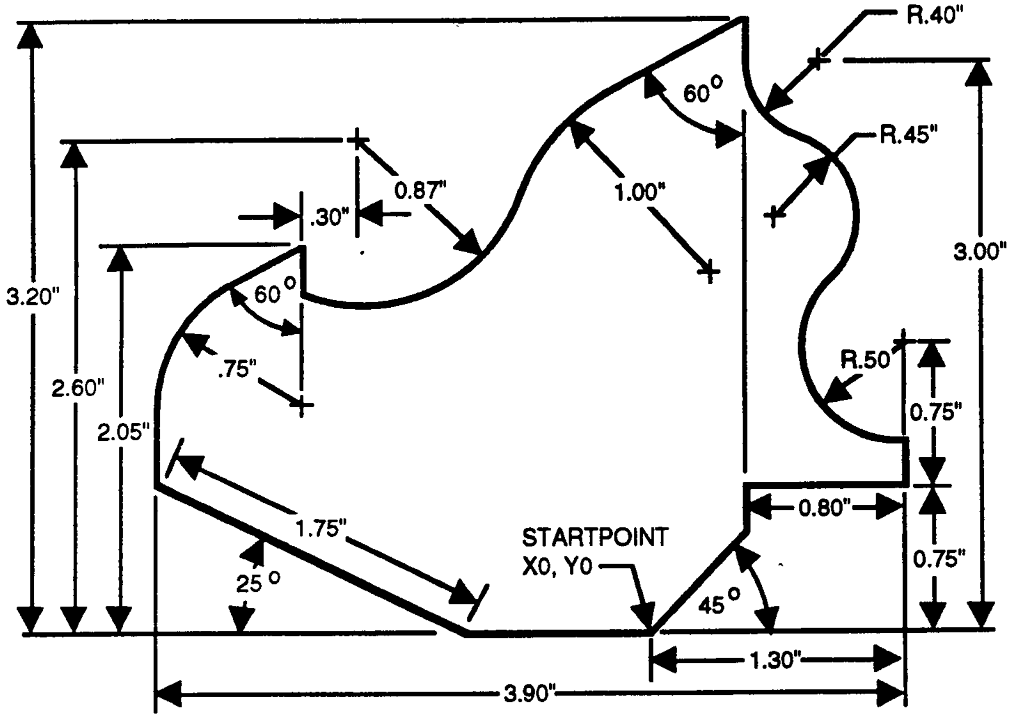
2) **Move to Catalog mode and enter a name for the part.**

3) **Move to the Tool Tables and enter your tool numbers and dimensions.**

   For the sample part enter a .25 inch cutter diameter for the 3rd offset.

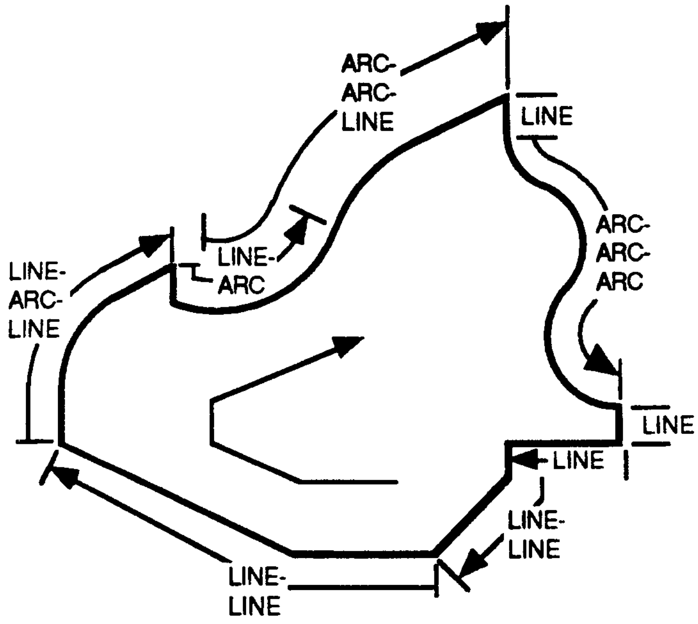
4) **Move to Program mode event display, but DO NOT turn Calc-Assist on.**

   Program the part on the next page using the program data that follows it.



```
PROGRAM DATA                            COMMENTS

N10  (9)  T3 H3 D3 M3 S1800
N20  (0)  X1.5 Y0                       Position for entry.
N30  (1)  X0/ C0, F10                   Dummy move; comp off.
N40  (1)  V180 C1                       Angle for first line in line-line join; comp on.
N50  (1)  V155 D1.75 X-2.6              Complete line-line with V, D and X
N60  (1)  V90                           Angle for first line in line-arc-line join.
N70  (2)  D0 R.75                       Floating arc by D and R.
N80  (1)  X-1.85 Y2.05 V30              Complete line-arc-line with fixed line.
N90  (1)  V270                          Angle for line in line-intersecting-arc join.
N100 (2)  I-1.55 J2.6 R.87 D1           Complete line-intersecting-arc with fixed arc.
                                        Also the first arc in an arc-arc-line join.
N110 (2)  R1 D0                         Floating middle arc for arc-arc-line.
N120 (1)  X.5 Y3.2 V30                  Complete arc-arc-line with fixed line.
N130 (1)  Y3                            Y-axis-only linear move.
N140 (2)  I.9 J3 D1                     First arc in arc-arc-arc join.
N150 (2) R.45 D0                        Floating middle arc for arc-arc-arc.
N160 (2) X1.3 Y1 I1.3 J1.5 D1           Fixed third arc in arc-arc-arc.
N170 (1)  Y.75                          Y-axis-only linear move.
N180 (1)  X.5                           X-axis-only linear move.
N190 (1)  V270                          First angle in a line-intersect-line join.
N200 (1)  V225 X-1 Y-1                  Fixed line completes line-intersecting-line.
N210 (1)  X0/ C0                        Dummy move; comp off.
N220 (9)  M30
```

The following illustration shows the location of various cases that ACA must solve as the control works through the program:



The overlapping Calc-Assist cases necessary to define this part is obvious in this illustration. Programming such overlapping cases is all right as long as all the information Calc-Assist needs to solve both cases can be programmed. For instance, in this example the overlapping occurs at N100 (see listing previous page), where the same arc is used as the second event in a line-intersecting-arc case and as the first event in a following arc-arc-line case. The arc must be defined as a fixed arc for the line-arc case; however, this doesn't mean it needs endpoints, and since the endpoints cannot easily be read from the part, we leave them out. The arc-arc-line case that follows will fill them in for us.

As the first arc in an arc-arc-line case, the shared arc needs to be defined only by I, J, and D (it need not be fixed); though the addition of a radius entry fixes the arc, the endpoints remain unknown and an arc-arc-line Calc-Assist case is valid.

5) **Return to Catalog mode and make a copy of your program to run.**

It's still good practice to make a copy of the program before running it so that you can later compare this original to the copy that was run through Calc-Assist.

6) **Set up the Calc-Assist information in program mode.**

Move to the Program mode Set Up values page by pressing the FRONT END GRAPHICS soft key and then the SET UP VALUES soft key. Enter the following values for the sample part:

ORIGIN:	X: set to X-3	**PLANE SELECT:**	P1 for the XY plane.
	Y: set to Y-1	**PLOT CONTROL:**	F: set to 0 to start at N0.
	Z: set to Z0		L: set to 9999.999.
SIZE:	V: set to V4 for a 4" vert. window.	**AUTO CALC:**	A2 - Calc On

7) **Press EXIT to return to the Front End Graphics screen.**

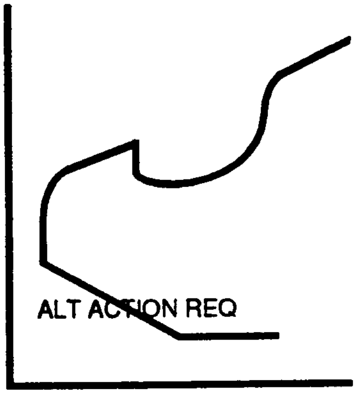
8) **Press the RUN CALC ASSIST soft key to start Calc-Assist.**

The program will start running and stop at N130 with an ALTERNATE CHOICE message. This is what the program graph and code will look like up to N130 (Calc-Assist generated values are denoted by *):

The Code:

```
N10  (9)  T3 H3 D3 M3 S1800
N20  (0)  X1.5 Y0 Z-.5 V0* D1.5* G0
          (V and D will vary with your Home position)
N30  (1)  X0/ Y0* V0* D0* C0 F10
N40  (1)  X-1.014* Y0* V180 D2.514* C1
N50  (1)  X-2.6, Y.7396*, V155, D1.75
N60  (1)  X-2.6* Y1.184* V90 D.4444*
N70  (2)  X-2.225* Y1.8335 I-1.85*
            J1.184* A180* B120.001* R.75 D0
N80  (1)  X-1.85 Y2.05 V30 D.433*
N90  (1)  X-1.85* Y1.7834* V270 D.2666*
N100 (2)  X-.7528* Y2.2515* I-1.55 J2.6 A249.828*
            B336.387* R.87
N110 (2)  X-.3365* Y2.717* I.1635* J1.851*
            A156.391*  B120.001* R1 D0
N120 (1)  X.5 Y3.2 V3, D.9659*
N130 (1)  X.5* Y3 V270* D.2*
```

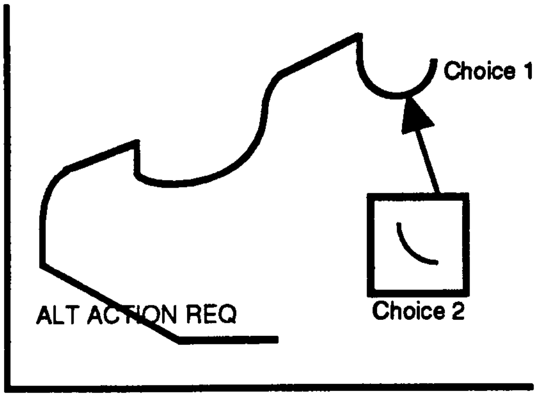
The Graph:



At N140 the Graphics Display plot stops and gives an ALTERNATE ACTION REQUIRED message, as shown in the above illustration.

9) **Complete any Alternate Choice selections or program editing necessary to process the run.**

Pressing the ALTERNATE CHOICE soft key once will cause the control to plot a half intensity arc as shown in the main illustration. Comparing this graph to the part print we can see that this is the wrong choice. To see the correct choice, press clear and press the ALTERNATE CHOICE soft key again. The graph up to N140 will disappear, but now the correct arc will be generated as shown in the inset. Pressing the PROGRAM DISPLAY soft key, we find that N140 contains:



```
X 0.7804*       A 180*
Y 2.6183*       B 252.602*
I  0.9          R  0.4*
J  3.0           D 1
```

Since this information describes the correct arc, press the STORE CHOICE soft key. The data that Calc-Assist computed (X, Y, A, B, and R) will then be stored as hard data and the asterisks will be removed.

Now return to the Front End Graphics and press CLEAR and the RUN CALC ASSIST soft key (but don't press RESET). This will regenerate the graph up to N140 and, having stored the correct arc as hard data, the control will finish plotting the graph. The remaining code is listed below.

```
N140 (2)  X.7804 Y2.6183 I.9 J3 A180 B252.602 R.4 D1
N150 (2)  X.9557* Y1.8626* I.6458* J2.1889* A72.596* B313.523* R.45 D0
N160 (2)  X1.3 Y1 I1.3 J1.5 A133.517* B270* R.5* D1
N170 (1)  X1.3* Y.75 V270* D.25*
N180 (1)  X.5 Y.75* V180* D.8*
N190 (1)  X.5* Y.5* V270 D.25*
N200 (1)  X-1 Y-1 V225 D2.1213*
N210 (1)  X0/ Y-1* V0* D0* C0
N220 (9)  M30
```

10) **Make a run in Graphics mode to check the final offset tool path.**

Enter Graphics mode. Here you have the option of viewing either the tool center path or the part profile. Select T1 - TOOL CENTER and move to the Graphics display. Press CLEAR, RESET, then CYCLE START. When the plot is complete, select T2 - PART PROFILE. Move back to the Graphics display and press RESET but **do not press CLEAR.** Press CYCLE START. The resulting graph will overlay a part profile plot onto the tool center plot.

11) **If you chose TM3 from Set Up mode, go back and set TM1. Your program should now be ready to cycle in Auto mode.**

This completes the sequence.

ADVANCED CALC-ASSIST ERROR MESSAGES

When an error condition halts your run, two lines of information will be displayed across the lower left corner of the graph. The first line identifies, as specifically as the data permits, which case Calc-Assist is looking at (i.e. ARC TO ARC TO ARC); the second line gives, within the case context, the suspected error (i.e. INCONSISTENT RADIUS). The N number shown at the top of the display is the number of the first event in the Calc-Assist sequence that failed.

ERROR RECOVERY

If the first line correctly identifies the case you are trying to solve at the given line number, review the case description in this appendix with careful attention to any suspected entries (if given) from the second line of the error message. If Calc-Assist does not specify which entry is at fault, double check all entries against both the case description and your part print. Always be sure the last event in the sequence has a defined place to end, such as a given endpoint, end angle or an immediately following Calc-Assist case. Also make sure your selected L entry on the Program mode Set Up screen doesn't break the ACA case.

If the first line of the message does not agree with your intended case, disregard the second line and refer instead to the case you attempted to use. Clear the error message and move to the Program Display screen. Determine which events make up your intended case and check them against both the part print and the case description for missing or redundant entries. Edit as required, then resume the run.

If this review shows that all data has been properly entered, suspect that the starting coordinates for the Calc-Assist sequence are in error. Here it is best to choose the last correct point and analyze all cases forward to the reported error. Any data stored as an improper Alternate Choice selection along the way must be manually cleared and recomputed by Calc-Assist.

It is usually best **not** to attempt to remedy a Calc-Assist error by working backwards as this may result in false joins and generate irrelevant error checking information.

ERROR MESSAGES - CASE IDENTIFICATION

The possible case-identification messages follow. Included are default messages that appear when CA cannot deduce your case structure.

ONE EVENT CA
BASIC LINE
BASIC ARC

TWO EVENT CA
LINE TO LINE
LINE TO ARC
ARC TO LINE
ARC TO ARC

THREE EVENT CA
LINE TO ARC TO LINE
LINE TO ARC TO ARC
ARC TO LINE TO ARC
ARC TO ARC TO LINE
ARC TO ARC TO ARC

ERROR MESSAGES - SUSPECTED ENTRIES

The messages that follow refer to suspected entries. Included are default messages that appear when Calc-Assist cannot deduce which entries are causing the fault.

INSUFFICIENT DATA
INCONSISTENT DATA
PARALLEL LINES
CONCENTRIC ARCS
INCONSISTENT AXIS
INCONSISTENT DIMENSION
INCONSISTENT ANGLE
INCONSISTENT CENTER
INCONSISTENT RADIUS
INSIDE OF ARC
RADIUS DOES NOT REACH END
ARC DOES NOT INTERSECT LINE
ARC DOES NOT INTERSECT ARC
V DOES NOT AGREE WITH AXIS DIR
AXIS MOVE GREATER THAN DIMENSION
LINE NOT EMPTY EVENT
ARC RADII OR DIR'S MISMATCHED
START AND END RADII NOT SAME
PROG AND END RADII NOT SAME
PROG AND START RADII NOT SAME
LINE DOES NOT INTERSECT LINE
LINE DOES NOT INTERSECT ARC

CUSTOMER PREPARED CYCLES

DYNAPATH DELTA 40, 50, 60 MILL CUSTOMER INFORMATION MANUAL

AUTOCON

INTRODUCTION

Macros and Customer Prepared Cycles (CPCs) define a range of executable machining instructions that are similar to the Subroutine and Repeat loops that you may already be acquainted with. However, macros can allow machining routines to be accessed by any program.

Customer Prepared Cycles further extend macro capability by allowing you to design your own machining routines that can accept variable data from the program that calls your cycle into operation.

DEFINITIONS

Repeat, Subroutine, and Macro Call events all specify a range of events that are called into sequential execution on command.

SUBROUTINE

A Subroutine event specifies a range of events by assigning the first (F) and last (L) events, causing all events in the F, L range to be executed. The range may only be executed once per call, and the events specified within the range must reside in the calling program.

REPEAT

A Repeat event specifies its range by a first (F) event entry. The Repeat event is the last event in the range. The range of events may be repeated (T) times, for as many as 999 repetitions per call. An intermediate X, Y, Z step or axis rotation is permitted between each repetition. The specified range of events, by design, must reside in the calling program.

MACRO CALL

A Macro Call event calls a range of events as does the Subroutine, but may call it to execute a given number of times, in resemblance to the Repeat event. Any required step or rotation may be specified prior to (or within) the macro program.

However, the Macro Call event calls a routine by a title that is cataloged much like a part program, and the routine that is stored under this title is not resident in the main program, but sheltered in protected storage within a separate Macro catalog. In this way any main program, or even another macro program, may call any program in the Macro Catalog.

CUSTOMER PREPARED CYCLE

A Customer Prepared Cycle (also referred to as a "User Macro" or "Advanced Macro") is an advanced Macro that shares all of the attributes of the standard Macro and resides in the same Macro Catalog.

However, the CPC has additional capabilities, the most exclusive of which is the ability to accept multiple parameters called "Symbolic Variables". These variables can be passed by a calling statement in the main program or another CPC.

For example, representative data for a G cycle, number of holes, incremental distance between holes, depth, feedrate, reference plane coordinate, starting angular orientation, and change in angular orientation over a specified number of repetitions are all examples of parameters that can be passed to a CPC that you might design to drill a radial hole pattern.

The CPC's capability is further extended by an instruction set that allows the use of Variables directly in X, Y, Z and other letter addresses to define the endpoint of a motion or any other data that may be expressed in a cycle.

As you might expect, the CPC can also designate local variables not accessible to the main program. Any variable within the CPC may be reassigned or adjusted by assignment or mathematical expression using the add, subtract, multiply and divide operators, which are further augmented by trigonometric, absolute, round off, and other functions useful in designing complex cycles.

Other CPC commands permit conditional and non-conditional branching, error handling, access to the machine's Tool Tables, Position Registers, Probe Registers and Digital Interface for interrogative or control functions, and extended access to the serial communications port.

PROTECTED or PROTECTED/HIDDEN MACROS/CPCs

Any program in the Macro Catalog may be specified as protected or hidden & protected.

With either of these settings, the machine operator may not edit or delete these files unless a machine parameter is changed in the control. You must consult your machine supplier for instructions on changing this parameter. Rules for protected or hidden & protected macro/CPCs are described below.

Protected Macro/CPCs

- are designated by an exclamation point character (!) that precedes the Macro Part ID. Example: the macro/CPC program (DYNAPATH) becomes a protected macro by adding the exclamation point to the title (!DYNAPATH).
- machine operators **can select and view** the macro/CPCs but cannot delete them from the Macro Catalog.
- are deleted when a Power Fail occurs on the control.
- cannot be written to the control's floppy disk drive or transmitted to a computer through the control's serial port.
- can be received from the floppy disk drive or through the control's serial port. This capability allows the machine operator to reload these macro/CPCs after a Power Fail.

Hidden and Protected Macro/CPCs

- are designated by an asterisk character (*) that precedes the Macro Part ID. Example: the macro/CPC program (DYNAPATH) becomes a hidden & protected macro by adding the asterisk to the title (*DYNAPATH).
- machine operators **cannot select, view, nor delete** these macro/CPCs from the Macro Catalog.
- are deleted when a Power Fail occurs on the control.
- cannot be written to the control's floppy disk drive or transmitted to a computer through the control's serial port.
- can be received from the floppy disk drive or through the control's serial port. This capability allows the machine operator to reload these macro/CPCs after a Power Fail.

WHAT YOU NEED TO KNOW

Since writing a Customer Prepared Cycle is more difficult than writing part programs, you should be experienced in creating, editing, and cycling part programs that reside in the main Catalog mode before you read this section. Additionally, you should be familiar with the topics described in the Macro Mode and Macro Call Event sections of this manual.

WHAT IS COVERED

Section 1 is an introduction to Customer Prepared Cycles

Section 2 is a detailed description of the Customer Prepared Cycle's instruction set (programming language). This section includes a description of variables and the feature's algebraic hierarchy.

Part 3 provides a step-by-step procedure for writing CPCs and gives several examples.

Part 4 describes the steps necessary to develop and test a complex 3D CPC.

CPC PROGRAMMING LANGUAGE

This section describes how to use the CPC programming language and instruction set to design and build your own machining cycles. The topics covered here are:

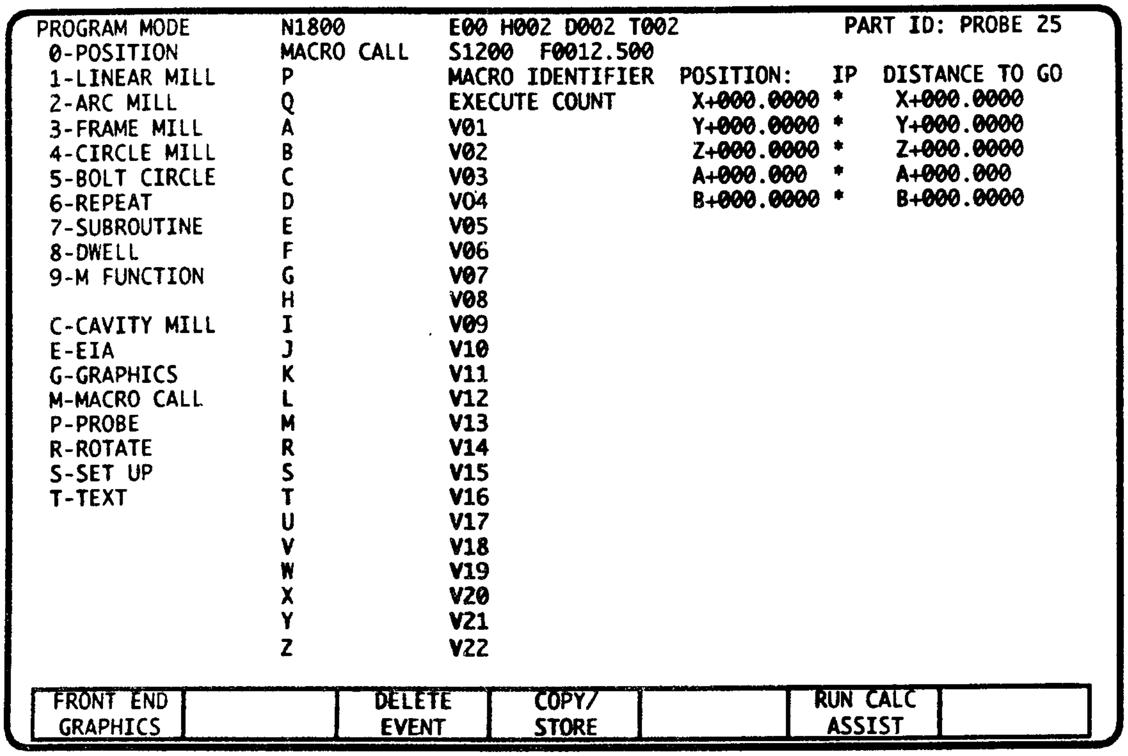
- Calling a CPC with the Macro Call event.
- Variable usage (including "pass", "global", and "local" variables.
- The Arithmetic event (event type A).
- Types of Arithmetic events (assignment, branching, procedural).

MACRO EDITOR MODE

You should already be familiar with the basic operation of the Editor mode that is described in part 2 of this manual. However, there is one feature, the SECOND FUNCTION soft key, that is present exclusively in the Macro Editor mode.

The SECOND FUNCTION soft key and the second function character set are available only if your control is equipped with the CPC option. These eight characters [( ) # = , + * /] are only used when you edit or create CPC code.

When the CPC option is present, the SECOND FUNCTION key descriptors appear on the second line of the Macro Editor display. Pressing the SECOND FUNCTION soft key highlights this line and activates the SECOND FUNCTION keys. The line will stay highlighted and the keys will remain active until you press the SECOND FUNCTION soft key again. Entering any of the letters A-H while the 2nd function definitions are active will yield that key's SECOND FUNCTION; the remaining keys, however, will be unaffected.



```
PROGRAM MODE       N1800        E00 H002 D002 T002              PART ID: PROBE 25
 0-POSITION        MACRO CALL   S1200  F0012.500
 1-LINEAR MILL     P            MACRO IDENTIFIER  POSITION:    IP  DISTANCE TO GO
 2-ARC MILL        Q            EXECUTE COUNT      X+000.0000 *     X+000.0000
 3-FRAME MILL      A            V01                Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL     B            V02                Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE     C            V03                A+000.000  *     A+000.000
 6-REPEAT          D            V04                B+000.0000 *     B+000.0000
 7-SUBROUTINE      E            V05
 8-DWELL           F            V06
 9-M FUNCTION      G            V07
                   H            V08
 C-CAVITY MILL     I            V09
 E-EIA             J            V10
 G-GRAPHICS        K            V11
 M-MACRO CALL      L            V12
 P-PROBE           M            V13
 R-ROTATE          R            V14
 S-SET UP          S            V15
 T-TEXT            T            V16
                   U            V17
                   V            V18
                   W            V19
                   X            V20
                   Y            V21
                   Z            V22
```

THE MACRO CALL EVENT

The Macro Call event is programmed in a "calling" program. A calling program can be a part program in the main Catalog, or a program in the Macro Catalog that is called, and then during execution, calls another Macro program[1].

The Macro Call event begins execution of the "called" program. The screen on the **previous page** shows the Macro Call event.

P - MACRO IDENTIFIER

Enter the Macro program title by pressing P, followed by the macro's part ID.

Q - MACRO EXECUTION TIMES COUNTER

Enter the number of times Q the Macro program is to be executed in immediate succession.

A - M and R - Z: THE MACRO "PASS" VARIABLES

There are 22 pass variables in the Macro Call event.

If you are writing a CPC, you can choose any of the A-M or R-Z entries in the Macro Call event to pass values[2] to corresponding Variables #01 - #22 in the called CPC (correspondence is shown on the previous screen) when the CPC begins execution.

If you're writing a CPC, note that A-M and R-Z in the Macro Call event will only modify variables #01 - #22 of the CPC. They have no relationship to any letter entry in other blocks or events in the calling program or called program. But if you, as the CPC writer, think that your CPC users find it easier to remember that K is an incremental peck depth, have the user enter the peck as K in the Macro Call event. K will then pass its value on to Variable #11 in the called CPC.

Data for any entry A-M or R-Z in the Macro Call event can be entered with up to 7 digits in + or - {0.7, 1.6, 2.5, 3.4, 4.3, 5.2, 6.1 or 7.0} decimal point format[3]. The data is passed to its corresponding variable #01 - #22 in the same format. As a CPC writer you will not be required to convert data into Inch/Metric fields or letter field formats.

If you are using a CPC written by someone else, you should fill in only the A-M or R-Z entries defined in the CPC writer's instructions, and make sure your data falls within a numeric range specified by the CPC writer.

For a peck increment of .25 to be entered as entry K in a Macro Call event, you would press K, then key in .25 followed by the ENTER key.

[1] As many as four Macro programs can execute in a nested arrangement, each called by the one before. The first Macro is called by the main program.

[2] In the main program, A-M and R-Z entries must be numeric constant values. If one CPC calls another, Macro Calling event entries A-M and R-Z in the calling CPC may use variables.

[3] Leading zeros or a decimal point with trailing zeros do not need to be entered.

CPC VARIABLES

500 variables can be used within Macro programs. They are key elements of the CPC.

VARIABLE TYPES

Variables #1 - #22 are called Global variables. Their values can be assigned by a calling program. Their values can also be assigned or modified within a CPC.

For any entry A-M or R-Z that you have not assigned a value in the Macro Call event, the initial value of its corresponding Variable #01 - #22 in the CPC will be indeterminate until the CPC assigns it a value.

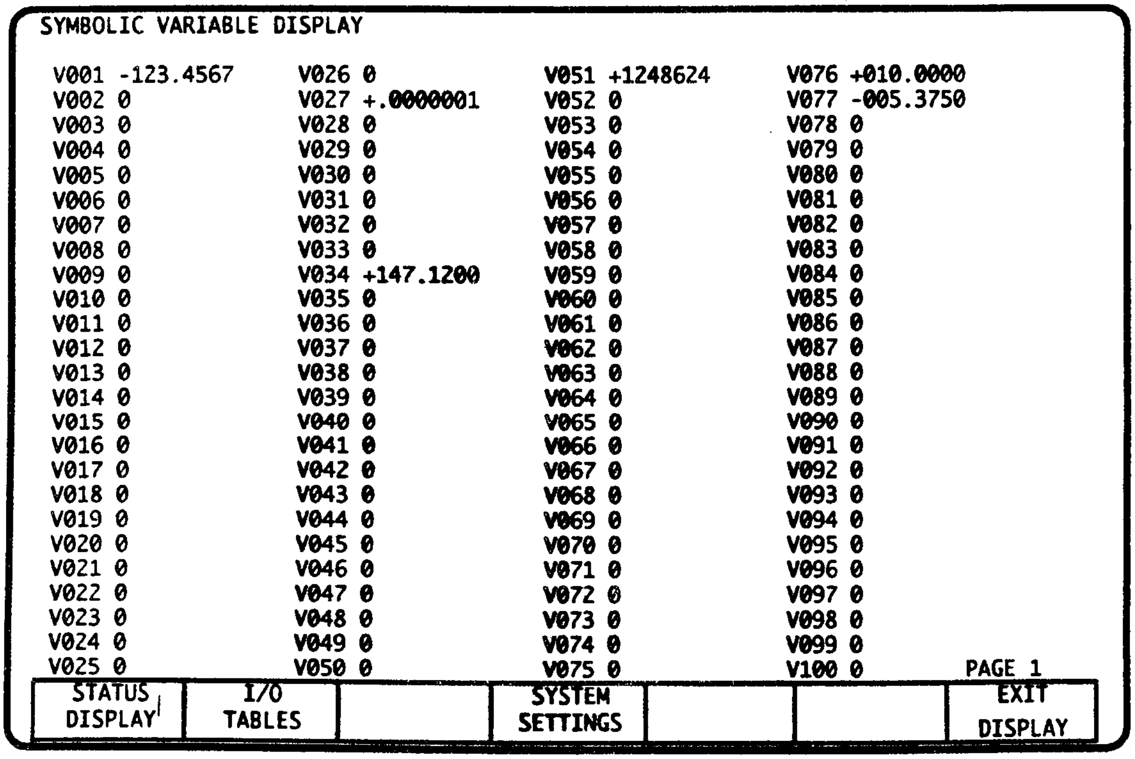
Variables #23 - #500 are called Local variables. Their values cannot be assigned by a calling program. Their values can only be assigned or modified within a CPC. Variables #23 - #500 will be indeterminate until the CPC assigns a value.

INITIALIZING VARIABLES WITHIN THE CPC

Any variable, #1 - #500, can be assigned a value within the CPC. The value may be up to 7 digits in + or - {0.7, 1.6, 2.5, 3.4, 4.3, 5.2, 6.1 or 7.0} decimal[3] format.

All variables, #1 through #99, are set to zero when the control is first turned On. Once a CPC has been executed, any variable that has had a value assigned or modified by the CPC will retain its last value. This is why we say variables not assigned by "pass" are indeterminate on entry to a new CPC.

If you wish to look at the present value of any CPC variable you can do so by first pressing the FAULT/STATUS key and then the SYMBOLIC VARIABLES soft key. (The EXIT DISPLAY soft key is used to return to your last screen.) The LEFT and RIGHT ARROW KEYS are used to move between symbolic variable pages.



```
SYMBOLIC VARIABLE DISPLAY

 V001 -123.4567    V026 0            V051 +1248624     V076 +010.0000
 V002 0            V027 +.0000001    V052 0            V077 -005.3750
 V003 0            V028 0            V053 0            V078 0
 V004 0            V029 0            V054 0            V079 0
 V005 0            V030 0            V055 0            V080 0
 V006 0            V031 0            V056 0            V081 0
 V007 0            V032 0            V057 0            V082 0
 V008 0            V033 0            V058 0            V083 0
 V009 0            V034 +147.1200    V059 0            V084 0
 V010 0            V035 0            V060 0            V085 0
 V011 0            V036 0            V061 0            V086 0
 V012 0            V037 0            V062 0            V087 0
 V013 0            V038 0            V063 0            V088 0
 V014 0            V039 0            V064 0            V089 0
 V015 0            V040 0            V065 0            V090 0
 V016 0            V041 0            V066 0            V091 0
 V017 0            V042 0            V067 0            V092 0
 V018 0            V043 0            V068 0            V093 0
 V019 0            V044 0            V069 0            V094 0
 V020 0            V045 0            V070 0            V095 0
 V021 0            V046 0            V071 0            V096 0
 V022 0            V047 0            V072 0            V097 0
 V023 0            V048 0            V073 0            V098 0
 V024 0            V049 0            V074 0            V099 0
 V025 0            V050 0            V075 0            V100 0        PAGE 1
```

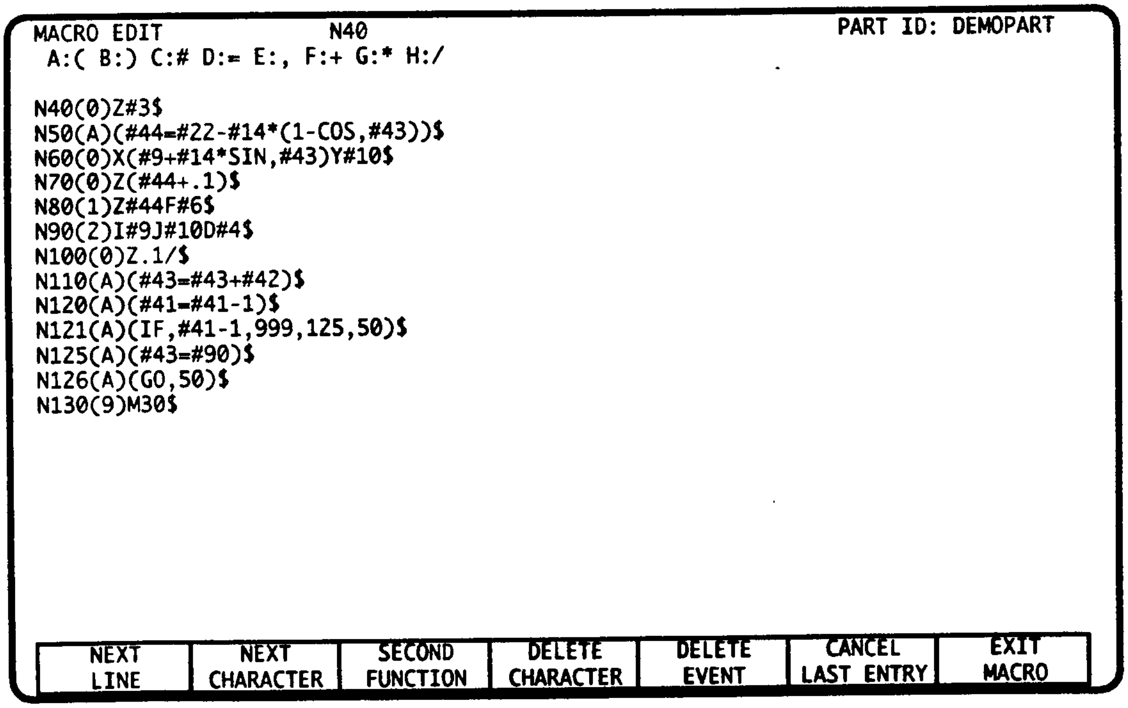
As the Macro writer, you should insure that all variables to be used are initialized by calling event or assignment statement[4] within the CPC before the tool enters stock. If one CPC will call another, it is improper to rely on a variable's last value in a calling CPC to initialize a called CPC's same variable. However, you can return a called CPC variable's last value to its calling CPC in this way.

USE OF VARIABLES WITHIN BLOCK LETTER ENTRIES

A variable or expression containing one or more variables (#1 through #500) can be entered in any letter (except the N sequence number or Event Type character) in any event or block within a Macro program.

Events using variables or expressions can be viewed from the Macro Edit mode screen. For example, N80 from the screen, **below**, is a Linear Mill event that uses variable entries for its Z axis endpoint Z and feedrate F.

When you assign a variable or an expression to a letter entry within a block or event, the Macro operating program will automatically adjust the variable's value to the correct letter field format for Inch or Metric mode. You will not need to perform field conversion operations.



```
MACRO EDIT              N40                                    PART ID: DEMOPART
 A:( B:) C:# D:= E:, F:+ G:* H:/

N40(0)Z#3$
N50(A)(#44=#22-#14*(1-COS,#43))$
N60(0)X(#9+#14*SIN,#43)Y#10$
N70(0)Z(#44+.1)$
N80(1)Z#44F#6$
N90(2)I#9J#10D#4$
N100(0)Z.1/$
N110(A)(#43=#43+#42)$
N120(A)(#41=#41-1)$
N121(A)(IF,#41-1,999,125,50)$
N125(A)(#43=#90)$
N126(A)(GO,50)$
N130(9)M30$
```

Although the CPC will perform error checking at runtime and flag any overflow conditions in letter entries, as the CPC writer, you can also employ range or error checking and generate Fault and Operator Message displays that are provided as procedural statement utilities. These are described further on.

Variables cannot be used in the main program's event or block letter entries, or be viewed from the Macro Program mode.

[4] Assignment statements are discussed further on.

Cutter compensation within CPCs: Delta 30, 40, 50, and 60 controls[5] allow you to use variables to define X, Y, and Z endpoints when cutter compensation is active. Your CPC must follow the rule that once cutter comp is active, only lines and arcs are permitted until cutter comp is turned off. Therefore, the start of your CPC should use Arithmetic events to calculate all required endpoints and store them in the user variables. Next, your CPC should program a continuous series of Linear and Arc mill events that activate cutter compensation and use the endpoint values that are stored in the user variables. Refer to the example at the end of section 3.

TERMS

Before proceeding, an explanation of several terms is in order.

Syntax - Syntax specifies the order or sequence in which mathematical operations are "solved" by the Macro program.

Constant - A constant is a number that is expressed in digits that remains unchanged throughout the CPC. It may be positive or a signed negative number and can be an integer or real (mixed, fractional or decimal) number.

Variable - As used in a CPC, "variable" refers to symbolic variables #1 through #500. Each is a symbol that may take on a range of values at various times as assigned or modified by an Assignment Statement within the CPC.

Assignment Statement - As used in CPCs, the "assignment statement" is an expression containing an equal sign (=). A variable, #1 through #500, to the left of the equal sign, is a "target variable" that is set equal to a value determined by the constant, variable, term, or expression directly to the right of the equal sign.

Branching Statement - This program instruction evaluates a variable, term or expression, testing its value relative to zero. The program can branch to as many as three different N Sequence numbers in the Macro depending whether the tested variable, term or expression is less than, equal to, or greater than zero.

Other instructions can test a specified PIC DC input at the machine interface and branch according to the input's status.

Factor - A "factor" can be any of the following: a constant, a variable, an expression or a function. A constant or variable is the simplest kind of factor. A factor is the simplest type of term (see next definition).

Term - In general usage, "term" often refers collectively to two factors that are to be multiplied, or two factors when one will be divided by the other. A term is the simplest kind of expression (see next definition).

Expression - In general usage, "expression" refers collectively to two terms when one term will be added to or subtracted from the other. One term is unsigned[6].

[5] The ability to use cutter compensation within a CPC is a feature that is unique to the D30 control, it is not available with the D20 CPC feature.

[6] Such as -4 + 3, with + indicating an addition operation with one signed term. Otherwise one or two sets of parentheses must be used, as in 3 + (-4). Syntax considers a signed constant to be an expression, since 4 in thiscase must be "negated" before the outer expression 3 + (-4) can be evaluated.

Argument - An argument can be a constant, variable, term or expression. It is the data to be operated on by Function or Procedure, and the Function or Procedure may limit the range of allowed values used in its argument(s).

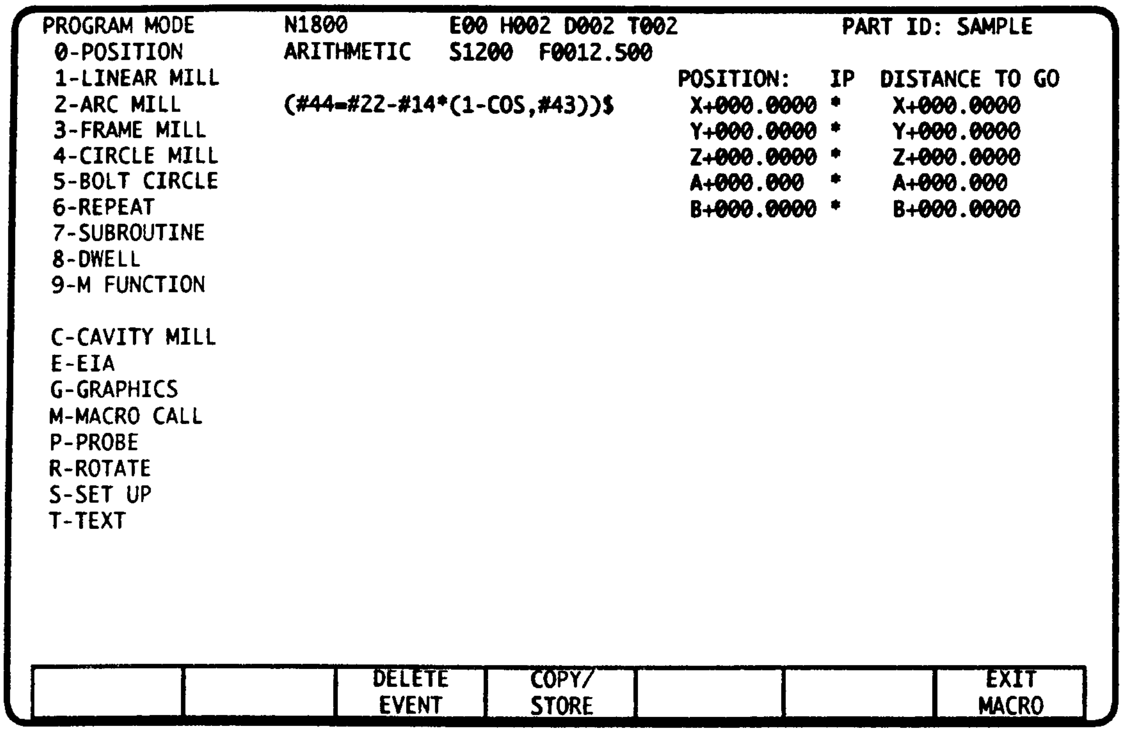
Function - A function follows syntax rules that are generally applied to a factor. As used in CPCs, it is identified by a three letter abbreviation followed by a comma, then one or more arguments. A Function always returns a value. In the assignment statement (#12=SQR,4) the SQR or "Square Root" function, for example, accepts 4 as its argument and assigns a value of 2 to variable #12.

Procedure - A procedure follows syntax rules that are generally applied to an assignment statement. It is similar to a Function because it has a three letter abbreviation followed by a comma, then one or more arguments. It differs, however, because it does not return a value, but instead performs some other type of specialized task. A print instruction is an example of a procedural statement. The print instruction uses text as its argument, and it performs the task of sending its argument (the text) to the serial communications port.

THE ARITHMETIC EVENT - EVENT TYPE A

Computations, branching and procedural operations can all be programmed within the Arithmetic event. This event is available only within CPC programs.

A typical Arithmetic event, as viewed from Macro Program mode, would appear as shown in the display **below**.



```
PROGRAM MODE         N1800         E00 H002 D002 T002              PART ID: SAMPLE
 0-POSITION          ARITHMETIC    S1200  F0012.500
 1-LINEAR MILL                                       POSITION:    IP  DISTANCE TO GO
 2-ARC MILL          (#44=#22-#14*(1-COS,#43))$       X+000.0000 *     X+000.0000
 3-FRAME MILL                                         Y+000.0000 *     Y+000.0000
 4-CIRCLE MILL                                        Z+000.0000 *     Z+000.0000
 5-BOLT CIRCLE                                        A+000.000  *     A+000.000
 6-REPEAT                                             B+000.0000 *     B+000.0000
 7-SUBROUTINE
 8-DWELL
 9-M FUNCTION

 C-CAVITY MILL
 E-EIA
 G-GRAPHICS
 M-MACRO CALL
 P-PROBE
 R-ROTATE
 S-SET UP
 T-TEXT
```

An Arithmetic statement cannot generate axis motion. Arithmetic statements can, however, compute complex instructions that assign values to variables in following executable events or blocks[7].

The display **above** shows an Arithmetic event programmed as an assignment statement.

To compute this statement, the Macro program will first evaluate the difference between a constant (1) and the cosine of variable #43. The result is then multiplied by the value of variable #14. This product is next subtracted from the value of variable #22, and the final value of the expression is assigned to variable #44.

Arithmetic events can be tailored to perform several important operations. You can program the event to:

1. Compute and assign values to selected variables;
2. Evaluate Functions;
3. Cause the Macro program to branch, unconditionally or by conditional test;
4. Perform Procedural operations.

These operations are described in the following sections.

[7] Expressions can also be entered directly in any block or event letter entry.

ASSIGNMENT STATEMENT

This section describes how the Arithmetic event can be programmed as an Assignment statement.

An assignment statements in Conversational and EIA/ISO format are shown below. Note that this is one way in which the Arithmetic event can be programmed.

CONVERSATIONAL[8]	Ndddd.ddd(A)(#XX=<expression>)$<CR><LF>
EIA/ISO BLOCK[8]	Ndddd.ddd(#XX=<expression>)<CR><LF>

FIELD DATA:	REPRESENTS	COMMENT
Ndddd.ddd	N Number	each d is a digit, N 4.3 decimal[9] format.
(A)	event type[10]	entered for Conversational format only[11]
(	left delimiter	identifies start of statement
#XX	variable	XX is variable number, range 1-99
=	equal sign	identifies this as an assignment statement
<expression>	assigned value	constant, variable, term or expression[12]
)	right delimiter	identifies end of statement
$	dollar sign	reqd. Conversational, marks end of event
<CR>	carriage return	character (optional)
<LF>	line feed	reqd. for EIA format, marks end of block

The following examples show the use of constants, terms, and expressions.

STATEMENT	COMMENT
N10(A)(#1=-4.3)$	constant assigned to variable #1
N20.25(#11=#72)	value of variable #72 assigned to #11
N220(A)(#44=#26/3)$	value of term #26+3 assigned to #44
N234(#21=3*SIN,30.5+#4/(3+#2))	value of expression assigned to #21

[8] As many as 80 characters may be entered in the character string sequence identified by underline. Spaces are not permitted. Commas may only be used within functions and only as shown by the syntax.

[9] Leading zeros or decimal point with trailing zeros need not be entered.

[10] (A) identifies an Arithmetic event.

[11] On Macro Transmit operations, the control sends all Arithmetic data out in EIA block format. In Macro mode 4, all Arithmetic data is displayed in Conversational event format.

[12] < and > or <expression> are, of course, not entered literally. They show where constant, variable, term or expressional data is located in the assignment statement character string.

FUNCTIONS

The function SIN,30.5 appeared in the N234(#21=3*SIN,30.5+#4/(3+#2)) example. In this example, 30.5 is the argument, representing 30.5 degrees. The function would return a value of .5075.

Syntax: SIN,<argument>

The Arithmetic statement's right delimiter “)” acts as the argument terminator in a simple assignment such as:

```
N20(#66=SIN,#99)
```

When a function is used as a factor or term in an expression[13], the its argument is terminated by one of the following operators: the addition sign ”+”, subtraction sign “-”, multiplication sign “*”, or the division sign “/’.

SIN,374.221*2	constant 374.221 as argument
COS,#13+3	variable #13 as argument

The argument of a function can be a constant, variable, term or expression. When the function's argument is a constant or variable, the argument need not be enclosed in parentheses. In the first example above, the “*” operator terminates the argument. In the second example above, the “+” operator acts as a terminator.

SIN,(#11/3.1416)	term (#11/3.1416) as argument
COS,(#11/3.14/2+#52)*2	exp. (#11/3.14/2+#52) as argument
SIN,(-45)	negative term as argument[14]

When the function's argument is a term or expression, the argument must be enclosed in parentheses. In the third example, factor #11 is divided by 3.1416, so the resulting term (#11/3.1416) must be enclosed, or the “/’ operator would terminate the function's argument. The same is true, where the argument is extended into an expression, adding the term #52. Thc last example shows an argument that must be “negated” before it can be evaluated.

Multiple levels of parentheses may also be used in the function argument. More information about levels of parentheses follows later.

The functions include:		
	SQR	square root function
	ABS	absolute function
	SIN COS TAN ATN	trigonometric functions
	INT FDW FUP	rounding functions

SQR - THE SQUARE ROOT FUNCTION

Syntax: SQR,<argument>

The square root of a number “a” is a number which, when multiplied by itself, yields the number “a”.

Every positive real number has two real number square roots. This function returns the non-negative square root of a positive argument, which is also called its principal square root.

[13] The syntax is the same for a function within an assignment statement, or a functional expression that appears in any block or event letter entry within a CPC program.

[14] Parentheses are optional but recommended in this case.

The function returns zero when the value zero is evaluated from its argument.

A runtime error occurs if a negative value is evaluated from its argument.

The argument can be a constant, variable, term or expression.

ABS - THE ABSOLUTE VALUE FUNCTION

Syntax: ABS,<argument>

The absolute value of a number is the unsigned value of the number.

- If the argument is non-negative, this function returns the argument itself.
- If the argument is zero, zero is returned.
- If the argument is negative, this function returns the argument as a positive value.

The argument can be a constant, variable, term or expression.

SIN - SINE FUNCTION

Syntax: SIN,<argument>

The following definitions refer to axes U and V. U and V are used annotatively to denote X and Y, respectively, in the XY plane, Y and Z, respectively, in the YZ plane, and Z and X, respectively, in the ZX plane.

Trigonometric functions relate an angle of a right triangle to the proportional lengths of its sides[15].

Generally, the Sine function is applied to find either the radial distance or V axis displacement from one point to another.

The relationship is: V = R * Sin Ø

If angle Ø is known, then either of the R radial distance or V axis displacement[16] can be found using the other (of R or V) and the Sin Ø function.

The argument can be a constant, variable, term or expression. The value of the argument is an angle of -9999.999 ≤ <argument> ≤ 9999.999 degrees. Fractional degrees will be read as decimal entries to the nearest .001 degree.

The function will return a value ranging from -1 to +1.

COS - THE COSINE FUNCTION

Syntax: COS,<argument>

Generally, the Cosine function is applied to find either the radial distance or U axis displacement from one point to another.

The relationship is: U = R * Cos Ø

[15] More precisely, if the initial side of angle Ø is the positive axis of a U,V plane, then Cos Ø and Sin Ø are the U,V coordinates where the terminal side of angle Ø intersects the unit circle $U^2+V^2=1$, with its center at the axes origins.

[16] Signed distance along axis.

If angle Ø is known, then either of the R radial distance or U axis displacement[16] can be found using the other (of R or U) and the Cos Ø function.

The value of the argument is an angle -9999.999 ≤ <argument> ≤ 9999.999 degrees. Fractional degrees will be read as decimal entries to the nearest .001 degree.

The function will return a value ranging from -1 to +1.

TAN - TANGENT FUNCTION

Syntax: TAN,<argument>

The Tangent function can be used when the angle between two lines is known and one of the axis displacements U or V must be found. The general relationship is:

Tan Ø = U/V; this may be more usefully expressed as
U = V(Tan Ø) or V = U/(Tan Ø)

The TAN function is undefined for Ø=90°, 270° or any angle (90+180n)° for all integer values of n. Always pretest Ø for values such as 90°, -90° and 270° to avoid run-time errors.

ATN - ARCTANGENT FUNCTION

Syntax: ATN,<argument>

Generally, the Arctangent function is applied to find the angular direction from one point to another when the U and V axes displacements[16] are known.

The relationship is: Ø = ATN (V/U)

The ATN function always returns an angle in the range of -90 < Ø < 90.

CPC Writer's Note

A divide by U = 0 error can occur when the net change in U displacement between points, or "ΔU" is zero. You should pretest ΔU before running this function. If ΔU is zero, do not run the ATN function. Instead, set an angle of 90 on positive ΔV, or an angle of 270 on negative ΔV. Run the function if ΔU is nonzero. Since ATN returns all angles in quadrants I and IV, test for angles in quadrants II and III as follows:

If the angle returned by ATN is zero or positive, check ΔV. If ΔV is also zero or positive, the angle is valid. If ΔV is negative, add 180 to the angle.

If the angle returned by ATN is negative, check ΔU. If ΔU is positive, the angle is valid. If ΔU is negative, add 180 to the angle.

The value of the argument can be -9999999 ≤ <argument> ≤ 9999999, but will be read as 7.0, 6.1...1.6 or 0.7 format.

A constant, variable, term, or expression can be used as the argument.

INT - INTEGERIZE FUNCTION

Syntax: INT,<argument>

When this function tests the value of its argument and finds that the decimal fraction is greater than .5000, the value is rounded to the next whole number. If its decimal fraction is less than .5000, the function truncates ("drops off") the decimal fraction. If the decimal fraction is exactly equal to .5000, then the nondecimal portion of the number is evaluated. If the nondecimal portion is odd, the value will be rounded up to the next whole number. If the nondecimal portion is even, the decimal fraction will be truncated.

The value .5001 would return 1.
The value .5000 would return 0.
The value 25.5000 would return 26
The value 26.5000 would return 26.
The value -.5001 would return -1.
The value -.5000 would return 0.
The value -12.5000 would return -12.
The value -13.5000 would return -14.

A constant, variable, term, or expression can be used as the argument.

FDW - FIX DOWN FUNCTION

Syntax: FDW,<argument>

When this function tests the value of its argument, it rounds down the decimal fraction.

The value .9999 would return 0.
The value 3.0001 would return 3.
The value 4.5000 would return 4.
The value 5.9999 would return 5.
The value -.9999 would return -1.
The value -3.0001 would return -4.
The value -4.5000 would return -5.
The value -5.9999 would return -6.

A constant, variable, term, or expression can be used as the argument.

FUP - FIX UP FUNCTION

Syntax: FUP,<argument>

When this function tests the value of its argument and finds a decimal fraction greater than .0000, it rounds the value up to the next whole number.

The value .0001 would return 1.
The value 3.0001 would return 4.
The value 4.5000 would return 5.
The value -.9999 would return 0.
The value -4.5000 would return -4.
The value -5.9999 would return -5.

A constant, variable, term, or expression can be used as the argument.

ALGEBRAIC HIERARCHY

When complex expressions must be defined, the following syntax specifies the order in which expressions will be evaluated by the CPC.

LEVELS OF PARENTHESES

When used, parentheses define lower and higher levels of operations. Data in the inner level of parentheses, level 4 shown below, will be evaluated first, and then level 3, then level 2, then level 1, and finally the data outside of level 1. The assignment statement below shows the maximum number of nested parentheses.

```
        D        1  2  3  4      4  3  2  1   D
Ndddd(A)(#XX=XXXX(XX(XX(XX(XXXXXX)XX)XX)XX)XXX)$
```

Parentheses shown with "D" labels are delimiters defining the start and end of the assignment statement. They do not represent a nesting level.

HIERARCHY OF OPERATIONS WITHIN LEVELS

Starting at the lowest level, (level 4 in the last example), operations are performed left to right as follows:

OPERATION	EXAMPLE
1. Functions are evaluated.	Sine, Cosine, Absolute Value
2. Factors are multiplied or divided	The "*" and "/" operators
3. Terms are added or subtracted	The "+" and "-" operators

After all levels of parentheses are evaluated:

4. Value is assigned to target variable	The "=" operator

INDIRECT VARIABLE ASSIGNMENT

In the previous examples, the "target" variable was defined directly by its number. The target variable can also be defined indirectly.

Given the following values: #2 is equal to 22
#16 is equal to 5

The statement: N20(A)(#(#2*3+2)=(4*#16))$ will assign a value of 20 to variable #68.

The value 20 results from the expression (4*#16).

Note that 68 is derived from (#2*3+2). Therefore #(#2*3+2) is variable #68.

Indirect variable assignment is useful, for example, when values must be assigned to a sequential group of variables.

Say that one variable in an expression, such as #1 is set to 50 to act as a base value in the variable expression #(#1+#2). Variable #2 can be initially set to zero, and then variable#2 can be incremented by (#2=#2+1) during each pass through a program loop. This results in an assignment to #50, then #51, #52. #53 etc. in each following run through the loop.

BRANCHING COMMANDS

Branching commands permit the normally sequential order of CPC execution to be altered by program instruction.

A branch instruction directs the Macro program to continue execution at an N sequence number identified within the instruction. The block or event programmed as a jump address in the branching instruction must reside within the same CPC. A runtime error occurs if no data is programmed in the block or event that is pointed to by the programmed jump address.

The Arithmetic event is used to program a branch statement. The branch can be of the type "unconditional" or type "conditional".

GO - UNCONDITIONAL BRANCH STATEMENT

An unconditional branching statement would be formatted as follows:

Statement: Ndddd.ddd(GO,<argument>)<CR><LF>

The argument represents a program sequence number[17] that will be the jump address. The block or event specified by the address may be located before or after this instruction in the CPC program. No return is implied by this instruction. The <argument> must be a numeric constant in 4.3 decimal[9] format. The character N is not used in the argument.

Example: N234(GO,1432) ;branch to N1432

IF - CONDITIONAL BRANCH STATEMENT

A conditional branching statement implies a test, and would be formatted off-line as follows:

Statement: ? exp. < 0 = 0 > 0

Ndddd.ddd(IF,<exp>,<arg1>,<arg2>,<arg3>)<CR><LF>

The value of expression <exp> is evaluated. If the value of <exp> is found to be less than zero, the program will jump to the N sequence number programmed as jump address <arg1>. If <exp> is equal to zero, jump address <arg2> will be used. If <exp> is greater than zero, jump address <arg3> will be used.

The <exp> can be a variable, term or expression[18]. Because of the comma delimiters, Factors or terms in the argument do not need to be enclosed within parentheses[19] . No spaces are permitted within <exp> or before the comma.

The jump addresses <arg1>,<arg2>, are each terminated by commas and <arg3> is terminated by ")" , the statement's right delimiter. Each must be a numeric constant in 4.3 decimal format. The N character is not used in any argument. No spaces are permitted within the statement's left or right delimiters.

Example: N420(IF,#4-1,430,440,430) ;comma ends <exp>

As the example shows, more than one argument can use the same jump address to provide "less than or equal", "greater than or equal" or "not equal" branching.

[17] Must be an absolute address.

[18] A constant is permitted here, but would force the statement to act as an unconditional branch instruction.

[19] Four levels of parentheses are permitted here.

Since all CPC variables are stored as real numbers, it is generally not good practice to use an IF statement to branch on strict equality. This is due to the fact that rounding or binary conversion errors may change the value of what you thought to be an integer (e.g. a counter) to a real number with a very small ($<10^{-5}$) decimal component. Testing for equality between two such numbers may then result in an erroneous branch.

Rather than testing for strict equality, try to use IF statements to branch on conditions such as "less than or equal to" or "greater than or equal to". For example, the statement N100(IF,#4-#5,110,110,230) will branch to N110 if the value of Variable #4 is less than or equal to the value of Variable #5 and to N230 if #4 is greater than #5. If you absolutely must check for equality, you can use nested IFs to check for a range of values surrounding zero:

```
N120(IF,#4-.00001,130,130,35)
N130(IF,#4+.00001,35,140,140)
N140
:
```

This sequence checks if the value of #4 is near 0 by first subtracting .00001 from it. If this result is less than or equal to 0, we know #4 ≤ .00001, so we check (at N130) to see if #4 is greater than -.00001 by adding .00001 to it. If this operation yields a positive number or 0, we have proved that #4 is within ±.00001 of zero and we may take care of the processing for this case (at N140). If either case fails, #4 is not near 0 and we branch to process the failure (at N35). You may, of course, check a closer range of values than ±.00001, but this should be adequate in most cases since the program resolution of the control is limited to .0001 inches or .001 mm.

GIN - CONDITIONAL BRANCH ON DC INPUT OR OUTPUT

This conditional branching statement tests the ON or OFF state of a selected digital input or output at the machine interface. It is formatted as follows:

Statement: ? I/O bit OFF ON

Ndddd.ddd(GIN,<iobit>,<arg1>,<arg2)<CR><LF>

The OFF or ON condition of the digital circuit at decimal address <iobit> from the DC or MTB interface is evaluated. If the condition of <iobit> is OFF, the program will jump to the N sequence number programmed as jump address <arg1>. If <iobit> is ON, jump address <arg2> will be used.

The I/O bit identifier is terminated by a comma. Interested customers should contact us for an I/O table directory, which will appear in our machine interface guides for systems with the PIC programmable interface controller. Any I/O bit in the PIC address range 0000. - 1535. may be accessed.

The jump address <arg1> is terminated by a comma and <arg2> is terminated by ")" , the statement's right delimiter. Each must be a numeric constant in 4.3 decimal format. The N character is not used in either argument. No spaces are permitted between the statement's left and right delimiters.

Example: N1720(GIN,<iobit>,1750,1800)

PROCEDURAL STATEMENTS

Procedural statements perform a specified task. Note that the procedural statement is another form of the Arithmetic event.

GAG - GET ACTIVE G-CODE

Use this statement to determine whether inch/metric and/or absolute/incremental G codes are active.

Statement: Ndddd.ddd(#dd=GAG,ARG1)<CR><LF>

ARG1 specifies whether the inch/metric, absolute/incremental or both G code(s) are to be examined. The table shows acceptable values for ARG1 and their effect on execution of the statement.

ARG1	EFFECT
1	Variable #dd is set to 0 if G70 (inch) is active and 1 if G71 (metric) is active.
2	Variable #dd is set to 0 if G90 (absolute) is active and 1 if G91 (incremental) is active.
0 or not programmed	Variable #dd will be set to reflect inch/metric status and variable #(dd+1) will be set to reflect absolute/incremental status.

Example: N220(#27=GAG,2)

In this example, ARG1=2 so we are checking the status of the absolute/incremental G code (G90/91). Variable #27 will be set to 0 if absolute (G90) is active and to 1 if incremental (G91) is active.

GAO - GET ACTIVE OFFSET

Use this command to find the number of an active T, H, D, E or Q code.

Statement: Ndddd.ddd(#dd=GAO,ARG1)<CR><LF>

This statement sets the variable specified by any variable #dd to the active tool, fixture offset or tool carousel number specified by ARG1.

ARG1 is the offset select letter code. This is the numeric code for the letter that normally represents the desired offset on your machine. It can be a constant, variable or the result of an expression. ARG1 accepts the following entries:

Enter 20 (T) to specify a T code.
Enter 17 (Q) to specify a Q code.
Enter 8 (H) to specify an H code.
Enter 5 (E) to specify an E code.
Enter 4 (D) to specify a D code.

Example: N250(#29=GAO,5)

In this example ARG1=5 which stands for Fixture offsets; the currently active Fixture offset select code would be stored in Variable #29 on execution of this event.

GOV - GET OFFSET VALUE

Use this command to gather data regarding a particular T, H, D, E or Q code.

Statement: Ndddd.ddd(#dd=GOV,ARG1,ARG2,ARG3)<CR><LF>

This statement sets the variable specified by any variable #dd to an offset value selected by ARG1, ARG2 and ARG3.

Each argument can be a constant, variable or the result of an expression.

ARG1 is the offset select letter code. This is the numeric code for the letter that normally represents the desired offset on your machine. ARG1 accepts the following entries:

Enter 20 (T) to access a T code.
Enter 17 (Q) to access a Q code.
Enter 8 (H) to access an H code.
Enter 5 (E) to access an E code.
Enter 4 (D) to access a D code.

ARG2 specifies the Tool or Fixture number that the offset should be retrieved for. It accepts the following entries:

Enter 1-199 for a particular T, H, or D code.
Enter 1-199 for a particular Q code.
Enter 1-32 for a particular E code.

ARG3 selects the dimension to be retrieved. It accepts the following entries:

With T code (ARG1 = 20) enter 4 to assign Tool diameter or 26 for Tool length.
With H code (ARG1 =8) always enter 26 for Tool length.
With D code (ARG1 = 20) always enter 4 for Tool diameter.
With Q code (ARG1=17) always enter 20.
With E code (ARG1 = 5) enter 24 to get the X axis dimension of the fixture, 25 for the Y axis dimension, or 26 for the Z dimension.

Example: N150(#37=GOV,20,9,4)

Here, ARG1=20, so the control will get a tool offset; ARG2=9 so it will look at Tool number 9; ARG3=4 so it will take the diameter. Therefore the diameter of tool number 9 will be stored in variable #37.

SOV - STORE OFFSET VALUE

Use this command to enter information into a particular T, H, D, E or Q code.

Statement: Ndddd.ddd(SOV,ARG1,ARG2,ARG3,ARG4,ARG5)<CR><LF>

This statement stores the value of a variable (given as an argument) into an offset specified by the other arguments.

Each argument may be a constant, a variable, or the result of an expression.

ARG1 is the offset select letter code. This is the numeric code for the letter that normally represents the desired offset on your machine. ARG1 accepts the following entries:

Enter 20 (T) to access a T code.
Enter 17 (Q) to access a Q code.
Enter 8 (H) to access an H code.
Enter 5 (E) to access an E code.
Enter 4 (D) to access a D code.

ARG2 specifies the Tool or Fixture number that the offset should be retrieved for. It accepts the following entries:

Enter 1-199 for a particular T, H, or D code.
Enter 1-199 for a particular Q code.
Enter 1-32 for a particular E code.

ARG3 selects the dimension to be retrieved. It accepts the following entries:

With T code (ARG1 = 20) enter 4 to assign Tool diameter or 26 for Tool length.
With H code (ARG1 =8) always enter 26 for Tool length.
With D code (ARG1 = 20) always enter 4 for Tool diameter.
With Q code (ARG1=17, always enter 20.
With E code (ARG1 = 5) enter 24 to assign the X axis dimension of the fixture, 25 for the Y axis dimension, or 26 for the Z dimension.

ARG4 is the value to be stored in the designated offset table for the parameter specified by ARG1, ARG2, and ARG3.

ARG5 is an optional entry which specifies whether the value given in ARG4 should be stored as absolute or incremental. If ARG5 is programmed as any nonzero digit, the offset value that is **already stored** in the parameter specified by ARG1-ARG3 is incrementally adjusted by adding the value given in ARG4 (offset value=offset value+ARG4). If ARG5 is 0 or not programmed, ARG4 is taken as an absolute offset value and it **replaces** the value currently stored in the specified offset.

Example: N200(SOV,20,5,26,#44,1)$

Here ARG1=20 so we are dealing with a Tool offset. ARG2 specifies Tool number 5; ARG3 = 26 which chooses the tool length. ARG4 specifies variable #44 and ARG5 is programmed 1 so the value of V44 will be taken as an incremental offset and added to the current value stored for the length of Tool number 5.

NOTE: The following SPP, TPP, SPD, TPD, and SPM instructions deal with CPC probing cycles. These commands cannot be executed if axis scaling, rotation or mirror imaging is active.

SPP - SPINDLE PROBE PROTECT[20]

Statement: Ndddd.ddd(SPP)<CR><LF>

Execution of this event causes the Spindle Probe to be placed in "stop on contact" mode. Use this command to protect the probe against unanticipated contact until you're ready to execute a probe sensing motion. If an unanticipated contact occurs with Probe Protect ON the control will halt axes motion and generate a flashing PROBE PROTECT Fault Display message. Jog will be enabled, permitting the probe to be moved away from the point of contact.

Fast debounce sampling logic in the control insures that electrical noise will not cause erroneous probe protect halts.

TPP - TABLE PROBE PROTECT[20]

Statement: Ndddd.ddd(TPP)<CR><LF>

This event is identical to SPP except that it applies to the Table Probe.

SPD - SPINDLE PROBE DEACTIVATE[20]

Statement: Ndddd.ddd(SPD)<CR><LF>

This event takes the Spindle Probe out of protect mode (cancels an SPP event). You would normally use this to cancel probe protect just before a probe sensing motion, then turn on protect as soon as the probe is withdrawn from the point of contact.

TPD - TABLE PROBE DEACTIVATE[20]

Statement: Ndddd.ddd(TPD)<CR><LF>

This event takes the Table Probe out of "stop on contact" mode (cancels a TPP event).

SPM - SPINDLE PROBE MEASURE[20]

Statement: Ndddd.ddd(SPM,EXP1,EXP2,EXP3,EXP4,EXP5)<CR><LF>

This statement moves the specified axis to a specified point. If probe contact occurs along the way, axis motion will be stopped, the part program coordinate[21] of the contact point is stored in a specified variable, and control is transferred to the next event. You must program a following withdrawal motion before you can turn probe protect on. If the endpoint is reached before contact is made, a NO PROBE CONTACT fault can be flagged, in which case control is *not* transferred to the next event. You must then reset and determine the reason contact was not made. Possible reasons include bad data in the macro, a logic error in the macro or a faulty part.

[20] Requires that optional probe, probe logic card and probe amplifier be installed.

[21] The part program coordinate(s) stored will include any G6/G92 Load Position or zero set presets but will exclude any PX, PY Set Up event offsets or cutter comp.

It is possible to set a parameter in this function so that a fault is not flagged if the probe reaches its endpoint without making contact. In this case control is passed to the next event once the specified point is reached.

Each EXP value may be a constant, variable or the result of an expression.

EXP1 specifies the axis number along which the probe is to move. It accepts the following entries:

Enter 1 to move along the X axis; enter -1 to move along X and store the contact point for each axis (see EXP3).
Enter 2 to move along the Y axis; enter -2 to move along Y and store the contact point for each axis (see EXP3).
Enter 3 to move along the Z axis; enter -3 to move along Z and store the contact point for each axis (see EXP3).

EXP2 specifies the endpoint where the probe is to stop if no contact is made.

EXP3 specifies the Symbolic Variable number at which the probe contact point value is stored. If EXP3 is a constant, variable #(EXP3) is used. If EXP3 is a variable number (#dd), the value stored in that variable number is used as the variable number in which the contact point will be stored[22]. If EXP3 is an expression, the expression is evaluated and the number obtained as its solution is the number of the variable in which the contact point will be stored. If EXP1 is negative (see above), all three axis contact points are stored:

The X contact point will be stored in the variable number specified in EXP3.
The Y contact point will be stored in #(EXP3+1).
The Z contact point will be stored in #(EXP3+2).

For example, if EXP3=56, X will be stored in V56, Y in V57 and Z in V58.
EXP4 specifies whether the contact point will be adjusted by datum offsets before being stored.

If EXP4 = 0 or is not programmed, the contact point is adjusted by the G0 datum offsets for the given axis before being stored.

If EXP4 <> 0, the contact point is stored unaltered by datum offsets.

EXP5 specifies whether or not a fault is flagged if the probe reaches the point specified by EXP2 without tripping. If EXP5 is excluded or programmed "0", the fault is flagged and part program execution is aborted. If it is programmed
non-zero, cycling will continue with the next statement if the probe reaches its
specified point without making contact.

Example: N170(SPM,1,34.5,29,1)

The X axis (axis number 1) will move from its current position to X34.5 and if the probe makes contact along the way the unaltered contact point will be stored in V29. If no contact is made, a fault will be flagged since EXP5 is excluded.

[22] For example, suppose that V37 holds a value of 20. If EXP3 is programmed "#37" (instead of "37") then V20 will store the contact point.

TPM - TABLE PROBE MEASURE[20]

Statement: Ndddd.ddd(TPM,EXP1,EXP2,EXP3,EXP4)<CR><LF>

This statement is identical to the Spindle Probe Measure statement except that it is used with a Table Probe.

SPO - GET SPINDLE PROBE DATUMING OFFSETS[20]

This command allows you to fetch datuming offsets obtained from the Probe event's G0 spindle probe datuming cycle.

Statement: Ndddd.ddd(#dd=SPO,EXP1)<CR><LF>

EXP1 allows you to specify a positive or negative axis whose offset will then be stored in #dd. Use +1 to specify +X, -1 for -X, +2 for +Y, -2 for -Y or -3 for -Z.

If EXP1 is not programmed, this statement gets all the Spindle Probe offsets and stores them at the five consecutive Symbolic Variables starting at #dd. The +X offset is stored at #dd, -X is stored at #(dd+1), +Y is stored at #(dd+2), -Y is stored at #(dd+3), and -Z is stored at #(dd+4).

TPO - GET TABLE PROBE DATUMING OFFSETS[20]

This command fetches the datuming offsets from the G8 datuming cycle.

Statement:

Ndddd.dd(#dd=TPO)<CR><LF>

Since the Table Probe only measures in -Z, the Z offset will be stored at #dd.

NOTE: *For this and the following"EP" and "CP" statements it is important to note that **offsets** include tool compensations (T, H, D codes), fixture offsets (E codes), and Set Up mode Part Offsets (PX, PY).*

EPD - GET AXIS END POINT IN POS. DISPLAY COORDS.

Statement: Ndddd.ddd(#dd=EPD,n)<CR><LF>

If n is 1, 2, or 3, this statement gets the endpoint of the last motion event for axis number n in Position Display coordinates and stores it in #dd. If n is 0 or not programmed, #dd will get the X axis coordinate, #(dd+1) will get Y, and #(dd+2) will get Z.

Position Display Coordinates reflect offsets (as defined above), axis scale, rotation and mirror imaging, G6/G92, and Part Program Zero Set.

EPP - GET AXIS END POINT IN PART PROG. COORDS.

Statement: Ndddd.ddd(#dd=EPP,n)<CR><LF>

Gets endpoint of last motion event for axis number n in Part Programming coordinates and stores it in #dd. If n is 0 or not programmed, storage will occur as described above.

Part Program Coordinates reflect G6/G92 and Part Program Zero Set but **do not** reflect offsets, cutter comp or axis scale, rotation or mirror imaging.

EPZ - GET AXIS END POINT IN MACHINE COORDS.

Statement: Ndddd.ddd(#dd=EPZ,n)<CR><LF>

Gets endpoint of last motion for axis number n in Machine coordinates and stores it in #dd. If n is 0 or not programmed, #dd will get the X axis coordinate, #(dd+1) will get Y, and #(dd+2) will get Z.

Machine Coordinates reflect the distance, along each axis, from machine coordinate zero.

CPZ - GET CURRENT AXIS END POINT IN MACHINE COORDS.

Statement: Ndddd.ddd(CPZ, arg1, agr2)<CR><LF>

Gets the current position (in machine coordinates) of the axis number specified in argument 2 and stores it in the variable specified in argument 1. If argument 2 is 0 or not programmed, the variable specified in arg1 will get the X axis coordinate, #(arg1+1) will get Y, and #(arg1+2) will get Z, etc. until coordinates for axes have been stored.

Machine Coordinates reflect the distance, along each axis, from machine coordinate zero.

PCP - GET LAST PROBE CONTACT POINT IN PART PROGRAM COORDINATES[20]

Statement: Ndddd.ddd(#dd=PCP)<CR><LF>

Get last probe contact point[23] in Part Program coordinates and store it in #dd.

PCZ - GET LAST PROBE CONTACT POINT IN MACHINE COORDINATES[20]

Statement: Ndddd.ddd(#dd=PCZ)<CR><LF>

Get last probe contact point[23] in Machine coordinates and store it in #dd.

SON - SET DC OUTPUT "ON" STATEMENT

Statement: Ndddd.ddd(SON,<outbit>)<CR><LF>

This statement sets one of 32 selected DC output circuits at the machine interface to the ON condition.

The <outbit> identifier specifies the decimal address of one of 32 CPC outputs in the address range 0768.-0799. (decimal). Interested customers should contact us for an I/O table directory, which will appear in our machine interface guides for systems with the PIC Programmable Interface Controller.

SOF - SET DC OUTPUT "OFF" STATEMENT

Statement: Ndddd.ddd(SOF,<outbit>)<CR><LF>

This statement clears one of 32 selected DC output circuits at the machine interface to the OFF condition.

The <outbit> identifier specifies the decimal address of one of 32 CPC outputs in the address range 0768.-0799. (decimal). Interested customers should contact us for an I/O table directory, which will appear in our machine interface guides for systems with the PIC Programmable Interface Controller.

[23] The point coordinates **do not reflect the probe's G0 datum offsets.**

MSG - DISPLAY "RUNTIME" OPERATOR MESSAGE

Statement: 16 characters

Ndddd.ddd(MSG,<charstring>)<CR><LF>

This statement displays an operator information message on the video screen. The argument <charstring> may be up to 16 characters in length as shown by underline in the statement.

Once executed, the <charstring> message appears as up to 16 characters on the right column display of the Auto or Single Event program execution modes below the normal Text event display field, and also on the large character Position Display screen to the left of any Text event data.

As the CPC writer, you may prefer to use the MSG statement instead of the Text event to alert the operator that the CPC program sent the message.

The message will remain displayed until replaced by a following message, or the message is "erased".

The statement: Ndddd.ddd(MSG) with no comma or <charstring> can be used to erase a displayed message.

ERR - ERROR CONDITION WITH DISPLAY MESSAGE

Statement: 32 characters

Ndddd.ddd(ERR,<charstring>)<CR><LF>

Execution of this statement actuates the flashing FAULT / STATUS display screen to alert the operator that something of note has been detected, for example, by a conditional test statement in the CPC program.

THE ERROR MESSAGE DISPLAY DOES NOT SET A FEEDHOLD OR E-STOP CONDITION.

If as the CPC writer you consider this to be appropriate reaction, program a following block to turn off the spindle and halt the program with an M05 Spindle Stop, M00 Program Stop sequence.

The message, up to 32 characters as shown by underline in the statement, will be displayed on the Fault screen with any other active Fault messages when the operator presses the FAULT/STATUS key.

The operator may respond to an error in either of the following ways:

1) The operator can press RESET to abort the program.
2) The operator can ignore the display and resume cycling.

If cycling continues, the operator's action can be anticipated by:

1) an M02 or M30 end of program code[24]
2) a CLR command from the CPC program (see next statement)
3) another ERR command from the CPC program
4) a jump back into the fault checking routine in the 3rd and 4th case the flashing Fault display will be regenerated.

Example

```
N0060(IF,#46,65,70,70)
N0065(ERR,V46 NEGATIVE)
N0066(9)M00$
N0067(#46=-#46)
N0070(CLR)
```

If variable #46 < 0, set V46 NEGATIVE fault with following Program Stop. On following Cycle Start, set V46=-V46 and continue the program in the normal way[25].

CLR - CLEAR ERROR CONDITION AND DISPLAY

Statement: Ndddd.ddd(CLR)<CR><LF>

This command cancels any active Fault condition and the flashing Fault display.

Note:The statement Ndddd(ERR,) with a space between the comma and right parenthesis will also clear an error condition, but not the flashing display until the operator presses the FAULT/STATUS key.

A CLR command, when no fault condition exists, can act as a "No Operation" statement.

[24] In this case an"End of Program" fault will be set if M02 or M30 is executed while a Fault Message is still active.

[25] While in some cases it is desirable to correct the error internally, if the error is in or has been caused by a user-entered value it is generally best to force the user abort the program and correct the entry. In cases like our example, you may save a few lines of code by using the ABS function on the value in question when it is not necessary to inform the user of the error.

PRT - PRINT TEXT MESSAGE

Statement: 80 characters

Ndddd.ddd(PRT,<charstring>)<CR><LF>

On execution, this statement will send its <charstring> argument to the RS-232C or optional RS 422 port. The message does not appear on the display screen as it is sent to the port. The argument, as shown by underline, can be up to 80 characters, and is terminated by the statement's right parenthesis.

CPC program execution will be suspended until <charstring> is sent out through the serial port. This means that you will not want to execute this command while the tool is in stock, or a dwell mark could occur. The axes Servo loops are closed while the data is being output from the port.

The data is output as RS-358 ASCII (7 bit) code. Parity and Baud rate will be the last entries selected from Load or Record mode. If no Baud rate is selected,

PRINT WAIT: NO BAUD RATE

will appear on the display. The operator can recover by pressing MOTION STOP, move to Transmit mode and set the Baud rate, return to Auto mode and press CYCLE START. Alternately, the output can be skipped by pressing the ABORT EVENT key after MOTION STOP, and execution resumed with the CYCLE START button.

If a Terminal mode, Receive or Transmit communication is attempted while the primary port is controlled by the CPC, the control will note the conflict in its COMMUNICATION LOG (see Terminal mode description) and the message PRINT WAIT: TERMINAL MODE or PRINT WAIT: TRANSMIT/RECEIVE will appear. You can allow the external communication to continue or abort it if you wish. The CPC program will wait, then resume after Terminal mode is done or you exit concurrent Receive or Transmit mode. If you wish to abort the waiting CPC print command, press MOTION STOP. Then go to Auto mode, press ABORT EVENT, then CYCLE START to resume the CPC without the pending print statement.

Note: If you are regularly handling external I/O or only occasionally will want to output print statements, use Block Delete to skip the Print statement. See the Program mode section in the Delta Series Customer Information Manual for more on Block Delete.

Examples:

```
N1234(PRT,VI5=EXPECTED CORNER X, V16=EXPECTED CORNER Y)
N1235(PRT,V17=CORNER RADIUS  R)
```

PSV - PRINT VARIABLES STATEMENT

Statement: one to five arguments permitted

```
Ndddd.ddd(PSV,<arg1>,<arg2>,<arg3>,<arg4>,<arg5>)<CR><LF>
```

The arguments here identify variables, the value of which will be sent to the serial port. Communication parameters must be preset as described in the previous section. One to five variables can be assigned as arguments. The PSV command and each following argument except the last must be immediately followed by a comma. The last argument is terminated by the statement's right parenthesis. The following example shows the assignment of several variables as arguments.

```
N1511(PSV,42,#30,1,#9+5,19)
```

<Arg1> print variable #42 and its value.
<Arg2> the value of variable #30 is the variable # to be printed with its value
<Arg3> print variable #1 and its value
<Arg4> 5 added to the value of var #9 is the variable # to be printed with its value <Arg5>
print variable #19 and its value

For convenience, a four digit number is printed as a record number counter before the variables and their values.

The statement: Ndddd(PSV)

should be programmed before N1511, for example, to set the record number counter to 0000 before any variables are output. The first output of N1511 would be with record number 0000. The next PSV statement will be output with record number 0001.

Examplc

```
N1510(PSV)
N1511(PSV,#30,42.1)
```

where:

Variable #1 = 50
Variable #8 = 81
Variable #30=8
Variable#42=-1000000

The output would be:

```
0000 V8+81.00000 V42-1000000 V1+50.00000<CR><LF>
```

Note that the decimal point is located according to the magnitude of the nondecimal portion of the number to be output.

If the symbolic variable number specified (or indirectly addressed) does not exist, the control will output all the variables up to that variable, then it will output a carriage return and linefeed. The nonexistent number and all numbers following it will be ignored.

CPC COMMAND SET SUMMARY SHEET

trig	SIN	sine	agr1	returns the sin of an angle (angle is in decimal degrees)	(#25=SIN,30) (#26=SIN,(#33+90))
trig	COS	cosine	arg1	returns the cos of an angle (angle is in decimal degrees)	(#22=COS,60) (#23=SIN,(#33-90))
trig	TAN	tangent function	arg1	returns the tan of an angle (angle is in decimal degrees)	value cannot be ...-270, -90, +90, +270...
trig	ATN	arc-tangent funciton	arg1	returns the arctan of a number (number cannot equal 0.0)	angle returned always between -90° and +90°
math	SQR	square root	arg1	returns the square root of any positive number	(#80=SQR,4.32) (#81=SQR,(#1*#1+#2*#2))
math	ABS	absolute value	arg1	returns a number's absolute value	(#43=ABS,-12.54) (#56=SQR,(ABS,#81))
math	INT	integerize function	agr1	rounds value off to nearest integer. (.5>>0, 1.5>>2)	(#78=INT,#21)
math	FDW	fix down	arg1	rounds value down to the lower integer. (.5>>0, 1.5>>1)	(#79=FDW,#22) (#76=FDW,(#22+11*.123))
math	FUP	fix up	agr1	rounds value up to the higher integer. (.5>>1, 1.5>>2)	(#55=FUP,#11) (#56=FDW,(1.234+#12))
branching	GO	unconditional goto	arg1	N Number in macro to jump to.	(G0,998)
branching	IF	conditional branch	agr1 arg2 arg3	N number to jump to if expression is less than zero N number to jump to if expression is equal to zero N number to jump to if expression is greater than zero	(IF, #12-4,30,40,999) if #12 is less than 4.0 goto N30, if #12 = 4.0 goto N40, if #12 is greater than 4.0, goto N999.
branching	GIN	conditional branch on DC I/O	iobit arg1 arg2	DC I/O bit to examine D20 (000-639) D30 (000-1535) N number to jump to if I/O bit is 0. N number to jump to if I/O bit is 1.	(GIN, 128,500,999) if I/O bit 128 is low (0), go to N500. if I/O bit 128 is high (1) go to N999.
probe	SPP	spindle probe protect		turns spindle probe protect On	(SPP)
probe	TPP	table probe protect		turns table probe protect On	(TPP)
probe	SPD	spindle probe deactivate		turns spindle probe protect Off	(SPD)
probe	TPD	table probe deactivate		turns table probe protect Off	(TPD)
probe	SPM	spindle probe measure	arg1 arg2 arg3 arg4 arg5	probe axis 1=X, 2=Y,3=Z. If minus,store XYZ endpoint where to stop if no contact is made variable number where contact point(s) are stored 0=store with G0 datum offset. 1=store without offset if no contact is made: 0=fault message, 1=continue	If arg1 is negative (-1, -2, -3) store XYZ contact point in the three consecutive addresses beginning with the variable specified in arg3. Arg4 only affects the axis specifed in arg1.
probe	TPM	table probe measure	arg1 arg2 arg3 arg4	probe axis 1=X, 2=Y,3=Z. If minus,store XYZ endpoint where to stop if no contact is made variable number where contact point(s) are stored 0=store with G0 datum offset. 1=store without offset	If arg1 is negative (-1, -2, -3) store XYZ contact point in the three consecutive addresses beginning with the variable specified in arg3. Arg4 only affects the axis specifed in arg1.

probe	SPO	get spindle probe datuming offsets	var #
			exp1
				variable # where result is stored
				1=+X, -1=-X, 2=+Y, -2=-Y, -3=-Z
					If exp1 is not programmed, store the 5 offsets at the 5 consecutive variables starting at var#

probe	TPO	get table probe datuming offsets	var	variable # where -Z table probe datum offset is stored	(#40=TPO)

CPC COMMAND SET SUMMARY SHEET

offset	GAG	get active G code	var#
			arg1
				variable # where result is stored
				1= G70/G71, 2=G90/G91, 0 = both. G70=0, G71=1 G90=0 G91=1
					if arg1=1 put G70/71 in var# if arg1=2 put G90/91 in var# if arg1=0 put G90/91 in var#, G70/71 in var#+1

offset	GAO	get active offset code	var#
			arg1
				variable # where result is stored
				20=T code, 17=Q code, 8=H code, 5=E code, 4=D code
					(GAO,20) - get active T Code (GAO,5) - get active E Code

offset	GOV	get offset value	var#
			arg1
			arg2
			arg3
				variable # where result is stored
				20=T code, 17=Q code, 8=H code, 5=E code, 4=D code
				offset number T,H,D (0-199), Q (0-99), E (1-32)
				T,H,D (4=dia, 26=length), Q (20), E (24=X, 25=Y,26=Z)
					N150(#37=GOV,20,9,4) get T09 diameter
					N230(#41=GOV,5,14,24) get E14 X dim
					N230(#42=GOV,5,14,25) get E14 Y dim
					N230(#43=GOV,5,14,26) get E14 Z dim

offset	SOV	store offset value	arg1 arg2 arg3 arg4 arg5	20=T code, 17=Q code, 8=H code, 5=E code, 4=D code offset # - T,H,D (0-199), Q code (0-199), E code (1-32) T,H,D (4=dia, 26=length), Q (20), E(24=X, 25=Y,26=Z) value (dimension) to store or add 0 = replace existing value, 1=add to existing value	(SOV,20,3,4,.500,0) - store .500 dia. for tool #3 (SOV,20,3,26,-.0024,1) - subtract .0024 from tool #3 stored length.
get probe contact:	PCP	in part program coordinates	var #	variable # where last probe contact point is stored	store without G0 Datum offsets
get probe contact:	PCZ	in machine coordinates	var #	variable # where last probe contact point is stored	store without G0 Datum offsets

get endpoint :	EPD	in position display coordinates	var # n	variable # where result is stored
				1=X, 2=Y, 3=Z, if 0 store XYZ at consecutive variables
					reflects T code, E code, part offset, scale, rotation, G6/G92, zero set, cutter comp, mirror.

get endpoint :	EPP	part program coordinates	var # n	variable # where result is stored
				1=X, 2=Y, 3=Z, if 0 store XYZ at consecutive variables
					reflects G6/92, zero set, but not T code, E code, part offset, scale, rotation, cutter comp, mirror.

get endpoint :	EPZ	machine coordinates	var # n	variable # where result is stored
				1=X, 2=Y, 3=Z, if 0 store XYZ at consecutive variables
					N300(#77=EPZ,3) store Z position in var #77 if arg1=0 store all axes at consecutive variables

get endpoint :	CPZ	current machine coordinates	arg1 arg2	variable where machine position is stored 1=X, 2=Y, 3=Z, 4=4th, 5=5th	N300(CPZ,24,3) store Z position in var #24 if arg2=0 store all axes at consecutive variables
misc.	SON	set DC output On	outbit	Turn on I/O bit: D20 (0384.-0415), D30 (0768.-0799.)	(SON,784)
misc.	SOF	set DC output Off	outbit	Turn off I/O bit: D20 (0384.-0415), D30 (0768.-0799.)	(SOF,785)
misc.	MSG	display runtime message	chars	up to 16 characters that appear on Auto/Single modes	(MSG, CALCULATING)
misc.	ERR	display error message	chars	up to 32 chars that appear on Fault/Status screen.	this message does not stop execution
misc.	CLR	clear error and display	chars		(CLR)
misc.	PRT	print text message	chars	up to 80 characters are output through serial port.	
misc.	PSV	print variables statement	arg1 arg2 arg3 arg4 arg5	first item to print second item to print third item to print fourth item to print fifth item to print	see manual for explanation (each statement goes out with a record number)

INTRODUCTION

This section explains how to write a CPC and includes several example CPCs.

DEVELOPING A CPC

The control's full featured Macro programming language allows the use of symbolic variables, conditional and unconditional branching, and numerous library functions. Developing useful Macros with it requires some preparation and planning.

You must gain a familiarity with any programming language before you attempt to use it and CPCs are no exception. Read through Parts 1 and 2 of this document carefully; they describe Macro syntax in detail.

Once familiar with the language, there are a number of steps to follow that can take the confusion out of writing a CPC. These steps are:

1. DEFINE THE PART

What is the shape of the part? What are its constant characteristics? Are there features or dimensions which may vary from job to job? If so, which ones? Can the task of programming this part be subdivided[1] (i.e. can submacros[2] or subroutines be used to cut a repeated shape)? Are there any machine or material related limitations that must be considered?

2. DETERMINE HOW TO CUT THE PART

Here you must choose the method by which the part will be machined. If there is more than one way to cut the part, which is more efficient? Which is better suited to the job you have in mind? Macros are especially useful when the part (or elements of it) can be defined mathematically (as in, for example, a sphere). Is this true of your part? Are any special tool compensations necessary? If you are nesting macros, now is the time to develop a preliminary algorithm or flowchart for their calling order. You should apply the following steps to each individual submacro without losing sight of the part as a whole. If you are using a macro with nested loops or subroutines, you should develop a preliminary algorithm for its operation. Take a modular approach to the design of *any* macro by grouping similar tasks together as often as possible. These modules can be code segments, subroutines or submacros; the important thing is that as separate entities they will be easier to develop and test. When each is done, they can be assembled into the main macro along with any necessary transitional code.

3. ASSIGN VARIABLES TO THE PART GEOMETRY

Here you must assign variables to the part dimensions that may change, as determined in step 1. It's also a good idea to note any constants.

[1] If you use nested macros to program sub-parts, be sure not to nest them more than four deep.

[2] A submacro is one which is called from within another macro (nested).

4. DETERMINE ERROR CONDITIONS

Do any of the variables have to be in a certain range of values (e.g. positive, negative, <>0)? What about the way they relate to each other (e.g. if you have a Z reference plane and a Z depth, the depth must be below the reference).

5. ASSIGN GLOBAL AND LOCAL VARIABLES AND EXPAND ALGORITHM

Global variables will be entered by the user. Since they are entered by letter in the calling program, it is wise to choose letters for your global variables that are representative of the variable's use; for instance, choose D for Direction, T for Tool number, etc.

Assigning local variables requires that you have completely thought-out how the part will be machined (step 2). It is wise to pick sequential variable numbers for local variables which are related. Since you'll usually find it necessary to add (and possibly remove) some Locals while writing and testing the macro, it is also wise to leave a few empty variables between your groups so you can add Locals to a group without breaking it up.

In the process of assigning local variables you will likely derive equations that define their contents. Use these equations in an expanded algorithm that also includes all foreseeable loops and branches. Do not be concerned with loop constructs or with using CPC language at this point; write the algorithm in plain English. At this level letter identifiers or even brief descriptions of the variable can be used in place of symbolic variable numbers. An example outline is shown later in this section.

If you are developing separate submacros and/or modules, construct an algorithm for each and write the code for each in this step. This will make step 6 a simple joining operation.

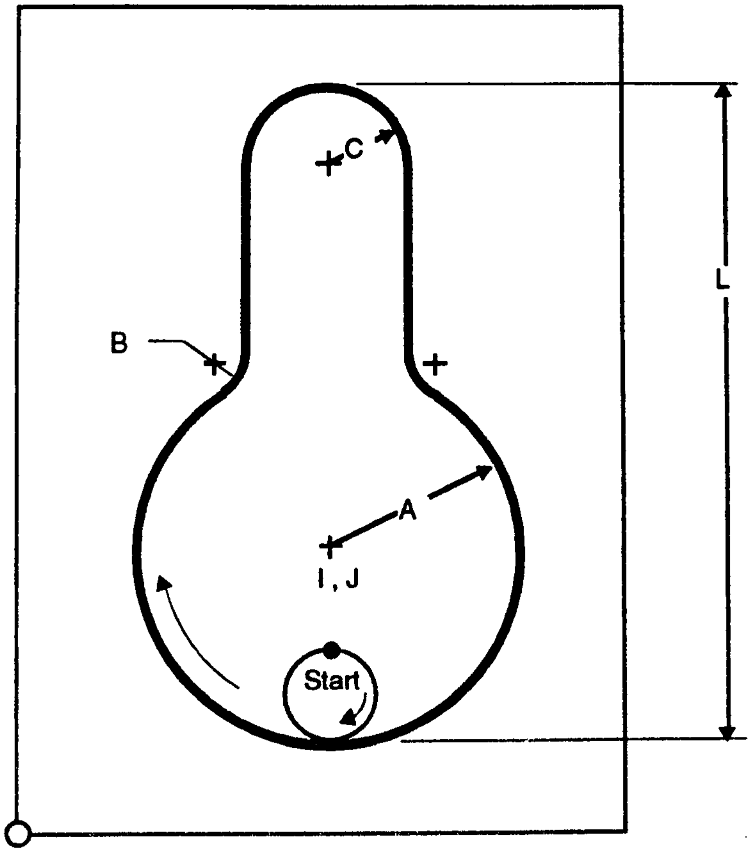
6. ASSEMBLE YOUR CPC

Now you are ready to join your various modules into a contiguous program. Write any joining code that is necessary (i.e. submacro or subroutine calls, main loops, initialization procedures) and renumber any module lines so that you get one block of code that can be called as a main macro from a part program. If your macro does not contain submacros and modules, this step will consist of simply translating algorithm(s) developed in step 5 into efficient macro code.

After this step you may proceed with testing and debugging.

EXAMPLE WITH CUTTER COMPENSATION

This example shows how cutter compensation can be used in a Customer Prepared Cycle that is designed for a Delta 40/50/60[3] control. The part milled by this CPC will always have the general shape shown below. The part's dimensions are specified by the variables (A, B, C, I, J, etc.) that are passed to the CPC from the main program.



Before starting the program, lets review the rules of using cutter compensation within a CPC.

1) Cutter compensation must be turned on and off within the CPC.
2) Once cutter compensation is active you may only program Linear and Arc Mill events until cutter compensation is turned off.
3) Variables and/or expressions may be used to describe X, Y, and Z endpoints while cutter compensation is active. The code segments below show valid and invalid uses of variables after cutter compensation has been turned on.

Valid sequence:

```
N110 (1) X#2 Y#3 Z-#22
N120 (1) X(#20*.123) Y(#21+(#3*#9))
N130 (1) X0 Y#88
```

Invalid sequence:

```
N110 (1) X#2 Y#3 Z-#22
N120 (A) (#99=#21+(#3*#9))
N130 (1) X(#20*.123) Y#99
N140 (1) X0 Y#88
```

The code segment at the right will cause cutter compensation to lose "look ahead" because it does not follow rule number 2.

[3] The ability to use cutter compensation within a CPC is a feature that is unique to D30 and D40/50/60 controls. It is not available with the D20 CPC feature.

PLANNING THE PART

This example will follow the six step development process that is described earlier in this section.

1) Define the part.

See illustration on the previous page.

2) Determine how to cut the part.

We will keep this CPC simple to highlight the use of cutter compensation. The tool motion will be commanded as follows:

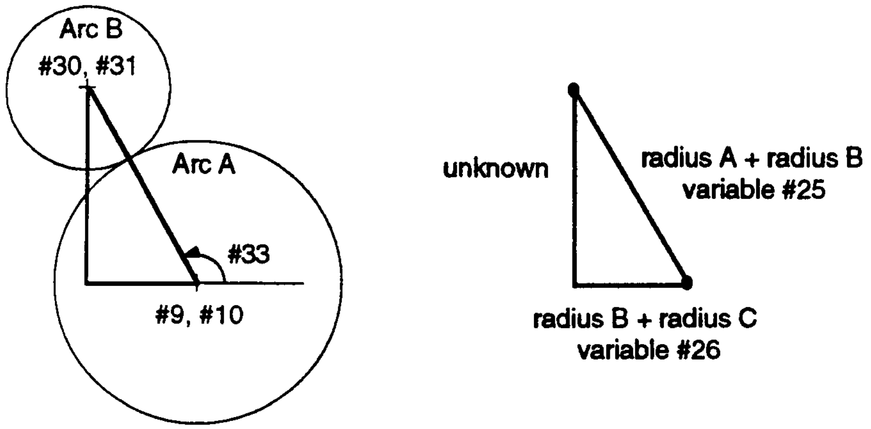
- plunge to depth at the start point (see label on the part print, previous page).
- turn on cutter compensation and tangentially enter the profile.
- feed in a clockwise direction around the part
- tangentially exit the profile and turn cutter compensation off.

3) Assign variables to the part geometry.

A.....	#1...	radius of lower arc
B.....	#2...	radius of blending arc
C	#3...	radius of upper arc
F.....	#6...	feedrate
I......	#9...	center X of lower arc
J	#10..	center Y of lower arc
H.....	#8...	arc radius for tangential entry into profile
L.....	#12..	length of part
Z.....	#22..	depth of cut (program assumes that top of part has the coordinate Z0.

The first thing that we notice about the keyhole print is that some geometric data must be calculated by the CPC. Most of the math is very simple (such as adding to variables together to obtain an endpoint coordinate) and will be handled on the fly. However, the math to locate the tangency point between the first two arcs is more complex - thus will be handled at the start of the CPC. As a general rule, all calculations that require storage of an intermediate value should be performed before the CPC commands any tool motion.

The illustration **below** shows the foundation for the calculations that are performed by the CPC. With the goal being to calculate the end angle for the first arc and store it in variable #33, the CPC firsts solves the right triangle whose hypotenuse starts at the center of arc A and ends at the center of arc B. Once all sides of the triangle are known, angle #33 can be found with the arctangent function. The CPC can now position the tool to the start point, activate cutter compensation then mill the part.



4) Determine error conditions.

The error checking portion of this CPC appears after the M30-end of program command and is called by a jump statement at N5. Notice that an M00-program halt command follows each ERR instruction since the the ERR command does not cause the control to drop out cycle.

5) Write the CPC program.

```
(KEY)
N001 (T) START CPC$
N005 (GO,2000)$                                - jump to error checking section
N010 (#30=#9-#3-#2)$                           - calculate X center of arc B
N015 (#25=#1+#2)$                              - hypotenuse of triangle
N016 (#26=#2+#3)$                              - horizontal side of triangle
N020 (#31=#10+SQR,((#25*#25)-(#26*#26)))$      - calculate Y center of arc B
N030 (#33=ATN,((#31-#10)/(#30-#9))+180)$       - find end angle of arc A
N040 (0) X#9 Y(#10-#1+#8*2) Z.1$               - rapid to start point
N045 (1) Z#22 F#6$                              - feed to depth
N050 (2) A90 B45 R#8 D0 C0$                    - turn cutter comp off
N060 (2) A45 B270 R#8 D0 C2$                   - turn comp on, tangential entry
N070 (2) A270 B#33 R#1 D0$                     - mill first arc (polar coordinates)
N080 (2) I#30 J#31 X(#30+#2) Y#31 D1$          - mill second arc (Cart. coords)
N090 (1) Y(#10+#12-#3)$                        - linear mill
N100 (2) A180 B0 R#3 D0$                       - arc mill (polar coordinates)
N110 (1) Y#31$                                 - linear mill
N120 (2) A180 B(360-#33) R#2 D1$               - arc mill (polar coordinates)
N130 (2) I#9 J#10 X#9 Y(#10-#1) D0$            - arc mill (Cartesian coordinates)
N140 (2) A270 B135 R#8 D0$                     - tangential exit away from profile
N150 (2) A135 B90 R#8 D0 C0$                   - comp off complete tangential exit
N160 (0) Z.1$                                  - retract
N999 (9) M30$                                  - end of CPC
```

```
N2000(T)START ERROR CHECK$

N2010 (IF,#1,2020,2020,2050)$            - check value of A
N2020 (ERR,RADIUS A <= 0.0)$             - generate error if ≤ 0
N2030 (9) M00$                           - program halt
N2040 (GO,1)$                            - return to beginning

N2050 (IF,#2,2060,2060,2110)$            - check value of B
N2060 (ERR,RADIUS B <= 0.0)$             - generate error if ≤ 0
N2070 (9) M00$
N2080 (GO,1)$

N2110(IF,#1,2120,2120,2150)$             - check value of c
N2120(ERR,RADIUS C <= 0.0)$              - generate error if ≤ 0
N2130(9) M00$
N2140(GO,1)$

N2150 (IF,#1-#8,2160,2190,2190)$         - check value of H and A
N2160 (ERR,TAN RADIUS H > RADIUS A)$     - generate error if H>A
N2170 (9) M00$
N2180 (GO,1)$
N2190 (GO,10)$                           - no errors: resume with CPC
```

Main calling program

```
(KEYHOLE)
N010  (9)  T01 S1250 M03                 •
N020  (1)  X0/ F13                       - set modal feedrate
N030  (M)  A4 B.67 C1 F14 H.75 I2        - Macro call event
           J4 L4 Z-1.25 PKEY
N040  (9)  M30                           - end of program
```

Notice that N10 in the main calling program will have to be modified if your control uses H and D codes. Refer to Part 1-Section 1.

6) Assemble your CPC code.

This step is not applicable for this short CPC.

HEMISPHERE MACRO

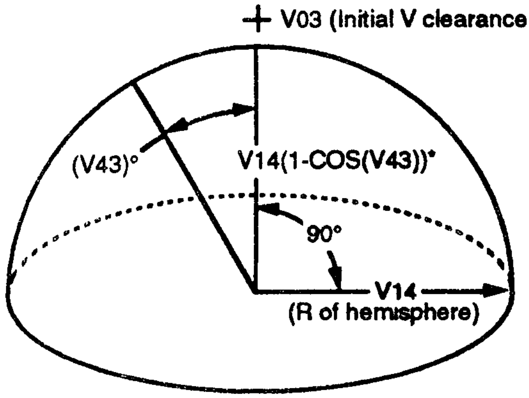
This macro cuts a hemisphere by milling successively larger arcs. It begins milling the arcs at the programmed Z first-pass depth (global variable C) and continues until a complete 90° hemisphere is cut (therefore the total Z depth of the hemisphere = its radius). The increasing size and depth of these arcs is controlled by the angle between the hemisphere center and its shell (held in the accumulator V43). This angle is incremented by a value passed as a parameter from the calling program. As the angle increases, its sine (which determines the arc mill radius) also increases; its cosine (which determines the distance between the hemisphere center and the plane of the arc mill) decreases, thereby increasing the Z depth of the arc mill. *This macro assumes that spindle speed is set prior to its calling. Also, it makes no cutter compensation.*

Global and local variable definitions follow.

SYMBOLIC VARIABLE NUMBER/LETTER	PURPOSE
V1/A	Incremental angle per Z pass.
V3/C	Z clearance position.
V4/D	Direction of arc mill: D0=CW, D1=CCW.
V6/F	Programmed feedrate[4].
V9/I	X-axis arc center.
V10/J	Y-axis arc center.
V14/R	Radius of hemisphere[5].
V22/Z	Z-dimension of first Z pass.
V41	Number of Z passes
V42	Corrected Z increment
V43	Total angle between hemisphere center and present cutter position
V44	Z depth of following arc mill

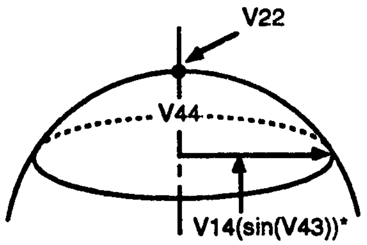
The illustration **below left** shows how a few of the variables are used to describe the complete hemisphere. Be careful not to confuse the Macro local variable V43 - which gives the total angle between the hemisphere center and shell – with the global variable V01 (A), which gives the incremental angle per Z pass.

OVERVIEW



*when V43=90°, 1-cos(V43)=1 and V14x1=V14, which is the Radius of the hemisphere.

CUTTING A COMPONENT



*We add this to V9 to get the X start/ end point of the arc mill. The Y dimension is determined by V10 and the Z depth by V22-V44.

[4] Be sure this conforms to the "dddd.d" format for feedrate specified in this manual.

[5] Not to be confused with the radius of each arc mill, which is computed for each pass in the macro.

The illustration **previous page, right** shows how the start point is determined for an individual arc mill. Since this is a 360° mill, its start point = its end point. The control can compute its radius by subtracting the I center dimension (V9) from the X start point (V14(sin(V43))).

The general algorithm follows.

```
MAIN PROGRAM calls
    HSPH-MAC which
        corrects the Z increment
        positions to the Z clearance
        while number of passes >0:
            computes Z depth of next pass
            mills next pass (arc event)
            increments the total angle between the hemisphere center and shell
            decrements the number of passes
        if number of passes=0 then:
            set angle =90°
            mill final arc
    End Macro;
End Program.
```

No error checking routines are provided with this macro.

The macro code follows, formatted and commented as before.

CODE	COMMENTS
N010(#41=1+FUP,(90/#1))	;#41 IS THE LOCAL VARIABLE CONTAINING THE NUMBER OF PASSES. THIS OPERATION DIVIDES 90° (THE ANGLE BETWEEN THE HEMISPHERE'S BASE AND CENTERLINE) BY #1 (THE INCREMENTAL ANGLE PER Z PASS) TO GIVE THE NUMBER OF PASSES, WHICH IS INTEGERIZED BY THE FUP FUNCTION AND STORED IN #41.
N011(#42=90/(#41-1))	;HERE THE INCREMENTAL ANGLE PER Z PASS IS RE-DERIVED AND STORED IN #42.
N020(#43=0)	;#43 IS INITIALIZED; IT WILL LATER STORE THE TOTAL ANGLE BETWEEN THE HEMISPHERE CENTER AND THE PRESENT CUTTER POSITION.
N040(0)Z#3$	;POSITION Z TO THE CLEARANCE DEPTH STORED IN #3.
N050(#44=#22-#14*(1-COS,#43))	;#44 IS THE Z DEPTH OF THE NEXT ARC MILL; TO DETERMINE IT WE TAKE THE COSINE OF THE TOTAL ANGLE BETWEEN THE HEMISPHERE CENTER AND SHELL (#43), MULTIPLY IT BY THE HEMISPHERE RADIUS (#14) TO DETERMINE THE LENGTH OF THAT LEG, AND SUBTRACT THAT FROM THE HEMISPHERE RADIUS (#14) AND THE Z DEPTH OF THE FIRST PASS (#22) TO GET #44.
N060(0)X(#9+#14*SIN,#43)Y#10$	;HERE WE FIGURE THE X START POINT FOR THE ARC MILL IN N90 BY MULTIPLYING THE SINE OF V43 BY THE HEMISPHERE RADIUS (V14) AND ADDING THE X-AXIS ARC CENTER (#9). THE Y START POINT IS THE SAME AS THE Y-AXIS CENTER (#10).
N070(0)Z(#44+.1)$	;POSITION TO .1 INCH ABOVE THE Z DEPTH CALCULATED IN N50.
N080(1)Z#44F#6$	;FEED TO THE Z DEPTH CALCULATED IN N50.

```
N090(2)I#9J#10D#4$         ;CUT THE ARC CENTERED AT I #9, J #10 IN THE DIRECTION
                           CONTAINED IN #4 (PASSED FROM CALLING PROGRAM).
N100(0)Z.1/$               ;RAPID UP .1 INCH IN Z.
N110(#43=#43+#42)          ;INCREMENT THE TOTAL ANGLE BETWEEN THE
                           HEMISPHERE CENTER AND SHELL (#43) BY THE
                           INCREMENTAL ANGLE PER Z PASS (#42).
N120(#41=#41-1)            ;DECREMENT THE NUMBER OF PASSES (#41) BY 1.
N121(IF,#41-1,999,125,50)  ;IF #41 IS < 0, THE HEMISPHERE IS CUT AND THE MACRO
                           RETURNS CONTROL TO THE CALLING PROGRAM.  IF #41 =
                           0, BRANCH FOR ONE MORE PASS; IF #41 > 0, BRANCH FOR
                           MULTIPLE PASSES.
N125(#43=90)               ;SET THE TOTAL ANGLE TO 90 (THIS PROVIDES A FINISH
                           PASS).
N126(GO,50)                ;BRANCH TO N50 FOR LAST PASS.
N999(9)M30$                ;END OF MACRO.
```

A simple program to cut one hemisphere using the above macro follows.

```
CODE                       COMMENTS

N10(9)M03S1500T1           ;TURN ON SPINDLE, SELECT TOOL
N20(M)A9C1D1F90I20J12R10
        Z0PHSPH-MAC        ;MACRO  CALLING EVENT. SPECIFIES AN INCREMENTAL
                           ANGLE A OF 9°, INITIAL Z CLEARANCE OF 1 INCH, CCW ARC
                           DIRECTION, A FEEDRATE OF 90 IPM, X AXIS CENTER AT 20,
                           J AXIS CENTER AT 12, HEMISPHERE RADIUS OF 10 AND A
                           FIRST PASS Z DIMENSION OF 0.
N30(9)M30$                 ;END OF PROGRAM
```

SPINDLE PROBE DATUMING MACRO

This macro is designed to perform exactly the same operations as the G0 probing cycle package; that is, it calibrates the spindle probe in each align direction (+X, -X, +Y, -Y, -Z). X and Y must be pre-positioned to the known center of the hole/slot coordinates in which the XY datuming is to take place. Optional probing hardware and software must be installed before this macro can be used.

Global and local variable definitions follow.

SYMBOLIC VARIABLE NUMBER/LETTER	PURPOSE
V1/A	Incremental distance from expected contact point at which feedrate changes from F to V.
V4/D	Holds first symbolic variable number of a contiguous series of five symbolic variable numbers in which the datum offsets will be stored in the following order: +X, -X, +Y, -Y, -Z[6]. Valid range for V4: 70-95.
V6/F	Modal feedrate at which all non-align motion in the cycle occurs.
V9/I	X position at which Z datuming takes place.
V10/J	Y position at which Z datuming takes place.
V11/K	Known Z contact position for Z datuming.
V12/L	Late contact tolerance.
V18/V	Feedrate at which align (find surface) motion in the cycle occurs.
V20/X	X dimension of hole/slot in which XY datuming takes place.
V21/Y	Y dimension of hole/slot in which XY datuming takes place. If X=Y, datuming takes place in circular hole.
V22/Z	Z depth at which XY datuming takes place.
V23	Contact point for axis during alignment.
V40	X start position in part coordinates (known X center of hole/slot in which XY datuming takes place).
V41	Y start position in part coordinates (known Y center of hole/slot in which XY datuming takes place).
V42	Z start position in part coordinates. Must be greater than or equal to the Z depth for XY datuming (V22) and greater than the "known" Z depth for Z datuming (V11). XY motion from XY datuming to Z datuming will take place at this Z depth.
V43	Known X positive contact point = X start point +V20/2.
V44	Known X negative contact point = X start point - V20/2.
V45	Known Y positive contact point = Y start point +V21/2.
V46	Known Y negative contact point = Y start point - V21/2.

The general algorithm is:

[6] For example, if V4=75, +X will be stored at 75, -X at 76, +Y at 77, -Y at 78, and -Z at 79.

```
MAIN PROGRAM
    activates tool number to identify probe
    positions to known center of the hole/slot coordinates in which XY datuming
    is to take place
    calls SPG0 which calls
        G0-ERR which
        checks for the following errors:
            D<70 or D>95
            A<=0
            F<=0
            L<=0
            V<=0
            X<=.5
            Y<=.5
            K>=Z start position
            current Z>Z start position
        End macro G0-ERR;
    initialize variable number held in V4 and next four variables to 0
    get current axis positions
    feed in Z to XY datum depth (V22)
    do XY datuming:
        put spindle probe in "Stop on Contact" mode
        if beyond approach position then feed axis to positive approach position
        activate align feedrate (V18)
        find surface of part (SPM)
        calculate probe datum offset for given axis, direction
        retract probe off contact
        repeat above for -X, +Y, and -Y offsets
    rapid retract Z to Z start position
    rapid XY to XY position for Z datum
    do Z datum:
        put spindle probe in "Stop on Contact" mode
        if beyond approach position then feed Z to approach position
        activate align feedrate
        find surface of part
        calculate probe datum offset for given -Z
        retract probe off contact to Z start position
        put spindle probe in "Stop on Contact" mode
        rapid X and Y to their start positions.
    End macro SPG0;
End Main Program.
```

The macro code follows, formatted and commented as before.

```
(SPG0)                      ;MACRO ID
N10(M)PG0-ERR               ;CALL ERROR CHECKING MACRO
N20(#(#4)=0)                ;SET VARIABLE NUMBER CONTAINED IN V4=0
N21(#(#4+1)=0)              ;SET NEXT FOUR VARIABLES =0...
N22(#(#4+2)=0)
N23(#(#4+3)=0)
N24(#(#4+4)=0)
N30(SPP)                    ;TURN ON SPINDLE PROBE PROTECT
N31(1)Z#22F#6$              ;FEED TO DATUMING Z DEPTH AT F
N40(#43=#40+#20/2)          ;COMPUTE KNOWN CONTACT POINTS...
N41(#44=#40-#20/2)
N42(#45=#41+#21/2)
```

```
N43(#46=#41-#21/2)
N50(IF,#40-(#43-#1),51,60,60)        ;IF BEYOND APPROACH POSITION
N51(1)X(#43-#1)$                     ;     THEN FEED TO APPROACH POS'N
N60(1)F#18$                          ;ACTIVATE ALIGN FEEDRATE
N61(SPM,1,#43+#12,23,1)              ;FIND SURFACE OF PART
N62(#(#4)=#43-#23)                   ;SUBTRACT CONTACT POINT (V23) FROM KNOWN
                                     CONTACT POINT (V43) TO GET +X OFFSET
N70(1)X#40F#6$                       ;RETRACT PROBE
N71(SPP)                             ;SPINDLE PROBE PROTECT
N80(IF,#40-(#44+#1),90,90,81)        ;IF BEYOND APPROACH POSITION
N81(1)X(#44+#1)$                     ;     THEN FEED TO APPROACH POS'N
N90(1)F#18$                          ;ACTIVATE ALIGN FEEDRATE
N91(SPM,1,#44-#12,23,1)              ;FIND SURFACE OF PART
N92(#(#4+1)=#44-#23)                 ;SUBTRACT CONTACT POINT FROM KNOWN
                                     CONTACT POINT TO GET -X OFFSET.
N100(1)X#40F#6$                      ;RETRACT PROBE
N101(SPP)                            ;SPINDLE PROBE PROTECT
N110(IF,#41-(#45-#1),111,120,120)    ;IF BEYOND APPROACH POSITION
N111(1)Y(#45-#1)$                    ;     THEN FEED TO APPROACH POSITION
N120(1)F#18$                         ;ACTIVATE ALIGN FEEDRATE
N121(SPM,2,#45+#12,23,1)             ;FIND SURFACE OF PART
N122(#(#4+2)=#45-#23)                ;COMPUTE +Y OFFSET
N130(1)Y#41F#6$                      ;RETRACT PROBE
N131(SPP)                            ;SPINDLE PROBE PROTECT
N140(IF,#41-(#46+#1),150,150,141)    ;IF BEYOND APPROACH POSITION
N141(1)Y(#46+#1)$                    ;     THEN FEED TO APPROACH POSITION
N150(1)F#18$                         ;ACTIVATE ALIGN FEEDRATE
N151(SPM,2,#46-#12,23,1)             ;FIND SURFACE OF PART
N152(#(#4+3)=#46-#23)                ;COMPUTE -Y OFFSET
N160(1)Y#41F#6$                      ;RETRACT PROBE
N161(SPP)
N170(0)Z#42$                         ;RAPID TO Z START POSITION
N180(0)X#9Y#10$                      ;RAPID TO XY POSITION FOR Z DATUM
N190(IF,#42-(#11+#1),200,200,191)    ;IF BEYOND APPROACH POSITION
N191(1)Z(#11+#1)$                    ;     THEN FEED Z TO APPROACH POS'N
N200(1)F#18$                         ;ACTIVATE ALIGN FEEDRATE
N201(SPM,3,#11-#12,23,1)             ;FIND SURFACE OF PART
N202(#(#4+4)=#11+#23)                ;COMPUTE -Z OFFSET
N210(1)Z#42F#6$                      ;RETRACT PROBE
N211(SPP)                            ;SPINDLE PROBE PROTECT
N300(0)X#40Y#41$                     ;POSITION X, Y TO START POSITIONS
N9999(9)M30$                         ;END MACRO
```

ADDITIONAL COMMENT:

The five symbolic variables starting at #(V4) can be used by other CPC probe cycles to determine true contact points; however, if they are to be used in this fashion, the user must be careful not to manipulate their contents from other macros. Also, they will be zeroed if the power is switched off.

The code for G0-ERR follows. Since this is a fairly straightforward error-checking macro, most comments have been omitted.

```
(G0-ERR)
N1(IF,#4-70,3,10,2)
N2(IF,#4-95,10,10,3)
N3(ERR,D<70 OR D>95)
N4(GO,90)
N10(IF,#1,11,11,20)
N11(ERR,A<=0)
N12(GO,90)
N20(IF,#6,21,21,30)
N21(ERR,F<=0)
N22(GO,90)
N30(IF,#12,31,31,40)
N31(ERR,L<=0)
N32(GO,90)
N40(IF,#18,41,41,50)
N41(ERR,V<=0)
N42(GO,90)
N50(IF,#20-.5,51,51,60)
N51(ERR,X<=.5)
N52(GO,90)
N60(IF,#21-.5,61,61,70)
N61(ERR,Y<=.5)
N62(GO,90)
N70(#40=EPP)                  ;GET AXIS POSITION, STORE X IN V40, Y IN V41 AND Z IN V42
N71(IF,#11-#42,80,72,72)
N72(ERR,K>=Z START POSITION)
N73(GO,90)
N80(IF,#22-#42,99,99,82)
N82(ERR,Z>Z START POSITION)
N90(9)M0$
N91(GO,90)
N99(9)M30$
```

A simple master program which prepositions X and Y and invokes the above macros for one datuming cycle follows.

```
(SPG0-MAS)                    ;MASTER PROGRAM NAME
N10(9)T10$                    ;SELECT PROBE
N20(0)X.4925Y4.2684Z-2.9$     ;POSITION TO CENTER OF COORDINATES WHERE XY
                              DATUMING IS TO TAKE PLACE
N30(M)A.2D95F40I1J5.3K-3.048
        L2V20X1.0023Y1.0095
        Z-3.25PSPG0$          ;CALL SPG0 TO OPERATE ON SPECIFIED PARAMETERS
N40(9)M0$                     ;HALT
N99(9)M30$                    ;END PROGRAM
```

SPINDLE PROBE FIND SURFACE MACRO

This macro uses a spindle probe to find the X, Y or Z surface of a part. It requires the align axis to be in prealign position (no obstructions between the probe and the part surface contact point) and the other two axes to be in position such that motion along the align axis will result in part contact. It also requires that optional probing hardware and software be installed.

Global and symbolic variable definitions follow.

SYMBOLIC VARIABLE NUMBER/LETTER	PURPOSE
V1/A	Incremental distance from expected contact point at which feedrate changes from F to V.
V2/B	Preset coordinate value for the true contact point in part program coordinates.
V3/C	Expected contact point for the axis of alignment.
V4/D	Symbolic Variable number of the datum offset value for axis and align direction. If spindle probe G0 datum offsets are to be used, V4 must equal 0.
V5/E	Fixture offset select code value where present offset is stored.
V6/F	Modal feedrate at which all non-align motion occurs.
V12/L	Late contact tolerance.
V18/V	Feedrate at which align (find surface) motion occurs.
V20/X	Axis designator. Enter 1 for X, 2 for Y, 3 for Z.
V23	Contact point for align axis inclusive of datum offsets.
V24	Active fixture offset select code value.
V40	X start position in part coordinates.
V41	Y start position in part coordinates.
V42	Z start position in part coordinates.

The general algorithm is:

```
MAIN PROGRAM
    activates probe's tool number
    positions axes so that probe has a clear path to the part contact point along the align axis
    calls SPG1 MACRO
        which calls G1-ERR MACRO
        checks for the following errors:
            A<=0
            D<>0 or D not in range 70-99
            E<=0 or E>14
            F<=0
            L<=0
            V<=0
            X<=0 or X>3
            align axis expected contact point=start point
            if Z is axis of alignment, Z expected contact point>Z start point[7]
        End G1-ERR;
    get current axis position (EPP)
    if align axis is further from expected contact point than approach distance then
        put spindle probe in protect mode
        feed align axis to approach position at feedrate V6
    set feedrate equal to align feedrate (V18)
    find surface of part (SPM)
    if V4=0 then add spindle probe datum offsets to contact point.
```

[7] Z is assumed to be in normal orientation (-Z down, +Z up).

```
        set V23=contact point for align axis
        if V4<>0 then V23=V23+value at Symbolic Variable number stored in V4.
        store absolute value of preset value (V2)- actual contact point (V23) as the fixture
            offset for the align axis (X, Y, or Z) and specified fixture offset select code (V5)
        position X, Y and Z to their respective start positions (V40, V41, V42)
        End SPG1 MACRO;
    End MAIN PROGRAM.
```

The macro code follows.

```
(SPG1)                              ;MACRO TITLE
N10(M)PG1-ERR                       ;CALL ERROR CHECKING MACRO
N20(#40=EPP)                        ;STORE AXIS POSITION AND STORE X IN V40, Y IN V41,
                                    AND Z IN V42
N30(IF,#20-2,31,100,200)            ;DETERMINE WHICH AXIS IS ALIGN AXIS AND BRANCH
                                    ACCORDINGLY
N31(IF,#3-#40,32,500,60)            ;{X IS ALIGN AXIS} IF EXPECTED CONTACT POINT IS  TO
                                    THE RIGHT OF THE START POSITION THEN
N32(IF,#40-(#3+#1),40,40,35)        ;   IF ALIGN AXIS IS FURTHER FROM EXPECTED
                                        CONTACT POINT THAN APPROACH DISTANCE
                                        THEN
N35(SPP)                            ;       SPINDLE PROBE PROTECT

N36(1)X(#3+#1)F(#6)$                ;       FEED ALIGN AXIS TO APPROACH
                                            POSITION AT FEEDRATE V6
N40(1)F(#18)$                       ;ACTIVATE ALIGN FEEDRATE (V18)
N45(SPM,1,#3-#12,23,#4)             ;FIND SURFACE OF PART
N46(GO,76)                          ;BRANCH PAST OPERATIONS FOR EXPECTED CONTACT
                                    POINT TO THE LEFT OF START POSITION
N60(IF,#40-(#3-#1),65,70,70)        ;IF EXPECTED CONTACT POINT IS TO THE LEFT OF START
                                    POSITION AND ALIGN AXIS IS FURTHER FROM EXPECTED
                                    CONTACT POINT THAN APPROACH DISTANCE THEN
N65(SPP)                            ;   SPINDLE PROBE PROTECT
N66(1)X(#3-#1)F#6$                  ;       FEED ALIGN AXIS TO APPROACH
                                            POSITION AT FEEDRATE V6
N70(1)F#18$                         ;ACTIVATE ALIGN FEEDRATE (V18)
N75(SPM,1,#3+#12,23,#4)             ;FIND SURFACE OF PART
N76(IF,#4,80,80,77)                 ;IF V4>0 THEN
N77(#23=#23+#(#4))                  ;   ADD SPINDLE PROBE DATUM OFFSETS TO
                                        CONTACT POINT.
N80(SOV,5,#5,24,#2-#23)             ;STORE FIXTURE OFFSET VALUE (V2-V23) FOR SPECIFIED
                                    ALIGN AXIS AND SPECIFIED FIXTURE OFFSET SELECT
                                    CODE (V5)
N85(GO,245)                         ;SKIP LOGIC FOR Y ALIGN AXIS
N100(IF,#3-#41,101,500,120)         ;{Y IS ALIGN AXIS} IF EXPECTED CONTACT POINT IS
                                    BELOW THE START POSITION THEN
N101(IF,#41-(#3+#1),110,110,105)    ;   IF ALIGN AXIS IS FURTHER FROM EXPECTED
                                        CONTACT POINT THAN APPROACH DISTANCE
                                        THEN
N105(SPP)                           ;       SPINDLE PROBE PROTECT

N106(1)Y(#3+#1)F(#6)$               ;       FEED ALIGN AXIS TO APPROACH
                                            POSITION AT FEEDRATE V6
N110(1)F(#18)$                      ;ACTIVATE ALIGN FEEDRATE (V18)
N115(SPM,2,#3-#12,23,#4)            ;FIND SURFACE OF PART
N116(GO,136)                        ;BRANCH PAST OPERATIONS FOR EXPECTED CONTACT
                                    POINT ABOVE START POSITION
```

```
N120(IF,#41-(#3-#1),125,130,130)  ;IF EXPECTED CONTACT POINT IS TO THE LEFT OF START
                                  POSITION AND ALIGN AXIS IS FURTHER FROM EXPECTED
                                  CONTACT POINT THAN APPROACH DISTANCE THEN
N125(SPP)                         ;    SPINDLE PROBE PROTECT
N126(1)Y(#3-#1)F#6$               ;         FEED ALIGN AXIS TO APPROACH
                                            POSITION AT FEEDRATE V6
N130(1)F#18$                      ;ACTIVATE ALIGN FEEDRATE (V18)
N135(SPM,2,#3+#12,23,#4)          ;FIND SURFACE OF PART
N136(IF,#4,140,140,137)           ;IF V4>0 THEN
N137(#23=#23+#(#4))               ;    ADD SPINDLE PROBE DATUM OFFSETS TO
                                       CONTACT POINT.
N140(SOV,5,#5,25,#2-#23)          ;STORE FIXTURE OFFSET VALUE (V2-V23) FOR SPECIFIED
                                  ALIGN AXIS AND SPECIFIED FIXTURE OFFSET SELECT
                                  CODE (V5)
N145(GO,245)                      ;SKIP LOGIC FOR Z ALIGN AXIS
N200(IF,#3-#42,=201,500,400)      ;{Z IS ALIGN AXIS} IF EXPECTED CONTACT POINT IS
                                  BELOW THE START POSITION THEN
N201(IF,#42-(#3+#1),210,210,205)  ;    IF ALIGN AXIS IS FURTHER FROM EXPECTED
                                       CONTACT POINT THAN APRROACH DISTANCE
                                       THEN
N205(SPP)                         ;         SPINDLE PROBE PROTECT

N206(1)Z(#3+#1)F(#6)$             ;         FEED ALIGN AXIS TO APPROACH
                                            POSITION AT FEEDRATE V6
N210(1)F(#18)$                    ;ACTIVATE ALIGN FEEDRATE (V18)
N215(SPM,3,#3-#12,23,#4)          ;FIND SURFACE OF PART
N236(IF,#4,240,240,237)           ;IF V4>0 THEN
N237(#23=#23+#(#4))               ;    ADD SPINDLE PROBE DATUM OFFSETS TO
                                       CONTACT POINT.
N240(SOV,5,#5,25,#2-#23)          ;STORE FIXTURE OFFSET VALUE (V2-V23) FOR SPECIFIED
                                  ALIGN AXIS AND SPECIFIED FIXTURE OFFSET SELECT
                                  CODE (V5)
N245(1)X#40Y#41Z#42F#6$           ;POSITION X, Y AND Z TO THEIR RESPECTIVE START
                                  POSITIONS
N250(SPP)                         ;SPINDLE PROBE PROTECT
N255(GO,999)                      ;BRANCH TO END
N400(ERR,Z EXPECTED>Z START)      ;FLAG INCORRECT Z FAULT
N405(GO,505)                      ;BRANCH TO PROGRAM HALT
N500(ERR,EXPECTED CONTACT POINT=START)  ;FLAG FAULT
N505(9)MO$                        ;PROGRAM HALT
N999(9)M30$                       ;END MACRO
```

ADDITIONAL COMMENTS:

Note that if the align axis = Z and the X start position is below (<) the expected Z contact point, a fault is flagged from the SPG1 macro. This is handled by branching at N200 and flagged at N400.

If the align axis is X or Y, the polarity of the difference (expected contact point-start point) determines the direction of alignment. See lines N31, N100 and N200.

If the expected contact point = the start point, SPG1 flags a fault. See N31, N100, N200 and N500.

The macro code for G1-ERR follows.

```
(G1-ERR)
N10(IF,#1,11,11,20)
N11(ERR,A<=0)
N12(GO,90)
N20(IF,#6,21,21,30)
N21(ERR,F<=0)
N22(GO,90)
N30(IF,#12,31,31,40)
N31(ERR,L<=0)
N32(GO,90)
N40(IF,#18,41,41,50)
N41(ERR,V<=0)
N42(GO,90)
N50(IF,#20,55,55,51)
N51(IF,#20-3,60,60,55)
N55(ERR,X<=0 OR X>3)
N56(GO,90)
N60(IF,#5,65,65,61)
N61(IF,#5-14,70,70,65)
N65(ERR,E<=0 OR E>14)
N66(GO,90)
N70(#24=GAO,5)
N71(IF,#24,75,80,75)
N75(ERR,ACTIVE FIXTURE OFFSET NOT E00)
N76(GO,90)
N80(IF,#4,85,99,81)
N81(IF,#4-70,85,99,82)
N82(IF,#4-99,99,99,85)
N85(ERR,D NEITHER 0 NOR IN RANGE 70-99)
N90(9)M0$
N91(GO,99)
N99(9)M30$
```

A simple master program which finds the X, Y, and Z coordinates of a part surface follows.

```
(SPG1-MAS)                    ;MASTER PROGRAM NAME
N10(9)E0T10                   ;SELECT FIXTURE OFFSET 0 AND PROBE TOOL NUMBER
N20(0)X11.7214Y3.0722Z0$      ;POSITION TO ALIGN Z AXIS WITH DESIRED SURFACE
N30(M)A.2B-.1C-.062D99E1F50   ;CALL MACRO SPG1 TO OPERATE ON SPECIFIED
      L.2V10X3PSPG1$           PARAMETERS
N40(0)X8.5Y3.14Z-.8$          POSITION TO ALIGN X AXIS WITH DESIRED SURFACE
N50(M)A.2B-2.5C9D95E1F50      ;CALL MACRO SPG1 TO OPERATE ON SPECIFIED
      L.2V10X1PSPG1$           PARAMETERS
N60(0)Y.64$                   ;POSITION TO ALIGN X AXIS WITH DESIRED
N70(0)X10$                    SURFACE
N80(M)A.2B-2.52C1.14D97E1F50  ;CALL MACRO SPG1 TO OPERATE ON SPECIFIED
      L.2V10X2PSPG1$           PARAMETERS
N90(9)E1$                     ;SELECT NEW FIXTURE OFFSET
N100(0)X0Y0Z0$                ;POSITION ALL AXES TO 0
N110(9)M0$                    ;PROGRAM HALT
N999(9)M30$                   ;PROGRAM END
```

DEVELOPING A CPC: AN EXAMPLE

To illustrate how the above steps work, we will now develop a macro to mill the four-drafted rectangular cavity shown on the **following** page.

1. DEFINE THE PART

The desired part is shown on the **following** page. It is a complex cavity that has a rectangular XY shape throughout. Each side wall of the cavity can have an independent draft angle between 0 and 90°. The cavity may have a variable XY position, Z depth and XY size.

Since this is a cavity, we must decide on a method of removing material not only on the perimeter of the rectangle but also on its interior.

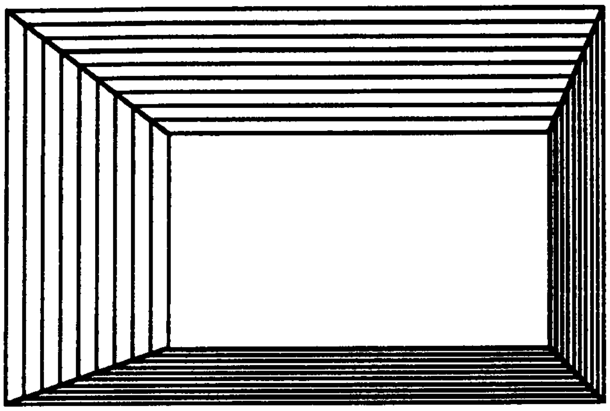
The cavity has one recurring characteristic at a given depth: its rectangular shape. Since its shape is made up of a progression of rectangles, we can use a Frame Mill event to shape and pocket the cavity. However, since the walls have draft angles (which are not programmable in the Frame Mill event) we must carefully size and locate each successive frame so that the end result (after milling several frames) is the desired cavity. The front and side views on the **following** page illustrate this approach.

The part has many variable characteristics. The XY size of the cavity at the surface of the stock is variable. The Z depth is variable. The four draft angles are independently variable. The location of the cavity in the XY plane is variable, as are the locations of the stock surface and the Z reference plane. The direction in which the cavity is cut should be programmable as Clockwise or Counterclockwise.

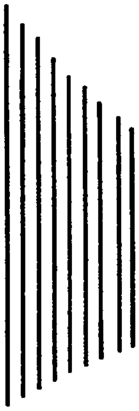
This part can easily be divided into sub-parts. We can, for instance, check for errors in a separate macro or macros. We can use the control's Frame Mill event to form and pocket the rectangular frame at successively deeper Z locations. Later, other aspects of the algorithm that lend themselves to the use of sub-macros and subroutines will become apparent.

We will also assume that this cavity may be milled in any material; therefore, we will pocket the cavity at every Z depth to insure that the cutter is not overloaded (as may be the case if pocketing were done in one pass).

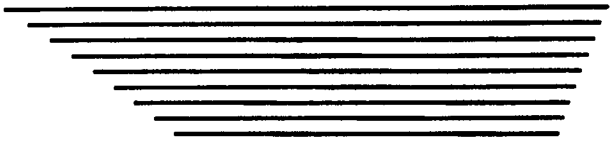
FOUR DRAFTED RECTANGULAR CAVITY



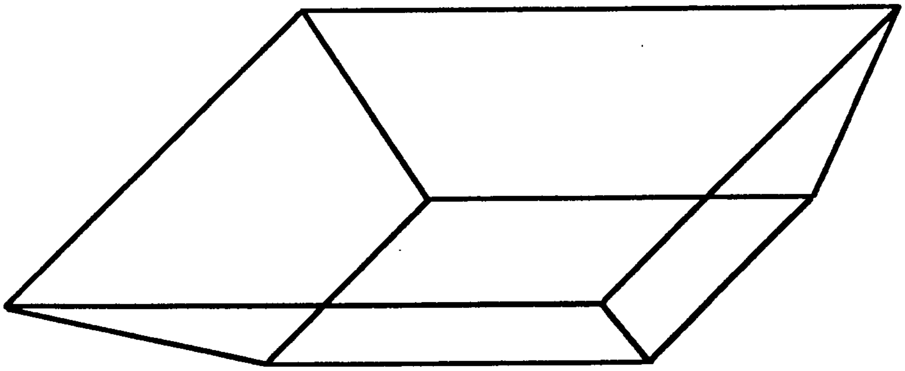
TOP



SIDE



FRONT



THREE DIMENSIONAL

2. DETERMINE HOW TO CUT THE PART

We will cut the part as a series of frames, each slightly deeper in the stock than the one before it. This will form a cavity, and each successive frame can be sized and offset so that the resulting final cavity has angled side walls.

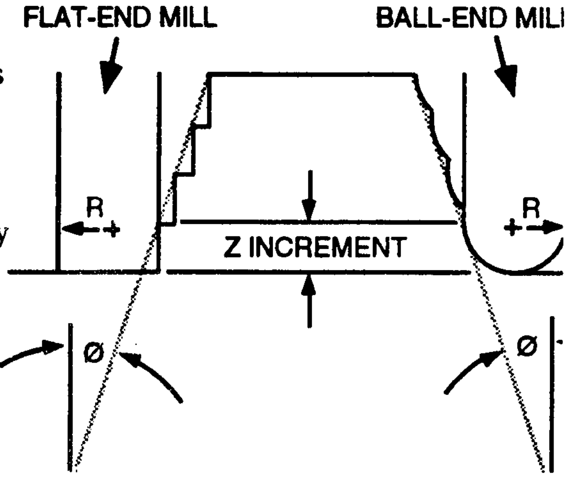
This requires that we determine specific dimensions and a new start point for each successive pass. Since the draft angles and the surface dimensions of the cavity are to be user-programmed, we must trigonometrically compute offset data for each pass based on this information. We will derive formulas to do this later in this section.

MODULES

Here we list the main "modules" of the macro. These modules may be segments of code in the main macro which accomplish a specific task, or they may be subroutines or submacros. Our plan is to give an overview of the purpose of each module, then to expand on each one individually later.

Our first module will be the least complex and the easiest to code. It is an error checking submacro which we call at the beginning of the main macro which checks all the user-entered variables to insure that they are in range. Later, after doing some computations on these variables, we will call another error checking macro to insure that our local variables (computed with the user-entered variables) are in range. Both of these macros are explained in more detail further on.

Our next module involves the actual machining of the part. We would like the angled sidewalls of our cavity to be smooth; however, truly smooth sidewalls are difficult to cut on a mill without special workholding fixtures. Cutting the cavity by the method we have chosen with a flat-end mill would produce tiny steps in the side of the cavity; this is undesirable. We can, however, minimize the steps by using a ball-end mill. As shown in the illustration at **right**, this yields a much smoother sidewall given identical cutter radii, Z increment and draft angle. Smoothness of the wall will increase proportionally as the cutter size increases and the Z increment decreases. We can trigonometrically derive the distance that a ball end mill must be offset so that it yields the same frame dimensions as a flat-end mill of the same radius; the formulas we come up with can then be called as a submacro. This will be discussed in greater detail further on.



While a ball-end mill may make our sidewalls smooth, it will have just the opposite effect on the cavity floor. If present (i.e. if the stock is thicker than the cavity is deep[1]), the cavity floor will have roughing marks from the ball end cutter. If we want a smooth floor, these marks must be removed with a flat-end mill. This will require us to compute and store the final cavity dimensions before we start adjusting things for the ball-end mill. When we expand our ideas into equations, algorithms and code, this should be the first module that we concern ourselves with.

[1] We will allow the user to disable the final flat-end mill pass by programming 0 into the symbolic variable which is to hold the tool number of the flat-end mill. In the program we will trap for this and only mill the last pass if a flat-end mill has been programmed.

Compensating for the ball end mill will give us a new cavity dimension at the stock surface (more on this later) from which we may compute the cavity size at each successive Z pass. As we shall see, this computation will require another submacro.

Our fourth job will be to determine a suitable Z increment. Although global variable S holds the user's Z increment, various tool-offsetting considerations will require us to rely on this value only as a *limit*. For reasons that will become apparent later, we will need to compute an ideal Z increment; this will require another submacro.

Now that we've determined how we will generate offsets, we must sketch a preliminary algorithm for the macro. This algorithm follows:

```
MAIN PROGRAM calls
    DRAFT ANGLE CAVITY MACRO which calls
        GLOBAL VARIABLE ERROR CHECKING MACRO;
    then calls
        FINAL CAVITY DIMENSION MACRO;
    then calls
        LOCAL VARIABLE ERROR CHECKING MACRO[2];
    then calls
        Z INCREMENT MACRO;
    then calls
        BALL-END MILL OFFSET MACRO;
    While Z depth is > bottom of cavity:
        calls
            FRAME RESIZE MACRO;
        Calls Frame event to mill Frame
        Increments Z depth
    When bottom of cavity is reached, change to a flat-end mill and finish cavity
    floor[3].
    END draft angle cavity macro;
END main.
```

To simplify further reference to these programs, at this point we can give them names which are indicative of their function.
We will name the main program DC, the main macro DCMAC, the user error checking macro USERR, the final cavity dimension macro FINFRM, the local variable error checking macro LOCERR, the Z increment macro ZINC, the ball-end mill offset macro BALLOFFS, and the frame resize macro PRESOFFS since it is finding the offsets for the frame at the present Z position.

Keep in mind that this is just a preliminary algorithm; we will add more macros and modules later when we expand it.

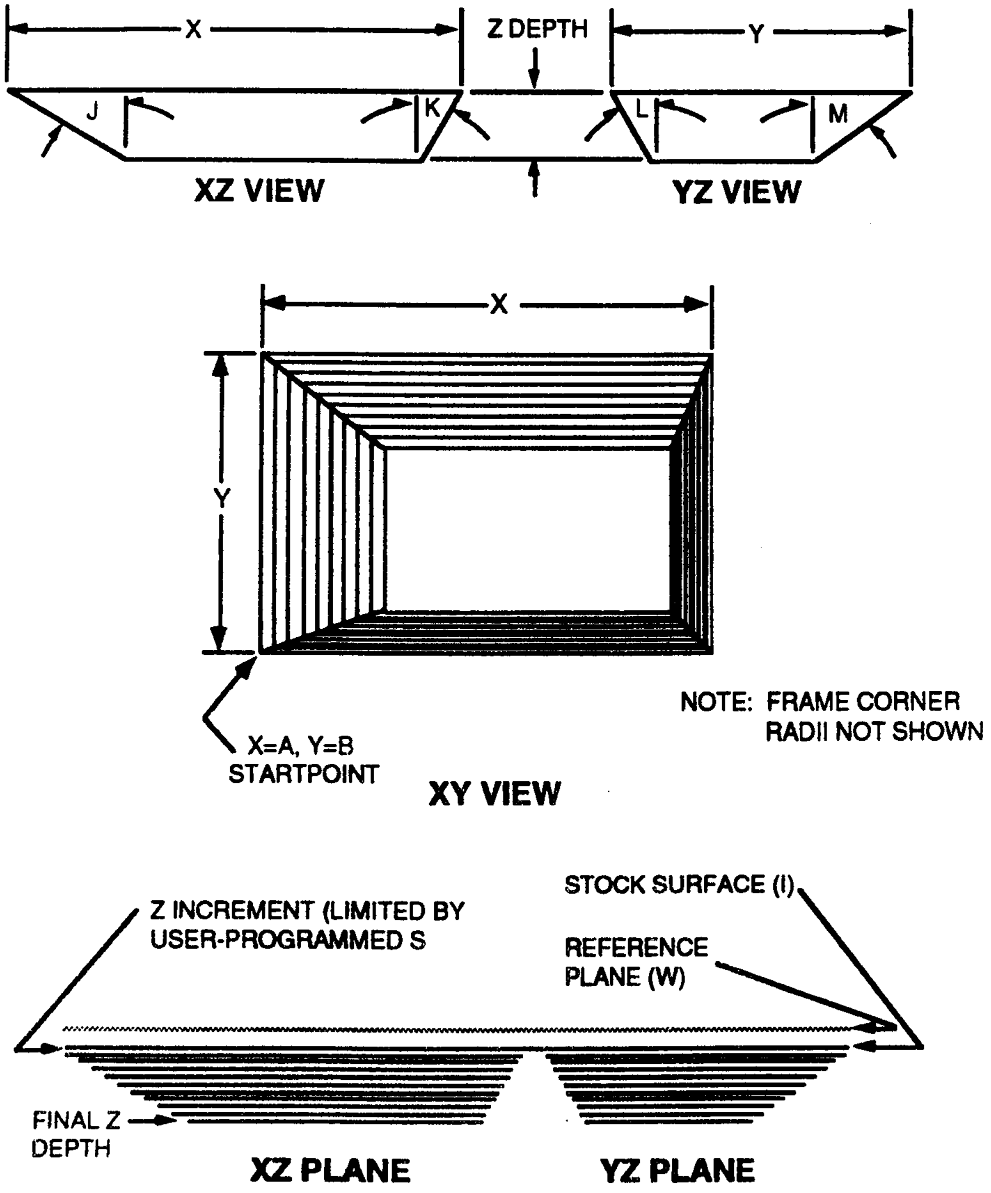
[2] This was mentioned when we discussed the user variable error checking macro above. As we shall see, all the local variables which need to be checked by this macro are computed at this point.

[3] The final ball end mill pass will leave "grooves" in the cavity floor (due to the shape of the mill); this finish pass with a flat end mill will smooth the cavity floor. When we expand our algorithm, we will make this pass optional.

3. ASSIGN VARIABLES TO THE PART GEOMETRY

The illustrations on the **opposite** page show how the dimensional variables discussed next are assigned to the part.

VARIABLE ASSIGNMENTS



This part requires a large number of Global variables, many of which are needed in the Frame event. However, since we are using the Frame event basically as a roughing event, there are a number of optional variables that we can leave unprogrammed. These include the peck increment K, corner roughing width J, and finish dimension and feedrate L and V. The cutter compensation variable C will not be user programmed but instead it will be set by the macro according to the user-specified direction D.

The remaining Frame event entries are filled in by the macro with symbolic variables that have either been directly entered by the user or that have been computed by the macro. The X and Y dimensions of the Frame, for example, are programmed with symbolic variables X and Y which the user enters and the macro adjusts to reflect necessary tool and Z depth offsets. The Z depth of the Frame is programmed with a *local* symbolic variable that accumulates the incremental Z depth for each pass. The Frame corner radius R, reference plane W, direction D and feedrate F are directly programmed with symbolic variables of the same name. Plunge feedrate P is programmed with symbolic variable V since P is not available as a global variable. For our offsets to work, Cutter Comp (C) must be programmed in the Frame event. Therefore the macro sets C according to D; for D0, C is programmed C2 (on right); for D1 it is programmed C1 (on left). The Frame event's G entry is always programmed G7 for pocket mill.

The remaining global variables are used by the macro as follows: A and B hold the X and Y start point of the cavity[4]; T will hold the T code of the ball end mill, U[5] will hold the T code of the flat end mill[6] (U may be programmed 0 if the user doesn't want to mill the final pass with a flat-end mill); G and H will hold the spindle speeds for tools T and U respectively; S will hold the Z increment limit; I will hold the Z position of the stock surface; finally, J will hold the left-side X draft angle, K the right side X draft angle, L the more negative Y draft angle and M the more positive Y draft angle. The reason we chose J-M to hold the angles will become apparent in Step 5.

4. DETERMINE ERROR CONDITIONS

There are a number of errors we must check for in our variables. For instance, a legitimate draft angle must be between 0 and 90°, so we have to check every programmed draft angle to insure that it falls within this range. Both the X and Y dimension of the cavity must be larger than twice the frame corner radius R. Since we will use D0 to imply a clockwise feed direction and D1 to imply a counterclockwise direction, we should flag an error for D <> [0,1]. The Z reference plane must be more positive than the stock surface (I), which must be more positive than the Z depth of the cavity. The Z increment limit S must be positive. Feedrates F and V and Spindle speeds G and H must also be positive. We will check for all of these errors from a submacro called USERR that is called at the start of DCMAC; if any are flagged, USERR will branch to a program halt and the error will be displayed on the Fault screen through use of the ERR statement.

While USERR checks for errors in the user-programmed values, as mentioned before, there may be errors in some of the values computed by DCMAC *after* USERR is called. Specifically, we want to flag an error if our final X and Y dimensions are smaller than twice the user-programmed corner radius or if they are smaller than the diameter of the finish tool. We will check for these conditions in a second error checking macro called LOCERR (for LOCal variable ERRor check) after we compute the X and Y dimensions of the final cavity and get the diameter of the finish tool.

[4] The Frame Mill event requires that this be the lower left corner of the cavity; for simplicity and consistency, we adopt this convention for our macro.

[5] We use variable U only because it immediately follows T in the list.

[6] For best results, the diameter of the ball end mill should equal the diameter of the flat end mill.

By using submacros to check for errors early in the main macro we greatly simplify the task of writing the actual offsetting and milling modules. If we are sure all of our variables are error-free, we needn't complicate the rest of our code with error checking logic and all the branching that comes with it.

5. ASSIGN LOCAL AND GLOBAL VARIABLES AND EXPAND ALGORITHM

Our global variables have basically been assigned in Step 3. For reference, they are listed along with their Symbolic Variable numbers and purpose below.

SYMBOLIC VARIABLES

LETTER	NUMBER	PURPOSE
A	1	X start point (lower-left corner of frame)
B	2	Y start point (lower-left corner of frame)
D	4	Direction to mill frame
F	6	Feedrate for Frame mill event
G	7	Spindle speed for ball-end mill
H	8	Spindle speed for flat-end mill
I	9	Z position of stock surface
J	10	Draft angle for more negative X wall
K	11	Draft angle for more positive X wall
L	12	Draft angle for more negative Y wall
M	13	Draft angle for more positive Y wall
R	14	Corner radius for Frame event
S	15	Z increment
T	16	Tool number of ball-end mill
U	17	Tool number of flat-end mill
V	18	Plunge feedrate (P in Frame event)
W	19	Reference plane Z position
X	20	Starting X dimension of cavity
Y	21	Starting Y dimension of cavity
Z	22	Z depth of cavity

These are the variables that are entered into the main program's macro calling event. Local variables are somewhat more difficult to define. To do so, we must look carefully at what our globals will be used for by the macro.

Local variables perform one of three primary functions: they can be used to store the result of an arithmetic operation that was performed on other variables; they can be used as sentinels such as counters or accumulators to control loops or to accumulate an incremented value (such as Z depth); or they can be used as flags which take on one of two possible values (usually 0 or 1). As we shall see, our local variables will consist of all three types.

The programmed draft angles are used in numerous calculations and therefore require a large number of subordinate local variables. This is the main reason we chose V10 through V13 (J-M in the calling event) to hold the user-programmed draft angles; we can now number related local variables in groups such as 30-33[7] and 40-43. This will allow us to quickly determine which angle we are dealing with by simply looking at the rightmost digit of the variable in question.

[7] We skipped 20-23 because 20, 21, and 22 are used as globals.

We cannot compile a list of the remaining local variables, or specify where we will use V30-33 and 40-43 until we expand on the ideas first put forward in Step 2 and develop each of the modules that make up DCMAC. So far these modules include the submacros USERR, FINFRM, LOCERR, ZINC, BALLOFFS and PRESOFFS. In addition to these, we will need a short initialization module at the beginning of DCMAC to take care of selecting the ball-end mill, turning on the spindle and setting the cutter comp variable, a main loop which does the actual machining, and a few other modules and macros, the need for which will become apparent as we develop our various offsetting routines. We now expand on each macro and module individually, starting with its purpose, working through any necessary equations and algorithms, and ending up with (in most cases) actual macro code. We will patch together these various segments of code in Step 6. Note that the following modules are ***not necessarily considered in their order of appearance in the algorithm given in Step 2***. Instead, they are presented in an order which is hoped to help you understand the process of writing submacros. When writing a complex macro like this, you will likely develop component submacros as you need them and figure out where they fit into the main macro (or calling program) afterwards.

SUBMACRO USERR

The purpose of USERR is to check the user-programmed variables to insure that they are all in range. Calling USERR should be the first thing we do in DCMAC because if any of the global variables are in error, executing any of DCMAC's code would be a waste of time.

The way any error checking macro works is simple: it will check a variable and branch to a program halt if the variable is out of range; if not, it checks the next variable and repeats the process until all the variables are checked.

The algorithm that implements this process is shown below.

```
Check for error
    if error then
        generate fault message
        branch to Program stop
    else
        check for next error...
            :
            :
    Branch to End
Program stop
Branch to beginning
End.
```

Notice that we branch to the beginning after the program stop. This is because if an M30 event (program end) immediately followed the M00 event (program stop) the user could press CYCLE START after detection of an error and the next event encountered would be M30; this would cause the submacro to return control to DCMAC without the error being corrected. This, of course, trivializes the usefulness of USERR. By branching to the beginning after the M00, we insure that an uncorrected error is trapped again and again until the user fixes the offending variable, RESETS and starts over again.

The global variables we need to check are:

D: Direction must be 0 (CW) or 1 (CCW).
J,K,L,M: All four draft angles must be between 0° and 90°.
X,Y,R: X and Y must be less than 2R (twice the corner radius of the frame)
S: User programmed Z increment must be >0.
W: Z reference plane must be above the final Z depth of cavity.
I: Stock surface (I) must be above Z depth of cavity.
F,V,G,H: Feedrate (F), plunge feedrate (V), spindle speeds (G,H) for each tool must be >0.

The macro code for USERR, which implements the given algorithm on these variables, follows. The code is presented in the off-line data preparation format required by the control and described in detail in an appendix of the Delta Customer Manual. Comments are included with the code following a semicolon (;). *It is important to note that these are comments only and must be removed from the code before the control will accept it in a Receive operation.*

```
CODE                              COMMENTS

(USERR)                           ;PART ID (APPEARS IN OFFLINE FORMAT ONLY)
N10(IF,(1-#4),20,50,20)           ;IF 1-D=0 GOTO 50 ELSE GOTO 20
N20(IF,#4,30,50,30)               ;IF D=0 GOTO 50 ELSE GOTO 30
N30(ERR,D<>0,1)                   ;FLAG ERROR
N40(GO,999)                       ;BRANCH TO PROGRAM STOP
N50(IF,#10,60,90,90)              ;IF J>=0 GOTO 90 ELSE GOTO 60
N60(ERR,J<0)                      ;FLAG ERROR
N80(GO,999)                       ;BRANCH TO PROGRAM STOP
N90(IF,(#10-90),120,100,100)      ;IF J>=90 GOTO 100 ELSE GOTO 120
N100(ERR,J>=90)                   ;FLAG ERROR
N110(GO,999)                      ;BRANCH TO PROGRAM STOP
N120(IF,#11,130,150,150)          ;IF K>=0 GOTO 150 ELSE GOTO 130
N130(ERR,K<0)                     ;FLAG ERROR
N140(GO,999)                      ;BRANCH TO PROGRAM STOP
N150(IF,(#11-90),180,160,160)     ;IF K>=90 GOTO 160 ELSE GOTO 180
N160(ERR,K>=90)                   ;FLAG ERROR
N170(GO,999)                      ;BRANCH TO PROGRAM STOP
N180(IF,#12,190,210,210)          ;IF L>=0 GOTO 210 ELSE GOTO 190
N190(ERR,L<0)                     ;FLAG ERROR
N200(GO,999)                      ;BRANCH TO PROGRAM STOP
N210(IF,(#12-90),240,220,220)     ;IF L>=90 GOTO 220 ELSE GOTO 240
N220(ERR,L>=90)                   ;FLAG ERROR
N230(GO,999)                      ;BRANCH TO PROGRAM STOP
N240(IF,#13,250,270,270)          ;IF M>=0 GOTO 270 ELSE GOTO 250
N250(ERR,M<0)                     ;FLAG ERROR
N260(GO,999)                      ;BRANCH TO PROGRAM STOP
N270(IF,(#13-90),300,280,280)     ;IF M>=90 GOTO 280 ELSE GOTO 300
N280(ERR,M>=90)                   ;FLAG ERROR
N290(GO,999)                      ;BRANCH TO PROGRAM STOP
N300(#95=2*#14)                   ;SET V95=2R
N310(IF,(#20-#95),320,350,350)    ;IF X<2R GOTO 320 ELSE GOTO 350
N320(ERR,X<2R)                    ;FLAG ERROR
N330(GO,999)                      ;BRANCH TO PROGRAM STOP
N350(IF,(#21-#95),360,380,380)    ;IF Y<2R GOTO 360 ELSE GOTO 380
N360(ERR,Y<2R)                    ;FLAG ERROR
N370(GO,999)                      ;BRANCH TO PROGRAM STOP
N380(IF,#15,390,390,410)          ;IF S<=0 GOTO 390 ELSE GOTO 410
N390(ERR,S<=0)                    ;FLAG ERROR
N400(GO,999)                      ;BRANCH TO PROGRAM STOP
N410(IF,(#19-#22),420,420,440)    ;IF W<=Z GOTO 420 ELSE GOTO 440
N420(ERR,W<=Z)                    ;FLAG ERROR
N430(GO,999)                      ;BRANCH TO PROGRAM STOP
```

```
N440(IF,#9-#22,450,450,470)     ;IF J<=0 GOTO 450 ELSE GOTO 470
N450(ERR,I<=Z)                  ;FLAG ERROR
N460(GO,999)                    ;BRANCH TO PROGRAM STOP
N470(IF,#6,480,480,500)         ;IF F<=0 GOTO 480 ELSE GOTO 500
N480(ERR,F<=0)                  ;FLAG ERROR
N490(GO,999)                    ;BRANCH TO PROGRAM STOP
N500(IF,#7,510,510,530)         ;IF G<=0 GOTO 510 ELSE GOTO 530
N510(ERR,G<=0)                  ;FLAG ERROR
N520(GO,999)                    ;BRANCH TO PROGRAM STOP
N530(IF,#8,540,540,560)         ;IF H<=0 GOTO 540 ELSE GOTO 560
N540(ERR,H<=0)                  ;FLAG ERROR
N550(GO,999)                    ;BRANCH TO PROGRAM STOP
N560(IF,#18,570,570,590)        ;IF V<=0 GOTO 690 ELSE GOTO 710
N570(ERR,V<=0)                  ;FLAG ERROR
N580(GO,999)                    ;BRANCH TO PROGRAM STOP
N590(GO,1000)                   ;BRANCH TO END (NO ERRORS)
N999(9)M0$                      ;PROGRAM STOP (THIS IS A CONVERSATIONAL  EVENT
                                AND THEREFORE MUST INCLUDE AN EVENT TYPE (9) AND
                                A TERMINATING DOLLAR SIGN $)
N999.5(GO,10)                   ;LOOP TO START IF ERROR FLAGGED TO FORCE
                                 CORRECTION OF ERROR
N1000(9)M30$                    ;END OF MACRO (CONVERSATIONAL EVENT)
```

Note that lines N10 through N40 check for D <> [0,1] by using nested IF statements. N10 subtracts D from 1 and checks the result; if #40 = 0, D=1 (1-1=0), which is in the specified range and this portion of the error checking macro is done. If not, D may be equal to 0. We check for this condition in N20. Both N10 and N20 branch to N30 (the error message) on failure. IF statements can be nested several deep to test conditions such as these.

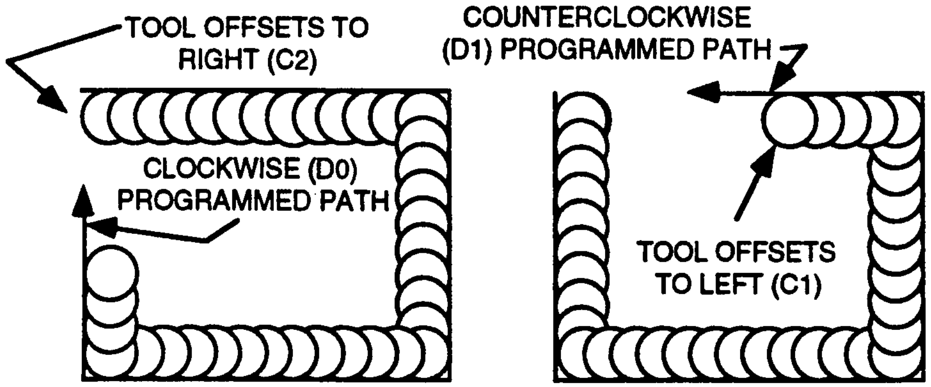
Also notice that we use variable number 95 as a "scratchpad" to hold 2*R (twice the corner radius of the frame) when we check 2R against the X and Y frame dimensions.

Once we are sure that our global variables are in range, we can proceed with DCMAC. The first thing we need is a small initialization module to turn on the spindle, select the ball-end mill and set our tool comp variable according to user-programmed direction (D).

INITIALIZATION MODULE

To maintain the proper external dimensions, the cutter should stay inside the programmed path. This means that if D is programmed 0 (CW), the cutter should travel to the right of the programmed path, so C (the Frame Mill event's cutter comp entry) should be set to 2. If D is 1 (CCW) the cutter should travel to the left of the programmed path and C should be set to 1[8]. Both cases are illustrated **below**.

[8] See the Frame mill event section of the Delta Customer Information Manual for more information on C and other Frame mill event parameters.


TOOL OFFSETS TO RIGHT (C2)
CLOCKWISE (D0) PROGRAMMED PATH
COUNTERCLOCKWISE (D1) PROGRAMMED PATH
TOOL OFFSETS TO LEFT (C1)

We will assign V27 to represent the Frame Mill event's C entry. We use V27 because it happens to be "free"; many of our local variables will be needed to hold other values because they in some way correspond to global variables that hold related values. This will become more clear further on.

Our initialization module must also activate the flat-end mill so that the submacro FINFRM can get its radius. However, we are allowing the user to specify that no flat-end mill is to be used by entering 0 for global variable U (which is to hold the tool number of the flat-end mill). Since 0 is an invalid T code, we must check for this condition before activating the tool; if U=0 we will simply activate the tool number stored in T instead (this must be a valid T code because it is the tool number of the ball-end mill).

The initialization module is simple enough that we can skip writing an algorithm and proceed right to the code. As a matter of fact, we can write the code from the beginning of DCMAC, commented as before:

```
CODE                    COMMENTS

(DCMAC  )               ;MACRO NAME
N10(CLR)                ;CLEAR FAULT/STATUS DISPLAY (WE BEGIN ALL MAIN MACROS
                        THIS WAY AS A MATTER OF CONVENTION)
N20(M)PUSERR$           ;CALL USERR
N25(IF,#17,35,35,30)    ;IF FLAT-END MILL IS PROGRAMMED THEN
N30(9)M06T#17$          ;ACTIVATE TOOL #17 (USER-PROGRAMMED FLAT-END MILL[9])
N32(GO,40)              ;SKIP NEXT INSTRUCTION
N35(9)M06T#16$          ;IF NO FLAT-END MILL IS PROGRAMMED, ACTIVATE BALL-END
                        MILL INSTEAD
N40(IF,#4,50,50,70)     ;CHECK V4 (D); IF <=0 GO TO 50 ELSE GO TO 70
N50(#27=2)              ;SET V27=2 (C WILL = 2 IN FRAME EVENT) IF CLOCKWISE
N60(GO,80)              ;SKIP NEXT INSTRUCTION
N70(#27=1)              ;SET V27=1 (C WILL = 1 IN FRAME EVENT) IF CCW
```

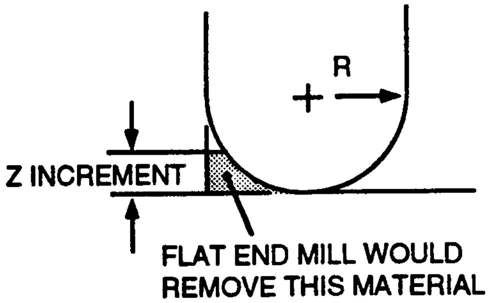
This gives us the first few lines of DCMAC. Before we can proceed we must determine how we are going to offset the ball-end mill and use this information to write our BALLOFFS submacro. Even though this doesn't follow our the calling order in our preliminary algorithm, some of the techniques we develop here will be used in following submacros, and BALLOFFS illustrates them particularly well.

[9] If you have a tool changer and Tool Life Management, you will need to use an M06 to select the tool. If Tool Life Management is not present you may use M03 to activate the spindle.

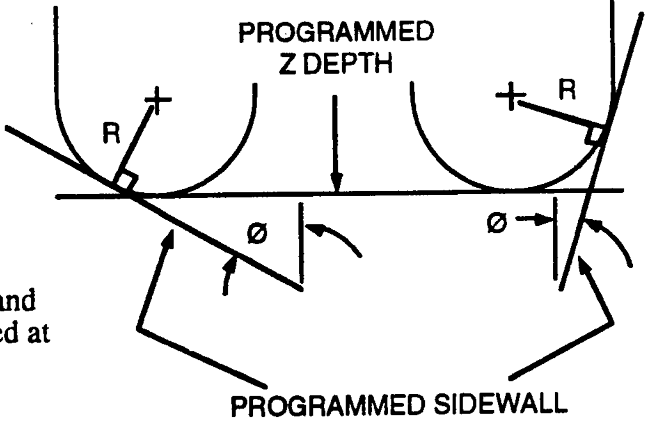
SUBMACRO BALLOFFS

This macro addresses the problem of making a ball-end mill work in an event (the Frame event) that thinks it's using a flat-end mill.

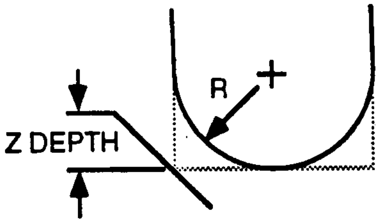
While using a ball-end mill minimizes step-marks on the walls of the cavity, it creates a problem when invoking the Frame mill's Cutter Compensation, which assumes a flat-end mill. When the Z increment is less than the tool's radius[10] a ball-end mill will not remove the same amount of material as a flat-end mill of the same radius (see illustration at **right**). The ball end mill requires an additional offset; its amount depends on what aspect of the tool is to contact the wall's profile.



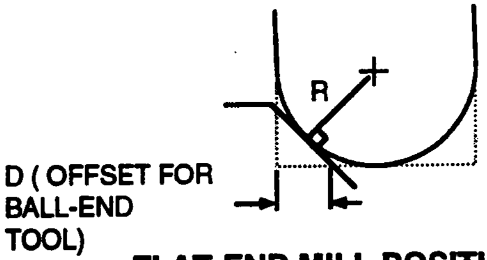
This situation can be handled in a number of ways, but at this point we will assume that the ball-end cutter is always to be tangent to the programmed sidewall. This means that the radius of the tool will always be perpendicular to the sidewall, as shown at **right**. Since we have eliminated all variables except R and Ø for a given offset, we may trigonometrically solve for the offset as a function of these.



As illustrated **below**, what we will call the D offset is just the amount the tool must move so that it becomes tangent to the programmed sidewall; it is constant for a given tool radius and draft angle when both tool shapes are compared at the same Z depth.



FLAT-END MILL POSITION

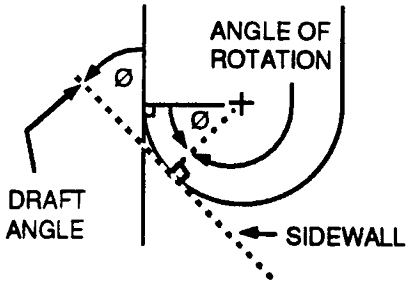


FLAT-END MILL POSITION PLUS D OFFSET

The procedure for deriving the equation for D is illustrated on the following page. We can develop this equation to specify a constant X or Y tool offset for *each* draft angle; since these offsets are valid for a given tool tangent to a given sidewall **at any Z depth**, we can add them to the overall size of the cavity before we begin cutting metal and forget about them afterward. If sizing successive cavities is necessary, they should be sized down from this initial offset cavity.

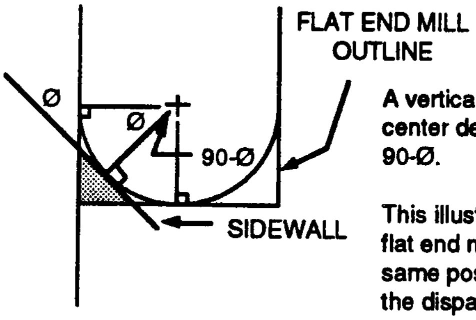
Since these offsets are fairly complicated to compute and since they must be computed for all four side walls, it will simplify our main macro if we designate a separate macro to do the computation.

[10] This is typically the case because small Z increments and large radius tools give smoother angled walls.



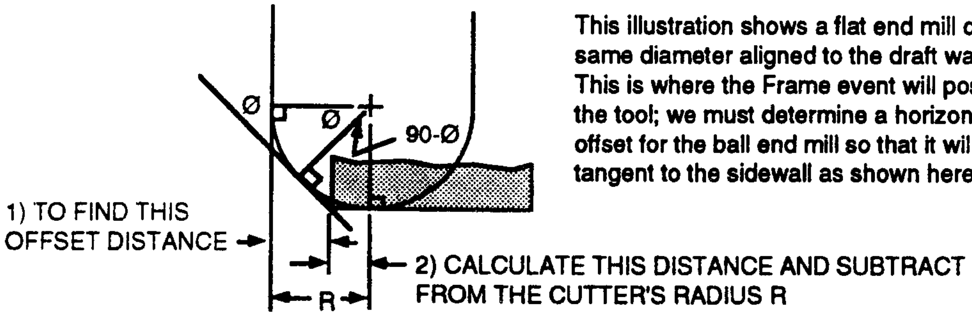
Rotate perpendicular lines through the ball end mill's center to align with draft angle Ø.

Notice that Ø is also the angle of rotation.

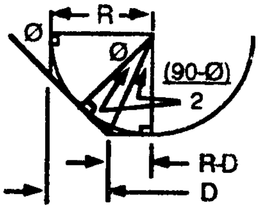


A vertical line through the ball end mill's center defines the complimentary angle 90-Ø.

This illustration also shows the outline of a flat end mill of the same diameter in the same position as the ball end mill. Notice the disparity in material removal.


Ø
Ø
90-Ø
1) TO FIND THIS
OFFSET DISTANCE
2) CALCULATE THIS DISTANCE AND SUBTRACT
FROM THE CUTTER'S RADIUS R
R

This illustration shows a flat end mill of the same diameter aligned to the draft wall. This is where the Frame event will position the tool; we must determine a horizontal offset for the ball end mill so that it will be tangent to the sidewall as shown here.



Bisect the angle 90-Ø to form two congruent triangles. If D is the offset distance, the horizontal leg of the rightmost triangle gives the amount by which R must be reduced to determine the offset D.

R-D=R*TAN((90-Ø)/2)
D=R-R*TAN((90-Ø)/2)

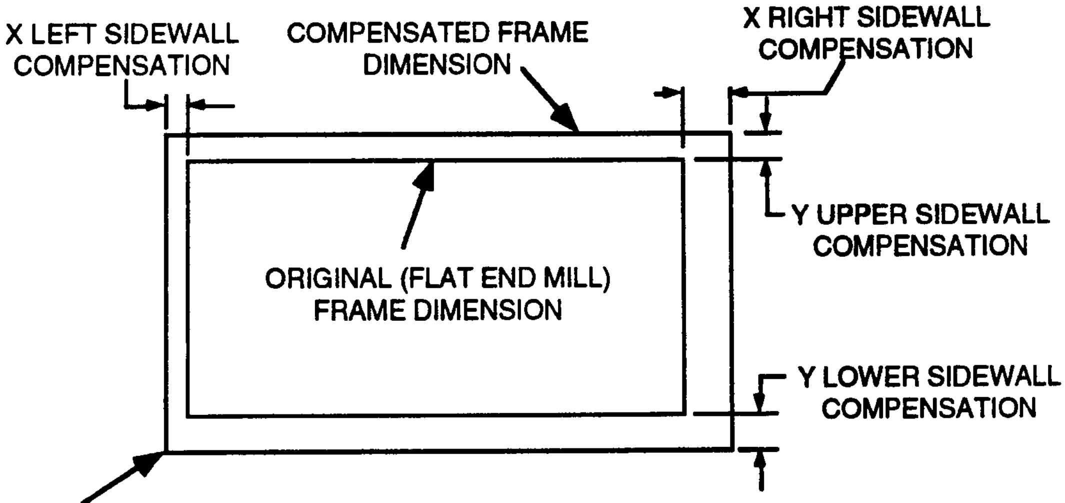
This macro, of course, is the submacro BALLOFFS. Once the equations for the offsets are understood, BALLOFFS is fairly simple to write; all we have to do is decide which variables will hold which values and plug them into our equations.

Recall our variable-numbering scheme for the draft angles; we will stick to that scheme in numbering our offset variables. Variable #50 will hold the offset for the left X sidewall, #51 for the right, #52 for the lower Y, and #53 for the upper. When we call BALLOFFS, V94 will hold the radius of our ball-end mill (this is found in ZINC which is discussed later but called before BALLOFFS in the actual program). The angle corresponding to #5x is in variable #1x for[11] x=0 to 3; we also know that our X and Y surface dimensions are in variables #20 and #21 (globals X and Y) respectively, that our startpoints (globals A and B) are in #1 and #2, and that we must scale and relocate *each successive* frame from these variables *after the variables have been offset for the ball-end mill.* This means we don't need new variables for the adjusted X,Y, A, and B

[11] For example, replace x by 1 and you get variable #11 or #51, which corresponds to the right X sidewall angle or offset. This is a convenient way to reference the various angles and offsets and we will see it again.

values; since we'll never need our originals again we can go ahead and store the adjusted values back into them. It also means we have all the variables we need, so we can proceed with writing the macro.

We don't need an algorithm for this macro because, like most of the modules we have written so far, it is fairly straightforward. We know the equation for the D offset from our derivation above; once it gets this value for each sidewall all BALLOFFS needs to do is add both X offsets from #20, add both Y offsets from #21, and subtract the left X and lower Y values from #1 and #2 (the startpoints) respectively. The reason for our choice of additions and subtractions is illustrated **below**.


X LEFT SIDEWALL COMPENSATION
COMPENSATED FRAME DIMENSION
X RIGHT SIDEWALL COMPENSATION
Y UPPER SIDEWALL COMPENSATION
ORIGINAL (FLAT END MILL) FRAME DIMENSION
Y LOWER SIDEWALL COMPENSATION

STARTPOINT MOVES IN THE NEGATIVE DIRECTION
IN X AND Y; **SUBTRACT** LEFT X AND LOWER Y
SIDEWALL OFFSETS FROM THE STARTPOINT

ABSOLUTE FRAME DIMENSIONS ARE **INCREASED** BY **BOTH** OFFSETS FOR EACH AXIS; **ADD** BOTH X LEFT AND RIGHT OFFSETS TO THE X DIMENSION AND ADD BOTH Y UPPER AND LOWER OFFSETS TO THE Y DIMENSION

The code to accomplish these offsets with the variables we have chosen is listed below:

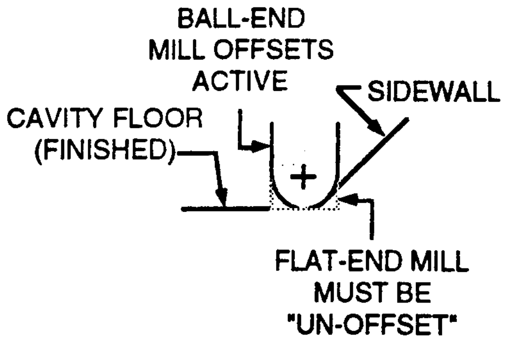
```
(BALLOFFS)                          ;MACRO NAME
N10(#50=#94-#94*TAN,((90-#10)/2))   ;COMPUTE OFFSET FOR LEFT X SIDEWALL
N20(#51=#94-#94*TAN,((90-#11)/2))   ;COMPUTE OFFSET FOR RIGHT X SIDEWALL
N30(#52=#94-#94*TAN,((90-#12)/2))   ;COMPUTE OFFSET FOR LOWER Y SIDEWALL
N40(#53=#94-#94*TAN,((90-#13)/2))   ;COMPUTE OFFSET FOR UPPER Y SIDEWALL
N50(#20=#20+#50+#51)                ;ADD OFFSETS TO X SURFACE DIMENSION
N60(#21=#21+#52+#53)                ;ADD OFFSETS TO Y SURFACE DIMENSION
N70(#1=#1-#50)                      ;ADJUST X STARTPOINT
N80(#2=#2-#52)                      ;ADJUST Y STARTPOINT
N90(9)M30$                          ;END MACRO
```

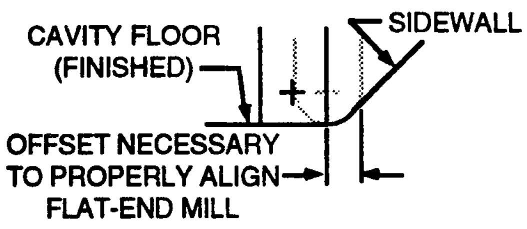
As you can see, this macro is simply a block of code that implements the trigonometric equations we derived above. The block could be placed directly in DCMAC, but for purposes of clarity and modularity we have made it a separate macro. In writing macros you will find that separating chunks of code like this one will make the macro easier for you to implement and debug and simpler for others to understand.

As we shall see, we'll need to cancel these offsets for the flat-end mill in the submacro FLATOFFS.

SUBMACRO FLATOFFS

The final frame of our Draft Angle Cavity will be cut twice; once by the ball-end mill to provide a smooth join between the angled walls and the floor, and once by a flat-end mill to remove roughing marks left on the floor by the ball-end mill. However, when either tool reaches the bottom, ball-end mill offsets will still be active unless we do something about them. The problem is illustrated **below**. The purpose of this submacro is to determine "un-offsets" for the flat-end mill so that the join is smooth between the flat cavity floor, the curved edge and the cavity walls.





For a flat-end mill equal in radius to the ball-end mill used for the rest of the cavity, this offset amount is equal to the offset we found in the BALLOFFS macro for keeping the ball-end mill tangent to the sidewall. However, if the flat-end mill is a different radius (and we allow for this), we must include the difference between the two radii in our "un-offset". We do this by subtracting the radius of the ball-end mill from that of the flat-end mill. If the flat-end mill is larger, this gives us a positive number which we add to the ball-end mill offsets to get our "un-offset". If the ball-end mill is larger, it gives us a negative number; this added to the ball-end mill offsets will again give us the correct value (we need to offset the flat-end mill *less* if it is smaller than the ball-end mill and *more* if it is larger).

Our formula is then: V6x=V94*TAN((90-#1x)/2+(V34-V94) where V6x is the flat-end mill offset, V1x is the draft angle and (V34-V94) is the difference between the ball-end mill radius (V94) and the flat-end mill radius (V34). V34 will be computed by FINFRM, which we discuss next but call prior to FLATOFFS in the actual program.

Since FLATOFFS simply computes four values, it does not need an algorithm. The code follows.

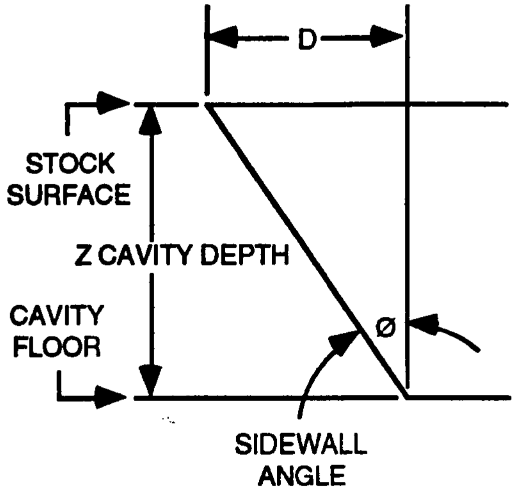
```
CODE                                        COMMENTS

(FLATOFFS)                                  ;MACRO NAME
N10(#60=#94*TAN,((90-#10)/2))+(#34-#94)     ;FIND OFFSET FOR LEFT SIDEWALL
N20(#61=#94*TAN,((90-#11)/2))+(#34-#94)     ;FIND OFFSET FOR RIGHT SIDEWALL
N30(#62=#94*TAN,((90-#12)/2))+(#34-#94)     ;FIND OFFSET FOR LOWER SIDEWALL
N40(#63=#94*TAN,((90-#13)/2))+(#34-#94)     ;FIND OFFSET FOR UPPER SIDEWALL
N50(#35=#20-#30-#31-#60-#61)                ;APPLY OFFSETS TO DIMENSIONS...
N60(#36=#21-#32-#33-#62-#63)
N70(#25=#1+#30+#60)                         ;APPLY OFFSETS TO START POINTS...
N80(#26=#2+#32+#62)
N90(9)M30$                                  ;END MACRO
```

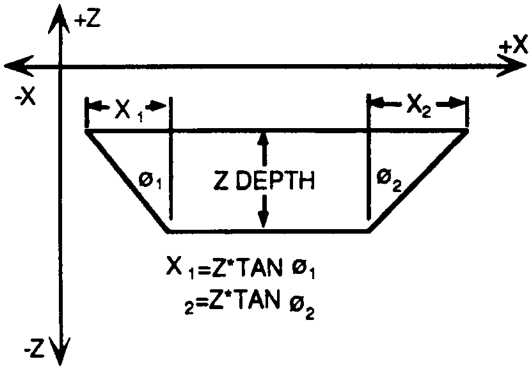
Next we must find the cavity floor dimensions (the dimensions of the last frame we mill). For this we need to write the submacro we named FINFRM in Step 2.

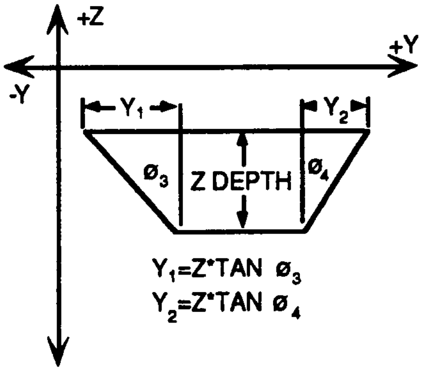
SUBMACRO FINFRM

FINFRM is simple enough that we don't need to write an algorithm for it; however, we do need to derive the equations that we will use to find the final cavity dimensions, and decide what variables we will use to store them.

The distance we need FINFRM to find is labeled D in the illustration at **right**. Since we know the Z cavity depth and the draft angle for each sidewall, we can trigonometrically solve for D using the formula D=Z*TAN Ø, where Z is the total Z cavity depth. We must find this distance for each of the four side walls as illustrated **below**.







Now we can make use of our variable numbering scheme for each of the four draft angles. Recall that V10 holds the left X sidewall draft angle, V11 the right X angle, V12 the lower Y angle and V13 the upper Y angle. Skipping V20-23 since they include globals, we use V30-33 to hold each sidewall's corresponding final cavity offset--in other words, the D value for that specific sidewall. From the formula we derived above, V3x=V22*TAN(V1x) (V22 is the user-programmed final Z depth). Substitute 0 through 3 for x in this formula and it will yield an offset for every angle V10 through V13. Now we can use these offsets—along with those we found in FLATOFFS—to find the X and Y start points of the final frame, which will be stored in V25 and V26 respectively, and the X and Y dimensions of the final frame, which will be stored in V35 and V36. (These are arbitrary variable numbers, but they stick to our convention of grouping similar values according to their variable numbers.)

To find the start points of the final frame we add[12] the left X and lower Y offsets (V30 and V32 from above and V60 and V62 from FLATOFFS) to the user-programmed X and Y start points (V1 and V2) to get the formulas V25=V1+V30+V60 and V26=V2+V32+V62. Similarly, to find the final frame dimensions we subtract both X and both Y offsets from the cavity's stock surface dimensions, yielding V35=V20-V30-V31-V60-V61 and V36=V21-V32-V33-V62-V63.

[12] The cavity is shrinking, therefore the startpoint is moving "inward"—to the right in X and up in Y, so we *add* these offsets to the original startpoint.

For purposes of error checking, we need FINFRM to find the radius of the flat-end mill (if one is programmed). Finding this value here will allow us to check the dimensions and corner radius of the final frame against it in the next module so that all our error checking is out of the way before we start computing tool offsets (remember, FINFRM is the second submacro we call and only the third module in the program; LOCERR will be the fourth). We will also need it in FLATOFFS, which we discussed earlier but which will not be called until after FINFRM in the actual program. If the user programmed 0 for the flat-end mill tool number, the ball-end mill will be active on entry to FINFRM and so its radius will be found and stored as the flat-end mill radius; however, this value will never be used because the final pass will never execute.

We will use V74 to store the tool number of the flat-end mill[13], V84 to store its diameter, and V94 to store its radius, which we shall compute. Since all we really want is V94 (V74 and V84 are necessary intermediate steps), we will use these variables again later to store similar values for the ball-end mill; therefore we will store the flat-end mill radius in V34 so it won't be lost when we get the ball-end mill radius later. Though this process may seem complicated, it is done to simplify things: it allows us to get each radius only once, and since the ball-end mill is the first tool used but the second radius we get, we can leave its radius safely in V94.

```
CODE                                   COMMENTS

(FINFRM)                               ;MACRO NAME
N10(#30=ABS,#22*TAN,#10)               ;LEFT SIDEWALL OFFSET
N20(#31=ABS,#22*TAN,#11)               ;RIGHT SIDEWALL OFFSET
N30(#32=ABS,#22*TAN,#12)               ;LOWER SIDEWALL OFFSET
N40(#33=ABS,#22*TAN,#13)               ;UPPER SIDEWALL OFFSET
N50(#25=#1+#30+#60)                    ;X STARTPOINT OF FINAL FRAME
N60(#26=#2+#32+#62)                    ;Y STARTPOINT OF FINAL FRAME
N70(#35=#20-#30-#31-#60-#61)           ;X DIMENSION OF FINAL FRAME
N80(#36=#21-#32-#33-#62-#63)           ;Y DIMENSION OF FINAL FRAME
N90(#74=GAO,20)                        ;GET ACTIVE TOOL NUMBER
N100(#84=GOV,20,#74,4)                 ;GET ACTIVE TOOL DIAMETER
N110(#94=#84/2)                        ;GET RADIUS FROM DIAMETER
N120(#34=#94)                          ;STORE RADIUS
N130(9)M30$                            ;END OF MACRO
```

After we call FINFRM early in DCMAC we will be ready to complete our error checking with LOCERR.

SUBMACRO LOCERR

LOCERR is another simple error checking macro. All it needs to do is check the X and Y dimensions of the final cavity (computed above in FINFRM) against twice the user-programmed corner radius (V14) and against the diameter of the flat-end mill which will be used to finish the cavity floor. If X or Y is less than either of these values, LOCERR should flag a fault and suspend execution just like USERR did for the variables it checked. Error-free execution of LOCERR means that *all* of our variables are within range and DCMAC may proceed without further checking.

The algorithm for LOCERR is identical for the one previously given for USERR, so we will proceed with writing the code:

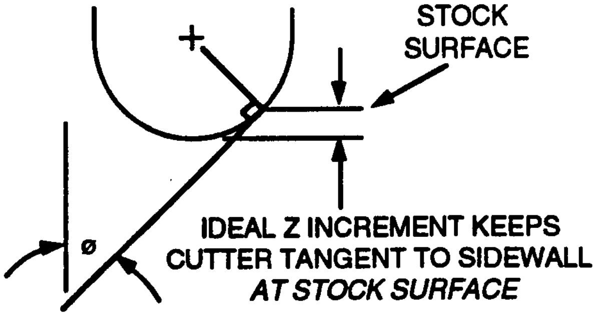
[13] Even though the user has programmed a tool number, if the Tool Life Management option is present the tool that is actually used may have a different number.

```
CODE                                  COMMENTS

(LOCERR)                              ;MACRO NAME
N05(#95=2*#14)                        ;SET #95=2*CORNER RADIUS
N10(IF,(#35-#95),20,40,40)            ;IF X<2R GO TO 20 ELSE GO TO 40
N20(ERR,X<2R FOR FINAL FRAME)         ;FLAG ERROR
N30(GO,120)                           ;BRANCH TO PROGRAM STOP
N40(IF,(#36-#95),50,70,70)            ;IF Y<2R GO TO 50 ELSE GO TO 70
N50(ERR,Y<2R FOR FINAL FRAME)         ;FLAG ERROR
N60(GO,120)                           ;BRANCH TO PROGRAM STOP
N65(#95=2*#34)                        ;SET #95=2*TOOL RADIUS
N70(IF,(#35-#95),80,80,100)           ;IF X<= 2* TOOL RADIUS OF FLAT-END MILL
N80(ERR,X<=TOOL DIA)                  ;THEN FLAG ERROR
N90(GO,120)                           ;AND BRANCH TO STOP
N100(IF,(#36-#95),110,110,130)        ;ELSE IF Y<=2*TOOL RADIUS
N110(ERR,Y<=TOOL DIA)                 ;THEN FLAG ERROR
N120(9)M00$                           ;AND STOP
N125(GO,10)                           ;BRANCH TO START
N130(9)M30$                           ;END MACRO
```

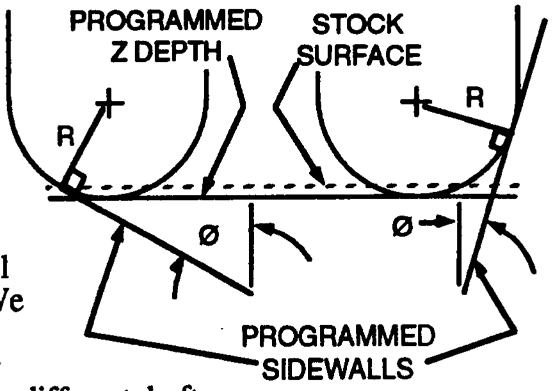
Notice that here, as in USERR, we used #95 as a scratchpad for comparison to X and Y, only this time for two values: in the first half of the macro it held twice the frame corner radius; in the second half (after N65) it held twice the tool radius. Remember that using a variable for more than one purpose like this is acceptable as long as what you are storing in it is for temporary use only.

SUBMACRO ZINC

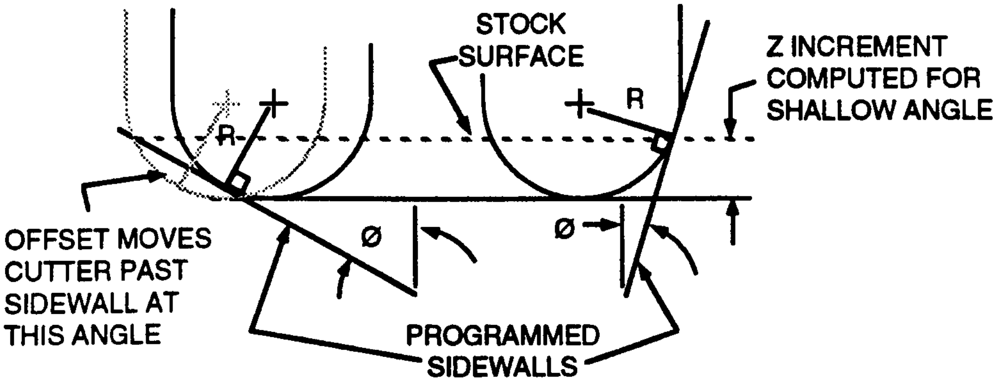
Before we can begin milling our cavity, we must determine our Z increment.


STOCK SURFACE
ø
IDEAL Z INCREMENT KEEPS CUTTER TANGENT TO SIDEWALL *AT STOCK SURFACE*

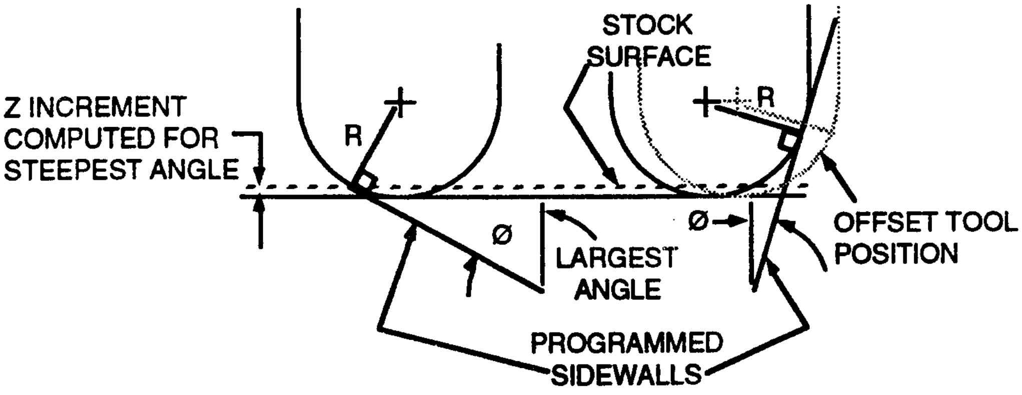
So far we have not considered how the Z depth of each frame will be determined. We could allow the user to program a Z increment, but, since we have compensated the cutter to be tangent to each sidewall, in nearly all cases this would create an error in the part's surface dimensions. The illustration at **right** shows that for our compensated cutter, an ideal first pass depth for a given draft angle places the cutter tangent to the sidewall at the stock surface; we cannot expect the user to compute this depth and even if we could the **lower right** illustration shows how a Z depth that is ideal for one angle is not ideal for any other angle. Since all four draft angles may be different, we must choose our increment so as to allow us to mill accurate surface dimensions in this "worst case" scenario.



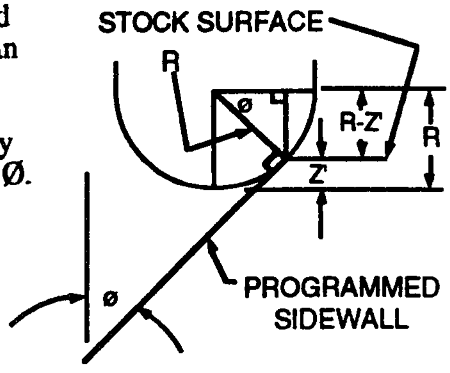
Unfortunately, there is no single Z increment that will allow us to do this with four different draft angles. We can, however, compute a Z increment that places our cutter perfectly tangent to one sidewall and then later compute offsets for the other three (assuming they use different draft angles).


STOCK SURFACE
Z INCREMENT COMPUTED FOR SHALLOW ANGLE
R
R
Ø
Ø
OFFSET MOVES CUTTER PAST SIDEWALL AT THIS ANGLE
PROGRAMMED SIDEWALLS

This raises the question of which sidewall we should choose the ideal Z increment for. The illustration **above** shows a Z depth computed for the (numerically) smallest draft angle. Using this increment, if we offset for a (numerically) larger angle we can obtain an accurate surface dimension but below the surface the cutter will exceed the sidewall at the larger angle. However, if we choose an increment based on the largest angle, the cutter can be offset to give accurate surface dimensions without exceeding any sidewall (see the **following** illustration).



Our Z increment, then, will be determined by the largest draft angle. Later, we will have to come up with a technique to find this draft angle, or alternately, the shallowest Z increment which it produces. We will use this information in a submacro discussed further on to generate additional offsets for all but the largest draft angle sidewall.



The illustration at **right** shows the ideal Z depth (denoted here as Z') for a given tool and given draft angle. We can compute Z' by finding the distance labeled R-Z' and subtracting it from R. R-Z' forms a right triangle with hypotenuse R and angle Ø; we can therefore find R-Z' by the equation R-Z'=R*SIN Ø; this gives us Z'=R-R*SIN Ø.

Notice that as the draft angle approaches 0 this formula causes our Z increment to approach the tool radius[14]. Since it is possible that all draft angles could be close to (or even equal to) 0, we may want to allow the user to limit the maximum size of the Z increment. This way the wall resolution can be increased on shallow angle cavities and the user need not worry about overloading the cutter if the macro computes a large Z increment.

[14] The SIN function approaches 0 as Ø approaches 0.

This, then, will give us a formal definition for our global variable S, which we previously called the Z increment. The user will program his largest acceptable Z increment (based on material and cutter limitations) into S; our macro will then compute its own ideal Z increment, test it against S, and use the smaller of the two. This will not bother our ball-end mill offsets (computed in BALLOFFS) because they are computed based on the cutter being tangent to each of the sidewalls, and this will be the case regardless of the Z increment we choose.

ZINC is more complicated than all the submacros we have discussed so far because it is required to make not only computations but some decisions based on those computations. Namely, after finding the radius of the ball-end mill, ZINC must use that radius to find the ideal Z increment for each sidewall angle, find the smallest and largest of those increments, compare this value to the user programmed limit and finally store the smaller of the two in our Z increment variable. Note that we must find both the smallest Z increment (the increment to be used between every frame) and the largest Z increment (which we will use in the following submacro TOPOFFS to signal the depth at which we no longer need to add stock surface offsets). It would therefore be useful to arrange the increments in order from smallest to largest; we can do this with a simple *sort* routine[15]. Because of its complexity, we will write an algorithm for ZINC, using the same style we used in writing our preliminary algorithm for DCMAC in Step 2:

```
ZINC is called by DCMAC; it will then
    find the active tool number[16];
    find the active tool diameter;
    find the active tool radius;
    compute Z increment for each angle (=R-R*SIN Ø);
    SORT these values to find smallest increment;
    if smallest increment is larger than S (user programmed limit) then
        store S in final increment variable;
    else
        store the smallest increment found in the sort in the final increment variable;
END ZINC.
```

We already know how to find the active tool number, diameter and radius (we did it for the flat-end mill in FINFRM). For this we will again use variable #74 for the tool, #84 for the diameter, and #94 for the radius, but this time, since we don't need #94 to hold another tool radius, we will leave the radius of the ball-end mill in it. Finding the Z increment for each angle is simply a matter of plugging symbolic variable numbers into the formulas we derived above; we will use variables #5x to hold the ideal increment for each angle #1x. Sorting the increments is a slightly more complicated task—so complicated, in fact, that we will designate *another* submacro (or sub-submacro if you prefer) to do it. This macro, which we will call SORT, is described below.

(SUB-) SUBMACRO SORT

Among the most abundant pre-written algorithms available to the computer programmer are those used to sort values, and the CPC feature can implement most of them. However, since processing time is not a major concern in CNC macros (the time required for axis motion is several orders of magnitude larger than the time required for a microprocessor to do most calculations) and since we are dealing with a small number of values in the first place, we will use a simple intuitive algorithm called the "bubble" sort.

[15] We must check every value to find the smallest and largest and it is actually quicker to put them in order than it would be to go through the list and pick them out.

[16] We shall insure that the ball-end mill will be active when ZINC is called.

The bubble sort gets its name from the way it works: the "lightest" (smallest) element "floats" to the top of the list much like bubbles float to the surface of water. By the same process, the rest of the elements get sorted in order so that the next smallest is second in the sorted list, the next is third, and so on, with the largest element ending up at the end.

The bubble sort works by going through the entire list once for every element (if there are 10 elements, it goes through the list 10 times). On each pass it compares the current element with the next element and moves whichever is smaller toward the top of the list. At the end of the last pass, the list is sorted. This process may be clearer if we look at the algorithm:

```
SORT is called by ZINC; SORT will then
    initialize a counter to 0;
    while counter <=4 (there are four elements in the list) do:
        initialize an increment number holder to 54 (V54 holds the first unsorted
        increment)
        while increment number < 57 (V57 holds last unsorted increment) do:
            if current increment value <= next increment value then
                increment counter;
            else
                store current increment in temporary variable;
                store next increment value in current increment holder;
                store old current increment (from temp) in next increment holder;
                add one to increment number holder (V5x);
        end while;
        increment counter;
    end while;
end SORT.
```

Notice the nested while loop structure. The inner loop goes through the entire list of increments (V54-V57), swapping the current with the next if the next is smaller and incrementing the increment number holder in either case. After the inner loop executes, the counter gets incremented and the process is repeated until the counter =4 (a condition checked by the while loop), at which point the list is sorted. Try this algorithm on paper with a scrambled list of four or five numbers to see exactly how it works.

The CPC language has no while loops as we have illustrated them above, but they are often useful in writing algorithms and they can be easily implemented using IF and GO statements. This is illustrated in the macro code:

```
CODE                                COMMENTS
(SORT    )                          ;MACRO NAME
N10(#98=1)                          ;INITIALIZE COUNTER
N20(#99=54)                         ;INITIALIZE INCREMENT NUMBER HOLDER
N30(IF,#98-4,40,40,160)             ;IF COUNTER ≤ 4 THEN (OUTER WHILE)
N40(IF,#99-57,50,140,140)           ;    IF INCREMENT NUMBER < 57 THEN (INNER WHILE)
N50(IF,(#(#99)-#(#99+1)),60,60,80)  ;        IF CURRENT INCREMENT ≤ NEXT THEN
N60(#99=#99+1)                      ;            INCREMENT COUNTER
N70(GO,30)                          ;            AND LOOP TO N30 (CHECK NEXT TWO)
N80(#97=#(#99))                     ;        ELSE STORE CURRENT INCREMENT IN #97
N90(#(#99)=#(#99+1))                ;            STORE NEXT INCREMENT VALUE IN
                                    ;            CURRENT  INCREMENT NUMBER
N100(#(#99+1)=#97)                  ;            STORE OLD CURRENT INCREMENT IN
                                    ;            NEXT INCREMENT NUMBER
N110(#99=#99+1)                     ;            INCREMENT INCREMENT NUMBER HOLDER
N120(GO,40)                         ;            LOOP TO N40
N140(#98=#98+1)                     ;    INCREMENT COUNTER
N150(GO,20)                         ;LOOP TO N20
N200(9)M30$                         ;END MACRO
```

It is important to remember that this macro makes use of indirect[17] variable assignments; #99 holds a value from 54 to 57 which represents the variable number of the variable that holds the current Z increment. For instance, if #99 = 55, then the current Z increment would be the value held in #55. In effect, the macro code "(#(#99))" means "the value contained in the variable number whose variable number is contained in #99". By using this feature we can sort the Z increments in one relatively short loop; in fact, we can sort just about any number of values by simply changing a few variables in the code listed above. For example, if we had a list of 12 numbers contained in variables #33 through #44 we could use this macro to sort them by simply initializing #99 to 33 at N20, subtracting 44 from #99 at N40 and subtracting 12 from #98 at N30 to check for the end of the loop. You may find this macro useful in other situations where sorting is necessary.

After running SORT, the smallest value is stored in V54; all ZINC has to do now is check this value against the user-programmed limit. If the limit is smaller, it replaces the value in V54 for use as the actual Z increment and ZINC is finished. If not, V54 stays the same and ZINC is finished.

Now we can write ZINC, based on the algorithm given prior to the SORT macro description:

```
(ZINC)                          ;SUBMACRO NAME
N10(#74=GAO,20)                 ;GET TOOL NUMBER FOR BALL-END MILL
N20(#84=GOV,20,#74,4)           ;GET DIAMETER OF BALL-END MILL
N30(#94=#84/2)                  ;COMPUTE RADIUS OF BALL-END MILL
N40(#54=#94-(#94*SIN,#10))      ;FIND IDEAL Z INCREMENT FOR LEFT X ANGLE
N50(#55=#94-(#94*SIN,#11))      ;FIND INCREMENT FOR RIGHT X ANGLE
N60(#56=#94-(#94*SIN,#12))      ;FIND INCREMENT FOR LOWER Y ANGLE
N70(#57=#94-(#94*SIN,#13))      ;FIND INCREMENT FOR UPPER Y ANGLE
N80(M)PSORT$                    ;SORT INCREMENTS
N90(IF,(#54-#15),110,110,100)   ;CHECK SMALLEST INCREMENT FROM SORT
                                 AGAINST USER-PROGRAMMED LIMIT
N100(#54=#15)                   ;STORE LIMIT AS Z INCREMENT IF SMALLER
N110(9)M30$                     ;END MACRO
```

SUBMACRO TOPOFFS

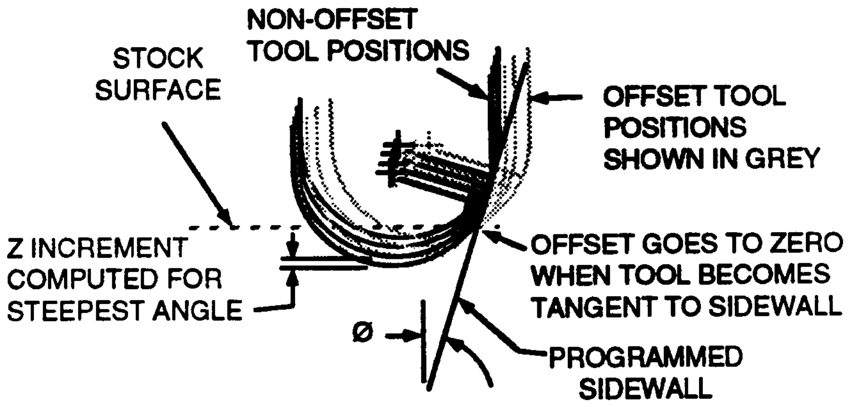
Now that we have our ideal Z increment, we must determine how to apply offsets to keep our stock surface dimension is accurate. This will be the job of the submacro TOPOFFS. The stock surface offsets determined by TOPOFFS are in addition to offsets that depend on the tool's Z depth (how much we must shrink each successive frame to give the angled sidewall effect). Unlike any of our previous macros, TOPOFFS will actually call the Frame event and do some machining; as we shall see, however, after a certain depth is reached in the stock TOPOFFS will no longer be necessary. The machining will then be done in DCMAC.

The point here is that we are dealing with two sets of offsets for the first few frames and one set for all the rest. The offsets computed by TOPOFFS only apply near the surface of the stock; the others apply at every Z depth (if they didn't, our cavity would not have angled sidewalls). This makes TOPOFFS much more complicated to design and write than any of our previous submacros; it also means we can simplify things a little by putting the offsets for the present Z position (which we need for every frame in the cavity) into a separate macro which we can call from both TOPOFFS and DCMAC.

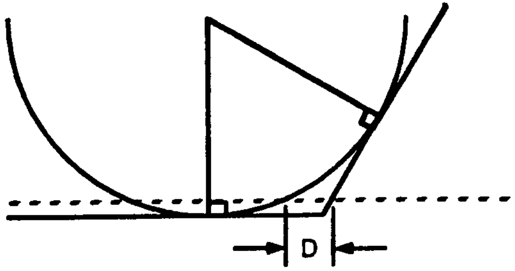
[17] Indirect variable addressing is more thoroughly defined in Part 2.

However, before we consider another submacro or the design of TOPOFFS, we should look at the stock surface offsets necessary to give our cavity true surface dimensions.

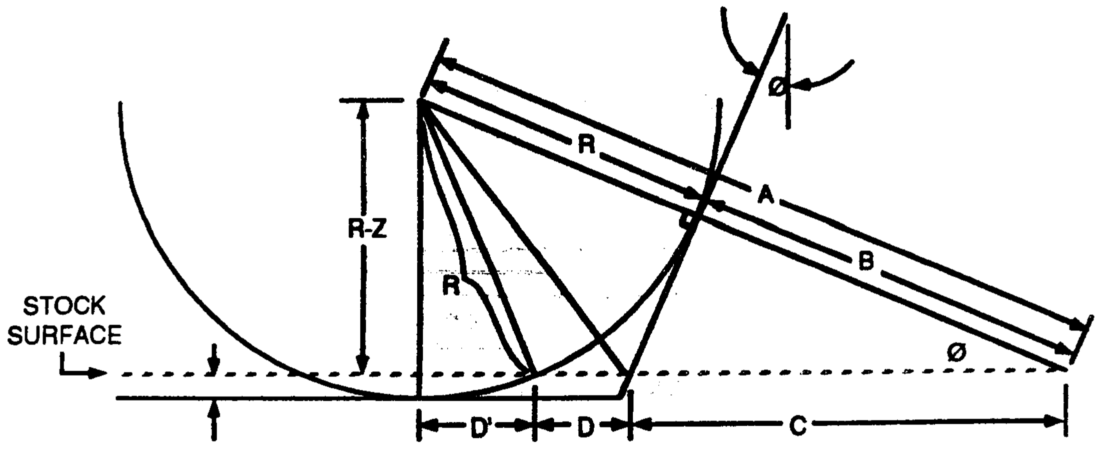
Our tool will be tangent to the steepest draft angle; we will offset the tool for the more shallow draft angles to provide an accurate stock surface dimension. On successive passes these offsets will decrease and eventually become zero. When they are all zero the tool is deep enough into the stock so that it is tangent to all the draft angles and we no longer need to be concerned with these offsets. This is shown in the following illustration.



Remember, our dimensions have been adjusted so that the tool is *tangent* to every sidewall; these new offsets simply allow for a more accurate surface dimension. The offsets go to zero at the ideal Z depth for a given angle; therefore our macro must stop adding offsets for a particular angle at this depth. When the ideal Z depth is reached for the shallowest angle (we will call this point the Z threshold), all the offsets will be zero.



The illustration at right clearly shows the offset. Our next task is to determine an equation for this offset. Since the trigonometry involved is rather complex, we need to expand our illustration as shown below[18].



[18] Do not confuse the A, B, C, and D values here with the macro's global variables A, B, C, and D.

To find our offset D we must find the distances D' and C and subtract them from E. We can find C as the hypotenuse of the triangle it forms with B and angle Ø, but first we must find B. B=A-R and A can be found as the hypotenuse of the triangle it forms with Ø and the line labeled R-Z. Since we know R (the tool radius), Z (the present Z depth) and Ø we can solve for A using the SIN function:

$$\text{SIN } \emptyset = (R-Z)/A \text{ so}$$
$$A = (R-Z)/\text{SIN } \emptyset$$

With A we can get B: $B=A-R$

And use it to solve for C: $\text{COS } \emptyset = B/C$ so
$C = B/\text{COS } \emptyset$

We can find E as the leg of the triangle it forms with R-Z and A:

$$(R-Z)^2 + E^2 = A^2 \text{ so}$$
$$E^2 = A^2 - (R-Z)^2 \text{ and}$$
$$E = (A^2 - (R-Z)^2)^{.5}$$

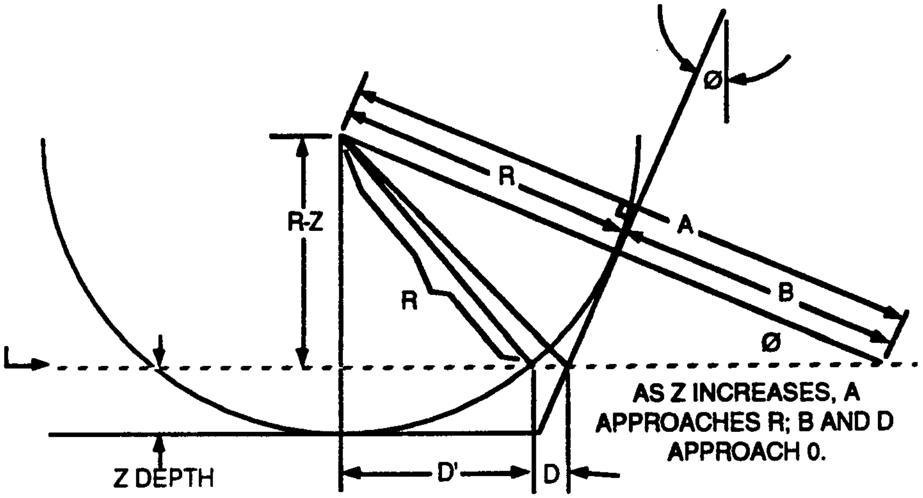
Similarly, D' is a leg of a triangle with R and R-Z. It can be found by the Pythagorean formula:

$$D' = (R^2 - (R-Z)^2)^{.5}$$

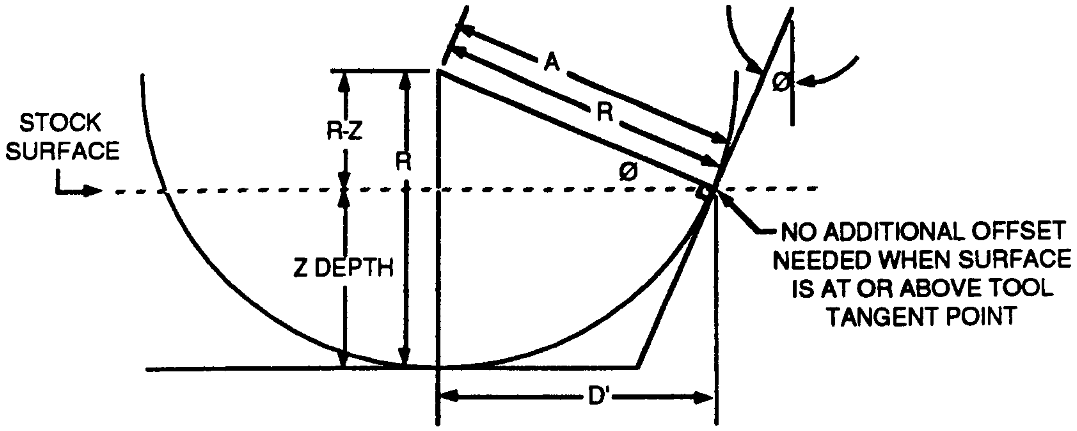
And finally, D = E - D' - C.

In the actual macro, we will compute the values A, B and $(R-Z)^2$ as intermediate steps; the D offset will then be computed by the substituting the E, D' and C formulas into D=E-D'-C to give:
$D=(A^2 - (R-Z)^2)^{.5}-(R^2 - (R-Z)^2)^{.5}-B/\text{COS } \emptyset$.

This, then, is the formula for our additional stock-surface offset. As the tool gets deeper in the stock, A approaches R and B approaches 0 (compare the illustration **below** with the first illustration).



This means when the offset is no longer needed, A=R and B=0; if B=0, B/COS Ø =0, and if A=R, $(A^2 - (R\text{-}Z)^2)^{.5}\text{-}(R^2 - (R\text{-}Z)^2)^{.5}$ equals 0, so D=0. This is how the offsets cancel themselves out when the tool gets deep enough into the stock so that they are no longer needed. This situation is shown in the **following** illustration.


A
R
Ø
Ø
STOCK SURFACE
R-Z
R
Z DEPTH
NO ADDITIONAL OFFSET NEEDED WHEN SURFACE IS AT OR ABOVE TOOL TANGENT POINT
D'

Our TOPOFFS macro will use the formula we derived for D and execute it for each of the four draft angles at each successive Z increment. Given different draft angles, we will start out with a D offset for each angle, and as we get deeper in the stock the D values for the smaller angles will start to reach 0. Eventually all the D values will equal 0, and at this point we can stop calling TOPOFFS (there's no sense in computing offsets if they're always going to be equal to 0). Since we have sorted the ideal Z increments for each draft angle, knowing when we no longer need special offsets is simple: just check the current Z depth against the largest Z increment value. Remember, when the cutter is tangent to the sidewall no more stock surface offsets are required for that angle, and this largest Z increment is the point at which the cutter becomes tangent to the sidewall for the steepest draft cavity, so at this point we are done with our stock surface offsets. An easy way to check whether we are past the threshold is to set a flag when we reach it and check the flag before calling TOPOFFS. We will see how this works in the algorithm and code.

Here is the algorithm for TOPOFFS:

```
DCMAC calls TOPOFFS which will then:
    call the submacro PRESOFFS to find the amount to shrink and relocate Frame
    for present Z depth;
    check to see if the current Z depth is below the biggest ideal Z increment {stored
    in V57 as a result of SORT};
    if above biggest Z increment depth then:
        initialize angle number counter to 10 {variable # of first sidewall angle};
        set top offset number counter to 70 {variable # of first sidewall top offset};
        while angle number counter <13 do:
            if current sidewall angle > 0° then
                A=(R-Z)/SIN Ø;
                B=A-R;
                if B<0 then
                    set B=0 {we may get a slightly negative B due to rounding error};
                F=(R-Z)*(R-Z)     {we add F as an intermediate variable; it does not
                                  appear in our derivation above};
                #7x=$(A^2 - (F)^2)^{.5}\text{-}(R^2 - (F)^2)^{.5}$-B/COS Ø   {find actual offset—#7x is
                                                       contained in the "top offset
                                                       number counter"};
                if #7x<0 then
                    set #7x = 0 {rounding error check};
            end if;
            increment angle number counter;
            increment top offset number counter;
        end while;
        adjust current frame dimensions to reflect these offsets;
        adjust current frame startpoints to reflect these offsets;
        clear threshold flag (indicate no more need for calling TOPOFFS);
        jump to end;
    end if;
    set threshold flag   {indicate Z threshold not yet reached and TOPOFFS must be
                          called again};
END TOPOFFS.
```

Note: all letter variables shown in the above algorithm relate to corresponding variables in our derivation on the previous page ***only****; do not confuse them with global symbolic variables (i.e. A as used here does* ***not*** *equal variable #1).*

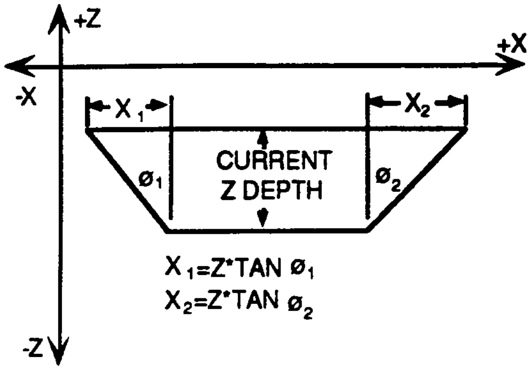
Before we can proceed with writing the code for TOPOFFS we must get our PRESOFFS submacro written.

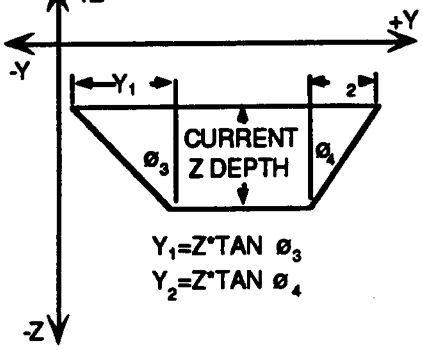
SUBMACRO PRESOFFS

The purpose of PRESOFFS is to find the amount to resize and relocate the current frame so that the final cavity (a series of progressively deeper frames) has angled sidewalls.

Notice from the algorithm for TOPOFFS that PRESOFFS is called once each time through the loop; this will also be true when TOPOFFS is done and we start calling PRESOFFS from DCMAC. What this means is that PRESOFFS is purely an offset-finding macro; we tell it where we're at in Z and it finds resize and relocate offsets for each sidewall for that position. It needn't be concerned with getting to that position or milling the frame.

This makes PRESOFFS very easy to write; as a matter of fact, FINFRM (which finds the position and size difference between the first and last frame) is virtually identical except for a few variable names and the portion which finds the radius of the flat-end mill. We therefore will only briefly recap the derivation of the necessary equations.





The values we must find are shown **above**. As with FINFRM, we must subtract the offsets for both X and both Y sidewalls from the corresponding surface dimensions to get the new frame dimensions for a given Z depth. The current frame startpoint is determined by adding the left X and lower Y offsets to their corresponding axis startpoints. The only major difference between this and FINFRM is that we are using the current Z depth (held in variable #64) instead of the final Z depth and we are using different variables to hold the offsets and adjusted starting point/size coordinates.

We will store the offset values for each sidewall in V4x; the adjusted frame coordinates in V45 and V46 (X and Y respectively) and the adjusted startpoint coordinates in V23 and V24 (A and B respectively). The code for PRESOFFS follows.

```
CODE                          COMMENTS

(PRESOFFS)                    ;SUBMACRO NAME
N10(#40=ABS,#64*TAN,#10)      ;LEFT SIDEWALL OFFSET FOR CURRENT Z DEPTH
N20(#41=ABS,#64*TAN,#11)      ;RIGHT SIDEWALL OFFSET
N30(#42=ABS,#64*TAN,#12)      ;LOWER SIDEWALL OFFSET
N40(#43=ABS,#64*TAN,#13)      ;UPPER SIDEWALL OFFSET
N50(#45=#20-#40-#41)          ;X DIMENSION OF CURRENT FRAME
N60(#46=#21-#42-#43)          ;Y DIMENSION OF CURRENT FRAME
N70(#23=#1+#40)               ;X STARTPOINT OF CURRENT FRAME
N80(#24=#2+#42)               ;Y STARTPOINT OF CURRENT FRAME
N90(9)M30$                    ;END OF MACRO
```

We have written PRESOFFS so that the offsets at each depth are found relative to the *surface* of the cavity. We could have found them relative to the last frame (in which case they would be constant for each resize/relocation and they would only have to be found once, at the beginning of DCMAC), but the cavity surface dimensions give us a known, constant starting point. This is very important because finding the offsets relative to the previous frame could build up round-off errors and degrade the accuracy of our cavity dimensions progressively as the program neared the bottom.

Now we can write the code for TOPOFFS based on the algorithm given previously. Keep in mind that V57 holds the largest Z increment (the point at which we stop computing surface offsets), V69 holds the SV number of the variable[19] that holds the draft angle we are dealing with, V68 holds the SV number of the variable that holds the current stock surface offset, V65 through V67 are intermediate variables for the computation of the stock surface offset (labeled A, B, and F in the section where we derived the equations), V45 and V46 are the X and Y dimensions of the current frame, V23 and V24 are the startpoints of the current frame and V99 is a flag which is 0 as long as we need stock surface offsets (our current Z position is above the Z threshold held in V57) and 1 when we no longer need to call TOPOFFS.

[19] Indirect variable addressing—see the description of SORT for further explanation.

```
CODE                                    COMMENTS
(TOPOFFS)                               ;SUBMACRO NAME
N005(M)PPRESOFFS$                       ;CALL PRESOFFS FOR CURRENT FRAME
                                        OFFSETS
N010(IF,-#57-#64,20,20,230)             ;JUMP TO END IF PAST THRESHOLD
N020(#69=10)                            ;INITIALIZE ANGLE NUMBER HOLDER TO 10
N030(#68=70)                            ;INITIALIZE OFFSET NUMBER HOLDER TO 70
                                        (OFFSET CORRESPONDING TO ANGLE 10)
N040(IF,#69-13,50,50,170)               ;IF NOT DONE WITH ALL SIDEWALLS:
N050(IF,#(#69),140,140,60)              ;    IF ANGLE >0°:
N060(#67=(#94-ABS,#64)/(SIN,#(#69)))    ;        FIND A (SEE DERIVATION)
N070(#66=#67-#94)                       ;        FIND B
N080(IF,#66,90,90,100)                  ;        IF B<0
N090(#66=0)                             ;            SET B=0
N100(#65=((#94-ABS,#64)*(#94-ABS,#64))) ;        FIND (R-Z)^2
N110(#(#68)=(SQR,((#67*#67)-#65)-SQR,((#94*#94)-#65)-(#66/(COS,#(#69)))))     ;FIND
                                           THE OFFSET AND STORE IN V7x
N120(IF,#(#68),130,130,140)             ;        IF OFFSET IS NEGATIVE
N130(#(#68)=0)                          ;            SET OFFSET=0
N140(#69=#69+1)                         ;        INCREMENT V68 (GET NEXT OFFSET)
N150(#68=#68+1)                         ;        INCREMENT V69 (GET NEXT ANGLE)
N160(GO,40)                             ;        LOOP BACK TO FIRST IF
N170(#45=#45+#70+#71)                   ;ADD X OFFSETS TO X FRAME DIMENSION[20]
N180(#46=#46+#72+#73)                   ;ADD Y OFFSETS TO Y FRAME DIMENSION
N190(#23=#23-#70)                       ;SUBTRACT LEFT X OFFSET FROM X ST PT.
N200(#24=#24-#72)                       ;SUBTRACT LWR Y OFFSET FROM Y ST PT.
N210(#99=0)                             ;CLEAR THRESHOLD FLAG
N220(GO,240)                            ;GO TO END
N230(#99=1)                             ;(BRANCH DESTINATION) SET THRESHOLD FLAG
                                        IF DONE WITH SURFACE OFFSETS
N240(9)M30$                             ;END MACRO
```

[20] Our reasoning for the addition and subtraction of offsets is illustrated in the section that covers BALLOFFS.

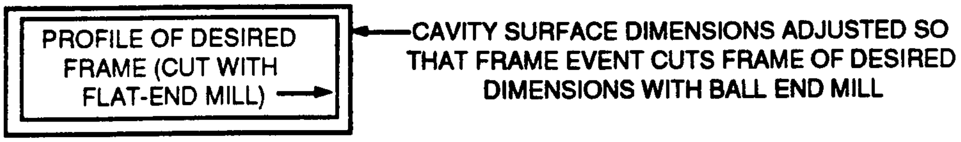
OVERVIEW OF OFFSETS

We have now described and written every submacro we need; all that is left is to finish writing DCMAC and the main (calling) program. Since most of the macros we have written have to do with tool offsetting in one way or another, we will now present an overview of these various offsets and how they relate to the cavity.

1. COMPENSATE THE TOOL:

BALL-END MILL OFFSETS (SUBMACRO BALLOFFS):

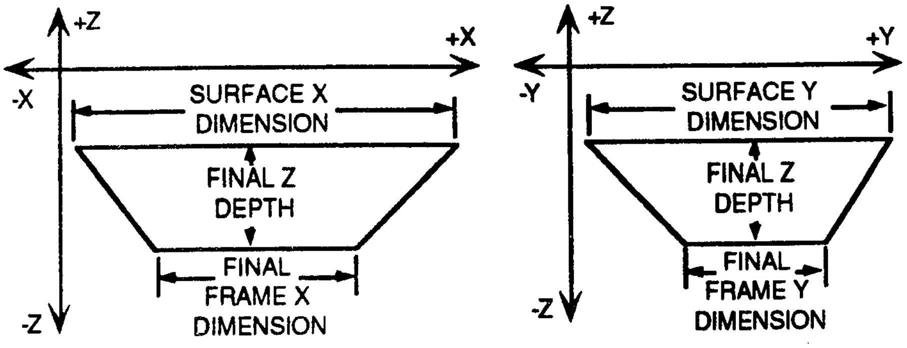
- FINDS OFFSETS TO MAKE BALL-END MILL CUT SAME SIZE FRAME AS FLAT END MILL.
- APPLIED ONCE TO CAVITY SURFACE DIMENSIONS AND START POINT. ALL OTHER FRAME FRAME EVENT ENTRIES ARE COMPUTED FROM THESE ADJUSTEDVALUES.


PROFILE OF DESIRED FRAME (CUT WITH FLAT-END MILL)
CAVITY SURFACE DIMENSIONS ADJUSTED SO THAT FRAME EVENT CUTS FRAME OF DESIRED DIMENSIONS WITH BALL END MILL

2. FIND THE FLOOR DIMENSIONS:

FINAL FRAME DIMENSION DIFFERENCES (SUBMACRO FINFRM):

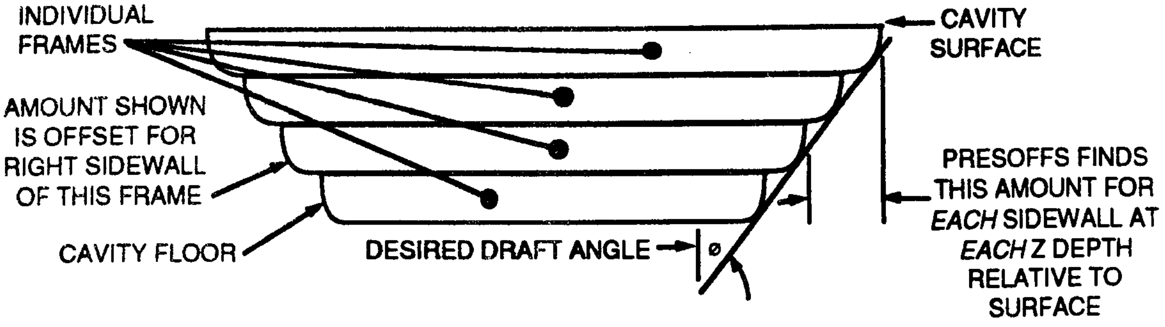
- COMPUTED EARLY IN DCMAC FOR ERROR CHECKING
- USED WHEN MILLING FINAL FRAME WITH FLAT-END MILL



3. FIND THE PER-PASS OFFSETS:

PRESENT Z LOCATION OFFSETS (SUBMACRO PRESOFFS):

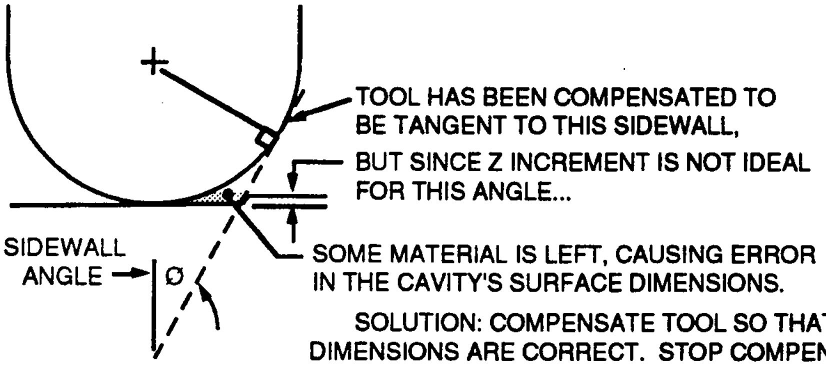
- COMPUTED FOR EVERY FRAME AT EACH SUCCESSIVE Z DEPTH
- TELL HOW MUCH TO RESIZE AND RELOCATE EACH FRAME SO THAT RESULTING CAVITY HAS ANGLED SIDEWALLS
- COMPUTED RELATIVE TO SURFACE DIMENSIONS FOR ACCURACY


INDIVIDUAL FRAMES
CAVITY SURFACE
AMOUNT SHOWN IS OFFSET FOR RIGHT SIDEWALL OF THIS FRAME
PRESOFFS FINDS THIS AMOUNT FOR EACH SIDEWALL AT EACH Z DEPTH RELATIVE TO SURFACE
CAVITY FLOOR
DESIRED DRAFT ANGLE
ø

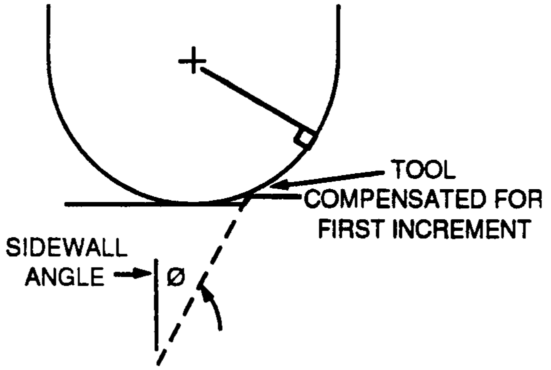
4. FIND THE STOCK SURFACE OFFSETS:

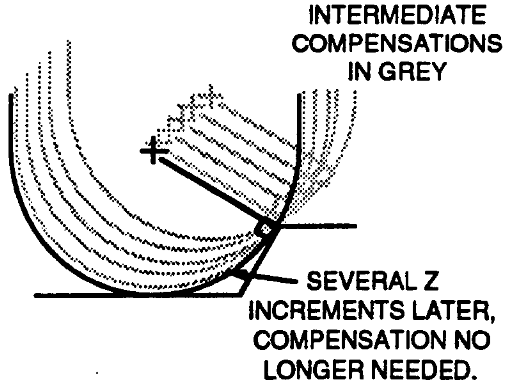
STOCK SURFACE OFFSETS (SUBMACRO TOPOFFS):

- COMPUTED FOR ANY SIDEWALL THAT CUTTER IS NOT TANGENT TO.
- WHEN CUTTER BECOMES TANGENT TO A SIDEWALL, THIS OFFSET GOES TO 0 FOR THAT SIDEWALL.
- WHEN CUTTER IS TANGENT TO SHALLOWEST-ANGLED SIDEWALL, ALL STOCK SURFACE OFFSETS=0.


TOOL HAS BEEN COMPENSATED TO BE TANGENT TO THIS SIDEWALL,
BUT SINCE Z INCREMENT IS NOT IDEAL FOR THIS ANGLE...
SOME MATERIAL IS LEFT, CAUSING ERROR IN THE CAVITY'S SURFACE DIMENSIONS.
SIDEWALL ANGLE
Ø

SOLUTION: COMPENSATE TOOL SO THAT SURFACE DIMENSIONS ARE CORRECT. STOP COMPENSATING WHEN TOOL TANGENT POINT IS REACHES SURFACE.




INTERMEDIATE COMPENSATIONS IN GREY
SEVERAL Z INCREMENTS LATER, COMPENSATION NO LONGER NEEDED.

Our final task before proceeding to Step 6 and synthesizing all this into one big macro (DCMAC) is to expand our preliminary algorithm for DCMAC (from Step 2) into a detailed outline we will be able to use to write the code.

As a starting point, we look once again at our Step 2 algorithm:

```
MAIN PROGRAM calls
    DCMAC which calls
        USERR;
    then calls
        FINFRM;
    then calls
        LOCERR;
    then calls
        ZINC;
    then calls
        BALOFFS;
    While Z depth is > bottom of cavity:
        calls
            PRESOFFS;
        Calls Frame event to mill Frame
        Increments Z depth
    When bottom of cavity is reached, change to a flat-end mill and finish cavity
    floor[21].
    END DCMAC;
END main.
```

This preliminary algorithm skipped calling the submacros TOPOFFS and FLATOFFS because we did not foresee a need for them at the time we wrote the algorithm. Our expanded algorithm will include calls to these two new submacros, and will also include tool selecting and positioning details.

```
MAIN PROGRAM will call
    DCMAC which will call
        USERR to check for errors in global variables;
    turn on spindle, set speed to user specified speed
    if flat-end mill tool number is programmed, activate flat-end mill[22];
    if user-programmed direction of cut for draft cavity is CW then
        set tool offset variable (V27) to "on right" (C2) for use in Frame event call;
    else if direction is CCW then
        set offset variable to "on left" (C1);
    call
        FINFRM to find the final frame dimensions and, if present, the flat-end mill
        radius;
    call
        LOCERR to check for errors in local variables;
    activate ball-end mill[23];
    call
        ZINC to find ball-end mill radius and the Z increment we will use;
    call
        BALLOFFS to find the ball-end mill offsets that will make it work in the
        Frame event;
    Z depth=Z position of stock surface-Z increment returned from ZINC;
    clear Z threshold flag (initially assume that TOPOFFS is necessary);
    while current Z depth > final Z depth do:
        while Z threshold flag is clear (threshold not yet reached) do:
```

[21] The final ball end mill pass will leave "grooves" in the cavity floor (due to the shape of the mill); this finish pass with a flat end mill will smooth the cavity floor. When we expand our algorithm, we will make this pass optional.

[22] FINFRM (the next macro called) needs the flat-end mill active to be able to find its radius.

[23] Now we activate the ball-end mill so that ZINC can find its radius.

```
            call
                TOPOFFS to compute stoek surface offsets and adjust frame
                startpoint and dimensions accordingly;
            position to start;
            call Frame event to mill frame;
            make current Z depth deeper by 1 Z increment;
            if current Z depth is below cavity floor then
                set current Z depth=cavity floor;
                rough final frame with ball-end mill;
        endwhile;
    endwhile;
    set Z depth accumulator=final Z cavity depth;
    call first part of while loop above (through Frame event) to mill final pass
    with ball-end mill at final Z depth {insure that roughing is done to bottom
    of cavity};
    if user-programmed flat-end mill <>T0 then
        activate flat-end mill;
        call
            FLATOFFS to un-offset flat-end mill for finishing cavity floor;
        set Frame event corner radius=flat-end mill radius;
        position for final frame;
        mill final frame;
    halt program;
    END DCMAC;
END MAIN.
```

We will use this algorithm in Step 6 to write our main macro DCMAC.

6. WRITE THE MACRO CODE

Now that we have all of our component submacros and modules written, we can write DCMAC using the expanded algorithm we developed at the end of Step 5 as a guide. Before we do this, however, we should insure that we have assigned all the variables that we need. A flowchart showing all the variables, their use and their relation to one another is shown on the **following** page. This chart will provide a useful reference while we are writing DCMAC.

GLOBAL VARIABLE LETTER/NUMBER **RELATED LOCAL VARIABLES**

A/1 ⇒ V23: X STARTPOINT ACCUMULATOR ⇒ V25: X STARTPOINT FOR FINAL PASS

B/2 ⇒ V24: Y STARTPOINT ACCUMULATOR ⇒ V26: Y STARTPOINT FOR FINAL PASS

D/4 ⇒ V27: FRAME EVENT CUTTER COMP PARAMETER

F/6 ⇒

G/7 ⇒

H/8 ⇒

I/9 ⇒

J/10 ⇒ V30: TOTAL ΔX FOR LEFT CAVITY WALL ⇒ V40: ΔX FOR LEFT CAVITY WALL AT PRESENT Z POSITION ⇒ V50: ΔX CUTTER OFFSET FOR LEFT CAVITY WALL ⇒ V60: LEFT ΔX FLAT END MILL OFFSET FOR FINAL FRAME ⇒ V70: LEFT ΔX STOCK SURFACE OFFSET

K/11 ⇒V31: TOTAL ΔX FOR RIGHT CAVITY WALL ⇒ V41: ΔX FOR RIGHT CAVITY WALL AT PRESENT Z POSITION ⇒ V51: ΔX CUTTER OFFSET FOR RIGHT CAVITY WALL ⇒ V61: RIGHT ΔX FLAT END MILL OFFSET FOR FINAL FRAME ⇒ V71: RIGHT ΔX STOCK SURFACE OFFSET

L/12 ⇒ V32: TOTAL ΔY FOR LOWER CAVITY WALL ⇒ V42 ΔY FOR LOWER CAVITY WALL AT PRESENT Z POSITION ⇒ V52: ΔY CUTTER OFFSET FOR LOWER CAVITY WALL ⇒ V62: LOWER ΔY FLAT END MILL OFFSET FOR FINAL FRAME ⇒ V72: LOWER ΔY STOCK SURFACE OFFSET

M/13 ⇒ V33: TOTAL ΔY FOR UPPER CAVITY WALL ⇒ V43: ΔY FOR UPPER CAVITY WALL AT PRESENT Z POSITION ⇒ V52: ΔY CUTTER OFFSET FOR UPPER CAVITY WALL ⇒ V63: UPPER ΔY FLAT END MILL OFFSET FOR FINAL FRAME ⇒ V73: UPPER ΔY STOCK SURFACE OFFSET

R/14 ⇒ V34: BALL-END MILL RADIUS

S/15 ⇒ V54: Z INCREMENT* ⇒ V64: Z DEPTH ACCUMULATOR

T/16 ⇒ V74: ACTIVE TOOL NUMBER ⇒ V84: DIAMETER OF ACTIVE TOOL ⇒ V94: RADIUS OF ACTIVE TOOL

U/17 ⇒

V/18 ⇒

W/19 ⇒

X/20 ⇒V35: X DIMENSION OF LAST PASS ⇒ V45: X FRAME DIMENSION ACCUMULATOR

Y/21 ⇒V36: Y DIMENSION OF LAST PASS ⇒ V46: Y FRAME DIMENSION ACCUMULATOR

Z/22 ⇒

V65-V67: INTERMEDIATE VARIABLES FOR STOCK SURFACE OFFSETS
V68: HOLDER FOR OFFSET VARIABLE NUMBERS V70-V73
V69: HOLDER FOR ANGEL NUMBER V10-V13
V95: SCRATCHPAD FOR USERR

SORT MACRO VARIABLES
V97-V98: SORT SCRATCHPAD
V99: HOLDER FOR VARIABLE NUMBER OF INCREMENT VALUE**
V54*-V57: UNSORTED/SORTED Z INCREMENTS

*V54 CONTAINS THE IDEAL Z INCREMENT OF

**ALSO USED IN MAIN MACRO AS Z THRESHOLD FLAG

Recall that the first few lines of DCMAC make up the initialization module which we wrote at the beginning of Step 5. This is presented below.

```
CODE                      COMMENTS

(DCMAC  )                 ;MACRO NAME
N10(CLR)                  ;CLEAR FAULT/STATUS DISPLAY (WE BEGIN ALL MAIN MACROS
                          THIS WAY AS A MATTER OF CONVENTION)
N20(M)PUSERR$             ;CALL USERR
N25(IF,#17,35,35,30)      ;IF FLAT-END MILL IS PROGRAMMED THEN
N30(9)M06T#17$            ;ACTIVATE TOOL #17 (USER-PROGRAMMED FLAT-END MILL[24])
N32(GO,40)                ;SKIP NEXT INSTRUCTION
N35(9)M06T#16$            ;IF NO FLAT-END MILL IS PROGRAMMED, ACTIVATE BALL-END
                          MILL INSTEAD
N40(IF,#4,50,50,70)       ;CHECK V4 (D); IF <=0 GO TO 50 ELSE GO TO 70
N50(#27=2)                ;SET V27=2 (C WILL = 2 IN FRAME EVENT) IF CLOCKWISE
N60(GO,80)                ;SKIP NEXT INSTRUCTION
N70(#27=1)                ;SET V27=1 (C WILL = 1 IN FRAME EVENT) IF CCW
```

According to our algorithm, the initialization module is followed by a series of submacro calls to FINFRM, LOCERR, ZINC, and BALOFFS. This segment of code is shown below:

```
CODE                      COMMENTS

N80(M)PFINFRM$            ;CALL FINFRM TO COMPUTE FINAL CAVITY DIMENSIONS
N90(M)PLOCERR$            ;CALL LOCERR TO FINISH ERROR CHECKING
N95(9)M06T#16$            ;ACTIVATE BALL-END MILL SO THAT ZINC CAN FIND ITS RADIUS
N100(M)PZINC$             ;CALL ZINC TO COMPUTE PROPER Z INCREMENT
N110(M)PBALLOFFS$         ;CALL BALLOFFS TO COMPUTE BALL-END MILL OFFSETS
```

Next we turn on the spindle, initialize the Z depth and threshold flag and construct the while loop which will rough out our entire cavity with the ball-end mill, calling TOPOFFS when appropriate to add stock surface offsets:

```
CODE                            COMMENTS

N120(#64=#9-#54)                ;INITIALIZE CURRENT Z DEPTH TO 1 Z INCREMENT;
N130(#99=0)                     ;CLEAR Z THRESHOLD FLAG;
N135(9)M03S#7$                  ;TURN ON SPINDLE AND SET SPEED;
                                ;WHILE CURRENT Z DEPTH > FINAL Z DEPTH[25]:
N140(IF,#99,150,150,170)        ;    WHILE Z THRESHOLD FLAG IS CLEAR:
N150(M)PTOPOFFS$                ;        COMPUTE AND APPLY STOCK SURFACE
                                         OFFSETS;
N160(GO,180)                    ;    ENDWHILE;
N170(M)PPRESOFFS$               ;    COMPUTE PRESENT Z POSITION OFFSETS;
N180(0)X#23Y#24Z#19$            ;    POSITION TO START;
N190(3)R#14G7X#45Y#46Z#64W#19D#4C#27F#6P#18$    ;MILL FRAME;
N200(#64=#64-#54)               ;    DECREASE POSITION IN Z BY 1 INCREMENT;
N210(IF,(#22-#64),140,220,220)  ;ENDWHILE;
```

At this point we have to mill a final pass with the ball-end mill at the final Z depth (it is quite likely that the Z increment we used did not divide the total Z depth evenly, in which case the last pass we milled in the module above was not actually at the final Z depth). We will therefore set

[24] If you have a tool changer and Tool Life Management, you will need to use an M06 to select the tool. If Tool Life Management is not present you may use M03 to activate the spindle.

[25] This while loop is controlled by the IF statement at N210.

the Z depth accumulator (#64) equal to the user-programmed final Z depth (#22) and call the sequence of events at N140-N190 as a subroutine to mill the final pass (this will also take care of the unusual case where the cavity is done before we pass the Z threshold). Next, if the user has programmed a value for U (the flat-end mill tool number) we must finish the cavity floor with the specified flat-end mill; if not, we can quit the macro.

Here, then, is our final module:

```
CODE                          COMMENTS

N220(#64=#22)                 ;Z DEPTH ACCUMULATOR = FINAL Z DEPTH;
N0230(7)F140L190$             ;CALL SUBROUTINE TO MILL FINAL PASS WITH BALL-END
                              MILL;
N240(IF,#17,340,340,250)      ;IF USER HAS PROGRAMED A FLAT-END MILL THEN:
N250(9)M6T#17$                ;    ACTIVATE FLAT-END MILL;
N260(M)PFLATOFFS$             ;    UN-OFFSET FLAT-END MILL;
N280(9)M3S#8$                 ;    TURN ON SPINDLE AND SET SPEED;
N290(#14=#94)                 ;    SET CORNER RADIUS=MILL RADIUS;
N300(0)X#25Y#26Z#19$          ;    POSITION TO START;
N310(3)R#14G7X#35Y#36Z#22W#19D#4C#27F#6P#18$      ;MILL FRAME;
N340(9)M30$                   ;END MACRO.
```

Now we present the complete program:

```
CODE                          COMMENTS

(DCMAC  )                     ;MACRO NAME
N10(CLR)                      ;CLEAR FAULT/STATUS DISPLAY (WE BEGIN ALL MAIN
                              MACROS THIS WAY AS A MATTER OF CONVENTION)
N20(M)PUSERR$                 ;CALL USERR
N25(IF,#17,35,35,30)          ;IF FLAT-END MILL IS PROGRAMMED THEN
N30(9)M06T#17$                ;ACTIVATE TOOL #17 (USER-PROGRAMMED FLAT-END
                              MILL[26])
N32(GO,40)                    ;SKIP NEXT INSTRUCTION
N35(9)M06T#16$                ;IF NO FLAT-END MILL IS PROGRAMMED, ACTIVATE
                              BALL-END MILL INSTEAD
N40(IF,#4,50,50,70)           ;CHECK V4 (D); IF <=0 GO TO 50 ELSE GO TO 70
N50(#27=2)                    ;SET V27=2 (C WILL = 2 IN FRAME EVENT) IF CLOCKWISE
N60(GO,80)                    ;SKIP NEXT INSTRUCTION
N70(#27=1)                    ;SET V27=1 (C WILL = 1 IN FRAME EVENT) IF CCW
N80(M)PFINFRM$                ;CALL FINFRM TO COMPUTE FINAL CAVITY DIMENSIONS
N90(M)PLOCERR$                ;CALL LOCERR TO FINISH ERROR CHECKING
N95(9)M06T#16$                ;ACTIVATE BALL-END MILL SO THAT ZINC CAN FIND ITS
                              RADIUS
N100(M)PZINC$                 ;CALL ZINC TO COMPUTE PROPER Z INCREMENT
N110(M)PBALLOFFS$             ;CALL BALLOFFS TO COMPUTE BALL-END MILL OFFSETS
N120(#64=#9-#54)              ;INITIALIZE CURRENT Z DEPTH TO 1 Z INCREMENT;
N130(#99=0)                   ;CLEAR Z THRESHOLD FLAG;
N135(9)M03S#7$                ;TURN ON SPINDLE AND SET SPEED;
                              ;WHILE CURRENT Z DEPTH > FINAL Z DEPTH[27]:
N140(IF,#99,150,150,170)      ;    WHILE Z THRESHOLD FLAG IS CLEAR:
N150(M)PTOPOFFS$              ;        COMPUTE AND APPLY STOCK SURFACE
                                       OFFSETS;
```

[26] If you have a tool changer and Tool Life Management, you will need to use an M06 to select the tool. If Tool Life Management is not present you may use M03 to activate the spindle.

[27] This while loop is controlled by the IF statement at N210.

```
N160(GO,180)               ;    ENDWHILE;
N170(M)PPRESOFFS$          ;    COMPUTE PRESENT Z POSITION OFFSETS;
N180(0)X#23Y#24Z#19$       ;    POSITION TO START;
N190(3)R#14G7X#45Y#46Z#64W#19D#4C#27F#6P#18$     ;MILL FRAME;
N200(#64=#64-#54)          ;    DECREASE POSITION IN Z BY 1 INCREMENT;
N210(IF,(#22-#64),140,220,220) ;ENDWHILE;
N220(#64=#22)              ;Z DEPTH ACCUMULATOR = FINAL Z DEPTH;
N0230(7)F140L190$          ;CALL SUBROUTINE TO MILL FINAL PASS WITH BALL-END
                           MILL;
N240(IF,#17,340,340,250)   ;IF USER HAS PROGRAMED A FLAT-END MILL THEN:
N250(9)M6T#17$             ;    ACTIVATE FLAT-END MILL;
N260(M)PFLATOFFS$          ;    UN-OFFSET FLAT-END MILL;
N280(9)M3S#8$              ;    TURN ON SPINDLE AND SET SPEED;
N290(#14=#94)              ;    SET CORNER RADIUS=MILL RADIUS;
N300(0)X#25Y#26Z#19$       ;    POSITION TO START;
N310(3)R#14G7X#35Y#36Z#22W#19D#4C#27F#6P#18$     ;MILL FRAME;
N340(9)M30$                ;END MACRO.
```

FOUR-DRAFTED RECTANGULAR CAVITY MAIN PROGRAM

Now that we have written our Draft Angle Cavity Macro, we need a master program to call it. The macro can be called from any part program or even from another macro; below is a simple two-line program that calls the macro to cut one cavity. It is formatted and commented as were the macros.

```
CODE                             COMMENTS

(DC)                             ;PROGRAM NAME
N10(M)A0B0D0F12G2000H2200        ;MACRO CALLING EVENT
    I0J6.5K3L10M0R.5S.1T1U2
    V5W.1X8Y7Z-1.2PDCMAC$
N20(9)M30$                       ;END OF PROGRAM
```

This program will use DCMAC to mill a draft angle cavity. The parameters specify that the cavity will have its lower left corner located at X0, Y0; it will be cut in the clockwise direction with an XY feedrate of 12 IPM; the ball-end mill will use a spindle speed of 2000 RPM; the flat-end mill will use 2200 RPM; the surface of the stock is located at Z=0; the left side draft angle is 6.5°, the right side draft angle is 3°, the bottom draft angle is 10° and the top draft angle is 0° (straight wall); the corner radius will be .5; the Z increment limit S is .1; Tool 1 is the ball-end mill and Tool 2 is the flat-end mill; the plunge feedrate is 5 IPM; the reference plane is located at Z=.1; the top-of-stock cavity dimensions are X=8 and Y=7 and the Z depth is -1.2.

This macro could be used to cut a cavity that was not parallel to the X axis by executing a Rotate event immediately before calling it just like the rotated frame example in the Frame event section of the Delta Customer Manual.

TESTING AND DEBUGGING

Macros need to be tested and debugged like any programs. Here are some guidelines to follow when testing and debugging.

TESTING

Testing concerns cycling the macro to observe its performance and to see if it can be made to flag a fault. Once a fault is flagged (unless it is a fault that is tested for within the macro and the fault conditions are valid) you must proceed to the debugging stage to isolate and correct the fault.

- Run an initial test in Auto or Graphics mode. If you don't check for a certain feedrate range within the macro, program the feedrate unrealistically high (i.e. 9999 IPM) to facilitate faster Auto mode execution. For the first run, *program all global variables within their specified range.*
- As with any other program, monitor the graphics and position displays to be sure the part is properly positioned.
- If no faults are flagged in the initial run and you have included some global variable range-checking within your macro, purposely enter out-of-range values for the variables you check to see if your range checking logic is functional.
- Due to the high number of statements that cause no axis motion in a macro, Graphics mode time estimates are generally less reliable than Auto mode run times. If you need a close estimate of the actual machining time for the part, use the Auto mode time.

DEBUGGING

Macros are more difficult to debug than ordinary part programs because they generally contain branches and other events that generate no axis motion.

- The first step to debugging any macro is understanding how it works. Often just tracing through a listing of the macro by hand will point out any problems that would occur during execution.

- The N number displayed when a fault occurs during a Graphics, Auto or Single Event mode run of the macro may not be anywhere near the code that generated the error. Branching is often to blame in cases like this. If there is nothing wrong with the N number which is flagged or the one immediately after it, suspect branching. Is the N number where the fault was flagged (or the one immediately following it) the target of a branch higher up in the macro? If so, look at the branching logic carefully; an erroneous branch may be the cause of your error. If the flagged N number (or the one immediately following it) is itself a branch statement, check that its destination(s) is/are valid. A bug buried deep within nested IF statements or even within a series of IF statements can be tricky to find.

- Watch for divide by 0 and illegal use of library functions (such as taking the square root of a negative number) when debugging macros. You can avoid these types of errors by testing variables before using them in such functions or in division and flagging the error yourself.

EIA/ISO PROGRAMMING

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON

EIA/ISO INTRODUCTION

Part 7 contains detailed descriptions of the EIA/ISO[1] Primitives used in this control and how they can be expected to function.

In addition to the "operator friendly" Conversational programming format provided with this control, the conventional EIA/ISO interchangeable Variable Block Data Format for CNC Programming is also provided.

The control can accept programs input with RS-358 (ASCII) data (with or without parity). Programs can be written and edited entirely in the EIA/ISO format or Conversational format. EIA/ISO and Conversational may also appear within a single part program. The control will examine each event (block) and determine if it is Conversational or EIA/ISO in nature and process the event accordingly. A single event (block), however, may not contain both Conversational and EIA/ISO formatted data.

The Transmit (output) mode or the Receive (input) mode will transfer data formatted by the operator as ASCII characters with or without parity, Parity selection must be made prior to initiating the input or output by selecting P0 or P1 from the Transmit mode or the Receive mode. The first time power is applied to the control, parity will be ON (P1 will be selected). Once an operator changes the parity to OFF (P0), it will remain OFF until the operator changes it back to ON (P1) or a total power fail (battery failure) occurs. PARITY OFF (P0) will ignore the parity bit on input and force the parity bit to zero on output.

M02 and M30 codes are not recognized as end of load characters. This enables the control to load a series of part programs with a single receive command. The control concludes a receive operation when the letter E is detected, or the operator presses the ABORT RECEIVE soft key. If a series of part programs is to be sent to the control as a single operation, each intermediate part part program must be terminated with the letter C and the last part program must be terminated with the letter E.

Part IDs should be placed at the beginning of each part program and will be loaded as indicated. The part ID must be enclosed inside of parentheses. Spaces are permitted within the parentheses. Any part received without an ID will be assigned the name P0000000.

EIA/ISO events (blocks) may be entered in Program mode by selecting EVENT TYPE E. The EIA event (event type E) program screen displays the Conversational event types in the left column and lists the available EIA/ISO entries in the right column.

Press the EIA/ISO ASSIST key to refer to EIA/ISO screen prompts. The LEFT and RIGHT ARROW keys are used to move back and forth between the EIA/ISO ASSIST pages. Page 1 lists the G Codes for preparatory functions. Page 2 contains references prepared by your machine supplier.

IMPORTANT NOTE ABOUT G CODES

The function that is performed by each G Code is determined by your machine supplier. For example, most machine builders use G70/G71 to switch between Inch an Metric modes; other machine builders will configure the control to use G20/G21 to perform the same operation. The G code functions described in this section refer to the "normal" set of Autocon G codes. Your control will have a different set of G Codes when an asterisk (*) appears before the SPECIAL G CODE item on the control's SYSTEM SETTINGS screen. Consult your machine supplier or refer to Part 1-Section 2 for more information.

[1] EIA/ISO refers to EIA RS-274D and ISO 1075 G Code format standards.

SUMMARY OF INTERCHANGEABLE VARIABLE BLOCK DATA

Preparatory Functions:.... G00, G01, G02, G03, G04, G08, G09, G10, G11, G17, G18, G19, G21, G22, G23, G27, G28, G29, G40, G41, G42, G53, G70, G71, G73, G76, G80, G81, G82, G83, G84, G85, G87, G88, G89, G90, G91, G92, G93, G94, G99, (G75-optional feature).

Miscellaneous Functions:.. M00, M01, M02, M03, M04, M05, M06, M30 See descriptions.

Feed Function range: ..... 0.1 to 9999.999 IPM[2] or (.006 min - 10 min with G93)

S Function Range: .... 1 to 9999 RPM, Some systems permit 1 to 65,535.

T Function Range: .... Tool Number. Also, if control does not use H and D codes, designates tool length and diameter offsets. Otherwise following H and D codes will designate tool offsets. Consult machine builder or refer to Part 1 - Section 2.

H Function Range:....... Tool Length Offset Hxx or Hxxx (used only if control does not use H and D codes. Refer to Part 1-Section 2).

D Function Range:....... Tool Diameter Offset Dxx or Dxxx (used only if control does not use H and D codes. Refer to Part 1-Section 2).

.... D is also used to load a G10 T Table diameter when control does not use H and D codes (refer to Part 1-Section 2).

E Fixture Offset:......... E01 to E32 from offset table. E00 cancels offset.

X,Y,Z: .... Programmed end points for the XYZ axes.

A,B,C,U,V,W: .... Programmed end points for optional auxiliary axes.

I,J,K:.................. Arc centers for Circular Interpolation.

R: .... 1st Z Axis Reference Plane in Fixed Cycles.

O:..................... 2nd Z Axis Reference Plane in Fixed Cycles.

L: .... Dwell Time in Fixed Cycles.

K:..................... Peck Increment in the Peck Drill Cycle.

Q:..................... Peck clearance, G73 or G83 cycles. Default is to 0.1".

X:..................... G04 Dwell function. The G04 X dwell format is 3.1.

Decimal Point Format: ... Control assumes decimal point programming. All values without a decimal point will be treated as integers. No other formats will be recognized.

[2] G75 Rigid Tapping option only: the modal feedrate range is temporarily stored. The G75 feedrate F specifies the thread pitch or lead for the Rigid Tap cycle.

INTERCHANGEABLE VARIABLE BLOCK DATA

ASCII RS-358	FUNCTION
A	Auxiliary Axis (per machine supplier)
B	Auxiliary Axis (per machine supplier)
C	Auxiliary Axis (per machine supplier)
D	Tool diameter offset[3]
E.	Fixture Offset
F.	Feedrate (Inverse Time with G93 active)
G	Preparatory Function
H	Tool length offset[3]
I	X Arc Center for Circular Interpolation
J	Y Arc Center for Circular Interpolation
K	Z Arc Center for Circular Interpolation or peck increment, peck drill cycle
Q	Peck clearance, peck drill cycle
M	Miscellaneous Function
N	Sequence Number
O	2nd Z Reference Plane Dimension for Fixed Cycles
R	1st Z Reference Plane Dimension for Fixed Cycles
S.	Spindle Speed
T.	Tool Function
U	Auxiliary Axis (per machine supplier)
V	Auxiliary Axis (per machine supplier)
W	Auxiliary Axis (per machine supplier)
X	Primary X Motion Dimension, G04-dwell time
Y	Primary Y Motion Dimension
Z.	Primary Z Motion Dimension
Line Feed	End of block for EIA/ISO data
Carriage Return	Ignored by Control
$	End of block for Conversational Data
(	Open Program or Open Event Type Identifier
)	Close Program or Close Event Type Identifier
-	Minus Sign

[3] This entry may not be allowed on all controls. Refer to Part 1-Section 1.

EIA/ISO EVENT FORMAT and OFF-LINE PROGRAMMING NOTATIONS

Notice that a Line Feed is the end of block character for EIA/ISO type data (EIA/ISO events) and a $ is the end of block character for Conversational data. If a $ appears in an EIA/ISO event, it will be ignored. When an EIA/ISO Event is transmitted (output), each block will be terminated with a Carriage Return and Line Feed. When a Conversational event is transmitted, (output) each block will be terminated with a $, Line Feed and CR.

Tape Format:	English	Metric
N	4.3	4.3
G	2	2
XYZ IJK R	±3.4	±4.3
ABCUVW[4] (linear axis)	±3.4	±4.3
ABCUVW[4] (rotary axis)	±3.3	±3.3
F	4.3	5.2
S	4 or 5	4 or 5
T	2 through 6	2 through 6
L	3.1	3.1
MHD	2 or 3	2 or 3
E	2	2
Q	.4	.3

SEQUENCE OR EVENT NUMBERS (N)

Each EIA/ISO block or event must be identified by programming a sequence number (the letter N). N numbers may range from 0 to 9999.999.

PREPARATORY FUNCTION (G)

When using the EIA/ISO Format, the letter G followed by two digits will determine the function to execute. One G Code may be programmed per EIA/ISO event. The following table shows the modal groupings of the various G Codes. G00/G01, G70/G71, and G90/G91 modal G Code groups have a default condition that is selected by the machine supplier. Pressing RESET or executing an M02 or M30 command will reset each G Code group to its default value.

G Code	Modal/Non-Modal	Conversational Interactive	Turn On
G00,G01	Modal Interpolation Type	no	Parameter
G00-G03,G21-G23	Modal Interpolation Type	no	see above
G04	Non-Modal Dwell	no	----
G08, G09	Modal Acc/Dec	yes	G09
G10, G11	Modal Offset Setting	no	G10
G17,G18,G19	Modal Plane Select	no	G17
G27,G28,G29,G53	Non-Modal Position to machine coord.	no	----
G40,G41,G42	Modal Cutter Comp Direction	yes	G40
G70,G71	Modal Inch/Metric	yes	Parameter
G73-G89	Modal Fixed Cycle	yes	G80
G90,G91	Modal Abs/Inc	no	Parameter
G92,G99	Non-Modal Preset Position	yes	----
G93	Non-Modal Inverse Time Feedrate	no	----
G94	Modal Per Minute Feedrate	yes	G94

[4] ABCUVW represent auxiliary axes that may be present on your control. The letter that is used to program an auxiliary axis is selected by your machine supplier. Notice that the format for these axes is listed twice since they may be defined as linear or rotary.

G00 - RAPID TRAVERSE

G00 commands the control to move all axes at the rapid traverse rate set by the machine supplier. The Z axis component of any rapid move occurs independent of other axis motions.[5] The sign of the Z endpoint dictates the motion's sequence. When the Z axis is to move in the positive direction, (away from the workpiece) the Z move occurs first. Once the Z axis is in position, all other axes will move. When the Z axis is to move in the negative direction, (toward the workpiece) the Z motion will occur after all other axes are in position.
This function is modal and cancels G01, G02, G03, G21, G22, G23. The last programmed feedrate will remain modal when G00 executes, but will not be operative until next block that designates motion at feed executes.

G01 - LINEAR INTERPOLATION

G01 causes the motion of all axes to be interpreted at a programmed feedrate (F) from their start position to the programmed end point. This results in a straight linear motion.

G01 is modal and cancels G00, G02, G03, G21, G22, G23.

G02 - CLOCKWISE CIRCULAR INTERPOLATION

G02 defines a clockwise circular arc from the start position to the programmed endpoint. The arc endpoint is defined with X and Y (or Z). The arc center is defined with I and J (or K). The feedrate is determined by the modal F entry.

If G91 is active, I and J (or K) are the signed incremental distances from the arc's start point to the arc's center. If G90 is active I and J (or K) are the absolute arc center coordinates.

G02 may designate up to 360° arc motion.

If an XY circular arc is programmed with a Z dimension, the control will move the X, Y, and Z axes in a helical motion. Helical interpolation will only occur if the XY plane is selected for circular motion (Z must be the linear axis).

G02 is modal and cancels G00, G01, G03, G21, G22, G23. The values for I, J or K are not modal and must be programmed in each EIA/ISO circular block.

G03 - COUNTERCLOCKWISE CIRCULAR INTERPOLATION

This function is identical to G02 except the arc direction is counterclockwise.

G03 is modal and cancels G00, G01, G02, G21, G22, G23.

G04 - DWELL

G04 initiates a dwell (or time delay) to occur in the EIA/ISO event containing the G04 command. The dwell time is described by the address letter for X. The dwell time can range from 0.1 to 9999.9 seconds.

An EIA/ISO event containing the G04 dwell command.cannot contain any other information such as axis motion, M Codes, S Codes, etc. G04 is nonmodal and effects only the event in which it is programmed.

[5] If desired, your machine supplier can set your control so that all axes (including Z) position simultaneously.

G08 - ACC/DEC ON for CONTOURING MOVES

Acc/Dec is a feature that must be parametrically enabled by your machine supplier. Acc/Dec regulates the rate at which the machine's axes accelerate and decelerate. This regulation will reduce the strain on the servo drives by cushioning starts and stops.

Do not use G08 and G09 when Acc/Dec has not been enabled by your supplier.

When Acc/Dec has been enabled by the machine supplier, it will always control moves commanded by G00 blocks, Conversational Position events, and manual jogs at the Rapid jog rate.

When G08 is active, Acc/Dec control is extended to G01, G02, G03, G21, G22, G23 blocks and Conversational Linear and Arc Mill events. G08 is canceled by G09, M02, M30, or by pressing RESET when the control is not in cycle.

G09 - ACC/DEC OFF for CONTOURING MOVES

G09 cancels a modal G08 Acc/Dec for linear and arc feeds. However, if your supplier has parametrically enabled Acc/Dec, it will continue to be operative within G00 blocks, Conversational Position events, and manual jogs at the Rapid jog rate.

G10 - TOOL/FIXTURE OFFSET SETTING

The G10 code lets you load Tool Offset or Fixture Offset information directly into the Tool Tables via part program commands. The G10 block (or any block that executes while G10 is active) must include a select code (e.g. E01, H04, or D04) followed by the data that will be stored in the specified select code. The example code below shows typical blocks that will load tool offset and fixture offset information into the Tool Tables. The control for this example uses H and D codes (refer to Part 1-Section 2).

```
N10 G10 H04 Z-2.5         - make G10 modal, store length at H04
N11 D04 X.5001            - store diameter at D04
N12 H05 Z-4.178           - store length at H05
N13 D05 X.3875            - store diameter at D05
N14 E01 X2 Y2 Z-2.125     - store fixture offset coordinates at E01
N15 E02 X5 Y8 Z-2.125     - store fixture offset coordinates at E02
N16 G11                   - cancel modal G10
```

The G10 code remains modal until it is canceled by a G11 block. Execution of any other G code while G10 is active will generate an error.

Only one select code (e.g. E01, H04, or D04) may appear in a single block. Any information that is not directly related to the programmed select code will be ignored by the control. Execution of M02, M30 or any Conversational programming event (any event other than event type E) will cause the control to cancel G10.

G11 - CANCEL TOOL/FIXTURE OFFSET SETTING

G11 cancels a modal G10 code. However, any select code and offset information that is present in a G11 block will be loaded into the Tool Tables – just as if it were programmed in a G10 block. For this reason, we usually recommend that the G code is the only information programmed in a G11 block.

G17, G18, G19 - PLANE SELECT

These functions select the plane for circular interpolation and cutter radius compensation. Clockwise and counterclockwise direction is determined by looking toward the selected plane from the positive axis normal to the plane.

G17 selects the XY plane. G17 is modal and cancels G18 and G19.
G18 selects the ZX plane. G18 is modal and cancels G17 and G19.
G19 selects the YZ plane. G19 is modal and cancels G17 and G18.

G21, G22, G23 - INTERPOLATION WITH IN POSITION

Similar to G01 - G03 with the exception that the next block will not begin execution until all axes have reached a programmed end point within the specified range. See the In Position topic in the Linear and Arc Mill event sections for additional details.

G21, G22, or G23 is modal and cancels another G Code in the G00-G03 or G21-G23 group.

G27 - POSITION TO MACHINE COORDINATES

This command is identical to G53.

G28 - RETURN TO REFERENCE POINT

The G28 command causes each programmed axis to rapid to its reference zero position after it first rapids to its programmed intermediate point. The intermediate point is programmed in part coordinates and may be defined with absolute or incremental coordinates. All offsets (such as tool length offset and fixture offsets) remain in effect during the move to the intermediate point. The offsets are temporarily canceled during the second portion of the move to ensure that the spindle nose moves to the reference zero position. The offsets will be reinstated upon execution of the first part coordinate block that follows the G28.

Since some machine suppliers may select a reference zero coordinate other than X0 Y0 Z0, it is important to note that the coordinates displayed at the end of a G28 block will be the reference zero coordinates selected by your machine builder.

You can command all axes to return directly to the reference position without moving through an intermediate point, by programming the tool's start position in the G28 block. This has the effect of locating the intermediate point at the tool's start position.

An error message will be generated if a G28 block executes while cutter compensation is active.

G29 - RETURN FROM REFERENCE POINT

The G29 code may only be programmed after a G28 block defines an intermediate point and moves the programmed axes to the machine's reference zero position. This code generates a two step move. First, the tool will rapid each programmed axis to its intermediate point (the intermediate point was defined in the previous G28 block). All offsets that were temporarily canceled during the G28 block will be reinstated during this move if the G29 block is the first motion block following the G28. Second, the axes will rapid to the part program coordinates that are programmed in the G29 block. The endpoint may be programmed with absolute or incremental coordinates; incremental coordinates are dimensioned from the intermediate point.

G40 - CANCEL CUTTER RADIUS COMPENSATION

G40 cancels cutter radius compensation. An EIA/ISO event containing a G40 command will move the center of the tool to the programmed path.

G40 is modal and will cancel G41 or G42.

This function is equivalent to programming a C0 in the Conversational Events.

G41 - CUTTER RADIUS COMPENSATION LEFT

This function will offset the center of the tool perpendicular and to the left of the programmed path. The amount of compensation will be one-half the diameter specified in active tool's Tool Table entry. In order to initiate cutter radius compensation a T Code (T01 to T99) must be active and the corresponding diameter D must be some number other than zero.

The radius offset will occur only in the selected plane (G17, G18, G19).

Inside corners will be linearly joined and outside corners will be circularly joined. The part programmer is only required to program the part surface and the control will compensate the tool accordingly. G41 is modal and will cancel G40 and G42. Refer to the Cutter Compensation Appendix.

This function is equivalent to programming a C1 in the Conversational Events.

G42 - CUTTER RADIUS COMPENSATION RIGHT

This function is similar to left cutter radius compensation (G41) except that the center of the tool is offset perpendicular and to the right of the programmed path.

G42 is modal and cancels G40 and G41.

This function is equivalent to programming a C2 in the Conventional Events.

G53 - POSITION TO MACHINE COORDINATES

A G53 block commands the control to rapid the spindle nose to a position that is programmed in machine coordinates. All active offsets (such as tool length offset and fixture offsets) will be temporarily canceled during this move to ensure that the machine moves to the correct position. These offsets will be reinstated upon execution of the first part coordinate block that follows the G53.

G70 - INCH MODE

This function interprets all programmed dimensions as inch values. All dimensions (X, Y, Z, I, J, K, etc.) will be interpreted as inch values. A following feedrate F entry will represent inches/minute.

The execution of a G70 command will also cause Tool Table and Set Up mode values to be converted and displayed in inch format.

G70 is modal and cancels a G71.

This function is identical to selecting INCH in Set Up Mode.

G71 - METRIC MODE

This function interprets all dimensions as metric values. All dimensional values (X, Y, Z, I, J, K, etc.) will be interpreted as millimeter values. A following feedrate F entry will represent mm/min.

The execution of a G71 command will also cause Tool Table and Set Up mode values to be converted and displayed in millimeter format. G71 is modal and cancels G70.

This function is identical to selecting metric in Set Up Mode.

FIXED QUILL CYCLES

Available codes in the range G73 - G89, except G80, command fixed or quill cycles that occur at the end of G00 blocks, or Conversational Position and Bolt Circle events. Quill cycles perform drill, bore, tap, etc., operations.

G Codes in this range are modal. However, the modal quill cycle will only be performed when G00 is active, and the executing block contains an X and/or Y endpoint. When G00 and a quill cycle is active, execution of a block which does not contain an X or Y endpoint, will cause the control to update all of the modal values that are pertinent to the quill cycles, but no tool motion will occur.

The quill cycle will not be performed when a G1, G2, G3, G21, G22, or G23 block executes. However, any following G00 block will cause the modal quill cycle to execute.

Z, R, O, L, F, and K values that program each quill cycle are modal. They only need to be programmed for the first quill cycle in a program, or when one or more of the dimensions change. Modal 2nd reference plane and dwell values are reset on M02, M30.

G80 cancels any quill cycle in this range.

G80 - CANCEL FIXED QUILL CYCLES

G80 is modal and cancels any other fixed quill cycle. Any axis motion that is programmed in a G80 block will occur at the rapid rate. G80 does not cancel the modal G00-G03 setting.

G81 - DRILL CYCLE

This function is programmed with X, Y, Z, R, and O. G81 commands a simple drilling operation as follows:

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feeds to the programmed Z depth.
4) Rapid Z back to the O plane.

G81 is modal and cancels any other fixed cycle.

G82 - COUNTER BORE CYCLE

This function is programmed with X, Y, Z, R, O, and L. G82 commands a counter bore cycle that will dwell at the final Z depth before retracting.

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feed to the programmed Z depth.
4) Dwell at the Z depth for L seconds (0.1 to 999.9 sec).
5) Rapid Z to the 0 plane.

G82 is modal and cancels any other fixed cycle.

G83 - PECK DRILL CYCLE

This function is programmed with X, Y, Z, R, O, K, and Q. G83 is a peck drill cycle that retracts the tool to the reference plane R after every infeed.

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feed Z to the incremental K dimension. Z depth = R-K
4) Rapid Z to the R plane.
5) Rapid Z to the depth increment R-K+Q.
6) Feed Z to R-2K.
7) Rapid Z to the R plane.
8) Continue the Peck Cycle incrementing the depth increment 1K, 2K....NK until the Z depth remaining is less than 1K.
9) Rapid Z to R-NK+Q.
10) Feed to the programmed Z depth.
11) Rapid Z to the O plane.

G83 is modal and cancels any other fixed cycle.

G73 - PECK DRILL WITH CHIP BREAKER CYCLE

This function is programmed with X, Y, Z, R, O, K, and Q. G73 operates similar to G83 as the Z axis will progressively drill deeper into the workpiece, but retracts only Q units to break the chip before drilling the next K depth. Q reverts to the last modal Q, or 0.1" (2.5mm) if not entered.

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feed Z to the incremental K dimension. Z depth = R-K
4) Rapid Z to R-K+Q.
5) Feed Z to R-2K.
6) Rapid Z to R-2K+Q.
7) Continue the Peck Cycle incrementing the depth increment 1K, 2K....NK until the Z depth remaining is less than 1K.
8) Position Z to R-NK+Q.
9) Feed to the programmed Z depth.
10) Rapid Z to the O plane.

G73 is modal and cancels any other fixed cycle. The values for K and R are modal.

G84 - TAPPING CYCLE

This function is programmed with X, Y, R, O, and L. G84 commands a Tapping Cycle.

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feed Z to the programmed Z depth.
4) Output an M05 (spindle stop).
5) Output an M03 or M04. (Reverse spindle direction).
6) Dwell for .400 + L seconds.
7) Feed the Z axis to the R plane.
8) Output an M03 or M04 to reverse the spindle direction back to initial state.
9) Rapid Z to the O plane.

G84 is modal and cancels any other fixed cycle.

G85 - BORE CYCLE

This function is programmed with an X, Y, Z and R, and O. G85 commands a boring cycle. The boring tool will feed to depth and then feed back out of the hole.

1) Rapid X and Y to the hole's center.
2) Rapid Z to the R plane.
3) Feed Z to the programmed Z depth.
4) Feed Z to the R plane.
5) Rapid Z to the O plane.

G85 is modal and cancels any other fixed cycle.

SELF ORIENTING SPINDLE

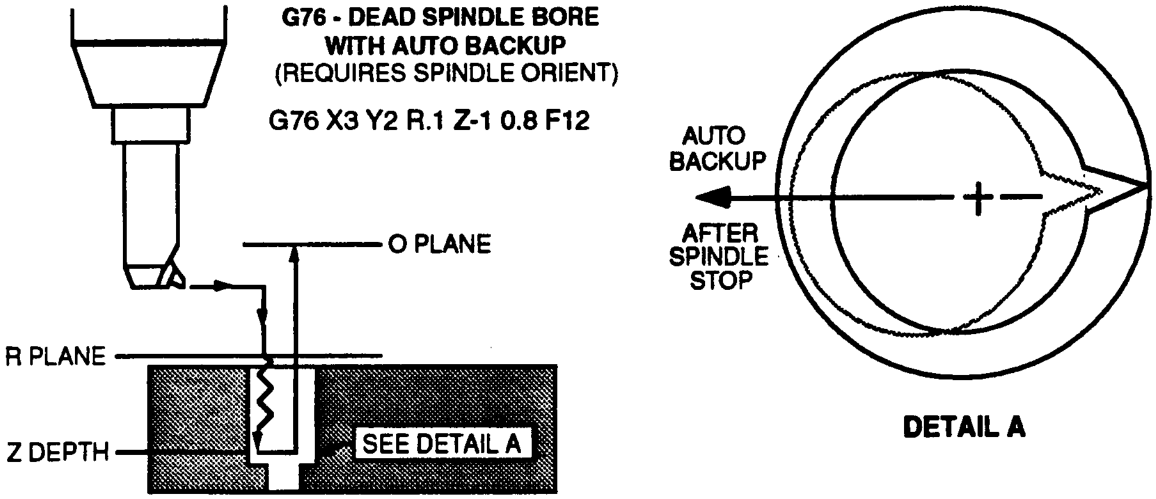
Some of the following cycles require a self-orienting spindle that is compatible with the control. Refer to the machine supplier to determine your machine's spindle orient capability. If provided, a self orienting spindle will have the following characteristics:

1) The spindle must be able to start (M03, M04) and stop (M05) under program control.
2) A precise directional orientation between the boring tool tip and spindle can be maintained. A flat on the boring bar shank, for example, is keyed to the spindle insert.
3) The spindle must orient in precisely the same direction whenever a "spindle orient" M Code executes. This M Code is selected by the machine supplier (usually M19 or M05) and is output by the control on all spindle orient quill cycles. This orientation must align the tool tip along the same axis. The supplier may choose the X axis or Y axis, and whether the tip will point in the + or - direction along the axis.

G76 - DEAD BORE CYCLE (AUTO BACKUP, AUTO RETURN)

This cycle requires a compatible self-orienting spindle. See Self Orienting Spindle topic.

The tool rapids XY over the hole center. Z then rapids to the the R plane, then feeds at F to the programmed depth Z. The control now outputs the spindle orient M Code to halt and orient the spindle. The spindle orients the tool tip, and the tip backs away from the hole circumference (direction and distance assigned by machine supplier). The tool rapids in Z to the R plane, then rapids in XY back to hole center. The spindle restarts with the same direction and speed commands that were operative prior to the start of the cycle. Finally, the tool rapids to the O plane. See illustration **below**.



To determine if your machine is equipped with a self orienting spindle, program the cycle and run it in Graphics mode (without motion), choosing the XZ plane (if tool tip orients in X) or YZ plane (if tool tip orients in Y) axes PLANE SELECT for plotting. Note the starting X (or Y) dimension and ending dimension during the infeed and withdrawal. This will give you the back-up distance, an important hole clearance dimension.

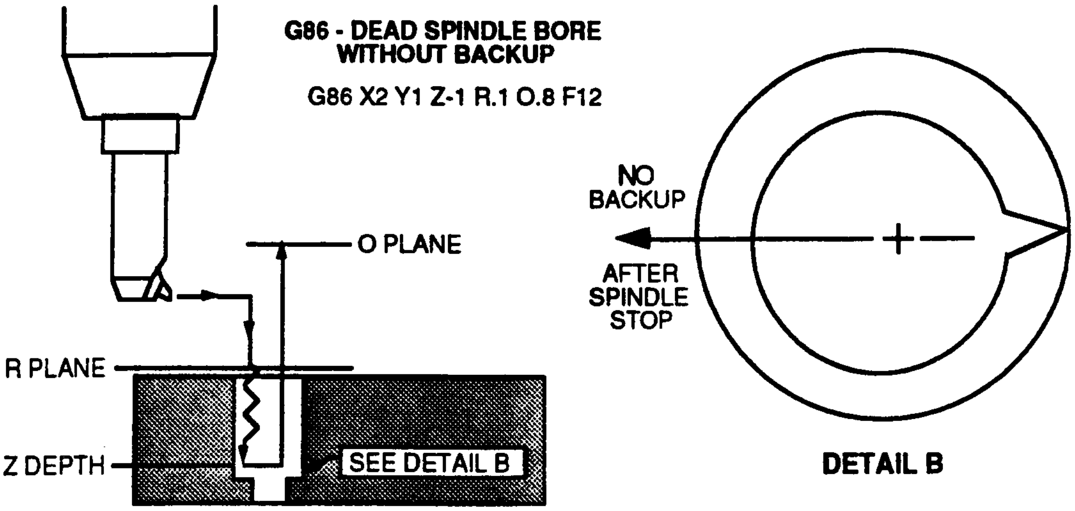
If there is no back-up your machine is not equipped with a self orienting spindle, the G76 cycle will operate with a zero backup distance. This means that G76 will be the same as the G86 Dead Bore Fixed cycle.

G76 is modal and cancels any other fixed cycle.

G86 - DEAD BORE FIXED CYCLE (NO BACKUP, AUTO RETURN)

This cycle operates as G76, but the tool tip does not back away before the retract. Without a self-orienting spindle, the tool tip direction will be random on withdrawal. The spindle will restart at the original speed and direction after the tool returns to the O plane.

G86 is modal and cancels any other fixed cycle.

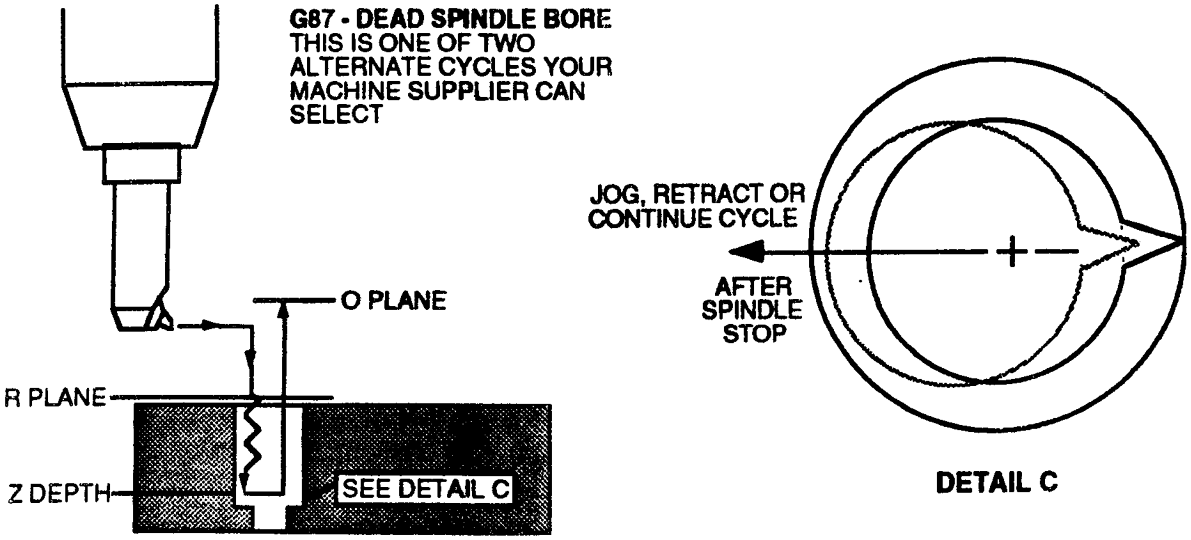


G87 - CYCLES

Your control will operate one of the following G87 cycles. Your machine supplier determines which cycle is operative.

G87 - DEAD SPINDLE BORE CYCLE (MANUAL BACKUP and RETURN)

The tool rapids XY over the hole center. Z then rapids to the the R plane, then feeds at F to the programmed depth Z. The control now outputs the spindle orient M Code (if available) to halt and orient the spindle and M00 command to halt program execution.



If desired, you can now move to Jog mode to back the tool tip away from the hole circumference. Unless your spindle is self orienting, you should first turn your boring bar to align the tool tip with the X and Y axis before jogging the axis to back the tool away. You can now jog in Z to withdraw or return to Auto mode.

With the cycle halted, you can withdraw the tool (if you have not already done so) by pressing the RETRACT button in Auto mode. The motion will proceed at rapid until you release the button.

Press CYCLE START to resume program execution. The first motion will be rapid in Z (either direction, depending on tool position) returning to the O plane. If you jogged to back the tool up before resuming the cycle, the next motion will be a rapid X or Y return to the hole center. The spindle then restarts the same direction and speed commands that were operative prior to the start of the cycle.

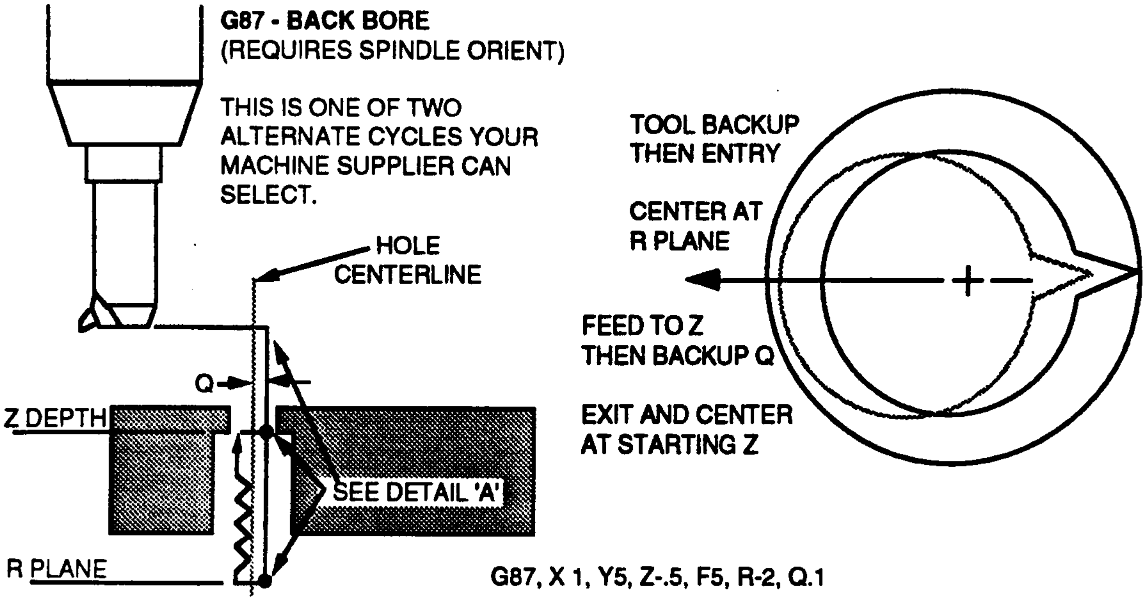
G87 is modal and will cancel any other fixed cycles.

G87 - BACK BORE CYCLE (AUTO BACKUP, AUTO RETURN)

This cycle bores the underside of through holes. A compatible self-orienting spindle is required.

You must enter the distance Q as the distance to move the tool in X or Y to obtain clearance to get the boring bar into the hole. Use caution because Q is programmed, and is modal with the peck depth in G73 and G83 cycles.

The tool rapids XY over the hole center. The control then outputs the spindle orient M Code. The cycle resumes, spindle off, with a short rapid motion to offset the tool distance Q (X or Y, direction opposite to the tool tip) clearing the inside of the hole. Then a rapid -Z motion positions the tool tip to the R plane. Notice that R is dimensioned as a clearance distance below the through hole exit in this cycle. (Set R a sufficient clearance for the spindle to reach operating speed before the tip contacts the material as it feeds back through the hole.)


G87 - BACK BORE
(REQUIRES SPINDLE ORIENT)
THIS IS ONE OF TWO ALTERNATE CYCLES YOUR MACHINE SUPPLIER CAN SELECT.
HOLE CENTERLINE
Q
Z DEPTH
SEE DETAIL 'A'
R PLANE
TOOL BACKUP THEN ENTRY
CENTER AT R PLANE
FEED TO Z THEN BACKUP Q
EXIT AND CENTER AT STARTING Z

```
G87, X 1, Y5, Z-.5, F5, R-2, Q.1
```

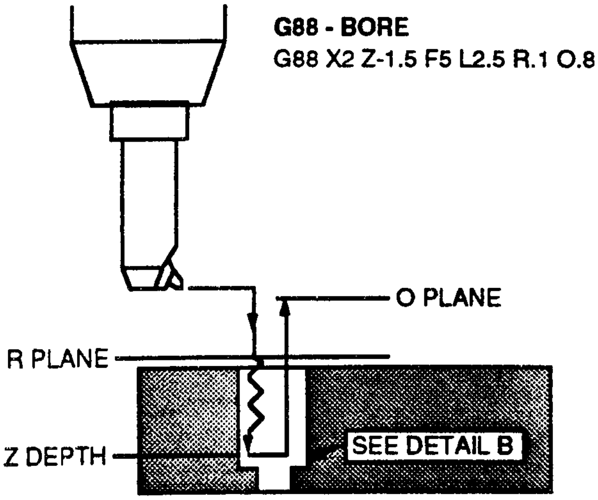
The spindle restarts at R with the same direction and speed commands that were operative prior to the start of the cycle. A short feed motion (in X or Y) returns the tool tip to the XY hole

center. The tool feeds in the +Z direction to the programmed Z end point. Note that Z is dimensioned above R in this cycle. At Z, the control outputs the spindle orient M Code. The spindle orients the tool tip then backs Q units away from the hole circumference.
A rapid withdrawal in +Z returns the tool tip to the starting Z coordinate, and is followed by a rapid X or Y motion, returning the tool to the hole center. The spindle restarts with the same direction and speed commands that were operative prior to the start of the cycle.

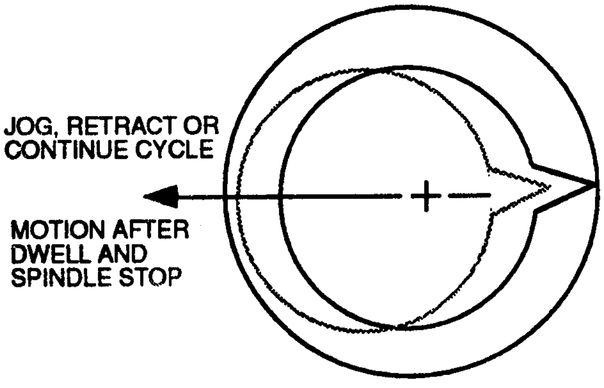
The method of determining whether G87 is configured for a back bore cycle or a dead spindle bore cycle is similar to the G76 cycle test. Program a G87 block with a Q value of .5". Then plot the cycle in Graphics mode (without motion) in the XZ plane (if tool tip orients in X) or YZ plane (if tool tip orients in Y). If the plot shows Q backup, G87 is the back bore cycle. If no backup occurs, the G87 will operate as a Dead Spindle Bore cycle.

G88 - DEAD SPINDLE BORE AND DWELL (MANUAL BACKUP, MANUAL RETURN)

The initial motion is XY rapid, positioning the tool over the hole center. A following Z motion begins entry at rapid, slowing to feedrate F at the R plane. The feed continues to depth Z, then the spindle continues to turn without linear feed for L (dwell) seconds to clean up the bottom of the bore. The control outputs the spindle orient M Code (M05 or M19) to halt the spindle and the M00 code to halt program execution.


G88 - BORE

G88 X2 Z-1.5 F5 L2.5 R.1 O.8


JOG, RETRACT OR CONTINUE CYCLE
MOTION AFTER DWELL AND SPINDLE STOP

DETAIL B

If desired, you can now move to Jog mode to back the tool tip away from the hole circumference. Unless your spindle is self orienting, you should first turn your boring bar to align the tool tip with the X or Y axis before jogging the axis to back the tool away. You can now jog to withdrawal, or return to Auto mode. With the cycle halted from Auto mode, you can withdraw the tool (if you have not already done so) by pressing the RETRACT button. The motion will proceed at rapid until you release the button.

Press CYCLE START to resume the cycle. The first motion will be a rapid in Z (either direction, depending on the tool position) returning to the O plane. If you jogged to back the tool up before resuming the cycle, the next motion will be a rapid X or Y return to the hole center. The spindle then restarts with the same direction and speed commands that were operative prior to the start of the cycle.

G88 is modal and will cancel any other fixed cycles.

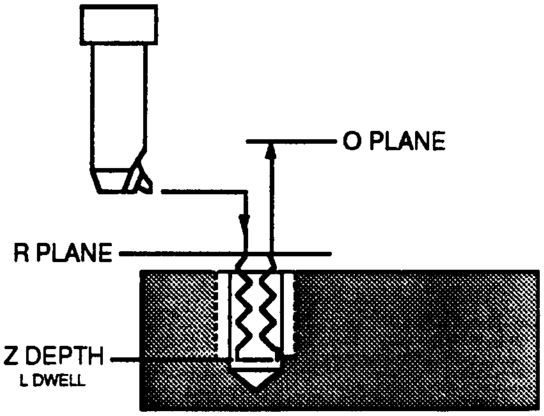
G89 - BORE CYCLE WITH DWELL (NO BACKUP, AUTO RETURN)

The initial motion is XY rapid, positioning the tool over the hole center. A following Z motion begins entry at rapid, slowing to feedrate F at the R plane. The feed continues to depth Z, then the spindle continues to turn without linear feed for L (dwell) seconds to clean up the bottom of the bore.

After the dwell, the spindle continues to turn and the tool begins a feed withdrawal. The tool withdraws until it reaches the R plane, ending the cycle.

G89 is modal and will cancel any other fixed cycles.

G89 - BORE
G89 Y3 Z-2 F7.5 R.1 O1.4 L1



G75 - RIGID TAPPING

Rigid tapping is an optional feature (refer to Part 1-Section 2 to learn if your machine can use the Rigid Tapping cycle) which allows tapping with a rigid tap holder. Your machine supplier can enable this feature if your machine's spindle can provide digital feedback signals to the control.

You must program the spindle's speed and direction prior to executing a Rigid Tap block. The control will calculate the tapping feedrate using your last spindle speed and the thread dimensions you enter in this event.

REQUIREMENTS

In order to use rigid tapping, you must have:

- A machine with a DC spindle.
- The PIC (Programmable Interface Controller circuit card) option.
- The Rigid Tapping option.

F - TAP THREAD DIMENSIONS - PITCH OR LEAD

Program F to specify the pitch or lead of your tap. When programming in inches, enter a sign positive F value to define the tap's pitch in threads/inch. When programming in millimeters, enter a sign negative F value to define the tap's lead in millimeters/thread.

The Lead/Pitch dimension F is modal to Rigid Tapping cycles only, and does not cancel the last F feedrate. For program clarity you should always program G75 and the Pitch/Lead F in any block that programs either entry.

The modal feedrate F will be restored when a non-Rigid Tap event or block executes.

FEEDRATE FORMULAS

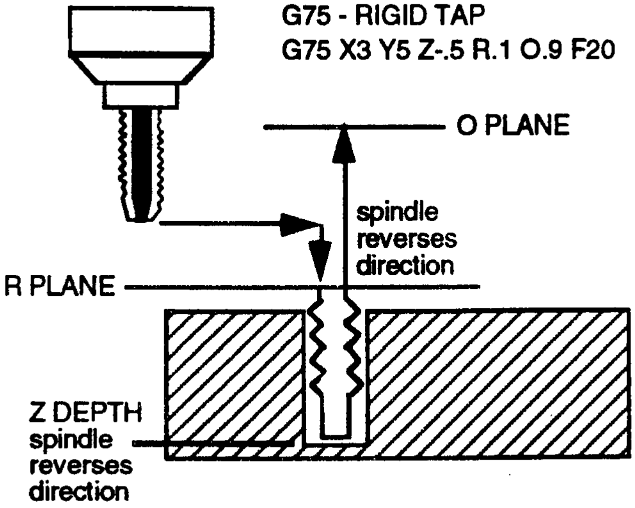
The control uses the following equations to calculate the tapping feedrate. The spindle speed in revs/min must set prior to executing a Rigid Tap cycle.

Inch feedrate: +F defines the pitch in threads/inch

Z Feedrate = (Spindle rev/min) / (threads/in) = in/min

Millimeter feedrate: -F defines the lead in mm/thread

Z Feedrate = (spindle revs/min)*(mm/thread) = mm/min


G75 - RIGID TAP
G75 X3 Y5 Z-.5 R.1 O.9 F20
O PLANE
spindle reverses direction
R PLANE
Z DEPTH spindle reverses direction

ORDER OF G75 CYCLE MOTION

1) Spindle direction and speed must be set in a prior block.
2) X, Y and any auxiliary axes rapid to their programmed endpoints.
3) The Z axis rapids to the reference plane R. During, or immediately after this move, the spindle will stop then resume at the original speed and direction.
4) Z feeds to the programmed Z depth. The Feedrate override pot is disabled at this time.
5) When Z reaches it's programmed depth, the spindle reverses direction.
6) Z retracts at feed to the reference plane R.
7) The control outputs M05 to halt the spindle.
8) Z rapids to the O plane.

G75 is modal and cancels any other fixed cycle.

G90 - ABSOLUTE

The control will interpret all end point dimensional data as absolute coordinate information.

G91 - INCREMENTAL

The control will interpret all end point dimensional data as incremental data relative to the starting dimension.

G92 - LOAD POSITION

The control will interpret all axis coordinates contained in a G92 block by substituting the values programmed with axis (XYZABCUVW letter addresses in place of the current axis positions. This command generates no axis motion. If no letter address has been programmed for a particular axis, no new position will be loaded for that axis. This command has no effect on machine zero, home or software travel limits. G92 is equivalent to a G6 in a Conversational Position Event. G92 is canceled by G99, M02, or M30.

G93 - INVERSE TIME

As an alternative to programming an inch or millimeter per minute feedrate for G01 or G21 blocks, you can specify the number of minutes that are required to complete the contouring motion described by each block. This type of feedrate definition is most useful when X, Y, Z and one or more linear or rotary auxiliary axes are programmed together.

To activate inverse time, program a G93 block. G93 is canceled by G94, M02, M30, or by pressing RESET when the control is not in cycle. When G93 is active, you may only program G00[6], G01 or G21 motion blocks.

To program the block time enter the inverse[7] of the time (in minutes) required to complete the programmed move in each G01 or G21 block. For example: enter F0.5 to program a move that must take 2 minutes to complete; enter F6.0 to program a move that must take 10 seconds (1/6 minutes) to complete. You can program a time that ranges from .006 seconds (program F9999.99) to 10 minutes (program F.01). You must enter F for all G01 or G21 blocks with G93 active.

Executing an inverse time F, or switching to G93, does not cancel the modal feedrate. The next contouring move with G94 active will use the last inch/millimeter per minute feedrate.

When an Inverse Time block executes, the control will calculate and command a feedrate to ensure that all axes arrive at their programmed endpoints at the same time. The actual feedrate for the move is displayed in the Auto and Single Event modes.

G94 - FEED PER MINUTE

G94 puts the control in a mode that sets inch per minute or millimeter per minute feedrate for all following G01, G02, G03, G21, G22, and G23 blocks. When G94 is active, programming F in one of these blocks will set a modal feedrate in inches or millimeters per minute.

G99 - CANCEL PRELOAD

The G99 function cancels the effects of a G92. Programming a G99 will reinterpret all axis positions previously modified by a G92 or G6 and restore the initial axis coordinate system.

[6] You may program a G00 block with G93 active as long as G80 - Cancel Fixed Quill Cycles is active. The G00 move will occur at the machine supplier's rapid rate.

[7] The inverse of a number (x) is equal to 1 divided by the number, or 1/x. In an EIA/ISO block the inverse time value must be a positive number. This is different from the negative Inverse Time value that must be programmed in a Conversational event.

M CODES - MISCELLANEOUS FUNCTIONS

These functions may be used by the machine supplier to control the supplier's machine functions. Coolant on/off and chip conveyor on/off are typical examples of functions performed by M Codes. The control outputs all M Codes (M00-M99)[8] to the machine tool interface but takes a prescribed action only with respect to the following M Codes:

M CODES RECOGNIZED BY THE CONTROL

M00 - Program Stop

Halts part program execution. Pressing Cycle Start after a halt will begin program execution with the program event appearing in the Next Event display. This display appears in Auto or Single event execution modes.

M01- Optional Stop

Operates as M00 is the OPTIONAL STOP soft key, or machine supplier switch is toggled On. If OPTIONAL STOP is toggled Off, M01 has no affect. The current status of Optional Stop is displayed on the Status screen.

M02 - End of Program

M02 execution will signify the End of Program. The spindle will turn off after all axes are returned to the Home position. This code does not have any effect on tape reader or load operations.

M03 - Activate Spindle Clockwise Spindle Rotation

M04 - Activate Counterclockwise Spindle Rotation

M05 - Spindle Stop

M06 - Tool Change Operation

Rapid to tool change position, then turn off spindle, and transfer to the next event after M Done confirmation from the machine tool interface.

M30 - End of Program

Processing this code will signify the End of Program when executed. Operation is same as the M02 command. This code will not have any effect on tape reader or load operations.

[8] Some systems permit M00 - M999 entry.

ORDER OF OUTPUT TO MACHINE INTERFACE

Axis motion and one G, M, S, T, H, D, and E code may be programmed in the same block. The order in which the control will execute these commands depends on whether the M code programmed in the block is Immediate or Deferred.

Immediate M Codes include M codes **except** M00, M01, M02, M06 or M30. Also **excluded** from this list are any M codes that your machine supplier has designated as a "deferred M code" and any M code that has been designated as a "Macro/CPC M Code"[9]. Your supplier's guide may list any deferred M Codes. Immediate M codes will execute before any programmed motion occurs.

Deferred M Codes include M codes M00, M01, M02, M06, or M30. Simply stated, deferred M codes include all M codes that are excluded from the Immediate M Code list above. Deferred M codes will execute after any programmed motion occurs.

EXECUTION ORDER FOR IMMEDIATE M CODES

1) G, H, D, E activate simultaneously
2) M Code
3) S Code
4) T Code[9]
5) Programmed motion.

EXECUTION ORDER FOR DEFERRED M CODES

1) G, H, D, E activate simultaneously
2) S Code
3) T Code[9]
4) Programmed motion.
5) M Code[9]

M and G CODE NOTES

There is a limit of one M Code (maximum) in any block. There is a limit of one G Code (maximum) in any block. An M06 block, however, may not contain a G Code or any motion entries.

If the block contains M, S, T, H, D, or E Code information, the allowed G Codes are limited to G00, G01, G02, G03, G21, G22, or G23.

While M02 and M30 are considered to be deferred M Codes, it is generally illogical to program E, S, or T within an M02 or M30 block. You may not program motion in an M02 or M30 block.

[9] Macro/CPC M Codes or Macro/CPC T Codes are advanced features that may be enabled by your machine supplier. Refer to Part 1-Section 2 to determine whether these features have been enabled. In general, when the corresponding feature is enabled, the control will execute an assigned Macro Catalog part program upon execution of a Macro/CPC M Code or a Macro/CPC T Code. Special M and T Codes will execute last when they are programmed with other codes. Technical information about these features is provided later in this section.

ACTIVATING TOOL OFFSETS - T, H, D CODES

Several different methods of programming tool offset codes are available for your machine supplier to select.

Your machine supplier should provide you with instructions describing how to change tools and activate tool offsets on your machine. Further information about the tool change capabilities of the control can be obtained by reading the Tool Table sections in Part 2 of this manual.

The option that has the most impact on a part programmer is the use of H and D codes to specify tool length and diameter offsets. You can determine whether your control uses H and D codes by referring to the instruction in Part 1-Section 2 of this manual.

APPENDICES

DYNAPATH DELTA 40, 50, 60 MILL
CUSTOMER INFORMATION MANUAL

AUTOCON

SCREEN MESSAGES

The following messages appear as screen prompts. These errors may occur when you are entering part program data in the Program mode, or you are copying, receiving, or deleting information in the controls part program memory. You must press the CLEAR key to remove the message and correct the condition before continuing operations.

CATALOG FULL: Available number of part titles has been exceeded. Delete any unwanted programs.

COPY FAULT: The N number was not recognized on an attempt to copy a program event. Be certain a valid N number is used.

DELETE FAULT: The Delete Program mode F or L sequence has been incorrectly entered or is illogical; program inhibit is in effect; or no N number was recognized on an attempt to delete an event from Program mode.

DUPLICATE ID: The title of an imported part program already exists in the Catalog mode. This message only occurs if the title is not selected in the Catalog - otherwise a merge receive will occur.

EDIT ERROR: Insufficient storage memory available for the control to relocate part program data following a Receive mode operation. Delete any unwanted programs and reload.

EXECUTION FAULT: Data is missing, improperly entered or illogical.

FORMAT FAULT: This fault is generated when an entry to an event contains an incorrect leading character, wrong decimal location, improper sign or incremental operand where absolute is required. Also flagged when an attempt has been made to copy a program event with an invalid N number or a specified F or L number is invalid.

INVALID CONCURRENT ID: Cannot select RS232C program ID from concurrent Catalog mode.

INVALID PART ID: Cannot copy or rename a program to RS232C part ID.

MACRO ERROR You attempted to merge a macro program while a Macro Inhibit was set on the CNC.

MACRO INHIBIT: A PI1 inhibit has been entered in the Set Up mode.

MEMORY FULL: Available part program storage has been exceeded. Delete any unwanted programs.

NEXT N FAULT: No event number was entered prior to pressing the NEXT EVENT or -NEXT EVENT key or if NEXT EVENT key was used the next event would be outside the 0-9999.999 range.

NO N FAULT: A program sequence number must be entered prior to any other data.

MEMORY LEFT: programs.

PARITY ERROR: A parity check during a Receive mode operation did not show even parity. The data is suspect.

PART ID ALREADY EXISTS: The selected program title is already in use.

PART ID IS CYCLING: The cycling program may not be selected from the concurrent Catalog mode.

PPS FORMAT FAULT: An internal parity check shows faulty program data in a memory location. Review event data from Display Program mode. Reenter the event if necessary.

PPS OVERFLOW: The attempted receive, edit or copy operation would exceed available part program storage. Delete unnecessary programs from storage.

PROG INHIBIT: A PI1 inhibit has been entered in the Set Up mode.

RECEIVE ERROR: The program contains one or more entries in wrong format. Error was detected at while receiving data from an off-line device. The Receive operation was aborted.

RESERVED PART ID: Part ID RS232C is reserved for buffered input operations.

SIZE OF PART ZERO NO COPY: A program of zero content cannot be copied.

STORE FAULT: See PROG INHIBIT.

VALUE FAULT: An entry to an event is outside of the allowable range (e.g. improper N number or nonexistent G cycle).

STATUS SCREEN MESSAGES

The following messages define the current condition of selected initialization parameters or EIA/ISO modal variables. These messages report the current status of the control and machine. The Status screen may be viewed at any time by pressing the FAULT/STATUS key twice.

X, Y, Z, 4, 5, 6 NOT REFERENCE ZEROED: Cautions that a reference zero operation has not yet been performed. Some applications may require that you Reference the machine before part programs can be executed.

G CODE STATUS MESSAGES: These messages report the current setting of several modal G code groups. Modal G Codes are changed by execution of an EIA/ISO part program block. They are also changed to the machine supplier's default values when the operator by presses RESET when the control is not in cycle, or when a part program execute an M02 or M30 end of program command.

OPTIONAL STOP ON/OFF: Displays the current status of the Optional Stop feature if the machine supplier has enabled the OPTION STOP soft key.

BLOCK DELETE ON/OFF: Displays the current status of the Block Delete feature if the machine supplier has enabled the BLOCK DELETE soft key.

FAULT SCREEN MESSAGES

The following errors will halt operations and generate a fault display. You must press RESET to clear the message and clear the fault condition resume operations.

EMERGENCY STOP: You must reset the control and correct the condition before continuing. Position information is retained unless a processor halt or memory failure has caused the fault.

X, Y, Z, 4, 5. 6 SERVO ERROR: Allowable following error has been exceeded, the axis has failed to respond to a servo command, a travel limit switch has been contacted, or the servo drive has shut down.

GANTRY CROSS ERROR FAULT: The allowable following error for one axis in your gantry system has been exceeded. This error should be investigated in the same manner as a servo error.

X, Y, Z, 4, 5, 6 TRAVEL LIMIT: A software travel limit has been exceeded. Limits are entered in the Set Up mode. RESET and jog off the travel limit.

4, 5, or 6 AXIS DISCONNECTED: The control has detected an attempt to execute an auxiliary axis command and the axis is not installed.

PIC I/O FAULT: An electrical circuit connecting a machine switch or relay to the control has failed. Report this error to your machine supplier.

MTB MESSAGES: Your machine supplier may designate halt, error, or status conditions specific to your machine. Refer to your supplier's specifications.

INTERRUPT: An operating system error has occurred. Do not attempt to continue operations. Report the control's serial number and circumstances to your machine supplier.

LOSS OF FEEDBACK: One or more of the axes feedback devices may have failed. Turn power Off, then On. If condition reoccurs, contact your machine supplier.

X, Y, Z, 4, 5, 6 AXIS RUNAWAY: An axis has moved without a command from the CNC. Turn power Off, then On. If condition reoccurs, contact your machine supplier.

CPU WATCHDOG: TIMEOUT An operating system error has occurred. Do not attempt to continue operations. Report the control's serial number and circumstances to your machine supplier.

LOSS OF COMM. WITH FRONT PANEL: An operating system error has occurred. If pressing RESET does not clear the problem, cycle power to continue operations. If the problem reoccurs, report the control's serial number and circumstances to your machine supplier.

PROBE CONTACT FAULT: A probe contact occurred while the Probe Protect signal was high.

FLOPPY DISK ERROR MESSAGES:

BAD DISK SECTOR: There is a bad sector on your floppy disk. Use another diskette.

DISK ACCESS DENIED READ ONLY DISK? You cannot transmit a file to the floppy diskette in the drive because it is write protected.

DISK FILE ALREADY EXISTS: You are attempting to overwrite a file on the disk. Either delete the disk file or transmit your file to a different filename.

DISK FILE NOT FOUND : The filename that you requested is not located on the diskette. Remember that you must press D ENTER whenever you insert a different diskette into the drive.

DISK INVALID MEDIA ID: Your floppy disk is not compatible with the disk drive. Use a disk that was formatted for a DOS computer.

DISK TIME OUT: DISK IN DRIVE? After you press D enter in the Transmit/Receive mode the control will read the floppy disk that is in the drive. This message indicates that there is no disk in the drive.

DISK WRITE PROTECTED: The control cannot transmit your part program to the floppy disk because the diskette is write protected. Slide the write protect tab to the "enable" position.

INSUFFICIENT DISK SPACE: There is not enough room on your diskette to store your part program. Insert a new, formatted diskette.

UNABLE TO INITIALIZE DFM MEMORY: A hardware or software error has occurred. Contact Autocon Technologies.

DMA ACCESS ACROSS 64K BOUNDARY: A hardware or software error has occurred. Contact Autocon Technologies.

DISK DMA OVERRUN: A hardware or software error has occurred. Contact Autocon Technologies.

ILLEGAL COMMAND TO FLOPPY DISK DRIVER: A hardware or software error has occurred. Contact Autocon Technologies.

CRC ERROR ON DISK READ: A hardware or software error has occurred. Contact Autocon Technologies.

DISK SECTOR NOT FOUND: A hardware or software error has occurred. Contact Autocon Technologies.

DISK SEEK FAILURE: A hardware or software error has occurred. Contact Autocon Technologies.

FAULT MESSAGES

The following error codes are generated when the control's error checking routines interpret a logical error when a part program event executes. Program execution is halted and a flashing FAULT/STATUS message is displayed. The N Number of the faulty event will most likely appear in the Auto or Single Event mode's NEXT EVENT display.

POSITION

001 Not used
002 An axis value entered for a G6 Position event is incremental.
003 (Aux. Axis Option) - the programmed rotary axis absolute value exceeds a maximum of 359.999°. This error code can occur in the Position, Linear Mill or Repeat event.
004 The value that you programmed for your rotary index axis is not a multiple of 1° or 5°.
005 - 009 Not used.

LINEAR MILL

010 A feedrate value must be entered when the Aux. axis is programmed by itself.

011 - 019 Not Used.

ARC MILL

020 End angle B has not been entered into the event.
021 The distance from center to start point does not equal the distance from center to end point of the arc. The actual start or end point may differ from either point presumed when the event was programmed.
022 The ratio of the feedrate to the radius is too large. The result will be a linear move with a possible servo error. Reduce the feedrate.
023 The generated Z feedrate relative to the arc feed would exceed the rapid rate and cause a servo error.
024 The Cam Mill start radius (E) must be greater than zero.
025 The Cam Mill start radius (R) must be greater than zero.
026 Illegal axis/center data
027 - 029 Not Used.

FRAME MILL OR CIRCLE MILL

030 Letter X, Y and/or Z was not entered into the event.
031 The value for X or Y is less than or equal to two times the corner radius. (Frame Mill Event only).
032 Cutter compensation off (CO) was programmed with a finish cut (L) specified.
033 A Pocket Mill (G7) was programmed with cutter compensation off (CO), cutter compensation value zero, or an outside cut specified.
034 Frame or Circle Pocket Mill - G7 programmed with an active cutter radius less than pocket overlap value (the value J or default .010"/.254mm).
035 A Peck value (K) equal to zero (0) was programmed.
036 The radius was not entered.
037 An inside cut was programmed and the finish dimension L is less than the active cutter's radius.
038 The cutter compensation (C) was not programmed or left blank.
039 Invalid pocket (G7) cutter diameter.

CIRCLE MILL

040 Radius value (R) equals zero (0).
041 - 049 Not Used.

BOLT CIRCLE

050 No hole (H) information entered, an H value of zero, or the number of holes to be processed is greater than the number of holes declared for the bolt hole pattern.
051 No Bolt Circle radius was entered or a radius value of zero was programmed.
052 You cannot program a 2nd reference plane in an event that cancels a quill cycle.
053 - 059 Not Used.

REPEAT

060 The sequence number (N) was not entered in a Repeat event.
061 The first event (F) or number of repetitions (T) was not entered in a Repeat event.
062 A rotational angle (C) and an X, Y or Z step value have been entered in the same Repeat event.
063 The first event sequence number (F) of a repeat event is greater than or equal to the Repeat event sequence number (the N value). A Repeat event may not include events programmed after itself.
064 Not Used.
065 No event in current program storage has a sequence number equal to the F value entered in a Repeat event.
066 A repeat loop is beyond the nesting limit. Repeats within repeats may only be programmed a total of four levels deep.
067 A combination of at least one repeat and one subroutine will probably transfer control to each other infinitely if allowed to continue.
068 - 069 Not used.

SUBROUTINE

070 The sequence number (N) has not been entered in a Subroutine event or in an event (of any type) being tested to see if it is the last event of a Subroutine.
071 The first event (F) or last event (L) defining the Subroutine has not been entered in the Subroutine event.
072 The first event (F) programmed in a Subroutine event appears later in program storage than the last event (L) that was programmed.
073 Not Used.
074 A Subroutine is calling a range of events including itself. Specifically, a Subroutine event has a sequence number that is between the values entered for the first and last event sequence numbers. The Subroutine would infinitely call itself.
075 A Subroutine event is calling a section of sequence numbers where a second Subroutine event is called. The second Subroutine is then calling the first Subroutine. The result would be an endless loop.
076 A Repeat event within a Subroutine has a first event value which is outside the Subroutine and causes the event that called the Subroutine to be executed again. The result would be an endless loop.
077 The nesting limit for Subroutines has been exceeded. You cannot program more than 4 levels of Subroutines within a Subroutine.

078 No event is stored that has a sequence number equal to the F or L value programmed into the Subroutine.
079 An N number within a Subroutine range of events is greater than the last N Number (L) specified by the Subroutine event.

DWELL

080 L was not programmed.

MACRO

081 You must enter the name of the macro you wish to call under the "P" parameter in the main program's Macro Call event.
082 The macro ID specified in the calling event cannot be found in the Macro Catalog.
083 You have nested macros greater than four deep.

084 Your macro calls itself; it may call itself directly or a call to itself may be nested within other calls. This is known as recursive calling and is not permitted.
085 The axis identifier in a CPC offset or probing command is out of range. See the appropriate command in Part 2 of this document for the acceptable range.
086 A CPC probing command has been programmed on a control in which the probing option has not been installed.
087 Cutter compensation must be turned off before a calling program calls a macro or CPC into execution.
088 You cannot program an Arithmetic event while cutter compensation is active.
089 Not Used.

M FUNCTION

090 An M Function event was programmed without either an M Code (M), spindle speed (S) or tool number (T) entered.
091 The tool offset identifier digits are outside the allowed range.
092 E Code exceeded the allowable range of E00-E14.
093 The first two digits in the T Code exceed the maximum tool number selected in Tool Life Management.
094 All tools within the selected group have expired. If tool life expires in the middle of a cycle, it will be allowed to finish the cycle.
095 Special M and T macros may not be programmed in the same event. Refer to Part 1-Section 1 for further information.
096 Not Used.
097 Your maximum spindle speed is 65535 rpm.
098 The tool you programmed is already in the spindle.
099 The tool request could not be located in Random Tool Table.
100 Tool number requested same as tool in spindle.
101 An error has been detected in parameter PROM entries for the tool change position. Contact your machine supplier.
102 - 109 Not Used.

SCALE, SET UP or GRAPHICS

110 A scale factor is greater than 99.9999 or a zero or negative absolute value is entered.

111 The programmed inch/metric option value was other than 0 or 1. Also, no other dimension whose values rely on the inch/metric declaration may be programmed into the same event with the declaration.

112 A mirror image option value other than 0 or 1 was programmed.

113 Software Travel Limit.

114 The plane select code (P) must be P1-P5.

115 The Type of Plot code must be T1 or T2.

116 The Graphic event scale factor must be greater than 0.0.

117 - 119 Not Used.

CAVITY MILL

120 The cross section radius (R) or angle (A) has not been programmed into the event.

121 The length (L) or linear increment (I) has not been programmed in a linear type of cavity.

122 The end angle (B) or rotary increment (C) has not been programmed in a rotary type of cavity.

123 Either R or D is less than the active cutter's radius.

124 A G Code has not been entered into the event.

125 An G Code other than G8 or G9 has been entered into the event.

126 A T value (ending side) is programmed, but an S value (starting side) was not programmed.

127 A K value (ending bottom) was programmed, but a J value (starting bottom) was not programmed.

128 The cavity's start point, start angle, and center of rotation are in conflict with the shape that is programmed (check L, I, C, A, B values).

129 Not Used.

EVENT FORMAT

130 An unrecognizable or undefined character has been entered into this event.

131 The first character in the stored event was not a sequence number ‘N’ or “/”. (View this event in Display Program mode. To recover, delete the previous through the following event and re-enter all three events).

132 The event type code is an invalid character; the left or right parenthesis is missing from the entry or incorrectly entered; or an M00, M02 or M30 was not present at the end of the program.

133 A dimensional entry has been entered incorrectly. The fault is related to a decimal point, incremental, or minus sign, too many digits, or an invalid letter programmed within the event.

134 A number in the current event is too large for the allowed range.

135 Multiple decimal points have been entered in a given number value.

136 A premature end of event ($) or end of program (END) has been detected. An M30 or M02 was not programmed at the end of the part part program.

137 A memory parity error occurred during part program cycling. Reenter the entire event from the keyboard; reload the program; run start up diagnostics to insure memory is intact.

138 Both center coordinates must be programmed in a Rotate event.

139 Not Used.

140 A polar event fault: the angle of motion and/or the radial distance of the move (R) has not been entered.

141 The Z reference plane (event variable W) has been programmed below the final depth of cut (Z).

142 No more than 10 consecutive M Codes can be programmed.

143 The cutter compensation logic cannot compute the compensated path in a Linear Mill or Arc Mill with cutter radius compensation. Usually, the cutter size is too large to follow a tool path in a confined geometric area. Typical problems include retracing a narrow tool path with compensation, attempting to mill a notch with a cutter diameter exceeding the notch width or the cutter radius is too large to reach the start point of an arc to mill around a corner without backing into the part.

144 The cutter radius is greater then the circle radius in a Linear Mill or Arc Mill with cutter radius compensation.

145 The plane select variable (P) is not P0 (zero), P1, or P2.

146 An Arc Mill event Direction of cut (D) variable has been entered outside of the range of D0-D1, or a Frame Mill or Circle Mill event Direction of cut D variable has been programmed outside the range of D0-D3.

147 The Cutter Compensation (C) variable has been programmed outside the range of C0-C2.

148 An intersection cannot be calculated along a path determined by the active cutter's radius while cuter radius compensation is on. Example; a line is not tangent to a circle within the program tolerance.

149 Not Used.

G CYCLES

150 The G value programmed is not one of the allowed values for G cycles in the faulty event.

151 The G value programmed is not in the range of G0-G6 for a Position event, or G0-G5 for a bolt circle event.

152 W may not be entered as an incremental value.

153 Letter W, Z, and/or K has never been entered as required for the declared G cycle in this event.

154 Contact Autocon Technologies with a description of events that caused error.

155 Invalid F value, feedrate value does not fit the ipm/mmpm format.

156 Report the error code, condition and serial number of the control to your machine supplier.

157 Report the error code, condition and serial number of the control to your machine supplier.

158 An internal hardware error (if unrelated to 154 handwheel operation) has occurred. Report the error code, condition and serial number of the control to your machine supplier.

159 Invalid Lead/Pitch (F) or RPM(S). Programmed or modal value is invalid or not programmed.

EIA EVENT

160 The system does not recognize this G code.

161 EIA: M S T or E with illegal G codes.

162 EIA: M06 with motion.

163 Dwell format has been improperly programmed.

164 W cannot be programmed with another axis. W cannot be programmed when a quill cycle is active.

165 G93 Inverse Time is modal and time F was not programmed. It must be programmed in every block that uses Inverse Time.

166 You may only program G01, G21 and G00 (with G80 active) blocks when G93 Inverse Time is modal.

167 Cutter Compensation cannot be active when a G27, G28, G29, or G53 block executes. Cancel it with a G40 block.
168 The intermediate point for a return from the reference point has not been defined in a prior G28 block for a programmed axis.
169 - 175 Not Used.

HARDWARE/OPERATIONS - SYSTEM FAILURE

176 - 179 Report the error code, condition and serial number of the control to your machine supplier.

PROBE EVENT

180 A required entry for the selected G cycle has not been programmed in the Probe event.
181 Scaling, axis rotation or mirror image cannot be active upon execution of the Probe event.
182 The spindle or table probe has not been calibrated.
183 The probe did not contact the target within the range specified in the event.
184 Not Used.
185 A fixture offset other than E00 was active upon execution of a G1 or G2 Probe cycle.
186 - 199 Not Used.

AUTO RADIUS - LINEAR MILL

200 The next event is not valid for the auto radius entry R.
201 The events to be joined with an auto radius form parallel lines. The auto radius cannot be performed.
202 - 206 Not Used.
207 The auto radius is too large for the programmed linear mill events.
208 The joining events have different P entries, are not in the same plane.
209 Auto radius is programmed for an out of plane axis.
210 The center for auto radius cannot be computed.
211 Incomplete event. D must always be programmed in combination with V unless a solution will be computed by Calc Assist. Without Calc Assist, the two together represent a complete move, and cannot be programmed independent of one another.
212 The V programmed does not agree with the direction specified, usually by axis endpoint.

213 - 218 See CPC Error Codes, below.

HARDWARE/OPERATIONS - SYSTEM FAILURE

219 Call your machine supplier with the error code and serial number of the control. Report the condition under which the error occurred.

BUFFERED INPUT

220 No baud rate selection has been entered

221 Incoming data failed a parity check.

222 Memory overflow or a character was not read. If 1600 or more characters are available (see Catalog mode) is available, select a lower baud rate.

223 Improper tape identifier format (for the program title). Correct format is a left parenthesis followed by up to eight characters or spaces followed by a right parenthesis.

224 An invalid event was detected. Subroutine and Repeat events cannot be executed through the buffer.

CPC ERROR CODES

213 The offset identifier in a GAO, GOV or SOV command is invalid. The offset letter code must equal 20 for Tool offset (T), 17 for the random tool number (Q) or 5 for a Fixture offset (E). No other values are allowed.

214 The Tool or Fixture (ARG2) number in a GOV or SOV command is out of range. T must be in the range 1-99; E must be in the range 1-32.

215 The offset letter (ARG3) in a GOV or SOV command is out of range. See the GOV or SOV topic in Part 2 of this document for a complete definition of acceptable values.

216 A GAG, GAO, GOV or SOV command is followed by an invalid number of arguments.

217 The offset value specified by a SOV command results in an offset value > 999.9999 inches or 9999.999 mm.

218 The argument for a GAG command must equal 0 or 1. No other values are permitted.

225 You have used an invalid mnemonic (such as a function) as a command (i.e. N100(A)(ABS,#34)).

226 An event contains a character string that the control cannot identify as a command or expression.

227 Your macro contains a Symbolic Variable that is not followed by an appropriate delimiter (i.e. parenthesis, arithmetic operator, etc.).

228 Your macro contains a stand alone expression that is not followed by an appropriate delimiter (in most cases a missing parenthesis or comma is to blame).

229 Your macro contains a pound sign (#) followed by a number (or an expression that when evaluated produces a number) that is not in the range 1-99. Symbolic Variable numbers must be within this range.

230 Your macro contains a function (i.e. SQR, TAN, etc.) that is followed by an invalid delimiter. Generally, functions must be immediately followed by a comma (sometimes by a right parenthesis); this error appears when one is not.

231 Your macro contains a constant followed by an improper delimiter. Constants may be followed by arithmetic operators, commas or a right parenthesis depending on how they are used.

232 Expressions may be terminated by a comma only if the number of right parentheses in the expression equals the number of left parentheses. Otherwise, this error is flagged.

233 The only mnemonics allowed within a mathematical expression are the trig functions, the rounding functions, INT, SQR, and ABS.

234 A multiply (*) or divide (/) operation has been detected without the first term (i.e. N10(A)(#5=*#10)).

235 Your macro contains a character that the control does not recognize. The only characters allowed are those that appear on the controls alphanumeric

keypad, the space, the slash (/), left paren {(}, right paren {)}, pound sign (#), equal sign (=), plus sign (+) and asterisk (*).

236 This error is flagged if the first character of a new expression level (the first character following a left parenthesis) is an End of Block marker.

237 Your macro contains expressions within expressions (ABS,(SQR,(TAN,(INT.....) beyond the limit of 4. Break the nested expression down into several separate expressions.

238 The last event of a Subroutine call is an Arithmetic branch event (IF, GO, GIN).

239 A single number contains more than one decimal point (i.e. 12.34.56).

240 The numeric result of an expression has overflowed the memory space allocated for it. Check that the expression is valid; it is unlikely that a valid expression would produce a number large enough to flag this error.

241 You can have no more than four levels of expression active at one time. This error is flagged when you exceed this limit (i.e. SQR(ABS(INT(COS(ATN(....))))))

242 On evaluating a division operation the control has discovered that the divisor is 0. In this case, it is likely that the divisor is itself the result of an expression; check that this expression is valid. If it is, you may have to somehow modify the macro to skip the division operation when the divisor is 0.

243 The control's interpreter has found an arithmetic event in the master program. This is not permitted; the event must be removed.

244 You must have the advanced macro feature to use arithmetic events; this error indicates that an arithmetic event has been detected by a control in which advanced macros are not installed.

245 The result of an expression is indeterminate. Check that the expression is valid; if it is, a variable contained in the expression must be at fault. If there is no way to avoid an offending value in the variable, you may have to modify the macro to allow for it.

246 A branch statement in your macro specifies an invalid destination. The destination N number must be between 0 and 9999.999.

247 A branch statement specifies its own N number as the destination.

248 A branch statement in your macro is attempting to branch to a nonexistent N number. Check that all the branch destinations exist.

249 Your macro is attempting to take the square root of a negative number. This number is most likely the result of an expression; check that the expression is valid. If it is, one way to avoid this error is to use the ABS function on the variable which is to be evaluated for its square root.

250 Your macro is attempting to evaluate a trig function that is undefined, i.e. TAN,90. When evaluating variables in such cases it is important to check that they are within a defined range before applying the trig function to them.

251 Decimal I/O address < 0 or >1535 in a GIN statement. For SON and SOF, the decimal address must be in the range 0768-0799.

252 An invalid baud rate is selected in the Transmit/Receive modes.

REFERENCE ZERO

254 A required reference zero operation prior to execution of the part program has been specified for this system by your machine supplier. A reference zero is required before cycling is allowed.

FUTURE EXPANSION

XXX "Refer to Manual" (error code number may vary)
Means that an error condition has been detected but there is no attached description of the error. This will generally be related to a new feature not documented at the time your manual went to print.

Report the error code number and firmware number from your control's start up screen to Autocon's Sales Support Staff. We will make a determination of the error in question and report back to you.

OFF-LINE COMMUNICATIONS INTRODUCTION

This section describes several issues involved with transferring data between the control and a serial communications device. These topics include:

1) Pinouts for the cable between your communications device and the CNC.
2) Format for data prepared on an off-line device:
   - Conversational part programs
   - EIA/ISO part programs
   - Tool Table data (Tool Table, Fixture Offsets, Random Tool Table, Tool Life Mngt).
3) Terminal mode ACK/NAK protocols.

CABLE CONSTRUCTION

Two RS-232 communications ports are available on the control. These ports are used for transmitting serial data to and from RS-232 devices. The first port, COM1, is used for transmitting part programs and Tool Table data to and from an external device. The second port, COM2, is used with the control's Terminal mode feature.

COMMUNICATIONS TERMS and DEFINITIONS

Device - any electronic serial communications device that can transfer data with the DynaPath control. Typical devices include personal computers and industrial diskette drives that act as tape reader emulators.

Transmit - any operation that transfers data **from** the control **to** a device.

Receive - any operation that transfers data **to** the control **from** a device.

Halt/Resume data flow control - a communications protocol that allows the device that is receiving data to command the sending device to halt, and then resume the transmission.[1] There are two common methods of halt/resume control: **Xon/Xoff** and **hardwired** control. Read the following definitions to determine which form of halt/resume control that your device (and communications software) supports. Some devices can support both forms of control.

Xon/Xoff data flow control - a common communications protocol that allows a receiving device to tell the sending device when it is OK to send data. The halt and resume signals are sent on the transmit and receive data lines. If the receiving device wants to temporarily halt data transfer, it outputs a DC3 (Xoff, ASCII 13 or Control S) stop character. To signify that it is ready to receive data, it outputs a DC1 (Xon, ASCII 11 or Control Q) start character. This is the preferred method of halt/resume flow control since it requires the simplest interconnect cable and it is supported by most communication software packages.

Hardwired data flow control - when a device is not capable of using Xon/Xoff control, it must use additional cable conductors to handle halt/resume control. With this form of control the devices sets a voltage high or low on a specific pin of the cable to identify when it is able to receive data. Hardwired control should only be used when your device does not support Xon/Xoff.

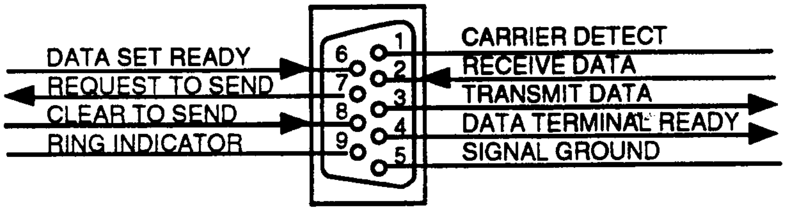
[1] Such halt/resume control allows the receiving device to stop the transmission when its buffer becomes full. Once the device clears its buffer (by writing to its memory or disk drive, for example) it will then command the sending device to resume the transfer.

Communications software - If your device is a computer, you must obtain a program that handles communications with the control. Your communications program must be able to use Xon/Xoff[2], send ASCII text data with 1 start bit, 7 data bits, 1 parity (even) bit, 1 or 2 stop bits, accept ASCII text data with 1 start bit, 7 data bits, 1 parity (even) bit[3], 1 stop bit, and send a <LF> character at the end of every line. All of these features are standard on a number of commercial and shareware communications programs.

THE DYNAPATH COMMUNICATION PORTS

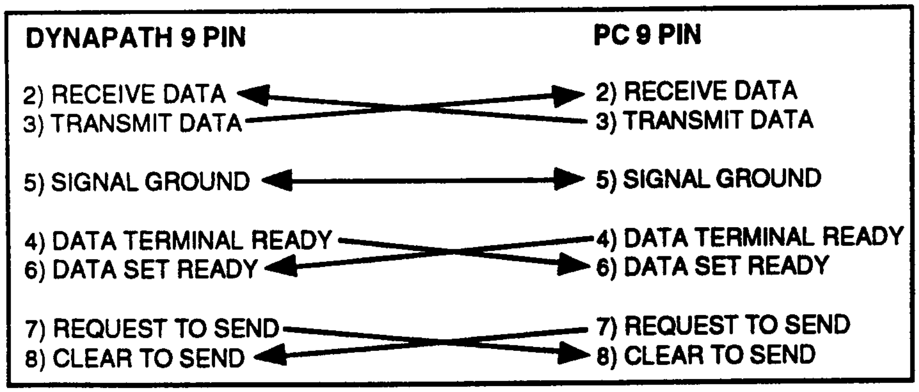
Pin assignments for the COM 1 and COM 2 ports on the Communications panel are shown **below.**

DYNAPATH 9 PIN COM1 and COM2 PORTS



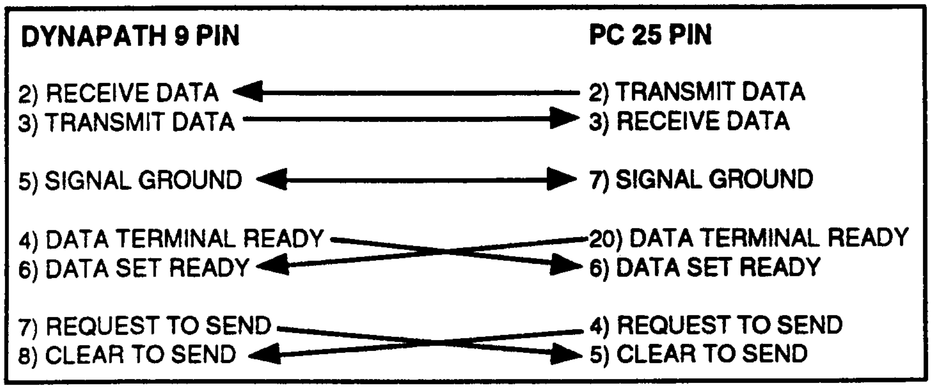
CONNECTING THE CONTROL TO A PERSONAL COMPUTER

The most common device that is used to communicate with the DynaPath CNC is an IBM PC® compatible personal computer. The cable that is required to connect the CNC to your computer's 9-pin or 25-pin communications port is shown **below and on the following page.**



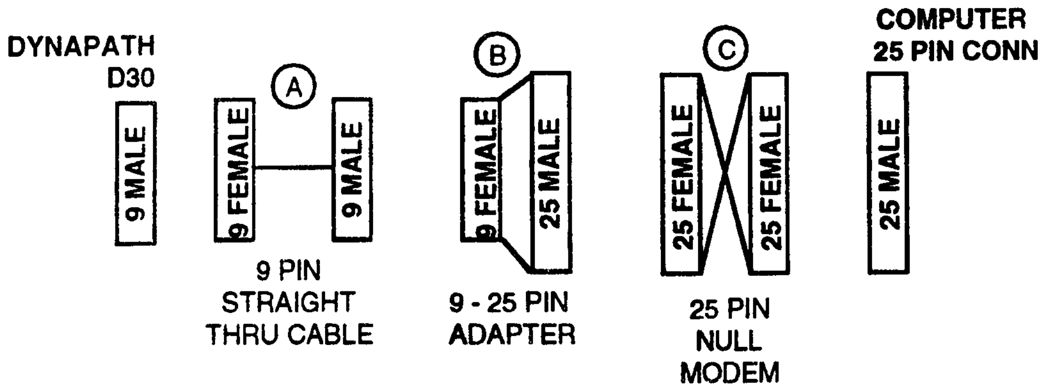
[2] If your device is a micro or personal computer, its communications program will have a group of "user" settings. Typically, one setting will allow you to select "Use Xon/Xoff Control" or "Do Not Use Xon/Xoff Control".

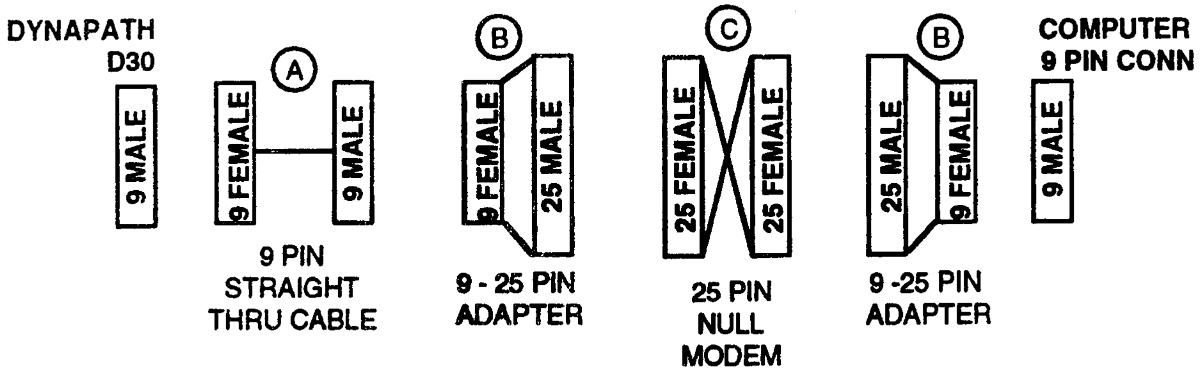
[3] Parity is selectable in the Transmit/Receive mode. If parity is on, all data is sent and must be received with an even parity bit. If parity is off, the control will send a 0 parity bit, and ignore the incoming parity bit.



These cables are referred to as "null-modem" cables throughout the computer industry.

If you prefer to purchase a communications cable, the illustrations **below** show typical setups using commercially available cables and adapter plugs.[4,5]





[4] Autocon has tested cables and adapter plugs that were supplied by Global Computer Supplies (phone: 800-845-6225). The reference letters in the illustrations relate to the following part numbers in the Global June 1992 catalog: A: 9-pin straight cable - part # C4155C, B: 9-25 pin adapter - part # C4338, C: null modem -part #C4915C. Similar items may be purchased from other vendors.

[5] The middle three adaptors in the lower cable may be replaced by a 9 pin null modem adaptor.

CONNECTING THE CONTROL TO ANOTHER DEVICE

If your device is other than a PC compatible computer, you will have to read the device's manual to determine the pinouts of its communications port(s).

When you are reading your device's manual, first locate the following pairs of signals:

Transmit data (TxD) and Receive data (RxD)
Clear to Send (CTS) and Request to Send (RTS)
Data Terminal Ready (DTR) and Data Set Ready (DSR)

Next, determine which signal in each pair is considered an output by the device, and which signal is considered an input by the device. This information should appear in the device's documentation.

Now you can construct your cable by connecting each pair's **output** with the corresponding **input** on the Autocon connector, and vice versa.

Example 1 - if your documentation indicates that device's **CTS signal is an output**, then it must be connected to the CTS input on the Autocon connector (refer to the Autocon connector illustration on the previous page). Logically following, the device's RTS signal must be in input and therefore connected to the RTS output on the Autocon connector.

Example 2 - if your documentation indicates that device's **CTS signal is an input**, then it must be connected to the RTS output on the Autocon connector (refer to the Autocon connector illustration on the previous page). Logically following, the device's RTS signal must be in output and therefore connected to the CTS input on the Autocon connector.

Finally, connect the device's **Signal Ground (SG)** pin directly to the Autocon Signal Ground (SG) pin.

TRANSMIT/RECEIVE COMMUNICATION

RECEIVE - PART PROGRAM or TOOL TABLE DATA - Xon/Xoff CONTROL

1) Set the device and the control to use Xon/Xoff.
2) Start the control by pressing the START OPERATION soft key. The control will now output a DC1 (Xon) character. [6]
4) Now start sending data from the device.
5) The control will not send any following control characters during the Receive.[7]

[6] If your device requires a DC1 character before it will send data, start the device first, then press a START OPERATION soft key on the control. Your device will use the DC1 character to initiate the load.

[7] Some communications programs may need to monitor the control's DTR and RTS signal during a transmission even when Xon/Xoff data flow control is used. When Xon/Xoff is used, these hardware signals are controlled as follows: the DTR ready signal will go high as soon as you enter the Transmit or Receive mode; the RTS signal will be set high when the operator presses the START OPERATION soft key. Both signals will remain high throughout the transmission.

RECEIVE - BUFFER INPUT - Xon/Xoff CONTROL

Buffer Input allows you to execute part programs in Auto mode as the data is sent to the control. Since the data is never stored, this feature is useful for running programs that would exceed the control's storage memory.

1) Select Buffer Input by entering the part title **RS232C** in the control's Catalog. The control will now set the DTR signal high.
2) Set the device and the control to use Xon/Xoff.
3) Start the transfer by moving to the Auto mode and pressing CYCLE START[8]. The control will now set the RTS signal high and it will output a DC1 (Xon) characters every 100 milliseconds until it starts to receive data.
4) Now start the sending device.
5) Once the controls receives 20 events, it will begin execution (while the device continues to send characters). When space remains for only 50 characters in the buffer, the control will output a DC3 (Xoff) character. The sending device must not send more than 50 characters after receiving a DC3 from the control.
6) After the control has run enough events to make room in the buffer for more than 200 characters, the control outputs a DC1 character. The device should then resume data transfer. This exchange will continue until the control detects an "E" (end of load) character. The control will halt the Receive and ignore any further incoming characters. It will output no further characters. In the case that the control executes events quickly enough during the transfer to exhaust the buffer, execution will not continue until sufficient events are loaded to allow cutter radius compensation logic to look ahead one motion.

TRANSMIT - PART PROGRAM and TOOL TABLE DATA - Xon/Xoff CONTROL

1) Set the device and the control to use Xon/Xoff.
2) Start the device first.
3) Start the transfer by pressing the START OPERATION soft key.
4) If the control's CTS input is high, the control will start sending data.[7]
5) The external device may then halt the operation by sending a DC3 (Xoff) character to the control. The device can resume the transmit by sending a DC1 (Xon) character to the control. The device may continue to halt and then resume the data flow, as desired. No control characters are sent by the control.[7]

RECEIVE -PART PROGRAM and TOOL TABLES - HARDWARE CONTROL

1) Set the control to X0 -Do Not Use Xon/Xoff. The control's DTR signal will be set high when you enter the Receive mode.
2) Start the control by pressing the START OPERATION soft key.
3) The control's RTS signal will go high, and remain high throughout the transmission.
4) Start the transmission from the sending device.

[8] **Caution:** The machine will start as soon as transmission begins. If you cannot attend to the machine while starting the sending device, you may wish to inhibit axis motion by turning the FEEDRATE OVERRIDE POT to the OFF position.

RECEIVE - BUFFER INPUT - HARDWARE CONTROL

1) Select Buffer Input by entering the part title **RS232C** in the control's Catalog. The control will now set its DTR signal high.
2) Set the control to X0 -Do Not Use Xon/Xoff.
3) Start the transfer by moving to the Auto mode and pressing CYCLE START[8]. The CNC will now set the RTS signal high.
4) Now start the sending device.
5) Once the control receives 20 events, it will begin execution (while the device continues to send characters). When space remains for only 50 characters in the buffer, the control will set RTS low. The sending device must not send more than 50 characters after RTS goes low. After the control has run enough events to make room in the buffer for more than 200 characters, the control sets RTS high. The device should then resume data transfer. This exchange will continue until the control detects an "E" (end of load) character. The control will halt the Receive and ignore any further incoming characters. In the case that the control executes events quickly enough during the load to exhaust the buffer, execution will not continue until sufficient events are loaded to allow cutter radius compensation logic to look ahead one motion.

TRANSMIT-PART PROGRAM and TOOL TABLES - HARDWARE CONTROL

1) Set the control to X0 -Do Not Use Xon/Xoff. The control's DTR signal will be set high when you enter the Receive mode.
2) Start the control by pressing the START OPERATION soft key. This will cause the control's RTS signal to go high.
3) The control will start to transmit the data if its CTS input is high.
4) The control will continue to transmit until its CTS signal goes low. It will continue transmitting when it sees its CTS input return high.

OFF-LINE PROGRAM PREPARATION

Part programs and Tool Table data may be prepared on an off-line device and then transmitted to the control. To do so, you must obtain a personal or micro computer and proper communications software that can send data in ASCII format to the control's RS-232C port. You may also use the CNC's floppy disk drive to transfer data between two devices. The data must be formatted as specified in this section.

THIRD PARTY VENDORS

We have evaluated a number of third party vendor's products and services which run on popular computer systems. Off-line features may range from program entry, edit, transmit and receive operations to extensive plotting or interactive plot and program construction and editing.

Inquiries regarding the availability of these systems may be directed in writing to:

Autocon Technologies, Inc.
Sales Department
38455 Hills Tech Drive
Farmington, MI 48331-5751

Please identify your telephone number and the name of the person we should contact regarding your inquiry.

DATA FORMAT

The information described below will be used if you are preparing part programs or Tool Table data on an off-line device.

When preparing a tape reader for teletype transmission, you should observe a 10 second delay. This is equivalent to sprocket only punch of about 1 foot (30cm) in length. RS232C transmission will not require a starting delay.

All data will be transmitted as 7 bit ASCII characters with the 8th parity bit set for even parity or zero parity as selected in Transmit and Receive modes.

Decimal point programming is assumed. This means that any number without a decimal point will be read as an integer. Enter the decimal point only when required. Leading zeros must not be entered. Trailing zeros (zeros following the last significant digit) should not be entered. No other formats are recognized.

Spaces are only allowed in program titles, Conversational Text events, and Tool Table data.

In the following topics, <CR> represents a carriage return (ASCII OD hexadecimal) and <LF> represents a line feed (ASCII OA hexadecimal).

The same part program can contain both Conversational events and EIA/ISO blocks. The control interrogates each line to determine if it is in Conversational or EIA/ISO format. Each line must conform to the corresponding formats described below.

CONVERSATIONAL PART PROGRAM FORMAT

TITLE

The first line of a part program contains the title. The title consists of up to eight characters enclosed within parenthesis '()'. The eight characters can be selected from the following list.

-The 26 alpha and 10 numeric characters
-The decimal point
-The minus sign
-The space character

The RS232C and DIGITIZE titles cannot be used (they are reserved to select special control features). If no title is entered, the control will automatically assign it the name P0000000. You should rename such a program once it has been transferred into the control's memory.

A following <CR> and/or <LF> are allowed after the close parentheses ')' but they are not required. No following spaces are permitted.

PART PROGRAM EVENTS

Each event must contain the following four items in the order shown below.

1) **N Number** - N followed by the sequence number in +4.3 format.
2) **Event Type** - the event type character must be enclosed in parenthesis. You can program an EIA/ISO event with event type E.
3) **Event Data** - the commands appropriate for the event type which is programmed above. Entries may appear in any order unless the Macro Call event is programmed. The Macro Call event requires that the P is the last entry before the end of line $.
4) **End of Line character is the $** - A <CR> and/or <LF> character may be sent. The control will discard these additional characters.

END OF PROGRAM

Two options are available to terminate the program.

When the control detects the E character after an event's end of event $, it will edit the part program into memory then display a DONE message.

When multiple part programs will be received, enter a C character after each part program. This causes the control to resume loading. The last program must be terminated by an E character, as described above. Once the E is detected, the control will ignore any further data and begin editing the programs into part program storage.

EXAMPLE CONVERSATIONAL PROGRAM

The underline is shown to identify areas where spaces will be permitted.

COMMENTS	PART PROGRAM
Program title	`(partname) <CR><LF>`
First event	`N10(0)X1/Y3.9$<CR><LF>`
Event with block delete	`/N20(1)Z-4.3221$<CR><LF>`
Following event	`N3002.452(T)16 characters$<CR><LF>`
Macro Call event	`N3074(M)Q99PGETMACRO$<CR><LF>`
End of program	`N9999(9)M30$<CR><LF>`
End of load	`E`

EIA/ISO PART PROGRAM FORMAT

TITLE

The first line of a part program contains the title. The title consists of up to eight characters enclosed within parenthesis '()'. The eight characters can be selected from the following list.

-The 26 alpha and 10 numeric characters
-The decimal point
-The minus sign
-The space character

The RS232C and DIGITIZE titles cannot be used (they are reserved to select special control features). If no title is entered, the control will automatically assign it the name P0000000. You should rename such a program once it has been transferred into the control's memory.

A following <CR> and/or <LF> are allowed after the close parentheses ')' but they are not required. No following spaces are permitted.

PART PROGRAM EVENTS

Each event must contain the following three items in the order shown below.

1) **N Number** - N followed by the sequence number in +4.3 format.
2) **Event Data** - enter any data that is required in the block. G, M, S, T, E Codes, and any dimensional data can be entered in any order. However, the program may be easier to edit at a later date if the G Code is first. No spaces or commas are allowed. Only one G, M, S, T, and E Code may be entered per block.
3) **End of Line character is the <LF>**[9] - A <CR> and/or $ character may be sent. The control will discard these additional characters.

END OF PROGRAM

Two options are available to terminate the program.

When the control detects the E character after a <LF>, it will edit the part program into memory then display a DONE message.

When multiple part programs will be received, enter a C character after each part program. This causes the control to resume loading. The last program must be terminated by an E character, as described above. Once the E is detected, the control will ignore any further data and begin editing the programs into part program storage.

EXAMPLE EIA/ISO PROGRAM

The underline is shown to identify areas where spaces will be permitted.

COMMENTS	PART PROGRAM
Program title	(partname) <CR><LF>
First event	N10G91X14.32<CR><LF>
Event with block delete	/N45G00X2<CR><LF>
Following events	N100G1X10Y10Z-1.35<CR><LF>
End of program	N9999M30<CR><LF>
End of load	E

[9] Most PC compatible computers automatically store the <LF> characters when you press the RETURN key on the keyboard. Macintosh computers, however, store the <CR> character. Most communications programs have a selectable option that automatically ends outgoing lines with <LF> and <CR> characters.

TOOL TABLE FORMAT

Tool table data (as described below) may not be joined with part programs since Tool Table and Program Receive are separate operations. Before a Tool Table receive, you should insure that the correct INCH or METRIC is selected in the Set Up mode.

Tool Table entries may be prepared off-line, and later loaded into the control. This type of coding differs from part program code in that no part ID equivalent need be specified at the beginning of the data.

A single file on the off-line device can include the T, H, D, E, Q, and L tables. The Q Table is present when your machine has a random tool changer. The L Table is present when your control has the Tool Life Management feature.

The off-line format for the T, H, D, E, and Q tables is described below. The format for Tool Life Management requires more organization and is described as a separate topic (even though all four tables can be in the same file).

DATA FORMAT

Each line of Tool Code data must follow this format.

1) Every line must begin with the "@" character (ASCII 40 hexadecimal, SHIFT-2 on most keyboards).
2) The next characters must identify the table (T, H, D, E, Q, L) and code number that is addressed.
3) Next enter the data you wish to store at the identified Tool Table entry. The data consists of a single character followed by a number. Entries can be separated by one or more space characters.
4) End each line with a <LF> [9]. A <CR> is permitted before or after the <LF>. No spaces may appear within or between lines.

TABLE CODE	PARAMETER(S)	REPRESENTS
T		• Tool number (range; T01-T99)
	Z	- Tool length (inch ±3.4) (mm±4.3)
	D	- Cutter diameter (inch ±2.4) (mm±3.3)
H		• Tool Length offset (range: H01 up to H199)
	Z	- Tool length (inch ±3.4) (mm±4.3)
D		• Tool Diameter offset (range: D01 up to D199)
	X	- Tool length (inch ±3.4) (mm±4.3)
E		• 32 fixture offsets (E01-E32)
	X	- X axis offset from X0 (inch±3.4) (mm±4.3)
	Y	- Y axis offset from Y0 (inch±3.4) (mm±4.3)
	Z	- Z axis offset from Z0 (inch±3.4) (mm±4.3)
Q		• Tool pocket (range 1-?[10]; 99 max)
	T	- Tool number (range; T01-T99)

[10] Your machine supplier assigns the number of tool pockets, Q, corresponding to the number of pockets in your tool carousel.

CLEARING T, H, D, E, Q TABLES

The following data will erase the noted table when sent to the control from an off-line device.

Clear Tool Table	@T-1T/<LF>
Clear H Table	@H-1H/<LF>
Clear D Table	@D-1D/<LF>
Clear Fixture Offset Table	@E-1E/<LF>
Clear Random Tool Table	@Q-1Q/<LF>

The control edits the data into the Tool Tables after the Receive operation is complete. Only parameters that are received by the control will be stored in the Tool Tables. No values will be be erased unless they are overwritten. To illustrate this rule, assume the following line of code is sent to the control: @E02X23<LF><CR>. Notice that Y and Z are not programmed. When the control enters this line into the Fixture Offset table, any existing Y and Z values will not be overwritten. Therefore, the values in E02 will be a combination of the old values and the new values.

SAMPLE TOOL TABLES DATA

T, E, and Q Table codes may appear in any order. Your final line of code should be the "E" character (not @E) to signify the end of the file.

A sample tool table might appear as follows:

```
@T-1T/<LF><CR>                    - Clear entire T table (Z and D)
@E-1E/<LF><CR>                    - Clear entire E table (X, Y and Z)
@Q-1Q/<LF><CR>                    - Program when it is desirable to clear the Q table.
@T1 Z-1.2342 D.75<CR><LF>
@T2 Z-.3778 D1.5<CR><LF>
@T3 Z-2.1 D1<CR><LF>
@T4 Z2.25 D.5<LF><CR>
      --
      --
@E1 X1.25 Y.75 Z-.25<LF><CR>
@E2 X-4.85 Y-8.25 Z3<LF><CR>
      --
      --
@Q1 T1<CR><LF>
@Q3 T6<CR><LF>
@Q2 T9<LF><CR>
@Q14 T2<LF><CR>
      --
E
```

TOOL LIFE MANAGEMENT OFF-LINE FORMAT

Tool Life Management Table entries have the same format as T, H, D, E, and Q Table entries. Every line must begin with a '@' character and end with a <LF> character. Four different types of Tool Life Management Table commands can be sent from an off-line device. The four commands and their functions are described below.

IDENTIFY TABLE SIZE

Program **@Lx<LF>** to identify the TABLE SIZE for all following data. The L value must agree with the TABLE SIZE that is selected on the control (this setting is displayed every time you enter the Tool Life Management Table). If you do not wish to change the TABLE SIZE on the control, this command in not required.

CLEAR ALL VALUES IN A SPECIFIED GROUP

Program **@P-xxP/<LF>** to clear all of the entries in the group xx. This command will clear (set to zero) all remaining life and tool numbers in the specified group.

ENTER EXPECTED LIFE FOR ENTIRE GROUP

Program **@Pxx.00 M________or F________<LF>** to set the remaining life for group xx to number of minutes M, or frequency F. This command will not alter the tool numbers in the specified group. The R entry in the following command will be assumed to be in the units (M or F) entered here. For this reason, this group command should precede any line that addresses an individual member of the group.

ASSIGN TOOL NUMBER or REMAINING LIFE TO A GROUP MEMBER

Program **@Pxx.yy Tzz R________<LF>** to change the values of a member (yy) entry in group (xx). The value zz assigns the tool's Tool Table number, and R assigns the remaining life for the tool expressed in the units (minutes or frequency) which is currently selected for the entire group. If R or T is not programmed, the control will not overwrite any existing value.

A line by line example is as follows:

```
@L2              - confirms 32 groups of 8 tools each.
@P-1P/           - clears all values in group 1.
@P1.00 F100      - sets Frequency=100 for all tools in group 1.
@P1.01 T1 R80    - assigns a T1 offset and changes the Frequency to 80 for P1.01.
@P1.02 T2 R80    - assigns a T2 offset and changes the Frequency to 80 for P1.02.
@P1.03 T3        - assigns a T3 offset and keeps the Frequency at 100 for P1.03.
@P-2P/           - clears all values in group 2.
@P2.00 M16       - sets  Minutes=16 for all tools in group 2.
@P2.01 T20 R20   - assigns a T20 offset and changes the Minutes to 20 for P2.01.
@P2.02 T21 R25   - assigns a T21 offset and changes the Minutes to 25 for P2.02.
@P2.03 T22       - assigns a T22 offset and keeps the Minutes at 16 for P2.03.
E
```

TERMINAL MODE PROTOCOLS

A brief description of the proprietary Terminal mode protocol follows.

ESTABLISHING COMMUNICATIONS

Communications can be established by either the host computer or the CNC.

HOST ESTABLISHED

The host computer will initiate communications with the CNC by transmitting an ENQ (ASCII 5 hex) character followed by the ASCII code for the two-digit CNC ID. This string should be transmitted every .5 seconds for six seconds or until the CNC responds by:

1) Returning an ACK (ASCII 6 hex) character followed by the CNC ID number, in which case communications have been established. The CNC will display the status message HOST INITIATED COMMUNICATIONS and the host may continue.

2) Returning a NAK (ASCII 15 hex) character followed by the CNC ID, in which case the CNC is unable to communicate because it is in Program mode, plotting in Graphics mode, jogging in Jog mode, or in Auto or Single event mode with identical cycle and edit part IDs.

3) Not responding, in which case host access has been inhibited, the control is inoperative, or the CNC is transmitting or receiving from the second serial port. In this case the host must rely on the six second ENQ timeout to signal that communications cannot be established.

In either of the last two cases, the operator would be totally unaware that the host attempted to initiate communications.

When the host successfully establishes communications the CNC keyboard is locked. The ABORT COMM and PAGE soft keys are the only keys recognized by the control and can be used any time during the communication. Pressing ABORT COMM sends an EOT (ASCII 4 hex) character followed by the CNC ID number to the host, prematurely aborting communications.

CNC ESTABLISHED

The CNC will initiate communications with the host when:

1) The operator selects R and then presses the TRANSMIT PROGRAM , TRANSMIT ALL PROGS or TRANSMIT TOOLS soft key.
2) The operator selects L and then presses the RECEIVE PROGRAM or RECEIVE TOOLS soft key.
3) The operator selects M, types in a message, and presses the ENTER key.

In any of these cases, the CNC will initiate communications with the host by transmitting an ENQ followed by the CNC ID. It will repeat this every .5 seconds for six seconds or until the host returns an ACK followed by a matching CNC ID number. The CNC will ignore all transmissions from the host except ACK.

If the host doesn't respond with an ACK within six seconds the control will stop transmitting and display the status message CANNOT ESTABLISH COMMUNICATIONS.

Once the CNC has established communications, it will proceed to carry out the selected operation as described below.

DATA TRANSFER PROTOCOLS

Once communications have been initiated and established, the initiating machine will begin data transfer. All part program and Tool Table data transmitted from the host must be formatted as described earlier in this section. This formatted information is what is referred to by "data" in the examples that follow. All data transmitted from the CNC will follow the same format.

Both host and CNC will preface all commands with an ACK followed by the CNC ID number (from now on referred to as ACKid). The command will consist of a string of ASCII characters to describe what operation is to take place next (i.e. Transmit, Receive, etc.). The machine receiving this string will return an ACKid and the operation may then take place. Either machine may return a NAKid here or anytime it receives bad data from the other. If the CNC receives a NAKid at any time during its transmission, it will retransmit its command string and wait for the host to respond with an ACKid. Then it will retransmit the data. If it receives a second NAKid the CNC will abort with an EOTid. Likewise, no response from the host within ten seconds will cause the control to issue an EOTid. The EOTid string can be used to abort at any time by either the host or CNC.

The string containing the operation to be performed must start with an ACKid. Next the operation is specified by the ASCII code(s) for one or two characters from the list below.

Character(s)	Meaning
L1	Receive one or more part programs from selected host filename
L2	Receive tool data
L3	Receive macros from host
R1	Transmit selected part program
R2	Transmit all part programs
R3	Transmit tool data
R4	Transmit macro program to host
R5	Transmit all macro programs to host
C1	Activate part program ID (host to CNC)
C2	Delete part program ID (host to CNC)
C3	Activate a macro program (host to CNC)
C4	Delete macro program ID (host to CNC)
C	Activate filename (CNC to host)
M	Message

The letter describing an operation is followed by a number identifying the data type except in the case of a Filename[11] or Message. A carriage return (<Cr> is ASCII 0D hex) and line feed (<Lf> is ASCII 0A hex) must always follow the letter or letter-number sequence.

[11] Filenames, denoted by the character "C", are not followed by a number when sent from CNC to host; this simply activates a filename in the host's memory. When sent from host to CNC, however, the character "C" denotes a part program ID and may be followed by "1" to active the PPID or "2" to delete it.

A command string from CNC to host to receive part program data would look like this:

ACK01L1<Cr><Lf>

Broken down into its components:

Character	Meaning
ACK	Preface to all commands
01	ASCII character code for CNC#1
L	Receive the data specified by the next character from CNC to host
1	Part program data
<Cr>	Required by protocol
<Lf>	Required by protocol

A command string from host to CNC to transmit tool data would look like this:

ACK01R3<Cr><Lf>

Broken down into its components:

Character	Meaning
ACK	Preface to all commands
01	ASCII character code for CNC#1
R	Transmit data specified by the next character from CNC to host
3	Tool data
<Cr>	Required by protocol
<Lf>	Required by protocol

Note that the command characters L and R always mean the same thing no matter who's initiating: L always causes a Receive from the host to the CNC; R always causes a Transmit from the CNC to the host.

These command strings are only a fraction of the total transmission. Examples of complete transmissions, from ENQ (start) to EOT (break) will be presented later.

If Xon/Xoff control is enabled in Receive or Transmit mode, either machine can temporarily stop data transmission during a Receive or Transmit by sending a DC1 (ASCII 11 hex) character and restart transmission with a DC3 (ASCII 13 hex) character. The CNC uses this facility to avoid buffer overflow.

FILENAMES

The HOST FILENAME menu choice allows the CNC to specify a filename in the host's memory. The CNC may then do one of two things:

1) Transmit the currently active part program, all programs, or tool tables into the specified filename in the host's memory;

2) Receive the contents of the specified filename into the CNC's Catalog or Tool Tables. In this case, the incoming data would be formatted such that the part program ID(s) of the part program(s) it contained would be extracted by the control and used as PPIDs in the control's own catalog.

If no filename is specified in a CNC-initiated receive, it is up to the host to determine which (if any) file is sent. See your host computer software documentation.

Prior to a host-initiated transmit (where the host is taking a program from the CNC), the host may specify an up-to-eight character part program ID by using the character "C" in its command string. This **should not** be confused with the host filename as transmitted from the CNC; when the host uses the "C" command it is selecting a PPID from the CNC catalog. If the "C" is followed by a 1 the part program will be activated; if it's followed by a 2 it will be deleted. If no PPID is specified prior to the host-initiated transmit operation, the host will receive the currently active file.

When the host wants to perform more than one operation (i. e. send a PPID to the CNC, then transmit that part program from the CNC) it must perform them as separate communications. Therefore the following example would probably be followed immediately by a reinitiation of communications by the host which would include a string of commands to transmit a part program from the CNC (an example like this is given in the TRANSMITTING section). The program that would be transmitted is the one the host chose with the PPID transmission.
The following example shows the transmissions necessary for the host to activate a part program ID at CNC #01. The host could then send transmit instructions, for example, in a following transmission.

Characters	Meaning
ENQ01	Host requesting to establish communications with CNC# 1 (same command would be used if CNC was initiating).
ACK01	CNC acknowledge; link established.
ACK01C1<Cr><Lf>	Host acknowledge; select PPID to follow
ACK01	CNC acknowledge
:	:
Part Program ID	String of up to eight ASCII characters for PPID
:	:
ACK01	Host signals end of PPID
ACK01	CNC acknowledges end
EOT01	Host sends End of Transmission character.

For CNC-initiated communications this could be regarded as a "Send Filename" transmission and "C" instead of "C1" would be used. A CNC-initiated receive or transmit operation would then follow. An example of this appears in the Receiving section.

The host may delete a part program from the CNC catalog as shown below.

Characters	Meaning
ENQ01	Host requesting to establish communications with CNC# 1 (same command would be used if CNC was initiating).
ACK01	CNC acknowledge; link established.
ACK01C2<Cr><Lf>	Host acknowledge; host will send ID to be deleted
ACK01	CNC acknowledge
:	:
Part Program ID	(from host)
:	:
ACK01	Host signals end of PPID
ACK01	CNC acknowledges receipt, deletes program
EOT01	Host sends End of Transmission character.

It is not possible for the CNC to delete a filename from the host's catalog.

RECEIVING

For a CNC-initiated Receive of part program data from the host, the following interaction between CNC and host would take place (this example includes selection of the file to be received by the CNC). In order to produce the following communications, the CNC operator would select F, enter a host filename, then select L and press the RECEIVE PROGRAM soft key. At that point the CNC would initiate and, assuming no errors, the following exchange would take place:

Characters	Meaning
ENQ01	CNC #1 requesting to establish communications with host.
ACK01	Host acknowledge; link established.
ACK01C<Cr><Lf>	CNC acknowledge; CNC will send filename.
ACK01	Host acknowledge.
:	:
Filename	CNC sends ASCII filename characters
:	:
ACK01	CNC signals end of filename
ACK01	Host successfully received filename
ACK01L1<Cr><Lf>	CNC acknowledge; request host to Receive part program data identified by filename
ACK01	Host acknowledge
:	:
data	Host transmits part program data
:	:
ACK01	Host signals end of data
ACK01	CNC successfully received data
EOT01	CNC sends End of Transmission character.

The command strings are virtually identical for a **host-initiated load.**

Characters	Meaning
ENQ01	Host requesting to establish communications with CNC#1.
ACK01	CNC acknowledge; link established.
ACK01C1<Cr><Lf>	Host acknowledge; host will send PPID.
ACK01	CNC acknowledge.
:	:
Part Program ID	Host sends ASCII PPID characters
:	:
ACK01	Host signals end of PPID
ACK01	CNC successfully received PPID, will activate title.
EOT01	Host terminates communication
ENQ01	Host requests to establish communications with CNC#1.
ACK01	CNC acknowledge; link established.
ACK01L1<Cr><Lf>	Host acknowledge; host will send PPID data
ACK01	CNC acknowledge
:	:
data	Host sends data for PPID selected in last communication
:	:
ACK01	Host signals end of data
ACK01	CNC successfully received data
EOT01	Host sends End of Transmission character.

The host does not need to activate a PPID in the control before a download. In this case, however, if the host downloads a part program with the same PPID as the presently active[12] part program in the control, the control's version will be merged with the host's. However, if the host downloads a part program with the same PPID as a part program in the CNC catalog that is not active but presently stored, an OPERATION ERROR will occur. DUPLICATE ID ERROR will appear on the Communications Log screen.

[12] Or with concurrency, the concurrently active program.

TRANSMITTING

For a CNC-initiated transmit of tool table data into a CNC-specified file in the host memory, the following communications would take place:

Characters	Meaning
ENQ01	CNC #1 requesting to establish communications with host.
ACK01	Host acknowledge; link established.
ACK01C<Cr><Lf>	CNC acknowledge; select filename to follow.
ACK01	Host acknowledge.
:	:
:	:
Filename	String of ASCII character codes for filename
:	:
:	:
ACK01	CNC signals end of filename
ACK01	Host successfully received filename
ACK01R3<Cr><Lf>	CNC acknowledge; Transmit tool data to host.
ACK01	Host acknowledge
:	:
:	:
data	CNC transmits tool data
:	:
:	:
ACK01	CNC signals end of data
ACK01	Host successfully received data
EOT01	CNC sends End of Transmission character.

An example of a host-initiated Transmit follows. Notice that the host requires two separate communications--one to specify a part program ID and one to transmit that file. If the host wanted to transmit the whole catalog, the tool table or a program whose filename was currently active it would not need to specify a PPID. In any case, it is up to the host software to determine the file in which to put the data it is transmitting.

Characters	Meaning
ENQ01	Host requesting to establish communications with CNC#1.
ACK01	CNC acknowledge; link established.
ACK01C1<Cr><Lf>	Host acknowledge; host will send PPID.
ACK01	CNC acknowledge.
:	:
Part Program ID	Host sends ASCII PPID characters
:	:
ACK01	Host signals end of PPID
ACK01	CNC successfully received PPID, will activate title.
EOT01	Host terminates communication
ENQ01	Host requests to establish communications with CNC#1.
ACK01	CNC acknowledge; link established.
ACK01R1<Cr><Lf>	Host acknowledge; request CNC send PPID data
ACK01	CNC acknowledge
:	:
data	CNC sends data for PPID selected in last communication
:	:

ACK01	CNC signals end of data
ACK01	Host successfully received data
EOT01	Host sends End of Transmission character.

MESSAGES

Messages may be transmitted from host to CNC or vice-versa. They require no special formatting; once the receiving machine acknowledges that it is ready to capture, the sender should transmit the ASCII characters (up to 32) that make up the message and signal the end with an ACKid. When the host sends the CNC a message, it immediately appears on the Terminal mode primary screen under TERMINAL STATUS. If the CNC is not in terminal mode (but no lock-out conditions exist) when it receives a host-initiated message it will automatically switch to Terminal mode and the message will be displayed as usual. In either case, the message will be transmitted and noted as a Message on the Communications Log (Terminal mode page 2) screen. For a host-to-CNC message, the communication would look like this:

Characters	Meaning
ENQ01	Host requesting to establish communications with CNC#1.
ACK01	CNC acknowledge; link established.
ACK01M<Cr><Lf>	Host acknowledge; host will send Message.
ACK01	CNC acknowledge
:	:
data	Host transmits message--up to 32 ASCII-code characters.
:	:
ACK01	Host signals end of message
ACK01	CNC successfully received message
EOT01	Host sends End of Transmission character.

The following example exchange would take place (assuming no errors) if the CNC operator entered Terminal mode, selected M, typed in a message and pressed ENTER:

Characters	Meaning
ENQ01	CNC #1 requesting to establish communications with host.
ACK01	Host acknowledge; link established.
ACK01M<Cr><Lf>	CNC acknowledge; CNC will send Message.
ACK01	Host acknowledge
:	:
data	CNC transmits message--up to 32 ASCII-code characters.
:	:
ACK01	CNC signals end of message
ACK01	Host successfully received message
EOT01	CNC sends End of Transmission character.

MACRO MODE

If your control has Macro mode you will also find Terminal mode in the Macro mode menu. This will permit the transfer of Macro programs.