ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

User's Manual

Series B
8400 MP/Bandit IV
Revision F Firmware

Important User Information

Because of the variety of uses for the solid state equipment described herein, and because of the differences between it and electromechanical equipment, you must satisfy yourself as to its acceptability for each of your applications. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages that may result from the installation or use of this equipment.

The illustrations, charts, and layout examples shown in this manual are intended solely to help you understand the text, not to guarantee operation. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company will not assume responsibility for actual use based upon illustrations of applications.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment or software described in this text.

Table of Contents

Chapter	Section	Title	Page

Chapter	Section	Title	Page

Chapter	Section	Title	Page

Chapter	Section	Title	Page

Chapter	Section	Title	Page

List of Figures

Figure *Title* *Page*

List of Figures

Figure *Title* *Page*

List of Figures

Figure *Title* *Page*

List of Figures

Figure *Title* *Page*

List of Tables

List of Tables

Chapter 1 -- Using this Manual

1.0 Chapter Overview

This chapter tells you about this manual. In this chapter you'll learn:

- how the manual is organized and what information you'll find in it
- who we wrote this manual for and what background we expect you to have
- how we define certain key terms
- how to use to this manual

1.1 Manual Overview

We've written this manual to help you program and operate your control. Table 1A shows how this manual is organized.

Table 1.A
Manual Organization

Chapter	Title	Summary
1	Manual Overview	Manual overview, intended audience, definition of key terms, how to proceed.
2	Control Description	Location and function of the control's physical features.
3	Powering Up the Control	How to power up the control, power up conditions.
4	Page Selection	How the paging system and softkeys work; main menu page.
5	Manual Operate Page	How to use the manual operate page, machine home, jog hand-wheel, jog continuous, jog increment, and MDI.
6	Job Set Up Page	How to use the job setup page, inch metric, block delete, optional stop, mid program start, next program, tool and fixture offsets.

Chapter	Title	Summary
7	*Program Edit Page*	*How to create or edit a part program. Display and list edit, search, insert P, G codes, M codes, and directory.*
8	*Check Out Page*	*How to use quick check, feed check, dry run, dry Z run.*
9	*Auto Operate Page*	*How to use auto operate, jog in-auto, jog-and-return.*
10	*Graphics Page*	*How to use the graphics page. Erase, set window, window smaller, window larger, set limits and rapid.*
11	*Load/Save Page*	*How to use the load/save page, load, save, verify, multi-program load, multi-program save.*
12	*Support Page*	*How to use access control, passwords, logging on, AMP, control parameters, control fault parameters, data entry and programming, program protect.*
13	*Introduction to Programming*	*Structure and requirements of the programming language for your control.*
14	*Coordinate Control*	*G words that define how the control treats the coordinate system.*
15	*Positioning/Dwell*	*G words that define positioning and dwell data blocks.*
16	*Axes Modes*	*G words used to define axes modes.*
17	*Autocycles*	*G words used to define auto-cycles.*

1.2 Audience

This manual is intended for those who must program and/or operate the Bandit IV. We assume that you are familiar with the basic operation and programming of a CNC.

1.3 Terms We Use

To make this manual easier to read and understand, we try to avoid repeating full product and feature names where possible. Here are the names we shorten:

- Adjustable Machine Parameters -- AMP
- Cathode ray tube -- CRT (the monitor screen)
- Central Processor Unit -- CPU (the computing part of your control)
- Emergency Stop -- ESTOP
- Manual Data Input -- MDI
- Programmable Application Logic -- PAL
- Computerized Numerical Control -- the control.
- Random Access Memory -- RAM
- Eraseable Programmable Read Only Memory -- EPROM

Information that is especially important to note is called out by three names:

- **Important:** This is information that is important for successful application of the control.
- **Caution:** Cautions inform you of circumstances or practices that can lead to damage to the control or other equipment.
- **Warning:** Warnings inform you of circumstances or practices that can lead to personal injury as well as to damage to the control or the machine or other equipment.

1.4 Manual Design

This manual is divided into two major sections. Chapters 2 through 12 tell you how to operate the control. Chapters 13 through 21 describe how to program your control.

You'll find section headings in the left margin of each page. We include illustrations and examples that will help you program and operate the control.

We provide an appendix of error codes and operator messages. They are presented in alphabetical order for quick reference.

1.5 Chapter Summary

In this chapter we discussed how to use this manual. Chapter 2 explains the physical features of the control.

Chapter 2 – Control Description

2.0 Chapter Overview

This chapter tells you the location and function of the control's physical features. After reading this chapter, you will recognize the:

- pendant
- front panel
- control package
- axis drives
- axis motors
- drive enclosure
- machining options

2.1 Pendant Control

The pendant is the focal point for the use of the control. The pendant's front panel contains a 12 inch cathode ray tube (CRT), a full alphanumeric keyboard, and controls for machine operation.

The pendant can be mounted to the machine with a mounting arm at a location that is convenient for you to reach the front panel.

2.1.1 Front Panel and Softkeys

This section describes the front panel of the control as shown in figure 2.1.

CRT

The CRT allows you to visually monitor and control the functions of the control. The CRT displays various "pages" of information. You will use the main menu page to access all other pages. It is shown in figure 2.1

Softkeys

The vertical group of 9 keys, which have software defineable functions, is used to access a number of different functions, depending on the page currently being displayed.

Alphanumeric Keyboard

Use the alphanumeric keyboard to enter word addresses and numeric data for part programs, as well as other data for control operation.

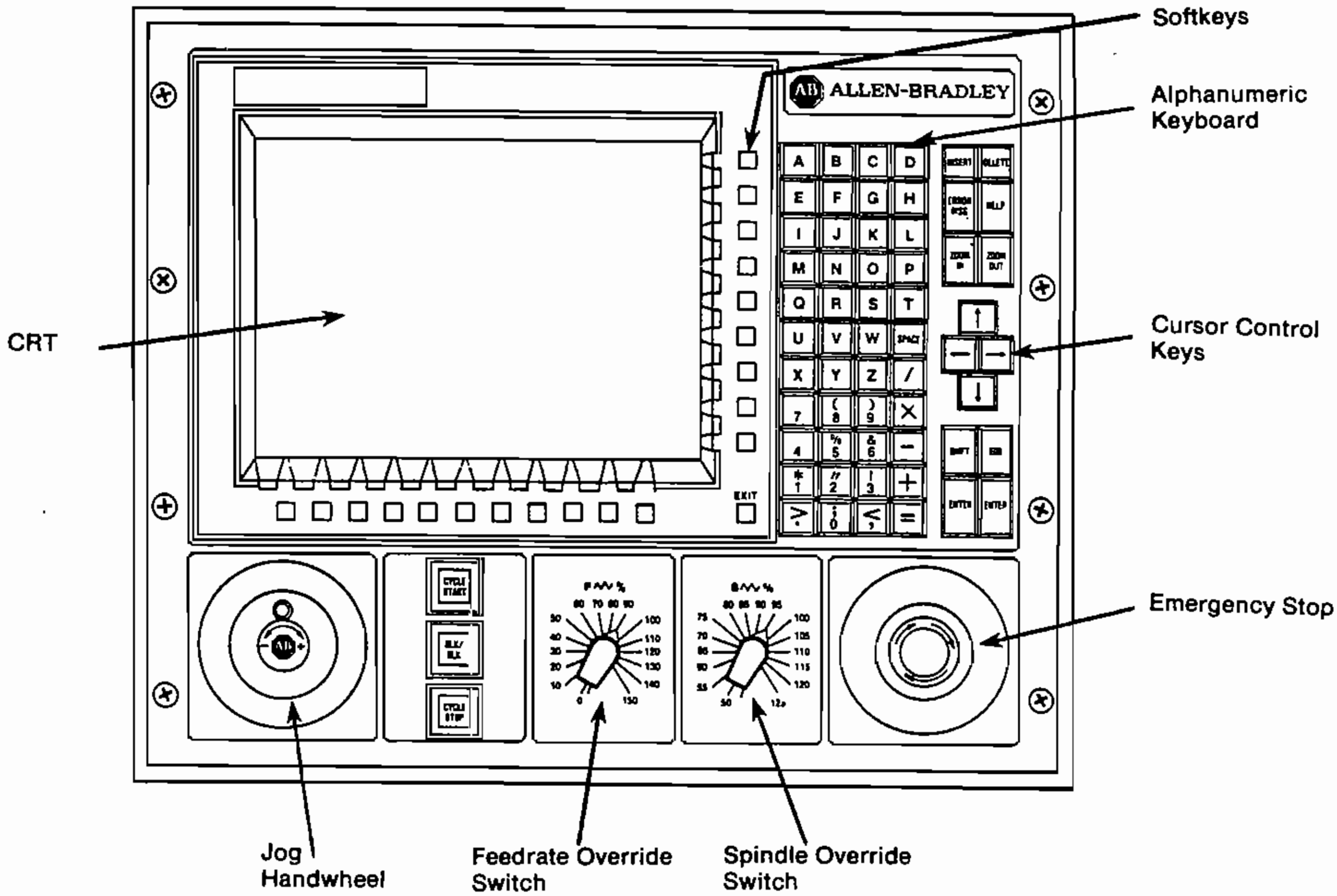
Insert and Delete Keys

Use the insert and delete keys during program editing. You can use them in conjunction with the cursor to insert or delete individual characters or complete data blocks.

Error Message and Help Key

The error message and help key are not implemented in this release. These keys are reserved for future use.

Figure 2.1 - Front Panel



Zoom In and Zoom Out Keys

Use the zoom in and zoom out keys to make the graphics window smaller and larger.

Cursor Control Keys

Certain pages have a moveable cursor to make data entry more convenient. The cursor control (arrow) keys move the cursor up, down, right, or left.

Press and hold a cursor control key to move the cursor continuously in a given direction. Press and release the cursor key to move the cursor one character or one line.

You can enter information by using the characters from the keyboard and then pressing the [ENTER] key.

Shift Key

The shift key is a special key that has no set function of its own. You must use the key with another key or button from the front panel to perform a certain function. Press and hold the [SHIFT] key and then press the other key or button on the front panel.

Here is a summary of SHIFT functions:

Pressing the [SHIFT] key and

- **letter**-- gives a lower case letter. Used mainly for nonmodal incremental axis moves.
- **numeric key** – puts a secondary character at the current location of the cursor.
- **cycle stop button** – aborts the current function of the control and restores several previous conditions.
- **cycle start button** -- bypasses the jog and return function.
- **exit key** -- generally will bring the control back to the main main menu page from anywhere in the softkey menu.
- **enter key** -- changes the mode of the axis position displays.
- **insert key** -- allows you to insert a new data block.
- **delete key** -- allows you to delete the entire current block. It also cancels the insert mode. Under prompt edit, it deletes an entire block. Under list edit it deletes everything from the cursor to the # character.
- **cursor up key** -- displays the data block just before the current data block. It also cancels insert mode. In prompt edit, the cursor up key will step you back to the previous page. When you are in list edit, it will jump to the start of the program.
- **cursor down key** -- allows you to display the next data block in prompt edit. In list edit, it will step to the last block in the program. It also cancels the insert mode.

EOB Key

You must terminate each data block in a part program with an end of block (#) character. An end of block is specified when you press this key. A new data block can begin following the # character.

Enter Key

Press the enter key after you enter a data block or a value.

Exit Key

Use the [EXIT] key to move "or exit" from a page that is currently displayed on the screen.

If you press the [EXIT] key once, the previous page is displayed.

Each time you press the [EXIT] key, the control takes you back one level until you reach the main menu page. If the [SHIFT] key is held down and the [EXIT] key is pressed the control will return to the Main Menu Page.

Emergency Stop Button

When you press the emergency stop button, the control enters the ESTOP state; all power to the axis and spindle drives is removed (the drives enable relay opens) and any movement of the axes and spindle stops. A flashing "EMERGENCY STOP" message appears on the screen.

To re-activate the control, turn the knob in a clockwise direction until it pops out. In most cases, the CRT then displays the page that was on the screen when you entered emergency stop.

Spindle Speed Switch

The spindle speed switch affects the programmed spindle speed. You can modify programmed spindle speed in 5% increments from 50% to 125%.

Programs should be written assuming a 100% setting. Spindle speed override is inhibited during a G84 tapping cycle.

Feedrate Override Switch

You can modify the programmed rate of moves by adjusting the feedrate override switch. The switch is graduated in 10% increments from 0% (FEEDHOLD) to 150%.

Programs should be written assuming a 100% setting. Feedrate override is inhibited during a G84 tapping cycle.

Cycle Start Button

When you press the [CYCLE START] button, it initiates continuous execution of a program or other function.

If this green button is lit, [CYCLE START] is active and the function is running continuously. If the green light and red light are on, the control is halted.

Block-by-Block Cycle Button

Each time you press the [BLK/BLK] cycle button it executes a single program data block or a portion of a single block.

If you press the [BLK/BLK] button repeatedly, you can run the part program one block at a time. If the yellow light is lit, the control is executing a single part program block or a portion of an autocycle or autoroutine.

Cycle Stop Button

Press the [CYCLE STOP] button to halt the execution of a program. When pressed, the axes will decelerate to a stop. When the red button is lit, the cycle is halted.

Press the [CYCLE START] or [BLK/BLK] button to re-activate the cycle.

This button is disabled during a G84 tapping cycle. A G62 program command can inhibit this button and the feedrate override and/or spindle speed override controls.

Jog Handwheel

Use the jog handwheel to move all or any combination of axes in any direction.

The axis moves in the +- direction depending on which direction you rotate the handwheel. Incremental movement of the handwheel will move an axis a given incremental amount depending on the setting of the feedrate override switch.

2.2 Control Package

The control package is the principal operating component of the system. It contains the CPU and CRT modules that monitor and direct virtually all CNC functions.

The CPU module processes system information and directs axis movements.

The CRT module controls the CRT and monitors the machine.

For more information on the control package, refer to the installation manual.

2.3 Chapter Summary

In this chapter we discussed the location and function of the control's physical features. In the next chapter we discuss how to power-up the control.

Chapter 3 -- Powering Up the System

3.0 Chapter Overview

This chapter discusses:

- how to power up the control.
- how to exit standby.
- how to log on to the control.
- the conditions automatically assumed by the control at power up.
- the power-up conditions of the control.

3.1 Incoming System Power

See the Series B 8400 MP Installation Manual, publication no. 8420-4.1.3.

WARNING: To avoid potential harm to personnel and equipment, turn the incoming AC off if you open the drive enclosure. Refer all service problems to qualified service personnel.

3.2 Turn Power On

To power up the control:

1. Push the emergency stop button into the emergency stop position.
2. Make sure that all peripheral devices, such as the tool changer, are OFF.
3. Turn the AC power ON. Remember, the power switch is not located on the pendant.

When you turn the power ON, the control performs self diagnostic tests and checks its memory. A message, "CHECKING MEMORY" appears in small white on black characters near the top of the screen.

The "CHECKING MEMORY" message will appear on the screen for 2 to 6 seconds depending of the user memory size. Then, the screen will go blank, the red [CYCLE STOP] button will come on, and the control will assume the standby mode.

3.3 Standby

In standby, these conditions exist:

- drive power is OFF
- screen is blank
- green and yellow buttons are OFF
- red [CYCLE STOP] button is ON

To remove the standby condition, press [EXIT].

3.4 Log On Screen

After you press [EXIT], a log on screen appears unless a user is defined with an **** password. (See Access Control in section 12.2)

You will see "ACCESS CODE" with **** beneath the message.

Figure 3.1 shows the log on screen as it appears on the screen.

Figure 3.1 - Log on Screen

```
 ________________________________________
|                               STANDBY |  *
|                                       |  -
|                                       |  -
|                                       |  -
|             ACCESS CODE:              |  -
|               >****                   |  -
|                                       |  -
|                                       |  -
|_______________________________________|  -
```

3.4.1 Gaining Access to the Control

You must gain access to the control by typing in the user code that was assigned to you by the system installer. Up to eight users can be assigned an individual code for each machine.

1. Type in your assigned code.

2. Press [ENTER].

If the password you type in does not match any of those in the control, you will not be able to gain access to the control. The access page will remain on the screen.

3.5 Emergency Stop

After you log on with your valid password, and you have pressed in the [EMERGENCY STOP] button, "EMERGENCY STOP" appears on the screen. If you haven't pressed the [EMERGENCY STOP] button, the message, "DRIVES NOT ON" appears on the screen. In either case, press [EMERGENCY STOP] and then release it to turn the drives on.

Figure 3.2 shows the main menu screen as it appears on the CRT. "Emergency Stop" flashes in the lower left corner of the screen.

Figure 3.2 - Main Menu Screen

```
| < operator messages>     STANDBY | *
|                   MANUAL OPERATE | *
|                     PROGRAM EDIT | *
|  ALLEN-BRADLEY         LOAD/SAVE | *
|   REV F1-003           JOB SETUP | *
|    GRAPHICS            CHECK OUT | *
|  MILL CONTROL       AUTO OPERATE | *
|                                  | -
| <  error messages  >     SUPPORT | *
```

3.5.1 Removing Emergency Stop

To remove emergency stop, rotate the emergency stop button clockwise until it pops out.

You can press the [EMERGENCY STOP] button at any time to stop unexpected or hazardous movement.

3.6 Main Menu Page

The main menu page, shown in figure 3.2, should appear on the screen after power up. The main menu page is the main page from which other pages and various functions are accessed. These pages and functions are described in section 4.2.

3.7 Power-up Conditions

When you start up the control, or display the main menu page, there are a number of conditions assumed by the control. These are listed below:

- All axes will be in absolute
- G01, Linear Interpolation is assumed
- Circular Interpolation plane is returned to XY
- Cutter Compensation cancelled
- Fixture and tool offsets are cancelled (shown as 0)
- Tool selection is shown as 0
- Override inhibits are cancelled
- Feedrate will default to the value specified in AMP
- Mirrors are turned off
- Inch/Metric is **not** modified; the inch/metric mode will be as shown on the job setup page.
- Program zero is **not** modified.

3.8 Chapter Summary

In this chapter we discussed how to power up the control. Chapter 4 tells you how to use the main menu page and the softkeys.

Chapter 4 -- Page Selection

4.0 Chapter Overview

After reading this chapter, you will know how to:

- use the main menu page to access other pages and functions of the control
- use the softkeys

4.1 Pages And Softkeys of The Control

The control is equipped with a variety of functions that you can access by using the softkeys..

A "page" that appears on the screen when you press a softkey lets you access different functions. The functions appear as a "menu". These softkeys give you access to the most likely functions while using the control.

Refer to figure 2.1 which shows the screen and the vertical row of softkeys to the right of the screen.

Each softkey addresses a function on a currently displayed screen. To select a particular function, press the key that corresponds to that function.

Throughout this manual we tell you to press a certain softkey. We put the current label of the softkey in brackets. For example, suppose the main menu page is displayed on the screen and you want to edit a program. You would press the [PROGRAM EDIT] softkey to gain access to the program edit page.

4.2 Main Menu Page

The main menu page serves as a "home base" from which you can access all other pages. The other pages available to you, through the main menu page, will be explained in later chapters.

Figure 4.1 shows the main menu page as it appears on the screen.

This page appears after you have correctly logged on from standby. You can return to this page at anytime by:

- Pressing [EXIT] repeatedly until it appears on the screen or
- Pressing [SHIFT] and [EXIT] at the same time.

Figure 4.1 - Main Menu Page

```
| < operator messages>      STANDBY | *
|                   MANUAL OPERATE | *
|                     PROGRAM EDIT | *
|  ALLEN-BRADLEY         LOAD/SAVE | *
|   REV F1-003           JOB SETUP | *
|    GRAPHICS            CHECK OUT | *
|  MILL CONTROL      AUTO OPERATE  | *
|                                  | -
| <  error messages  >     SUPPORT | *
```

The main menu page softkeys are listed below. The functions are described in detail in the following chapters.

- **[Standby]** -- powers down the drives and enters the standby mode.

  When you select standby, the screen will go blank and the red [CYCLE HALT] button lights.

  Press [EXIT] to come out of standby.

- **[Manual Operate]** -- accesses the manual operate page.

  Manual operate allows you to "manually" move the axes using jog, manual data input (MDI), or machine home. These operations prepare the machine to perform a particular job.

  We discuss the manual operate page in detail in chapter 5.

- **[Program Edit]** -- accesses the program edit page.

  You can use program edit to manage part program memory, to select an active program (for auto operate or check out, or to enter or edit a program from the keyboard.

  We discuss the program edit page in detail in chapter 7.

- **[Load/Save]** -- accesses the load/save page.

  Load/Save lets you load (input) and save (output) programs and control parameters using an external data storage device such a Kaypro II computer.

  We discuss the load/save page in Chapter 11.

- **[Job Setup]** – calls up the job setup page.

  Job setup establishes machine and control conditions for running a part program.

  We discuss the job setup page in chapter 6.

- **[Check Out]** – calls up the check-out page.

  Check out tests a program before you put it into production.

  We discuss the check out function in chapter 8.

- **[Auto Operate]** – accesses the auto operate page.

  Auto operate allows automatic execution of part programs.

  We discuss auto operate more in chapter 9 and 10.

- **[Support Page]** -- accesses the support page.

  The support page gains access to the features of the control that are customized to your machine.

  We discuss the support page in chapter 12.

4.3 Chapter Summary

In this chapter we discussed how to select a page and how to use the softkey functions. Chapter 5 tells you how to use the manual operate functions.

Chapter 5 -- Manual Operate Page

5.0 Chapter Overview

After you read this chapter you will know about:

- Machine Home
- Jog Handwheel
- Jog Continuous
- Jog Incremental
- MDI

5.1 Manual Operate Page

Press the [MANUAL OPERATE] softkey on the main menu page to call up the manual operate page on the screen.

Manual operate allows you to "manually" move the axes and perform the other functions by using jog and Manual Data Input (MDI), or you can home the machine by using machine home. These operations prepare the machine to perform a particular job.

When the screen appears, you will have five options available to you. You can gain access to these options by pressing the corresponding softkey. These options are explained in the following sections.

Figure 5.1 shows the manual operate page as it appears on the screen.

Figure 5.1 - Manual Operate Page

```
| MANUAL OPERATE      JOG:HANDWHEEL | *
| < operator messages>  CONTINUOUS | *
|                      INCREMENTAL | *
|                                  | -
|                                  | -
|                              MDI | *
|                                  | -
|                                  | -
| <  error messages  >MACHINE HOME | *
```

5.2 Machine Home

Press the [MACHINE HOME] softkey to access the machine home page.

Press [START] to begin the operation. Each axis is homed individually in a sequence established in control parameters. The rate of homing varies through the process. Control parameters can also establish whether machine home is required for a specific axis.

An * will appear on the screen in reverse video, after the axis letter of each axis after it is homed.

Figure 5.2 shows the machine home page as it appears on the screen.

Figure 5.2 - Machine Home Page

```
| MACHINE HOME-IDLE                    | -
| < operator messages>                 | -
|                          0.0000 U*   | -
|                          0.0000 V*   | -
|                         -0.0001 W*   | -
|           MACH ZERO:     0.0000 X*   | -
|                          0.0002 Y*   | -
| Rate=     20.0          -0.0006 Z*   | -
| <  error messages  >                 | -
```

You must perform a machine home when power is initially applied to the control. This allows the control to establish its position reference, which is accomplished by moving the machine's axes to switch-set "home" locations.

There are certain operations which require a machine home before they can function. These are:

- MDI (from Manual Operate)
- Dry Run
- Auto Operate

If you need to perform a machine home, the error message "MACHINE HOME REQUIRED" appears on the screen and you will not be allowed to continue operation.

The control maintains an absolute position reference from zero point designated as machine zero (absolute XO, YO, ZO). The system installer establishes this point during installation. The "home" switches are located along the axes.

During a machine home, each axis proceeds until the home switch is detected, it reverses direction to come off the switch and find an encoder marker, and then reverses again to stop on the marker edge.

Once you have performed a machine home, it will stay in effect indefinitely unless power is turned off or certain motion errors occur.

5.3 Jog Handwheel

Press the [JOG HANDWHEEL] softkey on the manual operate page, to move a given machine axis using the jog handwheel on the front panel.

Figure 5.3 shows the jog handwheel page as it appears on the screen.

Figure 5.3 - Jog Handwheel Page

```
| JOG HANDWHEEL MACH ZERO:      AND | *
| IDLE                        MINUS | *
| < operator messages>   0.0000 U+ | *
|                        0.0000 V+ | *
|                       -0.0001 W+ | *
|                        0.0000 X+ | *
|                        0.0002 Y+ | *
|  Resolution 0.0020    -0.0006 Z+ | *
| <  error messages  >              | -
```

CAUTION: When the handwheel is being used while the feedrate override switch is set to 10%, 20%, or 30%, and the control is configured for 'coarse' resolution, small axis movements can be made that do not appear on the axis displays. This occurs because the handwheel move increment (as controlled by the feedrate override switch setting) is smaller than the display resolution.

For example, if your control is configured for:

```
LDEC = 4                    resolution = coarse
feedrate override = 30%     active mode = Handwheel Jog
```

then the control will display 0.000 for linear axes, but the axis will move 0.0005 with each pulse of the handwheel. Every pulse from the handwheel will result in a move, but only every other move will result in a change in the axis display.

This same situation, but with different amounts of motion, will result if LDEC is changed to 3, 4 or 5, or if the feedrate override switch is 10%, 20% or 30%.

Important: The handwheel relies on a changing magnetic field that creates pulses as it is rotated. If it is rotated very slowly, the amplitude of the pulses will be reduced, and the axis may not move.

The axis will move in the + or - direction, depending on which direction you rotate the handwheel.

5.3.1 Select a Single Axis

To select a single axis, press the desired axis. For example, press the [X] softkey for the X axis. When you access the jog handwheel screen, none of the softkeys will appear in reverse video. However, when you press an axis softkey, that axis appears in reverse video. This indicates that you have selected this particular axis for handwheel jog.

Press the desired axis to change the axis selection. The axis chosen previously will de-select and will not appear in reverse video.

5.3.2 Select More Than One Axis

To select two or more axes, complete the following steps:

1. Select a positive axis. (The crank will run in either direction; positive sense means plus direction for clockwise crank rotation.)
2. Press AND. AND appears in reverse video.
3. If the next axis is to go in the minus direction (for counterclockwise handwheel rotation), press [MINUS].
4. Press the other axis letter. This axis appears in reverse video which indicates that it was selected. The appropriate sign will follow your selection.

Repeat steps 2-4 to select additional axes.

The handwheel is "detented" so that an incremental movement of the handwheel will move the axis a certain amount, depending on the setting of the feedrate override switch and the rate set in the adjustable machine parameters (AMP).

The incremental movement of the handwheel is displayed on the screen as resolution.

To change back to a positive axis, press [AND] then select the axis you want to change. Or hit the softkey next to the selected axis this will change it from - to +.

5.4 Jog Continuous

Press the [JOG CONTINUOUS] softkey on the manual operate page to call up jog continuous.

WARNING: To avoid potential harm to personnel and equipment, remember that pressing a softkey while this page is displayed will move the axes of the machine. The message, "SOFTKEYS MAY MOVE AXES" is shown blinking on this page.

As long as you press the + or - softkey, the axes will move in a + or - direction "continually".

You will also see axes on the right side of the screen. The first axes you will see is: UVW. When you press the [UVW] softkey a similar screen, with the U,V,W axes appears on the screen. Note, however, that the system installer must have assigned these axes for your particular control. If the installer didn't assign them, they are not enabled. Figure 5.4 shows the jog continuous page as it appears on the screen.

When Jog motion is requested (from a soft-key or from PAL), the axis will move an increment, then delay, then jog continuous until the request is removed. The length of the increment is displayed in the 'Initial Move' field. The Feedrate Override input controls the initial move amount. The number displayed in the field represents units of motion resolution, '100' represents .01 inches or .1 degree for example. If the number displayed is zero, no increment is moved, no delay takes place, and the axis jogs continuously as soon as the jog motion is requested.

Table 5.A - Feedrate Override effect on Initial Move Distance

Feedrate Override %	Initial Move Distance
0	0
10	1
20	2
30	5
40	10
50	20
60	50
70	100
80	200
90	500
100	1000
110	0
120	0
130	0
140	0
150	0

Figure 5.4 - Jog Continuous Page

```
| JOG CONTINUOUS MACH ZERO:     UVW | *
| IDLE     SOFTKEYS MAY MOVE AXES   | -
| < operator messages>   0.0000 X+ | *
|                                X- | *
|                        0.0002 Y+ | *
|                                Y- | *
| Initial Move     2    -0.0006 Z+ | *
| IPM            20.0            Z- | *
| <  error messages  >              | -
```

5.5 Jog Incremental

Press the [JOG INCREMENTAL] softkey on the manual operate page to activate jog incremental.

Jog incremental allows you to specify an increment of axis movement, and apply this increment to any direction of the axes.

Figure 5.5 shows the jog incremental page as it appears on the screen.

Figure 5.5 - Jog Incremental Page

```
| JOG INCREMENTAL MACH ZERO:    UVW | *
| IDLE     SOFTKEYS MAY MOVE AXES   | -
| < operator messages>   0.0000 X+ | *
|                                X- | *
|                        0.0002 Y+ | *
|                                Y- | *
|  Rate=        25.0    -0.0006 Z+ | *
| >Inc.=     0.0000              Z- | *
| <  error messages  >              | -
```

When the jog incremental page is displayed, "Inc." appears in the lower left portion of the screen, immediately followed by the cursor. The axes is on the right side of the screen. The first axes you will see is: UVW. When you press the [UVW] softkey a similar screen, with the U,V,W axes appears on the screen. Note, however, that the system installer must have assigned these axes for your particular control. If the installer didn't assign them, they are not enabled.

The jog incremental rate appears on this page just above the increment. Its value is based on the position of the feedrate override switch.

Table 5.B gives jog rates and the upper limit on the machine home rates vs. the feedrate override setting.

Table 5.B - Feedrate Override effect on Jog Rates

SWITCH POSITION	RATE
150%	Rapid
140%	93% Rapid
130%	87% Rapid
120%	80% Rapid
110%	73% Rapid
100%	67% Rapid
90%	60% Rapid
80%	53% Rapid
70%	47% Rapid
60%	40% Rapid
50%	33% Rapid
40%	27% Rapid
30%	20% Rapid
20%	13% Rapid OR 5 ipm, whichever is least.
10%	.5 ipm (minimum jog rate)
0%	---

To use jog increment:

1. Type a numeric value at the cursor. The increment appears in reverse video in the "Increment" space.

2. Press [ENTER] to activate the increment you specified. This increment remains until you change it.

If you type in a wrong increment, but you have not yet pressed [ENTER], correct the entry by:

- pressing the [DELETE] key to remove the last entered character

OR

- pressing one of the cursor keys to restore the previous increment.

After setting an increment, an axis can be moved by pressing the + or - softkey. The axis starts moving as soon as the key is pressed and will continue moving until it moves the increment amount, or you press the [CYCLE STOP] or [E-STOP] button.

Pressing the [E-STOP] button will abort the the incremental move. You cannot resume an aborted jog move.

Pressing the [CYCLE STOP] key stops the move. Resume the the stopped jog move by pressing [CYCLE START]. Abort the stopped jog move by pressing and holding the [SHIFT] key and pressing [CYCLE STOP].

Axis motion or exiting the incremental jog mode will be inhibited until one of these actions is performed.

5.6 How to Change the Axis Position Display

A display of axis position is shown next to each set of + or - softkeys. The display usually shows the axis positions as "PROG ZERO"; however, you can change the function of the display by completing the following steps:

1. Press and hold the [SHIFT] key while you press the [ENTER] key.
2. Select one of the following axis position displays. Each time you press [ENTER], the display changes.
   - **Machine Zero** – Machine zero is the position of the axes as referenced to machine zero.
   - **Lag Error** – Lag error is the "following" error developed by the system due to velocity control; the amount that actual position lags commanded position. It is useful in system installation.
   - **Program Zero** – Program zero is the position of the axes as referenced to an established zero coordinate within a program.
   - **Distance to Go** - Distance to go is the distance needed to reach the final destination of the axis; It is always a positive number that counts down to zero.

Important: The distance to go display does not apply to the jog continuous or jog handwheel modes. You can select it for these modes, however, it is meaningless and will stay at zero. It has significance only when used on the jog incremental page.

5.7 MDI

Press the [MDI] softkey on the manual operate page to access Manual Data Input (MDI). MDI allows you to enter a desired data block for immediate execution by the control.

This screen is similar to the prompt edit page; all block editing features apply.

Figure 5.6 shows the MDI page as it appears on the screen.

Figure 5.6 - MDI Page

```
| MDI                     IDLE    < operator messages>    GRAPHICS | *
|                                 <  error messages  >             |
| POCKET MILL--- G26              ABS/inc XYZWVUIJKABCR            | -
| LENGTH--------_X1.256                                            |
| WIDTH---------_Y2.489                                            | -
| DEPTH---------_Z-2.6                                             |
| X FINISH CUT--I                                                  | -
| Y FINISH CUT--J                                                  |
| Z FINISH CUT--K                                           STATUS | *
| EXIT RADIUS---Q                                                  |
| Z ROUGH CUT---_U.2                                               | -
| NO. OF Z CUTS-_L5                                                |
| XY ROUGH CUT--W                                          G CODES | *
| NO OF XY CUTS-D                                                  |
| ROUGH RATE----F                                          M CODES | *
| FINISH RATE---H                                                  |
|                     DOWN ARROW FOR MORE PROMPTS        DIRECTORY | *
|                                                                  |
```

5.7.1 How to Enter and Execute an MDI Block

To enter and execute a block in MDI:

1. Make sure that you have homed the machine and that the control is not in emergency stop.
2. Enter a single data block.
3. Press the green [CYCLE START] button or the yellow [BLK/BLK] button.

For example, to establish the incremental mode and move X-2. at 10 ipm:

1. Type in [G91].
2. Press [CYCLE START]
3. Enter G01 X-2. F10. to move 2 inches in the negative X axis direction at 10 ipm.

Before you press [CYCLE START], the control displays the "idle" message. When you press [CYCLE START], the control displays "running" until the command is completed.

4. Press [CYCLE START] to execute the block.

If the control detects an error in coding, the data block, along with an error message,"EXIT TO CLEAR ERROR", appears.

When this occurs, you should check the data block for erroneous coding. Press the [EXIT] key to remove the error message.

Press [CYCLE START] to repeat an MDI block that you have entered. If you enter another block, the currently displayed block will be replaced.

When you exit the MDI function, modal conditions for the main menu page are restored. Refer to section 3.7 for a list of modal conditions.

For example, assume you are in the incremental mode in MDI, and you return to the main menu page. If you re-enter the MDI mode, the control is now in the absolute mode; you will have to re-establish the incremental mode.

5.7.2 Options Available in MDI

Notice the options available through the MDI page. When you need to use any of these operations, press the softkey that corresponds to the command.

- **[Graphics]** -- accesses the graphics capabilities of the control.

  You can execute MDI blocks on the graphics page, but you must enter the blocks while you are on the MDI page.

  There are many other graphic options. Refer to Chapter 10 for more details about using graphics in your operation.

- **[Status]** -- calls up the status page. This page informs you, at any time, of the status of any operation currently being performed by the control.

  You can execute MDI blocks on the status page, but you must enter the blocks while you are on the MDI page.

- **[G Codes]** -- access a list of G codes used with this control.

  You can use this page for reference, or you can type in a G code number and press [ENTER] to enter the G code for the block. We discuss the G code page in detail in Chapter 7.

- **[M Codes]** -- access a list of standard M codes that may be used with this control.

  You can use this page for reference, or you can enter an M code by typing a code and pressing [ENTER].

  We discuss the M code page in detail in Chapter 7.

- **[Directory]** -- accesses a list of programs currently stored in memory.

  You can use this page for reference, or you can type in the program number you want to access and press [ENTER].

  We discuss the directory page in detail in Chapter 7.

- **ABS/inc** -- Displays the current status of the word address fields. Either Absolute(upper case) or Incremental (lower case).

5.8 Chapter Summary

In this chapter we discussed the manual operate functions. In the next chapter we discuss how to use the job set-up functions.

Chapter 6 -- Job Setup Page

6.0 Chapter Overview

In this chapter we discuss the job setup page. After reading this chapter, you will know how to:

- change between inch and metric modes.
- use the block delete function.
- use the optional stop function.
- use the tool offsets page.
- use the fixture offsets page.
- initialize mid-program start.

6.1 Job Setup

The functions associated with the job setup page prepare the control for part production.

After you press the [JOB SETUP] softkey on the main menu page, you will see other programming options. These options are explained in the following sections.

Figure 6.1 shows the job setup page as it appears on the screen.

Figure 6.1 - Job Setup Page

```
| JOB SETUP              INCH METRIC | *
| < operator messages>BLOCK DELETE   | *
|  COMMENTX_DIM       OPTIONAL STOP  | *
|  Program      153       NEXT PROG  | *
|  Line #         1  MID PROG START  | *
|  Tool Offset    0    TOOL OFFSETS  | *
|  Fixture        2 FIXTURE OFFSETS  | *
|  Last Fixture 12           STATUS  | *
| <  error messages  >    DIRECTORY  | *
```

While you are on the job setup page, you can enter the program number of a program you want to execute.

Program 0 is for external input. If you select program 0, the program to be executed will come from the I/O port using the currently active load device as selected on the load/save page.

6.2 Inch/Metric

Press the [INCH METRIC] softkey on the job setup page to change between inch and metric programming units. The mode currently in effect will appear in reverse video.

You can also change modes through the part program using a G70 (inch mode active) or a G71 (metric mode active).

When you change modes, all dimensional parameters, such as axis displays, feedrates, and offsets are displayed in terms of the chosen mode.

Carefully enter values according to the current mode. For example, if the current mode is metric, you must enter tool offsets with metric values. Note the tool offsets and fixture offsets do not convert.

6.3 Block Delete

Press the [BLOCK DELETE] softkey on the job setup page to activate the block delete function. When the block delete function is active, it appears in reverse video.

When block delete is active, the control will ignore any data in a programmed data block that follows a slash (/).

Therefore, if you have used the slash (/) in the first position of any block in a program, and you do not want this block executed, you can press the [BLOCK DELETE] softkey to activate block delete.

You can ignore certain parts of a block by placing the slash in front of the data to be ignored. For example programming G01 X2./Z5. F10# will execute normally if block delete is not active. If the block is executed with block delete active, the Z move will not be made and the X move will execute at a previously declared feedrate instead of the 10. feedrate that follows the slash (/).

6.4 Optional Stop

You can press the [OPTIONAL STOP] softkey on the job setup page to allow the control to respond to an M01 optional program stop command. When you select this key, "OPTIONAL STOP" appears in reverse video on the screen. With optional stop active, an M01 command in a program will have the same effect as an M00 program stop.

With an M00, the control automatically stops program execution until you press the [CYCLE START] or [BLK/BLK] button. The message "HALTED" and "AWAITING START CYCLE" appears on the screen and axes are held in position until you press the [CYCLE START] button to resume program execution.

A programmed M01 or M00 can cause machine functions, such as turning the coolant off; however, any action at this point depends on what is programmed in Programmable Applications Logic (PAL). PAL is the program that controls the miscellaneous functions of the machine. Standard PAL consists of the codes listed on the M code page plus a few general purpose codes. A custom PAL can be provided to implement specific M codes for specific functions.

If an M01 is in a program, but you do not select optional stop, the M01 code is ignored by the control. M00 and M01 are not active during quick check.

6.5 Tool Offsets

Tool offsets allow the programmer to generate the part program without concern over the exact tool length and diameter. Using an O code, the programmer can specify the number of a specific offset in a table of up to 48 that can be entered into the control. This offset will generally correspond to a tool currently selected by a T code.

The machine operator will make entries in the tool offset table that define the diameter and length offset for each tool.

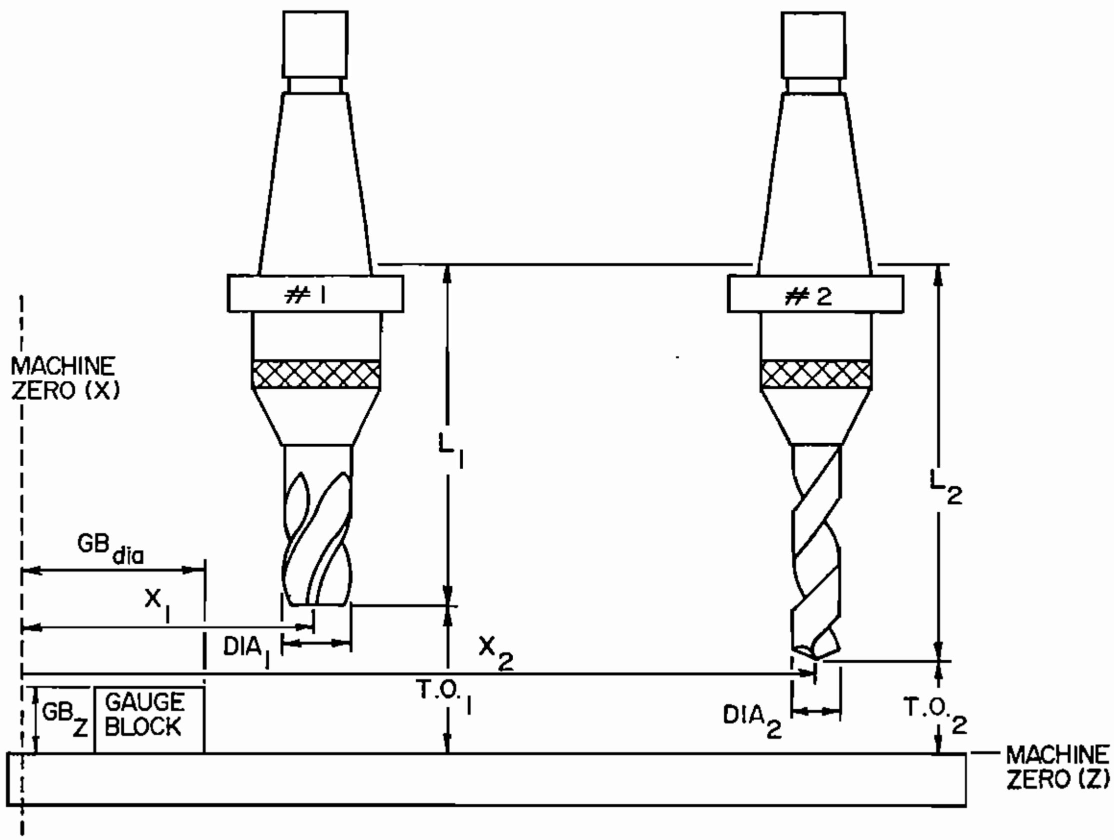
The control will adjust the programmed part path to compensate for entered tool dimensions.

The actual tool offset move will be executed on the next Z axis command (there is no motion because of the O code specifying the tool offset itself). The entire move is made at the rate programmed for that Z axis move. When the offset is added to the move, the result is executed at the current rate in effect.

Similarly, the O code does not initiate motion in the XY plane to compensate for the diameter. The compensation is incorporated when cutter radius compensation is invoked.

Figure 6.2 shows typical tools and tool holders and the dimensions associated with them. The dimensions are also explained below.

Figure 6.2 - Tools and Tool Holder Dimensions



Dimension "L." is the tool length, the distance from the gauge line on the holder taper to the tip of the tool.

Dimension "T.O." is the tool length offset, the distance from the tip of the tool to the machine zero in the Z plane. Note that the location of the machine zero plane is arbitrary and can be chosen to coincide with program zero, making the G92 preset for Z able to be set to 0.

Dimension "G.B." is the distance from the positive facing surface of an optional gauge block for digitizing to the machine zero plane.

The offsets may be manually determined or they may be digitized as described in section 6.5.3. Once each tool's length offset has been determined, it is entered at a corresponding location on the tool offsets pages. If cutter radius compensation is programmed, then each tool's <u>diameter</u> should be entered at the same O location.

Caution: To avoid a rapid movement of the tool into the part, use care in establishing tool length offsets and reference planes. Always test programs thoroughly with quick check and feed check.

A particular set of length and diameter values can be applied in a program by specifying a data block containing an appropriate command.

If you have an automatic toolchanger with your control, the T command causes the tool with the number "tt" to be loaded.

- "Ttt" corresponds to the number of the tool in the toolchanger.
- "Ooo" corresponds to the length and diameter entry on the tool offsets page.

Press the [TOOL OFFSETS] softkey on the job setup page to access the tool offsets page.

Figure 6.3 shows the tool offsets page as it appears on the screen. When you call up this page, you will see columns that are labeled as O#, Dia., Z

Figure 6.3 - Tool Offsets Page

```
| TOOL OFFSETS              < operator messages>     OFFSET #  2 | *
| Resolution   .010          <  error messages  >               |
|              O#     Dia       Z                    NEXT OFFSET | *
|               1   0.2473   -1.3567                             |
|               2   0.3740   -2.5640               INC -0.0012   | *
|               3   0.0000    0.0000                             |
|               4   0.0000    0.0000                             | -
|               5   0.0000    0.0000                             |
|               6   0.0000    0.0000                         JOG | *
|               7   0.0000    0.0000                             |
|               8   0.0000    0.0000                   HANDWHEEL | *
|               9   0.0000    0.0000                             |
|              10   0.0000    0.0000                    ZERO ALL | *
|              11   0.0000    0.0000                             |
|              12   0.0000    0.0000                  ADJUST ALL | *
|                                                                |
| GAUGE BLOCK: GB   0.0000    0.0000                    DIGITIZE | *
| MACH ZERO:   XZ  -1.2340  -10.0123                             |
```

Up to 48 offsets can be entered. Twelve of these are displayed at a time. They are selected by an O command in the program. The Z entries offset the Z axis positions. The Dia entry is used for radius compensation.

The entry for Z is the distance from the end of the tool to machine zero. The entry for Dia is the diameter of the tool. Typically the bed of the machine will be chosen for machine zero. Program zero may be offset from this by a fixture offset. A gauge block may be placed on the bed of the machine and the distance from its positive facing surface to machine zero entered in GB. When Digitize is executed, the GB entry is added to the current coordinate to compute the offset. For Dia, which is digitized in the X axis, this offset value is doubled. See Section 6.5.3 on digitizing for additional information.

Upon entering the Tool Offset screen from Job Setup, the Offset Number will be zero and the cursor will be on it. The other fields will retain their previous values. The cursor arrows can be used to access and enter any of the individual fields. The Offset number will reset to zero anytime the cursor is moved into the offset table, but will be restored if you press [OFFSET #].

The coordinates versus machine zero for X and Z respectively, are displayed on the bottom line of the screen.

[SHIFT] [UP] and [SHIFT] [DOWN] arrows will advance the screen to the previous or next set of twelve offsets, respectively. The screens will wrap around; that is, 37-48 is before 1-12 and 1-12 is after 37-48. The Offset Number will reset to zero when this is done.

Pressing the [OFFSET#] soft>key will bring the cursor to the field following the # to allow entry of an Offset number for editing or digitizing.

The cursor arrows can also be used to position the cursor on this field.

When a number is entered, the page containing that offset will display.

Pressing the [NEXT OFFSET] softkey increments the OFFSET Number by 1. If the number was on 48, it will reset back to 1. If the cursor was in the table, it is moved to the Offset Number field and the field is incremented to one. The page containing that offset will be displayed.

[NEXT OFFSET] is generally used in conjunction with Digitize to sequence through the tool offsets.

There are 3 ways you can enter offset values:

- enter the values manually.
- use the softkeys to increment the values.
- digitize the values using the softkeys.

6.5.1 Direct Cursor Entry of Tool Offsets

You can enter or edit tool length or diameter values by moving the cursor to the appropriate number. To use this method, complete these steps:

1. Move the cursor to any length (Z) or diameter (Dia) space by using the cursor control keys.

2. Type in the value. If you want to enter a negative number, type [-] and the number.

3. Press [ENTER].

If you move the cursor away from its position while you are entering the value, the previous value will be restored.

Change existing data with the above steps. The new value will replace the old value.

If the desired offset number is not on the screen, it can be accessed using [OFFSET #], [SHIFT] [UP]/ [SHIFT] [DOWN] cursor arrows, or [NEXT OFFSET], as described above.

6.5.2 Incrementing an Offset Entry

You can also increment (plus or minus) data by using the softkeys. This allows the operator to adjust an offset by a specifiable amount. For example, if he has determined that his part is in error by a measured amount, he can enter that amount as an increment and then adjust selected offsets with the control performing the math for him.

To increment a length offset or a diameter using the softkey method, follow these steps:

1. Position the cursor on the field to be incremented. If the desired offset number is not on the screen, it can be accessed using [OFFSET #], [SHIFT] [UP]/ [SHIFT] [DOWN] cursor arrows, or [NEXT OFFSET], as described above.

2. Press the [INC] soft key. Pressing this soft key will position the cursor on the [INC] field to allow entry of an incremental amount.

If the cursor was in the table when the soft key was pressed, the field it was on will change to reverse video as the cursor is jumped to the [INC] field. This flags that field as the one to be incremented.

3. While the cursor is on the [INC] field:

   a) If the incremental amount is the desired amount, the operator simply presses [ENTER].

   b) If a different incremental amount is desired, the amount can be entered with the pressing of [ENTER] registering the amount.

When [ENTER] is pressed, the cursor returns to the reverse video field, adds the incremental amount to it, and restores it to normal video. If no field is displayed in reverse video (the cursor didn't come from the table), no field is adjusted.

Any action other than pressing [ENTER] (another soft key, a cursor arrow, exit) will restore the reverse video field to normal video without modifying it.

6.5.3 Length Offset Digitizing

You can also use the softkeys to "digitize" a tool offset. When you digitize an offset, the applicable axis (with tool) is jogged to a given position from machine zero. The tool is typically touched to a surface or a gauge.

The jogging may be done from the JOG pages or it may be done using the handwheel while on the tool offsets page.

If the surface being touched is not at machine zero, such as with a gauge block, an entry can be made below the tool offset table that represents the distance from the positive facing surface of the gauge block to machine zero.

The [DIGITIZE] soft key initiates the digitizing of the tool offsets per the jogged location. Offsets can be digitized either one axis at a time or all axes at once.

When [DIGITIZE] is pressed, the control will add the current location versus machine zero to the gauge block entry to arrive at the basis for the offset value.

With the desired tool loaded in the machine, follow the following steps to digitize its offsets:

1. If the location you will position to for digitizing is not the Machine Zero location, cursor to the GB offset and enter the distance from the surface of the gauge block to machine zero. This value will be added to the absolute coordinates to form the entry when you press digitize.

   For diameter, the X axis is used. The computed value will represent the radius, which the control will double to convert it to diameter.

2. If you can digitize all axes at once, either cursor to the "OFFSET #" field or press the [OFFSET #] soft key to get to the Offset Number field and enter the desired offset number. Note that this will require positioning both axes to the digitize position before digitizing them.

   If you need to digitize one axis at a time (such as touching off on each reference surface), cursor to the appropriate axis of the desired offset number.

   If the desired offset number is not on the screen, it can be accessed using [OFFSET #], [SHIFT] [UP]/ [SHIFT] [DOWN] cursor arrows, or [NEXT OFFSET], as described above.

3. Jog to the desired position. You have two choices of the means to do the jogging. The handwheel can be enabled while on the tool offset page. If you do not wish to use the handwheel or you need to jog more than the X and Z axes, the jogging can be done from the regular jog pages.

To use the Handwheel:

Position the cursor to the column in the table next to the axis you would like to jog. It does not matter which offset you are on, only the column.

Important: If you are set up for digitizing all axes at once, the cursor was probably on the [OFFSET #] field. This field will display as zero when you go to the table. It is not lost and will be restored when you press [OFFSET #] again.

With cursor positioned in the table, press the [HANDWHEEL] soft key. This will change its label to reverse video and enable the handwheel. The handwheel is enabled for the axis column the cursor is on in the table. For the Diameter column, the X axis is enabled.

Pressing the [HANDWHEEL] softkey again or performing any cursor motion away from the current column or pressing any other active soft key will disable the handwheel and return the soft key label to normal video.

The handwheel resolution is controlled by the feedrate override switch and is displayed on the screen.

To use the Jog Pages:

Press the [JOG] soft key of the tool offset page. Then select the jog mode you want and position to the location you wish to digitize.

Then [EXIT] back to the Tool Offset screen.

4. If you are digitizing all axes at once, press [OFFSET #], to restore this field. It should now display the offset number you entered to be digitized.

   If you are digitizing only one axis, double check that the cursor is on the proper field (column and offset number).

5. Press the [DIGITIZE] soft key. The desired entry should be automatically made.

6. If digitizing all axes at once and you wish to digitize the next offset in sequence, press [NEXT OFFSET] to advance to the next tool offset and then digitize it in a like manner. If digitizing one axis at a time, cursor to the next field to digitize and then digitize it in a like manner.

6.5.4 Zeroing Tool Offset Table

The [ZERO ALL] soft key can be used to zero the entire tool offset table (1 through 48).

This key brings up a screen to confirm whether to zero all tool offsets.

If the operator presses the [YES] softkey, the control will zero the entire tool offset table but not the Gauge Block entries.

The Offset Number will be reset to zero and the cursor placed on its field.

6.5.5 Adjusting Tool Offset Table

The [ADJUST ALL] softkey is intended for adjusting the entire offset table by the same amount necessary to adjust a selected offset.

Suppose the existing offset table is known to be correct relative to each other but is shifted relative to machine zero. This can easily happen if the machine zero has been shifted, such as by manually moving the knee of the machine.

To correct the table:

1. Select one offset to use as reference.
2. Following the procedure described for digitizing, redigitize that one offset, except do not press [DIGITIZE].
3. Press [ADJUST ALL].

This will cause all of the entries in the tool offset table (1 through 48) except those that have zero for the Dia and Z to be adjusted by the amount that axis needed to be shifted to make it correct for the selected tool offset.

This procedure would normally be used for digitizing the Z axis but can also be used for diameter or for both offsets simultaneously.

6.6 Fixture Offsets

In generating a part program, it is generally desired to have a zero coordinate or program zero location at some convenient location relative to the part, such as the 'lower left corner' of the drawing.

Seldom is a machine such that its machine zero location is a suitable point for program zero. Thus program zero must be offset from machine zero to execute that program.

There are four general means of accomplishing this.

1. The programmer can define the offset in the part program. This is generally undesirable since the programmer then needs to be concerned with the machine characteristics and the program becomes machine dependent.

2. The program can begin with a G92 preset command to select the current location as the program zero. The operator then can jog to the desired program zero location prior to starting the program.

3. The operator can set the program zero using MDI under AUTO OPERATE or using JOG under AUTO OPERATE and aborting the return-from-jog by pressing [SHIFT] [CYCLE START] to start the program.

4. The operator can define one or more fixture offsets which the program can select.

The different fixture offsets may be used for different part programs or they may be used for repeated copies of the same part program.

The [FIXTURE OFFSETS] soft key selects a page upon which the operator may view or create the fixture offsets.

Figure 6.4 shows the fixture offsets page as it appears on the screen.

Figure 6.4 - Fixture Offsets Page

FIXTURE OFFSETS			< operator messages>		FIXTURE # 2	*
Resolution		.010	< error messages >			
O#	X	Y	Z	W	NEXT FIXTURE	*
1	0.0000	0.0000	0.0000	0.0000		
2	0.0000	0.0000	0.0000	0.0000	LAST O# 5	*
3	0.0000	0.0000	0.0000	0.0000		
4	0.0000	0.0000	0.0000	0.0000	MIRROR	*
5	0.0000	0.0000	0.0000	0.0000		
6	0.0000	0.0000	0.0000	0.0000	JOG	*
7	0.0000	0.0000	0.0000	0.0000		
8	0.0000	0.0000	0.0000	0.0000	HANDWHEEL	*
9	0.0000	0.0000	0.0000	0.0000		
10	0.0000	0.0000	0.0000	0.0000	ZERO ALL	*
11	0.0000	0.0000	0.0000	0.0000		
12	0.0000	0.0000	0.0000	0.0000	ADJUST ALL	*
FZ	0.0000	0.0000	0.0000	0.0000	DIGITIZE	*
MZ	2.3457	-1.2340	-10.0123	0.0000		

Fixture Offsets represent the distance from Machine Zero to Program Zero. Up to 12 offsets can be entered. Each offset can be entered with or without mirroring.

The program can request a specific offset number or it can sequence through the list of offsets.

In a program,·a specific offset is invoked by programming a G45 and an Onn where nn is the desired offset number. A G45 without an O field will increment the currently active fixture number.

Any movements associated with fixture offsets are applied with the next associated axis moves of the program. Offsets are cancelled by programming G45 O0# and are removed with the next programmed moves or by performing a machine home.

The [LAST FIXTURE] field controls the use of fixture offsets as follows. If Last Fixture is set to zero, Fixture Offsets are not used, even if the program requests a specific offset. If Last Fixture is nonzero, the program can explicitly select any of the 12 fixture offsets or it can request incrementing to the next fixture number. When incrementing, if the currently active fixture number was greater than or equal to that specified in Last Fixture, the next number will be number one and, if the G45 includes a branching command, the branch will not occur.

If suitable rules are followed in the creation of the program (see G45 in Chapter 16), the operator may determine whether to use fixtures and how many, without having to make any changes to the program. If a part requires several tools, the program can be created so that all fixtures are machined with one tool before changing to the next one rather than go through the tool change for each fixture location.

Near the bottom of the table, is a separate offset labeled FZ (Fixture Zero) that is used in digitizing the fixture offsets. This entry is the distance from the gauge location on the fixture to the program zero location. See Section 6.6.4 on digitizing for additional information. The SAVE and LOAD functions on the Load/Save page will save and load FZ and Last Fixture as well as the twelve offset values and their mirror flags.

Upon entering the Fixture Offset screen from Job Setup, the Fixture Number will be zero and the cursor will be on it. The other fields will retain their previous values. The cursor arrows can be used to access and enter any of the individual fields. The Fixture number will be reset to zero anytime the cursor is moved into the offset table.

The coordinates versus machine zero are displayed on the bottom line of the screen.

6.6.1 Direct Cursor Entry of Fixture Offsets

You can enter or edit fixture offset values by moving the cursor to the appropriate number. To use this method, complete these steps:

1. Move the cursor to any axis column in the table by using the cursor control keys.
2. Type in the value. If you want to enter a negative number, type [-] and the number.
3. Press [ENTER].

   If you move the cursor away from its position while you are entering the value, the previous value will be restored.

To change existing data, follow the above steps. The new value will replace the old value.

6.6.2 Mirroring a Part

Occasionally it is desirable to create 'as programmed' parts at some fixture locations and mirror image parts at other fixture locations.

Using the mirroring feature this is possible. To do this:

1. Using the cursor arrows, position the cursor on the axis entry to be mirrored for the desired fixture number.
2. Press the [MIRROR] soft key. A lower case 'm' will appear following the offset entry, indicating that the offset will be mirrored in that axis.

Pressing the [MIRROR] soft key with the cursor in or on a field that is set for mirroring will cancel the mirroring and remove the 'm' indicator.

The mirroring affects the program moves, not the offsets, in the same manner as the programmable mirrors. That is, if the mirror flag is set and the program calls for mirroring, then mirroring is off.

Any or all of the axes of a fixture can be mirrored.

6.6.3 Selecting the Number of Fixtures

Fixture Offsets are designed to allow programs to be written which allow the operator to determine if the program will be run without fixture offsets or on a number of fixtures of the operators choosing without modifying the part program.

Programs can easily be written which will execute the programmed pattern for one tool on as many fixture locations as the operator selects before proceeding to the next tool, thus saving tool change time.

Setting LAST O# to zero will disable fixture offsets, even if the program calls for a specific offset number.

Setting LAST O# to a non-zero number will allow a program to increment the active fixture number on command. If the active fixture number is equal to the Last Fixture number when commanded to increment, the active fixture number is reset to one and the execution of a conditional branch, if present, is inhibited. See Section 16 for details on the programming.

To set LAST O#, press the [LAST O#] softkey. This moves the cursor to the LAST O# field to allow entry of the Last Fixture number. The cursor keys can also be used to access this field. Enter the desired Last Fixture number and press [ENTER].

6.6.4 Fixture Offset Digitizing

You can also use the softkeys to "digitize" a fixture offset. When you digitize an offset, the applicable axis is jogged to a given position from machine zero. A gauging or reference tool is typically touched to a surface or a gauge or located into an alignment hole.

The jogging may be done from the JOG pages or it may be done using the handwheel while on the tool offsets page.

If the surface being touched is not at program zero, an entry can be made below the fixture offset table that represents the distance from the location you will digitize to program zero. This value will be added to the absolute coordinates from machine zero to form the entry when you press [DIGITIZE].

The [DIGITIZE] soft key initiates the digitizing of the fixture offsets per the jogged location. Offsets can be digitized either one axis at a time or all axes at once.

When DIGITIZE is pressed, the control will add the current location versus machine zero to the fixture zero (FZ) entry to arrive at the offset value.

The MACH ZERO row at the bottom of the Tool Offset page displays the current position of the axis with respect to its zero position. The value displayed is an absolute distance unaffected by any active offsets or G92 presets.

With a suitable tool or instrument loaded in the machine, follow the following steps to digitize the fixture offsets.

To digitize an offset:

1. If the location you will position to for digitizing is not the Program Zero location, cursor to the FZ offset and enter the distance from the location you will digitize (Fixture Zero) to program zero. This value will be added to the absolute coordinates to form the entry when you press digitize.

   Two typical situation for this would be:

   1) An 'L' shaped fixture with positive facing surfaces located at program zero. A known diameter tool is touched to the surfaces. The FZ entry would be minus the radius of the tool.

   2) A fixture with an alignment hole in its frame to which alignment tool can be positioned and inserted. The FZ entry would be the distance from the center of this hole to the program zero location.

2. If you can digitize all axes at once, either cursor to the [OFFSET #] field or press the [OFFSET #] soft key to get to the Offset Number field and enter the desired offset number. Note that this will require positioning all axes to the digitize position before digitizing them.

   If you need to digitize one axis at a time (such as touching off on each reference surface in sequence), cursor to the appropriate axis of the desired offset number.

3. Jog to the desired position. You have two choices of the means to do the jogging. The handwheel can be enabled while on the fixture offset page. If you do not wish to use the handwheel or you need to jog more than the X, Y, Z, or W axes, the jogging can be done from the regular jog pages.

To use the Handwheel:

Position the cursor to the column in the table next to the axis you would like to jog. It does not matter which offset you are on, only the column.

Important: If you are set up for digitizing all axes at once, the cursor was probably on the [OFFSET #] field. This field will display as zero when you go to the table. It is not lost and will restore when you press [OFFSET #] again.

With cursor positioned in the table, press the [HANDWHEEL] softkey. This will change its label to reverse video and enable the handwheel. The handwheel is enabled for the axis column the cursor is on in the table.

Pressing the [HANDWHEEL] softkey again or performing any cursor motion away from the current column or pressing any other active softkey will disable the handwheel and return the softkey label to normal video.

The handwheel resolution is controlled by the feedrate override switch and is displayed on the screen.

To use the Jog Pages:

Press the [JOG] soft key of the tool offset page. Then select the jog mode you want and position to the location you wish to digitize.

Then [EXIT] back to the Fixture Offset screen.

1. If you are digitizing all axes at once, press [OFFSET #], to restore this field. It should now display the offset number you entered to be digitized.

If you are digitizing only one axis, double check that the cursor is on the proper field (column and offset number).

2. Press the [DIGITIZE] soft key. The desired entry should be automatically made.

3. If digitizing all axes at once and you wish to digitize the next offset in sequence, press [NEXT FIXTURE] to advance to the next fixture and then digitize it in a like manner. If digitizing one axis at a time, cursor to the next field to digitize and then digitize it in a like manner.

6.6.5 Zeroing Fixture Offset Table

The [ZERO ALL] soft key can be used to zero the entire fixture offset table.

This key brings up a screen to confirm whether to zero all fixture offsets.

If the operator presses the [YES] soft key, the control will zero the entire tool offset table including the fixture zero FZ.

The Offset Number will be reset to zero and the cursor placed on its field.

6.6.6 Adjusting Fixture Offset Table

The [ADJUST ALL] soft key is intended for adjusting the entire offset table by the same amount necessary to adjust a selected offset.

Suppose the existing offset table is known to be correct relative to each other but is shifted relative to machine zero. A typical situation would be a multistation fixture for which a previous set of fixture offsets has been loaded but the fixture assembly is located at a different position relative to machine zero.

In this situation, all offsets are incorrect by the same amount. The [ADJUST ALL] function allows you to adjust the entire offset table without having to independently redigitize each field. To correct the table:

1. Set [LAST FIXTURE] to the number of stations in the fixture. This sets the range of how many offsets will be adjusted.
2. Select one offset to use as reference.
3. Following the procedure described for digitizing, redigitize that one offset, except do not press [DIGITIZE].
4. Press [ADJUST ALL].

This will cause all of the entries in the fixture offset table number 1 through the [LAST FIXTURE] number to be adjusted by the amount that offset needed to be shifted to make it correct for the selected tool offset.

This procedure may be used for one axis at a time digitizing or for all axes simultaneously.

6.7 Mid Program Start

Mid Program start provides a means to force the control to start at a block other than the first block of the program.

For mid program start to function, you must enter the job setup page from the auto operate page. If you get to the job setup page from the main menu page, the start is reset to the beginning of the program when you exit through the main menu page.

Press the [MID PROGRAM START] softkey on the job setup page to activate the mid-program start feature.

CAUTION: To guard against potential hazard to the machine or to personnel when performing a mid program start in auto operate, all necessary program parameters (G codes and M codes) must be established first by using MDI and jog from auto operate. You must set up control status, machine axis position (if program is incremental) and other machine functions to be compatible with the operations that follow the mid program start.

When you use mid program start, the control uses the block search page of the control to locate the starting block. When you must press the [MID PROG START] softkey on the job setup page you will get the Block Search page as shown in figure 6.5.

Figure 6.5 - Block Search Page

```
| BLOCK SEARCH                   START | *
| < operator messages>           LINE# | *
|                                RESET | *
| >< search string/ line # >           | -
|                                      | -
|   211G81Z-2.5D2.                     | -
|                                      | -
| Program  153 COMMENTX_DIM            | -
| <  error messages  >                 | -
```

The block search page can be used to locate a block by searching for a known text string in the block or it can locate a specified line number in the program. In either case, the program start point will be at the start of the located block.

6.7.1 How to Search for a String

To initiate mid program start, complete these steps:

1. Access the job setup page through either AUTO OPERATE or CHECK OUT.

2. Press the [MID PROG START] key on the job setup page to display the block search page.

3. Type in the character string that you want the control to search for. The characters appear in reverse video on the middle of the screen.

4. Press [ENTER] to complete the entry. The characters change to normal video.

5. Press the [START] softkey to scan through the program. The control stops its search when the first match is found. The message, "MATCH FOUND" appears on the screen.

   Press the [START] softkey again to continue the search if the block found is not the one you want. If you reach the end of the program and the control doesn't find a match, the message "NO MATCH FOUND" appears on the screen.

If no match is found, the control resets back to the first block in the program. You can force it back to the first block at any time by pressing the [RESET] soft key.

Once the correct matching sequence of characters in the program is found, you may exit back to the auto operate or checkout page (but **not** to the main menu page) and press either the green [CYCLE START] button for continuous execution or press the yellow [BLOCK BY BLOCK CYCLE] button for single block execution.

6.7.2 How to Search for a Line

You can also search for a specific line number in a program by completing the following steps:

1. Enter the line number you want the control to find.
2. Press [ENTER].
3. Press the [LINE #] softkey. When you press this softkey, the control will go directly to the specified line number and automatically exit to the page that called the block search.

If the line number is beyond the end of the program, the control will display 'no match found'.

6.8 Next Program

Press [NEXT PROG] on the job setup page to call the next program from the program directory. The control advances to the next program in numerical order.

The program's name and number will be displayed on the screen.

6.9 Directory

Press the [DIRECTORY] softkey to call up the directory. The directory is explained in detail in chapter 7.

You can use this page for reference, or you can enter a enter a program number to activate that program.

Complete the following steps:

1. Press [DIRECTORY]
2. Type in the desired program number and press [ENTER]. The control returns to the job set up page with the program you just chose as the active program.

6.10 Chapter Summary

In this chapter we discussed how to prepare the control for part production. In the next chapter we tell you how to enter a program and how to change a program that is already stored in memory.

Chapter 7 – Program Edit Page

7.0 Chapter Overview

This chapter tells you how to use the program edit functions.

After reading this chapter you will be able to:

- create a new program
- display a stored program
- edit a stored program
- assign a new number to an existing program
- delete a stored program

This chapter is divided into separate sections for each of the softkey selections available on the Program Edit page.

7.1 Program Edit Page

Press the [PROGRAM EDIT] softkey on the main menu page to access the program edit page.

Program edit allows you to manage program memory, enter a new program, or edit a program in storage.

When the screen appears, you will have nine options available to you. Press the corresponding softkey to call up an option.

Figure 7.1 shows the program edit page as it appears on the screen.

Figure 7.1 - Program Edit Page

```
| PROGRAM EDIT                  DISPLAY | *
| < operator messages>             EDIT | *
|                                 TEACH | *
| COMMENTX_DIM                 NEW PROG | *
| Program      153            NEXT PROG | *
| Flags                        RENUMBER | *
| Size          22          DELETE PROG | *
| Available 143918           DELETE ALL | *
| <  error messages  >        DIRECTORY | *
```

Prompt edit is the first edit page to appear after you press [EDIT].

Notice the [LIST] key. You can enter a program using prompt edit (the page you are on) or list edit. When you are in the prompt edit mode, you can choose the list edit mode by pressing [LIST]. To return to the prompt edit mode press EXIT.

You can switch between prompt and list edit at will and the control will keep the cursor on the current block.

On either page, a white background on a single character space represents the cursor location. This is the current editing point. The location of the cursor is controlled by the four arrow keys.

The left and right arrows move the cursor left or right, respectively, within the confines of the current line. On list edit page, the cursor will not go to the right of the EOB (#). On the prompt edit page, the cursor will not go to the right of the first blank space.

On the list edit page, the up and down arrows move the displayed program listing up or down one line and move the cursor to the first column after the line number field.

On the prompt edit page, the up and down arrows move the cursor through the lines of the screen, positioning it immediately following the field on the line.

On the list edit page, holding the [SHIFT] and pressing the up or down arrows will position to the beginning or the end of the program, respectively.

On the prompt edit page, holding the [SHIFT] and pressing the up or down arrows will move you back or forward by one block.

The keyboard letters default to upper case. You can enter lower case letters for incremental motion parameters by holding the [SHIFT] and pressing the letter.

7.3.1 Prompt Edit

If you choose to enter a program using the prompt edit mode, a list of prompts appears each time you enter a code. An advantage of the prompt edit mode is that it allows you to see programming options available for a particular code.

Prompt editing is designed as an aid which, contrary to many such aids, should not get in your way as you become more proficient. It has the further advantage of the control verifying the syntax of your entry.

The prompt edit function shows you one block at a time. The prompts will appear ahead of each word address field and will define the word address.

Use the prompt edit mode to enter a program and to see the prompts of a particular code. You can type the data next to the prompt. You can then return to list edit to see this block along with all the blocks that fit on the screen that you have entered.

The prompt edit page is divided into a number of spaces, or fields, where only specific sets of characters are allowed.

Figure 7.2 shows the prompt edit page as it appears after you press the [EDIT] softkey on the Program Edit Page or press [EXIT] on the List Edit page. The fields of the prompt edit page are explained in detail in the following paragraphs.

1. Line number space

The first five characters of the second line of the screen displays the line number of the block currently being displayed.

This number is not a part of the block but simply a count of the number of blocks up to the current one in the program. Thus if you insert or delete blocks, the numbers of subsequent blocks will be different accordingly.

The line number is also displayed on the list edit page ahead of each block. This can be useful for editing long programs since block search allows quick access to blocks using this number.

When using line numbers for editing a 'marked up' listing, it is best to begin editing from the end and work your way to the start to avoid changing line numbers.

2. "/" space

The "/" space (same line as line number) is a space designated for a block delete code. At Job setup time, the operator can determine whether to execute or ignore all blocks that begin with a '/'. For the block to be ignored during execution, he must activate BLOCK DELETE in job setup.

To enter a slash, press the [/] key.

3. Sequence Number space

The sequence number space immediately follows the "/" space. It is designed to help you in locating specific blocks within a program. You may find sequence numbers useful for program documentation and program search purposes. They are necessary when you use program branching or subroutines.

To enter a sequence number, complete these steps:

1. Press [N] to enter this position. The 'N' will appear in its space followed by the cursor.

2. Type in a number up to four digits.

4. G space

Whenever you type in a G code, it appears in the G code position (below the '/' space). The cursor moves to the right of the G space.

Type in the G code digits.

You do not have to move the cursor to make these entries.

5. Prompt space

The prompt space displays the G code name and word address prompts. You can't enter this space with the cursor, or change anything in this space.

There are some G codes that have more prompt comments than can appear on the screen at one time, due to the limits of the screen. A "#" character appears in the lower right corner of the screen when the last prompt is on the screen.

A 'DOWN ARROW FOR MORE PROMPTS' message appear at the bottom of the screen (in the comments space) if there are more prompts below those on the screen.

Press the DOWN cursor to advance to the rest of the prompts if the DOWN ARROW FOR MORE PROMPTS message appears instead of the "#".

6. Word Address Space

If letters other than N or G appear in a data block that is not a comment, they will be listed vertically in this space located below the G space. Enter a value following the letter address.

7. Optional Parameter Space

When prompts are displayed for certain G codes, several letter addresses may be considered optional. These letters will be listed vertically and will "flash" to indicate that they are not required.

If you specify an optional letter in the data block, the letter will stop flashing and will move right one space (into the word address space), immediately followed by the cursor. You can then type in a value.

8. Comment (;) Space

You can use the ; space to specify a comment in a part program or define the name of a program if the comment block is the first block in the program. Program names are limited to 12 characters. Other comment blocks can be up to 70 characters long.

For example, you can use this ";" space to name the example program of section 7.3.1.1 you are about to enter. Name the program, "EXAMPLE." The control will ignore the program name when you execute the program, but it will be displayed on the pages of the control and in the directory.

A second ";" on the program name line will make the balance of the line be a comment.

1. Press [;] to put a ";" in this position. The cursor will appear immediately after the mark.

If you have cursored to the comment line, typing any character will cause the ";" to automatically be inserted into the block.

2. Type in [EXAMPLE].

3. Press [EOB].

Entering and Editing a Program in Prompt Edit

When you enter a program in the prompt edit mode, certain prompts appear when you select G codes. The control recognizes that the G code entry is complete and brings up these prompts.

The control recognizes the completion of a G code entry when:

- you enter 2 digits with a G code.
- you press [ENTER] after you enter a G code.
- you press a letter other than a G.

For example, to get the block G81Z-2.F20, complete the following steps:

1. Press [G] key. The G will appear with the cursor following it.

2. Enter [81]. The prompts for G81 will automatically appear since this was a 2 digit G code. If you were entering G1, you can either press [ENTER] after the 1 or enter G01, which is equivalent.

3. The Z will appear, since it is a required block, and the cursor will follow it. If you want to press the [Z] you may, but it is not necessary to do so.

4. Enter [-2]. for the Z entry.

5. Press [F]. The flashing F, indicating an optional field, will disappear and a non-flashing F will appear in the next column, indicating it has been added to the block.

Important: A flashing underscore will actually appear where the flashing prompt was, indicating it was an optional field.

6. Enter [20]. for the F entry.

Important: If you now press [ENTER], which is not necessary, the block will be redisplayed. The word address letters the are currently in the block (other than N and G) will be listed contiguously in the order they are in the block, under the G space. The prompts are added to these and then any other prompts are added below these entered fields.

7. Press [EOB].

   The control is now ready for the next block.

   If you press the [LIST] softkey, you will see the completed block.

Prompt edit entries can also be made by positioning the cursor to a line containing a prompt and making the numeric entry. The editor will automatically move the letter into the block when you begin the numeric entry.

Any time you press [ENTER], the block is redisplayed, the syntax is checked and the cursor is positioned wherever the control determines is most likely to need an entry.

Graphics

The GRAPHICS softkey selection provides a powerful program debuging tool. The first soft key from the top is labeled 'GRAPHICS'. Pressing this soft key takes you directly to the GRAPHICS mode of QUICK CHECK.

The program that was selected while you were under EDIT can be executed by pressing [CYCLE START]. When you EXIT the QUICK CHECK-GRAPHICS screen you will return to EDIT. Which block you are on will depend upon what happened while you were in QUICK CHECK-GRAPHICS.

If you did not attempt to execute the program, the editor will be on the block that it was on when you pressed [GRAPHICS].

If the program was executed to completion, the editor will be on the block of the program containing the M02 that terminated execution. Generally this is at the end of the program, and it is the 'best guess' as to where the operator might want to EDIT the program.

If the program aborts due to an error, or if it is aborted using SHIFT - CYCLE STOP, the editor will be on the block it was on when the error was detected or the program was aborted.

If the program was executing a subroutine at the time it aborts due to an error or due to the use of SHIFT - CYCLE STOP, the editor will be on the first block of the program being edited.

List Edit

List edit mode lets you type in a program block-by-block. No prompts appear, although line numbers are provided for you.

Press [LIST] to see a partial or complete program after you have entered it. As you enter each line, you will see the line on the screen. As you continue to type, the program will scroll up.

At the left of the second line of the screen is displayed the block type for the current block if it contains a G code in the first few characters.

Note: If the G code is edited or entry is being made with the cursor on the EOB (#), this field may not automatically update. An update can be forced by pressing [ENTER].

A space is displayed before each word address letter. This space is not really part of the block but is included in the display for added readability. The cursor automatically skips through these spaces as it is positioned along a line.

Holding [SHIFT] and pressing the left or right arrow will jump the cursor to the next word address letter in the corresponding direction.

The List Edit Page also has the Quick Check Graphics soft key selection. Operation of this debuging tool is described in section 7.3.1.

Figure 7.3 shows the list edit page as it appears on the screen.

Figure 7.3 - List Edit Page

```
| Program 153 COMMENTX_DIM        < operator messages>   GRAPHICS  | *
| BOLT CIRCLE                     <  error messages  >             |
|                                                                  | -
|                                                                  |
|                                                                  | -
|                                                                  |
| 00001;COMMENTX_DIM,I_I_INC#                                      | -
| 00002/N123 G24 I2.5 J-2.125 R.6 A25.#                            |
| 00003M02#                                                        | -
|                                                                  |
|                                                                  | -
|                                                                  |
|                                                                  | -
|                                                                  |
|                                                                  | -
|                                                                  |
|                                                                  | -
|__________________________________________________________________|
```

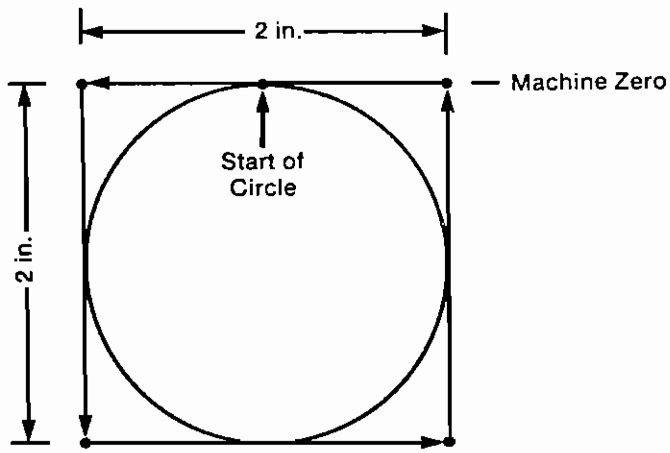
7.3.2 Example Program

Type in the following program:

Program This...	It Means This...
00001 ; EXAMPLE#	Names the program.
00002 G99#	Cancels any previous position preset.
00003 G00 X0 Y0 Z0#	Rapid to machine zero (this assumes that machine zero is somewhere in the middle of machine travel).
00004 G91#	Establishes incremental programming mode.
00005 G01 X-2. F20.#	Move X-2. at 20 ipm.
00006 Y-2.#	Move Y-2. at 20 ipm. G01 remains in effect.
00007 X2.#	Move X2. at 20 ipm.
00008 Y2.#	Move Y2. at 20 ipm. These moves produce a square 2 inches on each side.
00009 X-1.#	Move X-1. at 20 ipm.
00010 G02 X0 Y0 I0 J-1.#	Make a full circle with 1 inch radius, starting at the current location. Move is made at 20 ipm.
00011 M02#	End of program.

Figure 7.4 shows the part you just entered.

Figure 7.4 - Example Program Result



7.3.3 Making Changes to a Part Program

You have just entered a program into the control using the prompt and list edit modes. When you list a program, you may find an error. You can delete, insert, or change any character or group of characters by following the steps listed below.

Insert a Character into a Data Block

To insert a character into a data block, complete these steps:

1. Move the cursor to the character where you want to insert the data.

2. Press [INSERT]. The characters on the right move to the right opening a one character insert space.

3. Type character(s). The insert space is maintained after the last character entered.

4. Press any cursor key or press [ENTER] to terminate the insert mode. The insert space is automatically closed by moving the characters on the right to the left.

For example, if your block reads G01X-2.Z0 and you want to insert a Y value before the Z value, you must put the cursor on Z and press [INSERT]. Type [Y] and its value, then press any cursor key.

Note that whenever the cursor is blank (not on a character) or is on an EOB (#), the editor is in the insert mode.

Insert a Block In List edit:

If you want to insert a block, complete these steps:

1. Move the cursor to the line below where you want to insert a block.

2. Hold [SHIFT] and press [INSERT].

   [SHIFT] [INSERT] inserts another block (normally in column one). When you press these keys, the balance of the current line is pushed down one line. An EOB (#) character is inserted at the cursor. The cursor will be on the (#).

   To insert a block above an existing block, place the cursor at the very beginning of the existing line and press [EOB] or [SHIFT] [INSERT].

   If you want to insert a block below an existing line, place the cursor at the # of the existing line. Press [SHIFT INSERT].

   Note that this can also be used to split a block into two blocks.

   Entering an EOB (#) over another character will also have the same effect as [SHIFT] [INSERT]. The EOB does not replace the character the cursor was on.

In Prompt edit:

1. Using up or down cursor arrows, position to the block immediately after the location where you want to insert a block.

2. Hold [SHIFT] and press [INSERT]. A blank prompt page will appear, ready to accept the new block.

   Pressing [EOB] instead of [SHIFT] [INSERT] will also set you up to insert a block but it will be after the currently displayed block.

3. Multiple blocks may be inserted in sequence. When you are done inserting, use [SHIFT] [DOWN ARROW] to terminate the insert. This will bring up the block you initially positioned to since it follows the last one you inserted.

Delete a Character

To delete a character in a block complete these steps:

1. Move the cursor to the character you want to change.

2. Press [DELETE].

   When you press [DELETE] once, one character is erased. If you hold the [DELETE] key, characters to the right of the cursor will be deleted. The control will automatically close in any blank spaces.

   The EOB (#) can not be deleted using the [DELETE] key alone, it required a [SHIFT] [DELETE]. Pressing [DELETE] with the cursor on the EOB (#) will delete the character to the left of the cursor.

   If you are in the insert mode, [DELETE] will delete the character to the left of the cursor.

Delete an Entire Block

Hold [SHIFT] and press [DELETE] to delete an entire block on the prompt edit page.

On the list edit page, hold [SHIFT] and press [DELETE] to delete characters from the cursor up to the next EOB (#).

Pressing [SHIFT] [DELETE] with the cursor on the EOB (#) will delete the EOB (#), thereby combining the next block onto the end of the current one.

Change an Incorrect Entry

To change an entry, complete these steps:

1. Move the cursor to the character you want to change.

2. Type in the correct character.

   The character you wanted to change disappears from the screen. The correct character appears in its place.

Syntax Checking

The prompt editor provides significant syntax checking to help protect against invalid entries. The editor will, however, allow several entry formats, including parenthesized parametric and arithmetic expressions, which your control may not be able to execute. These are included for compatibility with planned future features.

If the prompt editor detects a syntax error in the block it is attempting to display, it will display the balance of the block in the comments field along with a 'SYNTAX ERROR' message in the lower left corner.

Pressing [ENTER] or [EOB] causes the prompt editor to perform a syntax check on the current block before proceeding. The [ENTER] is normally used to force a syntax check on the current block. [EOB] is the normal termination of a block during entry. If the block does not have a detectable syntax error, it is accepted and the editor proceeds to the next block. If a syntax error is detected in the current block, the error condition is displayed and the editor remains on that block instead of advancing.

When the 'SYNTAX ERROR' message is being displayed on the prompt edit page, some otherwise active syntax checking or aids (such as automatic parenthesis balance control) is suspended. The cursor is on the first character of the comments field of the display, the best guess as to the location of the syntax error.

You should immediately correct the error. Then press [ENTER]. This will cause the editor to again attempt to display the block.

7.3.4 Search

Press the [SEARCH] softkey on the prompt edit page to access the block search page. This allows you to scan for a particular character or sequence of characters. You can also use the search mode to search for a certain line number.

On the block search screen, the block is prefixed with its line number in reverse video. The cursor remains at the start of the located or current block when you exit from the search mode.

On the list edit page, each block is preceded by its current line number. On the prompt edit page, the line number of the block being displayed is at the left of the second line.

SAVE devices that are defined for list mode prefix each block with its line number.

Figure 7.5 shows the search page as it appears on the screen.

Figure 7.5 - Search Page

```
| BLOCK SEARCH                    START | *
| < operator messages>            LINE# | *
|                                 RESET | *
| >< search string/ line # >            | -
|                                       | -
|   211G81Z-2.5D2.                      | -
|                                       | -
| Program  153 COMMENTX_DIM             | -
| <  error messages  >                  | -
```

When you press [SEARCH] from the prompt edit page, the line number and the characters of the current block appear in the middle of the block search page.

To use the block search function to find a character or sequence of characters follow these steps:

1. Type in the data block that you want the control to find.

   The data block appears in reverse video in the middle of the screen above the current block.

2. Press [ENTER].

   The data block changes to normal video.

3. Press the [START] softkey to scan from the next block forward through the program.

   If a corresponding code is found by the control the message, "MATCH FOUND" appears on the screen.

   The "found" data block and the line number in which it appears are shown in on the middle of the screen.

   The control stops its search when the first match is found.

4. Press [START] to resume the search if the block that was located is not the one you want. Programmed data blocks are examined in order. When the control reaches the end of the program, "NO MATCH FOUND" is displayed on the screen.

5. Press [EXIT] to return to the editor. The cursor will be on the last located block.

   If you need to return to the start of the part program to search again, press [RESET].

Search by Line Number

To look at the data of a certain line number, complete these steps:

1. Type in the line number you want to see. It will appear in reverse video in the middle of the screen.

2. Press [ENTER]. The line number field will change to normal video.

3. Press the [LINE #] softkey.

   If the program has at least as many lines as the line number you specified, the screen automatically advances to the prompt edit page with the block corresponding to that line number displayed.

7.3.5 Insert P

Press the [INSERT P] softkey on the edit page to access the program insert function.

Use the insert P function to transfer some or all of the data blocks from one program into another program. This function is useful if you have several similar data blocks in more than one program. If you use this function you will not need to re-enter the blocks that are the same.

You can also use insert P to copy an entire program; for example, you may want to edit one program, but keep the previous version.

Figure 7.6 shows the program insert page as it appears on the screen.

Figure 7.6 - Program Insert Page

```
| PROGRAM INSERT              INSERT | *
| < operator messages>        /START | *
|                      Begin at LINE# | *
| From Program>  25       120 \   N# | *
| STDPATHS                      / END | *
|                        End at LINE# | *
|                           200 \  N# | *
|                                     |
| <  error messages  >     DIRECTORY | *
```

To use the program insert function, complete the following steps:

1. Enable the part program that you want to insert into. Use list edit to position to the appropriate line number for program insertion. The currently active line # is where the data will be inserted. All data will inserted before (not after) this line. This line will be the first block after the inserted program. Return to prompt edit. This same line number is displayed on line 2 beneath the active program number.

2. Press the [INSERT P] softkey on the prompt edit page. You will enter data into the program currently being edited. After you press [INSERT P], the program insert page appears.

3. Type in the number of the program you are copying data from. As you enter the number, it will appear in reverse video in the "FROM PROGRAM" space.

4. Press [ENTER].

   The "From Program #" can't be the same as the currently active part program. You can also enter a "From Program #" by going to the directory page from the program insert page and entering the program number from there.

5. Select the starting point of the program you want to insert **from.** You have three choices:

   - Begin at the Start
   - Begin at a specified line number
   - Begin at a block containing a specified N field

6. Press the corresponding softkey. The option you select appears in reverse video. If you select [LINE #] or [N#], the cursor moves to an area below "Begin At".

7. Enter your line # or N# below "Begin At". A confirmation message, "MATCH FOUND" or an error message "NO MATCH FOUND" appears.

8. Select the point to "End At" in the program you are inserting **from**. You have three choices:

   - End at END
   - End at a specified line #
   - End at a block containing a specified field

Note that the insert function will copy all lines from the start number up to, but not including, the last number. In other words, if you enter lines 10 to 16 into the program insert mode, lines 10 to 15 will be copied to the designated program.

9. Press the corresponding softkey. The option you select appears in reverse video. If you select [LINE #] or [N#], the cursor moves to an area below "End At".

10. Enter your line # or N# below "End At". A confirmation message, "MATCH FOUND" or an error message "NO MATCH FOUND" appears.

11. Press the [INSERT] softkey. The data will be copied to the new program. The data remains unchanged in the original program.

7.3.6 G Codes

Press the [G CODE] softkey on the edit page to access a list of G codes used with this control.

The lists are divided into categories according to their functions:

- axes modes
- coordinate control
- positioning dwell
- autocycles
- autoroutines
- CNC control functions

Figure 7.7 shows the first page of the G code page as it appears on the screen after you press the [G CODE] softkey.

Figure 7.7 - G Codes Page (first page)

```
AUTOCYCLES              AUTOROUTINES            POSITIONING/DWELL
G75 CAVITY              G22 HELIX CW            G00 POINT-POINT
G79 PROGRAMMABLE        G23 HELIX CCW           G01 LINEAR
G80 CANCEL              G24 BOLT CIRCLE         G02 ARC CW
G81 DRILL               G25 STEP/REPEAT         G03 ARC CCW
G82 COUNTER BORE        G26 POCKET MILL         G04 DWELL
G83 PECKDRILL           G27 POST MILL           G05 ARC TANGET
G84 TAP                 G29 EXEC AUTOCYCLE
G85 BORE                G39 PARA SUBPROG
G89 RESTORE

                                                VARIABLES
                                                G39 LOCAL
                                                G59 PAL
                                                G79 GLOBAL

                                                          PAGE 1 OF 2
                                                > G   0      NEXT PAGE
```

Figure 7.8 shows the second page of G codes as it appears on the screen. Press the [NEXT PAGE] softkey to return to the first page of G codes, as seen in figure 7.7.

Figure 7.8 - G Codes (second page)

```
COORDINATE CONTROL   AXES MODES          CNC CONTROL
G70 INCH             G17 XY PLANE        G60 ZONE INHIBIT
G71 METRIC           G18 ZX PLANE        G62 OVERRIDE INHIBIT
G72 SCALING          G19 YZ PLANE        G66 GRAPHICS CONTROL
G74 ROTATION         G30 CANCEL MIRROR
G90 ABSOLUTE         G31 AXES MIRROR
G91 INCREMENTAL      G40 CANCEL CTR COMP
G92 PRESET PROG 0    G41 CUTTER LEFT
G94 PER TIME FEED    G42 CUTTER RIGHT
G97 ACC/DEC INHIBIT  G45 FIXTURE OFFSET
G99 CANCEL PRESETS




                                                   PAGE 2 OF 2   *
                                       > G    0      NEXT PAGE
```

Press the [NEXT PAGE] softkey column to access the second page of G codes. Press the key again to return to the first list.

Examine a Code

When you are on the G code page, you may enter a G code. To enter a G code:

1. Type in the digits for the G code you wish to enter. The code you type in will appear in the lower right corner of the screen.

2. Press [ENTER].
   The screen will change immediately to the prompt edit page.

The G code you selected will be entered into the current block and its prompts displayed. If the block already contained other data, it will still be there. If it contains a G code, it will be replaced by your new G code.

7.3.7 M Codes

Press the [M CODE] softkey on the display page to access a list of M codes used with this control. These codes are divided into categories according to their function:

- spindle control
- program control
- coolant control
- clamps
- RPM changer
- tool changer
- spindle with flood coolant

The M codes page shows a typical standard set. Your custom PAL may exhibit a different list of M codes.

Figure 7.9 shows the M code page as it appears on the screen.

Figure 7.9 - M Code Page

```
SPINDLE CONTROL       PROGRAM CONTROL      TOOL CHANGER
M03 SPINDLE CW        M00 PROGRAM STOP     M06 PROG STOP MAN TC
M04 SPINDLE CCW       M01 OPT PROGRAM STOP M20 TOOL OUT   -ATC-
M05 SPINDLE OFF       M02 PROGRAM END      M21 TURRET CW  -ATC-
                      M30 PROGRAM END/REW  M22 TURRET CCW -ATC-
COOLANT CONTROL                            M23 TOOL IN    -ATC-
M07 COOLANT ON-MIST                        M27 TURRET HOME-ATC-
M08 COOLANT ON-FLOOD  CLAMPS
M09 COOLANT OFF       M10 CLAMPS ON        RPM CHANGER
                      M11 CLAMPS OFF       M24 RPM UP
SPINDLE W/FLOOD COOL                       M25 RPM DOWN
M13 SPINDLE CW                             M28 LOW GEAR
M14 SPINDLE CCW                            M29 HIGH GEAR



                                                PAGE 1 OF 1
                                     > M    0      NEXT PAGE
```

Select an M Code

When you are on the M code page, you may select a particular code.

1. Type in the digits for the M code you wish to enter.

   The code you type in will appear in the lower right corner of the screen.

2. Press [ENTER].

   There is only one list of M codes. The control will return to the prompt edit page with that M code entered into the current block. If there was already an M code in the block, it will be replaced with the one you entered. The rest of the block will be the same as it was before you generated the new M code.

7.3.8 Directory

The directory page is explained in detail later in this section.

If you select [DIRECTORY] from the prompt edit page, you can enter a Program Number into the current block by making the numeric entry on the directory page.

1. Type in the digits for the P code you wish to enter.

   The code you type in will appear in the lower right corner of the screen.

2. Press [ENTER].

   The control will return to the prompt edit page with that Program Number entered into the current block. If there was already a Program Number in the block, it will be replaced with the one you entered. The rest of the block will be the same as it was before you generated the new Program Number.

7.4 *New Prog Softkey*

Press the [NEW PROG] softkey to establish a program number before you begin editing or creating a program. Pressing [NEW PROG] gives you the first available unused program number.

If you want to assign your own number, you can enter it on the program edit page. Or, you can select [DIRECTORY] from the program edit page, locate an available number and enter it there.

7.5 Next Prog Softkey

Press [NEXT PROG] on the program edit page to call up the next program in numerical sequence.

The program's number and either its name or its flags and size will be displayed on the screen.

7.6 Renumber Softkey

Press the [RENUMBER] softkey on the program edit page to change a program number.

The new value must be different from any of those already in a directory. If you use a number already in use, an error message, "PROGRAM # IN USE" appears on the screen.

When you access the renumber page, you will see the number of the current program in the "Program" space. You will also see a space where you can enter another number.

Figure 7.10 shows the renumber program page as it appears on the screen after you press the [RENUMBER] softkey on the program edit page.

Figure 7.10 - Renumber Page

```
| RENUMBER PROGRAM                 | -
| < operator messages>             | -
|                                  | -
| COMMENTX_DIM                     | -
| Program      153                 | -
| Flags                            | -
|>Renumber To    0                 | -
|                                  | -
| <  error messages  >             | -
```

To renumber a program complete these steps:

1. Type in the new number, from 1 to 250. Make sure you do not have more than one program with the same number. If you do try to use the same number, an error message will appear on the screen.

2. Press [ENTER]. The program will now assume the new value and the control will return to the program edit page.

7.7 Delete Prog Softkey

Press the [DELETE PROG] softkey on the program edit page
to delete a currently selected program. Remember, you cannot delete a protected program.

Important: You can delete a program while it's running. When the program is finished running it is erased. The complete program will be executed before it's erased.

The screen shows the number and the name of the program that is currently active.

It asks if this is the program you want to delete.

- Press the [YES] softkey to erase the program from memory.

- Press the [NO] or [EXIT] softkey if you do not want to delete this program. The control will return to the program edit page without affecting the program in the editor.

  Figure 7.11 shows the delete page as it appears on the screen when you press the [DELETE] softkey.

Figure 7.11 - Delete Page

```
| DELETE PROGRAM                       | -
|                                      | -
|                                 YES  | *
| DELETE P 153 ?:PARA                  | -
|                                  NO  | *
|                                      | -
|                                      | -
| < operator messages>                 | -
| <  error messages  >                 | -
```

7.8 Delete All Softkey

When you press the [DELETE ALL] softkey on the program edit page, a screen similar to figure 7.11 appears. The flashing highlighted message "DELETE ALL?" appears along with [YES] and [NO] softkeys.

If you press the [YES] softkey, all unprotected programs will be deleted from memory. Press the [NO] or [EXIT] softkeys to return to the program edit page without deleting any programs. You cannot delete a protected program.

7.9 Directory Softkey

Press the [DIRECTORY] softkey on the program edit page to access the directory of stored programs.

There are a maximum of 138 programs in the directory. Each page will give you a list of the programs you have stored in memory.

When you access the directory, you will see an individual entry for each program stored. The directory toggles between showing each program's name or its size and flags.

Figure 7.12 shows the directory page as it appears on the screen with program names. Press [Flags] key to show the Flags.

Figure 7.12 - Directory Page

```
|      DIRECTORY      143926 Char Available                     FLAGS |  *
|      Program Name          Program Name         Program Name        |
| P002 PN476934234     P193 BFH_CAP                                   |  -
| P003 CALL_2          P237 A.B.LOGO                                  |
| P004 6-AXIS                                                         |  -
| P014 DIE-125A                                                       |
| P015 TOOL-125B                                                      |  -
| P153 COMMENTX_DIM                                                   |
| P155 TESTPROG                                                       |  -
| P156 EXTRACTOR                                                      |
| P157 JONES_CO                                                       |  -
| P158 SHUTTLE                                                        |
| P160 MOLD_CAVITY                                                    |  -
| P164 MOLD_PROFILE                                                   |
| P167 QR_TEMPLATE                                                    |  -
| P170 16PIN_IC                                                       |
| P171 68PIN_SOIJ                                   PAGE 1 OF 1       |  *
| P172 12-004-1345                    > P     0         NEXT PAGE     |
```

Figure 7.13 shows the directory page as it appears on the screen when it is toggled to show the program flags. Press [NAMES] key to show the Names.

Figure 7.13 - Directory Page (with program flags)

```
|       DIRECTORY      137246 Char Available                  NAMES | *
|         Size     Flags        Size     Flags        Size     Flags |
|  P002   81          R P193    144                                  | -
|  P003   22          P P237    677                                  |
|  P004   82                                                         | -
|  P014   50                                                         |
|  P015   50                                                         | -
|  P153   22                                                         |
|  P155  109                                                         | -
|  P156  163                                                         |
|  P157    2                                                         | -
|  P158 2145                                                         |
|  P160  126                                                         | -
|  P164 1453                                                         |
|  P167 1430                                                         | -
|  P170   70                                                         |
|  P171   80                                      PAGE 1 OF 1        | *
|  P172  455                        > P    0          NEXT PAGE      |
```

The directory lists each program's name (if any), its assigned number, the number of characters and status flags.

The highest number you can assign a program is 250.

If a program contains a checksum error (C flags) or a duplicate error (D flag), its number appears flashing with a red or reverse video background. and the C flag is set.

7.9.1 Select a Program From the Directory

To select a program in the directory, complete these steps:

1. Type in the number of the program you want to be the current active program. The number will appear in the lower right corner of the screen.

2. Press [ENTER].
   The control returns to the program edit page; the program number you selected from the directory appears on the screen as the active program.

7.10 Teach

Press the [TEACH] softkey on the program edit page to access the teach mode.

When you access this page, you will notice that the page is similar to the manual operate page.

When you use the teach mode, you are actually moving the machine to various locations. You can store the position of each location into program memory.

- If you are in jog mode, press [CYCLE START] to store the program.
- If you are in manual teach mode, press [DIGITIZE] softkey to store the program.
- If you are in MDI, press [CYCLE START] to store and execute a program, or press [EOB] to only store the program.

You are "building" a part program. The blocks generated in the jog or manual teach modes will consist of axis information only (X, Y, Z, U, V, W,). You can add additional codes (such as, G, F, S, T, etc.) to the part program by using the program edit function.

MDI teach blocks are added to the program in their entirety.

Figure 7.14 shows the teach page as it appears on the screen.

Figure 7.14 - Teach Page

```
| DIGITIZE PROGRAM  JOG: HANDWHEEL | *
| < operator messages>  CONTINUOUS | *
|                      INCREMENTAL | *
|  COMMENTX_DIM                    | -
|  Program      153   MANUAL TEACH | *
|  Flags                 MDI TEACH | *
|  Size          22    DELETE PROG | *
|  Available 143926                | -
| <  error messages  >   DIRECTORY | *
```

7.10.1
Jog Handwheel

Use the handwheel to jog to the desired location.
Press the green [CYCLE START] button on the front panel to store the program zero or current positions if they have changed.

7.10.2
Jog Continuous

Use the softkeys to jog to the desired location.
Press the green [CYCLE START] button on the front panel to store the current positions if they have changed.

7.10.3
Jog Incremental

Make the incremental entry and use the soft key to jog to the desired location. Press the green [CYCLE START] button on the front panel to store the current positions if they have changed.

7.10.4
MDI Teach

The MDI teach page is identical to the MDI page described in section 5.8. Refer to section 5.8 for the detailed functions of this mode. Each time you enter information into the control, the data will be entered into the program being digitized. If you press [EOB], the commands will be stored. If you press [CYCLE START], the command will be both stored and executed.

7.10.5
Manual Teach

Press the [MANUAL TEACH] softkey on the teach page to access the manual teach page.

The manual teach function allows you to digitize axes positions that are arrived at by moving the machine manually. For example, you may move axes by using the mechanical handwheels.

The control must be in ESTOP during the manual teach mode.

1. Press [E-STOP]. The E-STOP screen will appear on the screen. This should de-energize the drive to allow manual positioning.

2. Press [EXIT]. This will recall the teach (digitize program) page.

3. Press [MANUAL TEACH]. This will disable the drives automatically and the 'DRIVES NOT ON' message will appear.

4. Manually position the axes to the desired location.

   During manual teach, you are manually turning the axis shaft in order to provide an encoder feedback signal to the control.

5. Press the [DIGITIZE] softkey. The position you selected is recorded as a block in a the active program.

   Repeat these steps until you have finished all your digitized moves.

6. Press [EXIT] twice to return to the program edit page. If you view your program under list edit, you will note that this only builds axis moves. It does not put in G codes or M codes into the program.

   Figure 7.15 shows the manual teach page as it appears on the screen.

Figure 7.15 Manual Teach Page

```
 _____________________________________
| MANUAL TEACH MACH ZERO: DIGITIZE |  *
| < operator messages>              |  -
|                        0.0000 U   |  -
|  COMMENTX_DIM          0.0000 V   |  -
|  Program      153     -0.0001 W   |  -
|  Line          14     -0.0002 X   |  -
|                        0.0002 Y   |  -
|                       -0.0005 Z   |  -
|_<  error messages  >______________|  -
```

Important: When the Teach mode is active and the axis is defined as a Rotary axis, the control converts all moves it learns to shortest path (never greater than 180^{o}).

7.10.6 Delete Prog

Press the [DELETE PROG] softkey on the teach page to delete a currently selected program.

The delete page is described earlier in this chapter.

7.10.7 Directory

Press [DIRECTORY] on the teach page to access a directory of stored programs.

The directory is described earlier in this section.

7.11 Chapter Summary

In this chapter we discussed how to enter and edit a part program. In the next chapter, we will tell you how to check programs before you cut a part.

Chapter 8 -- Check Out Page

8.0 Chapter Overview

In this chapter we examine the methods to check a program for accuracy before you cut a part. After reading this chapter, you will know about:

- Quick Check
- Feed Check
- Dry Run
- Dry Z Run

8.1 Check Out

After you have entered a program into memory, you should check the program to make sure it performs exactly as you want.

Important: We recommend using Quick Check before you execute the part program with axis moves.

To check a program before execution, press the [CHECK OUT] softkey on the main menu page to bring the check out page to the screen.

When you access this page, you will have four check out options to choose from. Each check out option serves its own purpose. These options are:

- **[Quick Check]**-- There is no axis motion; the program is run as fast as the control can.
- **[Feed Check]**-- There is no axis motion; the program runs at the specified feedrate.
- **[Dry Run]**-- The program runs with motion on all axes except the Z axis. The executed feedrate will be the greater of either the programmed feedrate or the AMP dry run feedrate.
- **[Dry Z Run]**-- The program runs with motion on all axes. The executed feedrate will be the greater of either the programmed feedrate or the AMP dry run feedrate.

We explain these options in the following sections.

Figure 8.1 shows the check out page as it appears on the screen.

Figure 8.1 - Check Out Page

```
| CHECK OUT              QUICK CHECK | *
| < operator messages>  FEED CHECK | *
|                          DRY RUN | *
|                        DRY Z RUN | *
|                                  | -
|                                  | -
|                                  | -
|                                  | -
| <  error messages  >             | -
```

8.2 Quick Check

Press the [QUICK CHECK] softkey on the check out page to access the quick check mode. All of the program execution modes (QUICK CHECK, FEED CHECK, DRY RUN, and DRY Z RUN) have the same page format, which is equivalent to the auto operate page.

Quick Check involves no axes motion. The part program runs as fast as the control can execute it. Dwells and M functions are ignored during quick check.

Figure 8.2 shows the quick check page as it appears on the screen.

When you access the quick check page, you will see:

- the current program's number after the flashing cursor.
- the sequence number of the last executed command with an N field.
- the current axis positions (machine zero, lag error, program zero, distance to go, or target).

Figure 8.2 - Quick Check Page

```
| QUICK CHECK-IDLE       JOB SETUP | *
| < operator messages>         JOG | *
|                              MDI | *
|  P 153:COMMENTX_DIM              | -
| N    0  MACH ZERO:    PARAMETERS | *
| X  -0.0002  U   0.0000 AUTO-EDIT | *
| Y   0.0002  V   0.0000    STATUS | *
| Z  -0.0005  W  -0.0001  GRAPHICS | *
| <  error messages  >   DIRECTORY | *
```

Quick check allows you to quickly scan a program for programming errors. If the control finds an error during execution, the program will be stopped and the message "EXIT to CLEAR ERROR" appears on the screen.

Press [EXIT] to remove the error message. You can check the status page to determine where the error occurs. Press [STATUS] on the quick check page to call up this page. The status page shows the current block and the status of the modal functions.

8.2.1 Changing Axis Positions Displays

Press and hold [SHIFT] and then press [ENTER] to change the display of the current axis positions.
Each time you press [SHIFT] and [ENTER], the position displays will sequence through the following modal groups.

- **Machine Zero**-- Machine zero is the position of the axes as referenced to machine zero, the default mode of the displays except during program execution. During program execution, the default display is referenced to program zero.

- **Lag Error**-- Lag error is the following error developed by the system due to velocity control. It is the amount that the actual position lags the commanded position. Lag error is useful in system installation.

- **Program zero**-- Program zero is the position of the axis as referenced to an established zero coordinate in a program.

- **Distance to Go**-- Distance to go is the distance needed to reach the final destination of the axis. It is always a positive number that counts down to zero.

Note that not all of these display modes have a meaning for every check out mode. For example, if the axes do not move, the machine zero and lag error displays are not useful.

8.3 Executing a Program

To execute a program, type in the number of the program and press [ENTER], or use the directory to select a program number while on the directory page.

Important: We strongly recommend using quick check before you execute the part program with axis moves.

When you run a program, you can run a program continuously or block-by-block.

If you want to run the program continuously, press the green [CYCLE START] button to start program execution.

If you want to run the program block-by-block, press the yellow [BLK/BLK] button.

8.4 Stopping Program Execution

You can stop program execution by doing any of the following:

- Move the feedrate override switch to 0%. (The spindle continues to rotate). The G84 code inhibits this function. If ACC/DEC is active the control will decelerate the axis to a stop.

- Press the red [CYCLE STOP] button; however, execution may not stop during a G84 tapping cycle, threading cycle, or if inhibited by the execution of a suitably programmed G62. The G84 code inhibits this function. If ACC/DEC is active the control will decelerate the axis to a stop.
- Press the yellow [BLK/BLK] button, if you are in continuous execution. The control finishes the block it is currently executing and stops at the end of the cycle.
- Pressing the [EMERGENCY STOP] button.
- Executing an M0, M01, or M02.
- Press [SHIFT] and [CYCLE STOP] at the same time to abort program execution (Go to IDLE instead of HALTED).

8.5 Resuming Program Execution

To resume program execution:

- If you moved the feedrate override switch to 0%, resume execution by moving the feedrate override switch to a setting other than 0% .
- If you pressed the red [CYCLE STOP] button, resume execution by pressing the green [CYCLE START] button or the yellow [BLK/BLK] button.
- If you pressed the yellow [BLK/BLK] button or executed an M0, M01, or M02, resume execution by pressing the green [CYCLE START] button or the yellow [BLK/BLK] button.
- If you pressed the [EMERGENCY STOP] or the [SHIFT] [CYCLE STOP] button, the program cannot be resumed from the point of interruption..

8.6 Jog-in-auto and Jog-and-return

Jog-in-auto and Jog-and-return operate in Auto Operate and checkout modes, and in Feed Check if the axes are homed. If the control has not been homed the checkout modes treat jog moves like they are an offset being entered. If the control has not been homed upon exiting the jog mode under check out, the message "NO RETURN FROM JOG" is displayed indicating that the operator cannot automatically remove the jog distances under check out. Press [EXIT] to clear the message.

8.7 Feed Check

Press the [FEED CHECK] softkey on the check out page to run a program without axis movements or M functions. Programmed commands are executed at specified feedrates and the front panel override switches are active.

Figure 8.3 shows the feed check page as it appears on the screen.

Figure 8.3 - Feed Check Page

```
| FEED CHECK-IDLE          JOB SETUP | *
| < operator messages>           JOG | *
|                                MDI | *
| P 153:COMMENTX_DIM                 | -
| N     0  MACH ZERO:     PARAMETERS | *
| X  -0.0002  U    0.0000 AUTO-EDIT  | *
| Y   0.0002  V    0.0000     STATUS | *
| Z  -0.0005  W   -0.0001   GRAPHICS | *
| <  error messages  >     DIRECTORY | *
```

Select the program you want to execute as described under Quick Check, if it is not the one already displayed. You will also see the axis position displays. The function of this display can be changed by pressing [SHIFT] and [ENTER] at the same time.

You will also see the sequence number of the last executed command with an N field. You can access the status and graphics pages to check the execution of a program in detail.

Feed check allows you to run a program with the same time frame (programmed F words will be used) as an actual execution, but without moving the axes. This allows you to observe in detail the machine status and positioning without creating potentially hazardous conditions due to part program inconsistencies.

Feed check has all of the error detection capability of quick check and allows you to stop execution at any point to make desired checks.

8.8 Dry Run

Press the [DRY RUN] softkey on the check out page access the dry run mode. You will see the same page features as are found on the quick check and feed check pages. The dry run mode is also similar to the dry Z run described below. But Z does not move.

During dry run, axes will move not slower than the programmed dry run feedrate in AMP and not faster than the programmed rapid rate in AMP. In dry run:

```
dry run feedrate ≤ F ≤ rapid
```

While you are in dry run with the standard PAL, all M codes will be decoded. The output status of decoded M words is PAL dependant.

Figure 8.4 shows the dry run page as it appears on the screen.

Figure 8.4 - Dry Run Page

```
|     DRY RUN-IDLE         JOB SETUP | *
| < operator messages>           JOG | *
|                                MDI | *
|  P 153:COMMENTX_DIM                | -
| N     0  MACH ZERO:     PARAMETERS | *
| X  -0.0002  U    0.0000 AUTO-EDIT  | *
| Y   0.0002  V    0.0000     STATUS | *
| Z  -0.0005  W   -0.0001   GRAPHICS | *
| <  error messages   >    DIRECTORY | *
```

8.9 Dry Z Run

Press the [DRY Z RUN] softkey on the check out page to access the dry Z run mode. You will see the same page features as are found on the quick check and feed check pages. The dry run mode is similar to dry run described above.

Figure 8.5 shows the dry Z run page as it appears on the screen.

Figure 8.5 - Dry Z Run Page

```
|   DRY Z RUN-IDLE        JOB SETUP | *
| < operator messages>          JOG | *
|                               MDI | *
|  P 153:COMMENTX_DIM               | -
| N    0        MACH ZERO:          | -
| X  -0.0002  U   0.0000            | -
| Y   0.0002  V   0.0000     STATUS | *
| Z  -0.0005  W  -0.0001   GRAPHICS | *
| <  error messages  >    DIRECTORY | *
```

During dry Z run all axes move at feedrates that are equal to or greater than a minimum value specified in the control's personality. All M functions are executed; however, functions are somewhat PAL dependant.

The dry run execution provides a quick visual check of programmed movements after thorough checks have been made in quick check and feed check.

The dry run moves have full program error detection. You can check the graphics or status pages of the control at any time during execution.

8.10 Chapter Summary

In this chapter we told you how to check a program for program errors. In chapter 9 we tell you about the auto operation capabilities of the control.

Chapter 9 -- Auto Operate Page

9.0 Chapter Overview

In this chapter we discuss executing the part program. After reading this chapter, you will be able to run a program.

9.1 Auto Operate

After you have entered, tested, and de-bugged a part program, the program is ready for execution on the machine.

We assume that you have prepared the machine properly.

To begin program execution, press the [AUTO OPERATE] softkey on the main menu page.

When you access this page, you will see these options on the screen:

- Job Setup
- Jog
- MDI
- (blank) *
- Parameters
- Auto-Edit
- Status
- Graphics
- Directory

Job Setup, Jog, and MDI are necessary for establishing modal conditions which are reset when you access them through the main menu page.

Parameters information is found in Chapter 20, section 20.11.

Figure 9.1 shows the auto operate page as it appears on the screen.

The auto operate page is identical in format to the check out pages discussed in section 8. It provides all job setup, jog and MDI facilities prior to the execution of a program.

You can check the status of a program, or use the graphics features at any time. The status and graphics features are discussed in chapter 11.

* *Diagnostics Key (allows you to look at the PAL flags during PAL development). This key has the same function as the third softkey on the Support page.*

The auto operate page displays:

- the axis positions (machine zero, lag error, program zero, target, and distance to go). These axis positions are described in sections 5.7 and 8.2.1.
- the sequence number (N).
- the current program name and number (P). You can type in a program number and press [ENTER], or you can select [DIRECTORY] and enter a program number while on the directory page.

Simultaneous Edit and Execution

This feature allows the operator to execute a part program and edit a part program at the same time.

The 6th soft key on the auto operate page is labeled AUTO-EDIT. If a part program is running and the operator selects the AUTO-EDIT soft key, the running part program will continue to execute and the AUTO-EDIT page will be displayed.

The AUTO-EDIT page is very similar to the EDIT page although several of the soft keys are moved and the DELETE ALL and RENUMBER soft keys are not present. Additionally, the AUTO EDIT page provides the operator with LOAD and SAVE soft key selection. This allows the operator to LOAD and SAVE single programs the multi-program LOAD and SAVE feature is not allowed.

Caution should be taken when using the Simultaneous Edit Execute feature that the available memory space is not exhausted. When a program is edited using this feature, the control copies the program in memory to a temporary location. This uses available memory to make the temporary copy.

When the operator returns to the AUTO OPERATE page and no program is currently executing, the old part program that was being edited is deleted and the temporary copy becomes the new part program thereby returning that memory to the amount of available program memory. If several programs are edited before returning to the AUTO OPERATE page, the duplicated program storage will accumulate and the operator should be conscious of the remaining available space.

Figure 9.1 - Auto Operate Page

```
|     AUTO    -IDLE       JOB SETUP | *
| < operator messages>          JOG | *
| >P  12:PARA_NEST1             MDI | *
|                                   | *
| N    0  MACH ZERO:     PARAMETERS | *
| X  -0.0002  x  -0.0003 AUTO-EDIT  | *
| Y   0.0000                 STATUS | *
| Z   0.0000  W  -0.0003   GRAPHICS | *
| <  error messages  >    DIRECTORY | *
```

9.2 Power-up Conditions

When you are in the operate mode, prior to executing the part program, the powerup conditions are assumed. These conditions are:

- all axes will be in absolute
- G01, Linear Interpolation is assumed
- circular Interpolation plane is returned to XY
- cutter compensation cancelled
- fixture and tool offsets are cancelled (shown as 0)
- tool selection is shown as 0
- override inhibits are cancelled
- feedrate will default to the value specified in AMP
- mirrors are turned off
- inch/metric is **not** modified; the inch/metric mode will be as shown on the job setup page.
- program zero is **not** modified.

9.3 Executing a Program

To execute a program, type in the number of the program and press [ENTER], or use the directory to select a program number.

You run a program continuously or block-by-block by:

- pressing the green [CYCLE START] button for continuous execution.
- pressing the yellow [BLK/BLK] button for block-by-block execution.

The axes and spindle movements respond according to the settings of the front panel's override switches.

When you are in the auto operate mode, it is important to notice the green, yellow, and red lights on the front panel of the control. Table 9.A indicates the meanings of these lights if they are lit.

9.3.1 Executing a Program From An External Device

You can run a program directly from an external device. If sub programs are used they must be stored in RAM.

The execute from I/O feature can run up to a maximum baud rate of 9600. 2400 baud is typical. The control buffer holds about 1K (16 blocks of 64 characters) of commands that it continually fills as commands are executed. The buffer is made of two storage areas so it will not be emptied and have to wait for the I/O device to fill it again.

The required configuration for selected I/O devices is found in chapter 12 section 12.4.2 (I/O Configuration page).

Important: If you want the program to access information stored in the control you must be sure the information has been entered. This information previously stored in the control may include sub programs, fixture offsets, tool offsets, radius compensation, etc.

To run from an external device, follow this procedure:

1. Set up the device. Program the parameters into the device you select to run the program from. This example is for the Decitek tape reader 8000-XRDR. See the required switch settings for Decitek tape reader 8000-XRDR in section 12.4.2.

2. Select the external device from the Load/Save page. See section 11.2 for more information on Load Select.

3. Go to the Auto Operate page and select P0 as the program to be run.

The message "EXTERNAL" will be displayed next to P0.

4. Connect your I/O device to the RS232 port on the control. See the Installation manual.

5. Select Status or Graphics pages and press the [START CYCLE] push button.

The I/O device will begin sending the program to the control. The control will start executing the program as soon as it has enough information in the buffer (about 1 second).

The control will access previously stored information if the program requests it. This information, previously stored in the control may include sub programs, fixture offsets, tool offsets, radius compensation, etc.

9.3.2 PAL Prog. Selection

The PAL may select a program for execution by setting the PAL 'L' Arithmetic Variable 'L007' to the desired program number.

If L007 is set to a number in the range of 1 through 250, the corresponding program number is selected. If no such program exists, it will still select, with the name displayed as '* NEW PROG *'. Attempting to execute a nonexistent program will give an error message of INVALID PROGRAM NO. If the operator attempts to change the Program Number while the PAL is forcing a number, as soon as he presses [ENTER], the number will change to the PAL selected number.

9.6 Jog-In-Auto and Jog-and-Return

The jog-in-auto feature allows you to interrupt the part program during execution, to jog the tool away from the part. This move may be necessary for tool inspection or replacement or for part inspection. You can then return the tool automatically to the point where you interrupted the program, by using the jog-and-return feature.

Jog-and-return allows you to return the axes and continue the program. Complete the following steps for jog-in-auto and jog-and-return:

1. Press [CYCLE STOP] to stop execution as soon as possible or press [BLK/BLK] to stop at the end of the current cycle.

2. Press [EXIT] to go to the auto operate page if you are not already there.

3. If you want the spindle off, turn it off by programming an M5 in MDI or with the manual switch. If you don't have a switch go to MDI.

4. Access the JOG page from the auto operate page and then access the desired jog page.

5. Jog the tool away from the part to the desired location.

   Take extreme care in selecting the axes and directions first jogged since the cutter may come in contact with the work.

6. Perform the function you interrupted the program for (for example, tool or part inspection).

7. Press [EXIT] enough times to return to the auto operate page.

   The message "RETURN JOG ON START" should appear on the screen.

8. Press [CYCLE START] to cause the control to return the axes to the location they were when the jogging started.

   The Z axis goes, at rapid, to its machine zero position first; the other axes then reposition simultaneously at rapid to the point where they were when execution was halted; and then the Z axis feeds down to the position where it was when execution was halted.

CAUTION: The (step 8) axis return sequence of the Jog-In-Auto feature does not duplicate the jog moves used to move the tool away from the part (Step 5). You must determine if the Z axis rapid move to machine zero and the subsequent simultaneous rapid moves of the other axis jogged in Step 5 are safe for your machine and the machining job in progress.

The Jog-In-Auto feature always makes an axis rapid move to machine zero regardless of whether or not the Z-Axis was jogged in Step 5.

9. If you turned the spindle off in step 3, turn it back on.

10. Press [CYCLE START] to continue execution of the program.

If you do not want to return from jog, you can skip step 8 and in step 10 press [SHIFT] and [CYCLE START]. This will cause the program zero location to be shifted by the jogged amount and program execution to be resumed from the current location.

You cannot use the jog-in-auto or the jog-and-return features during MDI or Mid Program Start.

9.7 Chapter Summary

In this chapter, we discussed running a program using the auto operate functions of the control. In the next chapter we explain the graphics capabilities of the control.

Chapter 10 -- Graphics and Status Page

10.0 Chapter Overview

This chapter gives you an overview of the graphics capabilities available on your control.

After you read this chapter, you will know about these graphic functions:

- erasing a current display
- setting a viewing plane
- using the grid in programming and operations
- using the window functions
- examining a part program plot in detail ("zoom" functions)

10.1 An Overview of Graphics

The graphic functions of the control allow you to visually inspect a part program before you enable axis motion. You can also use graphics to display the tool path while the the axes are in motion. We recommend you use check out with graphics, specifically quick check. This is simply an exercise to ensure that you have not overlooked any needed movements, or that you have not made any mistakes in programming.

You can access the graphics capabilities in four ways:

- through the MDI page
- through the checkout page
- through the auto operate page
- through List or Prompt Edit

Through the MDI page

1. Press the [MDI] softkey on the manual operate page to call up the MDI Page.
2. Press the [GRAPHICS] softkey on the MDI page to call up the graphics page.

Through the checkout page:

1. Press the [CHECK OUT] softkey on the main menu page to access the check out page.

   When you see this page on the screen, there will be four options:

   - Quick Check
   - Feed Check
   - Dry Run
   - Dry Z Run

2. Select one of the options. We recommend quick check to set-up and use graphics.

3. Press the [GRAPHICS] softkey on any of the options shown to call up graphics.

Through the auto operate page:

1. Press [AUTO OPERATE] on the main menu page.

2. Press [GRAPHICS].

Through List or Prompt Edit:

1. Press [GRAPHICS].

When you access graphics and start to execute a part program, you will see one line of text at the top of the screen. This line displays the current program block.

At the bottom of the screen, you will see two lines. Reading from left to right, the top line at the bottom of the screen shows:

- the left boundary of the screen
- the axis that will be used as the horizontal axis with respect to the screen

- the current (Machine Zero) coordinate of that axis
- the right boundary of the screen
- the operator message field

The bottom line shows, from left to right:

- the lower boundary of the screen
- the axis that will be used as the vertical axis with respect to the screen
- the current (Machine Zero) coordinate of that axis
- a message indicating the current machine state (running, halted, idle, feedhold)
- the error message field

There are actually two graphic pages. We will explain each in detail in the following sections.

The first graphics page, or "viewing page", is shown in figure 10.1. It is mainly used to examine a part program plot after the settings have been selected in the second graphics page, or "set window page." The viewing page also lets you overlay a grid and display rapid moves.

The set window page is shown in figure 10.2. It is mainly used for setting limits for the viewing page. This set window page also provides a frame that can be made smaller or larger to view a part program in more detail.

10.2 First Graphics Page (Viewing Page)

When you first enter graphics, you will see other options. These are explained in the following paragraphs.

Figure 10.1 shows the viewing page as it appears on the screen.

Figure 10.1 - Viewing Page

```
 ;COMMENTX_DIM,I_I_INC#
                                                                ERASE  | *
                                                                       |
                                                                       | -
                                                                       |
                                                                       | -
                                                                       |
                                                                       | -
                                                                       |
                                                               STATUS  | *
                                                                 0.00  |
                                                                 GRID  | *
                                                                       |
                                                                RAPID  | *
                                                                  SET  |
                                                               WINDOW  | *
 >    0.0  X  -0.0002      0.0        < operator messages>     WINDOW  |
      0.0  Y   0.0002    IDLE         <  error messages  >        1    | *
```

- **[ERASE]**-- Press the [ERASE] softkey on the viewing page to erase the graphics display. The erase prompt flashes if the control knows you need to erase the display. If erase is flashing, it generally means the screen limit coordinates or plane no longer correspond to the graphics plot as displayed on the screen.

- **[STATUS]**-- Press the [STATUS] softkey to select the status page. The status page gives you current the information on the operation modes of the control. See section 10.6.

- **[GRID]**-- Each time you press the [GRID] softkey, the mode is toggled to show if the grid is active. If the grid is active, "GRID" appears in reverse video and grid lines appear on the screen. To erase the grid lines, first deactivate the grid, then press [ERASE]. All graphics plots and grid lines will be erased..

Grid resolution is automatically determined by the screen limits. The resolution of the grid is shown above the [GRID] softkey. (On color systems, the grid lines are displayed in blue.)

Table 10.A indicates grid line units as determined by window sizes. Inch units are shown. (Metric units are ten times that shown).

Table 10.A
Grid Line Units

Window Size Range (H)			Units per Grid Square
		.1	.01
		.2	.02
.3	to	.6	.05
.7	to	1.3	.1
1.4	to	2.9	.2
3.0	to	6.9	.5
7.0	to	13.9	1.0
14.0	to	29.9	2.0
30.0	to	69.9	5.0
70.0	to	139.9	10.0
140.0	to	299.9	20.0
300.0	to	699.9	50.0
700.0	to	1399.9	100.0
1400.0	to	2800.0	200.0

H = horizontal max. - horizontal minimum range

Table 10.A continued
Grid Line Units

Example: X is horizontal axis selected for graphics screen.

Left limit = -3.00", right limit = 5.00"

H = max-min
= 5 - (-3)
= 8

Now the value 8 is in the range of 7.0 to 13.9; therefore, the grid size = 1.00"

- **[RAPID]**-- Press the [RAPID] softkey to display rapid moves. If "RAPID" is off, only feed moves are displayed. Each time you press [RAPID], the mode toggles on off. When rapid moves are enabled, "RAPID" appears in reverse video. If you press [RAPID] while a program is running, only the rapid moves that occur after you press the key are shown.

  On color systems, the rapid moves are displayed in red and the feed moves are displayed in green.

- **[SET WINDOW]**-- Press [SET WINDOW] when you want to establish a setting for a window. A window (there are 4 available) consists of a plane defined by an axis pair and screen limit coordinates for that axis pair. When you press this key, the control will advance to the set window screen, described in section 10.3.

- **[WINDOW #]**-- Press the [WINDOW #] softkey to step through four available windows. Each time you press [WINDOW], you will be on one of the four available windows. The windows are covered later in this chapter sec 10.4.1

10.3 Second Graphics Page (Set Window Page)

When you press [SET WINDOW], the control advances to the next graphics page. When you access this screen you will see other options. These are explained in the following paragraphs.

Figure 10.2 shows the set window page as it appears on the screen.

Figure 10.2 - Set Window Page

```
| ;COMMENTX_DIM,I_I_INC#                                          |
|                                                          ERASE  | *
|                                                                 |
|                                                        AUTO SZ  | *
|                                                                 |
|                                                        ZERO SZ  | *
|                                                                 |
|                                                         LIMITS  | *
|                                                                 |
|                                                                 | -
|                                                                 |
|                                                                 | -
|                                                                 |
|                                                         LARGER  | *
|                                                                 |
|                                                        SMALLER  | *
| >    0.0  X  -0.0002     0.0      < operator messages>  WINDOW  |
|      0.0  Y   0.0002   IDLE       <  error messages  >       1  | *
```

- **[ERASE]**-- Press the [ERASE] softkey on the graphics page to erase the graphics display.

  The erase prompt flashes if the control knows you need to erase the display. If erase is flashing, it generally means the screen limit coordinates or plane no longer correspond to the graphics plot on the screen.

- **[AUTO SZ]**-- Press the [AUTO SZ] softkey to tell the control to automatically expand the screen limits if the graphics plot extends beyond the currently active limits. "AUTO SZ" appears in the reverse video when it is active.

- **[ZERO SZ]**-- Press [ZERO SZ] to force all screen limits to zero. Auto size will automatically be activated. Zero Sz shrinks the screen limits to minimal values. When you start program execution, you will see a small rectangular frame expanding on the screen. As the program is executed, this frame becomes larger. The control will automatically set the minimum limits for the part program plot.

ZERO SZ is allowed only in QUICK CHECK and FEED CHECK.

- **[LIMITS]**-- Press the [LIMITS] softkey to force the screen limits equal to the soft limits of the axes. The limits are displayed on the bottom left corner of the screen.

- **[LARGER]** or **[ZOOM OUT]** -- Press the [LARGER] softkey or [ZOOM OUT] key to move the horizontal maximum frame dimension by one grid line larger and the vertical maximum frame to maintain the proper frame proportions. The plotted frame and the horizontal maximum dimension change accordingly.

  The ZOOM OUT function works only on the set window graphics page.

- **[SMALLER]** or **[ZOOM IN]**-- Press the [SMALLER] softkey or the [ZOOM IN] key to move the horizontal maximum frame dimensions by one grid smaller and the vertical maximum frame to maintain the proper frame proportions. The plotted frame and the horizontal maximum dimension change accordingly.

  The ZOOM IN function works only on the set window graphics page.

  If you press and hold the [SHIFT] key then press the cursor keys, you can move the entire frame up or down, left or right, by one grid line each time you press the cursor keys.

- **[WINDOW]**-- Press the [WINDOW] softkey to step through the four available windows. The selected window number is displayed along with its window dimensions. A window consists of an axis pair and corresponding limits.

10.4 How to Use Graphics

Now that we've explained the different graphic functions, we'll cover what steps to complete to graphically display a part program.

10.4.1 After Initial Power Up

When you first power up the control, you must complete these steps, assuming you have entered a part program:

1. Press the [CHECK OUT] softkey on the main menu page to access the quick check page.
2. Press [QUICK CHECK] on the check out page.
3. Press [GRAPHICS] on the quick check page.
4. Press [SET WINDOW].

You should now be on the set window graphics pages, ready to complete the following steps to graphically display a part program:

1. Decide your window by pressing the [WINDOW] key. There are 4 possible windows. Each time you press the key, it sequences through the windows.
2. Choose the desired plane. This is done by positioning the cursor on the horizontal axis letter using the cursor arrows. Pressing [ENTER] then sequences the planes through four axes pair choices: XY, XZ, ZY, AND YZ. The top axis, as displayed on the bottom of the screen, represents the horizontal axis on the screen. The bottom axis represents the vertical axis of the screen.
3. Press [LIMITS]. The limits of the screen are set to the soft travel limits and are displayed on the bottom left side of the screen.

If you want the screen limits left at these numbers, you may [EXIT] to the viewing page and proceed to step 8.

4. Press the [ZERO SZ] softkey. After this is done the "AUTO SZ" will be reverse video.
5. Press [ERASE] to erase any old graphics plot.

6. Press [CYCLE START]. You will see a dotted frame (blue on color systems) "growing" around the part program plot. This frame represents the smallest window on grid multiples which will encompass the entire part plot.

As the frame size is being computed, some graphics plotting may not appear on the display. If you wish to fill in the gaps, press [CYCLE START] again to reexecute the program. Since no enlargening of the frame should be required the second time, the plot should show complete.

7. When the program has completed, determine whether you want the limits of the viewing screen to correspond to the frame size as displayed.

   If you do not, use the [LARGER] and [SMALLER] soft keys or the [ZOOM IN] and [ZOOM OUT] keyboard keys to set the size of the frame and the shifted cursor arrows to reposition it.

   Then [EXIT] to the viewing page. This will store the screen limits and plane for the current window in accord with the current frame size and plane.

8. Press [ERASE].

9. Press [CYCLE START]. You will now see the part program plot to the new window size.

10.4.2 How to Change a Window Setting

Once a window has been set, its plane and boundary setting will remain in memory until changed. To change a window you can:

- Directly enter a new value(s) for any of the 3 screen boundaries.

- Select a different plane for that window.

- Go to the second graphics page.

Directly enter a new value(s) for any of the 3 screen boundaries.

This also can be done from either the viewing or set window page.

To use this method, first use the cursor keys to position the cursor left of the boundary value you want to change. Now type in the desired value. You will see that field in reverse video as you type in the new value. Now press the [ENTER] key. The new value will be entered and displayed in normal video.

Select a different plane for that window.

This can be done in either the viewing or set window page. When you select a new plane for a window, the boundaries of the window default to the soft limits.

Go to the second graphics page.

Change the window plane and the limits using the [AUTO SZ], [ZERO SZ], [LIMITS], [LARGER], [SMALLER], or [SHIFT] cursor functions as described above.

If you are on the set window page, and you change the window settings by a method other than the [LIMIT] softkey or plane selection, the part program plot will still be displayed according to the old window settings.

If you want to view the plot under the new settings, press [EXIT] to return to the viewing page. If the program is now run it will be displayed according to the new settings. When you exit to the viewing page the new window settings are stored in memory.

10.4.3 How to Examine a Part Program in Detail

To examine a part program in detail, complete these steps:

1. Plot the part as described above.
2. Press the [SET WINDOW] softkey. This will take the control to the set window page and a frame will appear around the plot.

Adjust the size and position of the frame to enclose the area you would like to view.

You can make a frame smaller or larger by pressing the [SMALLER] or [LARGER] softkeys. The frame size can also be controlled by the [ZOOM IN] and [ZOOM OUT] keys.

You can use the shifted cursor keys to reposition the frame.

3. Press [EXIT] to return to the viewing page.
4. Press [ERASE] to remove the old plot.
5. Press [CYCLE START] to replot the portion you enclosed in the frame which now corresponds to the entire viewing screen.

10.5 Status Page

The status page, which is accessible through auto operate, checkout, or graphics, informs you of the information reflecting the status of the operation currently being performed.

Figure 10.3 shows the status page as it appears on the screen.

Figure 10.3 - Status Page

```
STATUS                 ABS/inc:XYZWVUIJKABCR  COMMENTX_DIM P 153
SPIN/COOL:CNC ONLY     Dimensions:INCH        Program 153  L   1
                      Interp Mode:
                       Path Mode :CONTINUOUS N     0  IPM   30.0
                      AutoRoutine:           G  0 M  0    FX  0
                        AutoCycle:           T  0 0  0 D   0.00
                       Circ Plane:XY
                      Cutter Comp:                            0
                      Ovr Inhibit:                            0
                                             U   0.0000 M  I  H
IDLE                                         V   0.0000 M  I  H
                                             W  -0.0001 M  I  H
< operator messages>                         X  -0.0002 M  I  H
<  error messages  >                         Y   0.0002 M  I  H
                                   MACH ZERO:Z  -0.0005 M  I  H

 ;COMMENTX_DIM,I_I_INC#
 G24I2.5J-2.125R.6A25.##
```

Status Page Messages Left Side

The left portion of this page displays:

- PAL status messages -- up to 8 rows of status messages generated by PAL can appear here
- Operating Status -- the control's operating status (RUNNING, IDLE, HALTED, FEEDHOLD, FAULT)
- Operator and error messages -- these appear directly below the control's operating status.
- Program blocks -- displays program data blocks during execution, on the bottom two lines. The next to the bottom line shows the current block. The bottom line shows the blocks immediately following it.

Status Page Messages Middle

The middle portion of the screen shows the control's programmed state:

- ABS/inc -- the top line displays absolute and incremental parameters. Parameters shown in upper case are absolute. Parameters in lower case are incremental.
- Dimensions -- inch or metric
- Interpolation Mode -- Rapid (G00), linear (G01), position (G73), helical (G22,G23), or circular (G02,G03).
- Path mode -- continuous, point to point, rapid, blend, or arc tangent (G05). See chapter 21.
- Autoroutine -- the name of any current autoroutine (Step/repeat, post, pocket, etc.
- Autocycle -- the name of any current autocycle (drill, bore, etc).
- Active plane -- indicates the active plane, XY (G17), (G18), and (G19)

- Cutter compensation -- indicates whether cutter compensation is active left (G41), active right (G42), or cancelled (G40).

- Override inhibit -- the status of the overrides: Q means the BLK/BLK and CYCLE STOP buttons are disabled, S means the spindle override switch is disabled, and F means the feedrate override switch is disabled.

Status Page Messages Right Side

The upper right portion of the screen will show:

- Active Program -- At the top right corner, the active program (or subprogram) number.

- Program -- to the right of the word "program" is the main program number, above it is the main program name.

- L -- the number of times the active program (or subprogram) has yet to be executed. If it is to be repeated 3 times, for example, the number 3 appears here first, then counts down to one.

- N -- the most recent sequence number

- G -- the most recent preparatory code number

- M -- the most recent miscellaneous function

- T -- active tool number

- O -- active tool offset number

- FX -- current fixture offset number

- IPM, MMPM -- the currently active feedrate (as affected by the feedrate override value).

- D -- current dwell value

Status Page Messages Lower Right

The lower right portion of the screen shows axis displays.

- Axis name -- the shows the axis name (X, Y, Z, V, U, W) followed by it absolute coordinates.
- M -- Axis mirrored, reverse video if active.
- I -- axis in position, reverse video if true.
- H -- axis homed, reverse video if true
- User selectable display:

  Mach Zero -- this displays the axis position relative to machine zero

  Prog Zero -- this displays the axis location relative to program zero

  Lag Error -- This displays the amount the axis is following the commanded position

  DTG -- This displays the distance from the current axis location to the end of the move

10.6 Chapter Summary

In this chapter we discussed the graphic capabilities and the status page of the control. In the next chapter we tell you how to load and save a part program.

Chapter 11 – Load/Save Page

11.0 Chapter Overview

In this chapter we describe the program input and output capabilities of the control. This assumes the control is interfaced to an external device.

We will examine how to:

- save a program or other data to a storage device
- load a program or other data from a storage device
- verify saved data
- use the multi save and multi load functions

11.1 Input/Output Overview

The control can be interfaced to an external device that receives or transmits data in serial fashion. The Mode of transmission may be RS232 or RS422 depending on system configuration at installation. The baud rate of transmission is also set at the time of system installation. See chapter 12 for setting up the control.

The control has a standard configuration included that interfaces an Allen-Bradley audio cassette interface. Up to 8 additional input and/or output devices can be configured. The operator can select which device is currently attached to the control using softkeys. The control automatically assumes the correct configuration for the device.

The input code conforms to EIA standards: RS 244-A (standard EIA code) or RS 358 (ASCII code). The RS244 accepts upper case only. The output code conforms to RS 358 (ASCII code), only.

Each character in EIA code has odd parity; that is, it has an odd number of 1's(in a binary representation) in the code representing each character. ASCII normally has even parity, or an even number of 1's in the code representing each character. The control can be set up for odd parity, even parity or no parity code on input. The control looks for the first end of block (EOB) character during input, determines the code being used then checks for parity errors, errors in coding, invalid characters, or damaged media.

All part programs prepared externally on tape should begin with a line feed character immediately followed by a carriage return (treated internally as an EOB (#)). Subsequent coding on tape should conform to the characters, word addresses and data block formats presented in this manual. For more information on tape preparation, see chapter 12. For more information on proper control/tape reader set-up modes, see section 12.4.

11.2 Load/Save Overview

To begin I/O operations, press the [LOAD/SAVE] soft key on the main menu page. When you access this page, you will see other options. Each of these options has a different meaning and is explained in the following sections.

Figure 11.1 shows the load/save page as it appears on the screen.

Figure 11.1 - Load/Save Page

```
| LOAD/SAVE        CASSETTE II SAVE | *
| < operator messages>       VERIFY | *
|                 TAPE READER LOAD  | *
|                                   | -
|  Program 153 COMMENTX_DIM         | -
|                                   | -
|                                   | -
|                       SAVE SELECT | *
| <  error messages   > LOAD SELECT | *
```

Note that you can enter a program number while you are on the load/save page.

On most of the pages you select from the load/save page, you will see additional fields as follows.

A 'Counter' field indicates how many characters have been loaded or are to be saved. It is mainly a status/activity indication.

An 'Available' field indicates how many total characters of program space remains. Note that individual programs are restricted to a little over 64,000 characters even if there is more total program space available.

The 'Flags' field lists the active flags associated with the selected program. These have the following meaning:

- **S** = means you have selected that program to be saved using multi save, or loaded using multi load.
- **C** = indicates a checksum error occurred during program load. This flag clears when you edit the program.
- **L** = indicates a load error occurred during program load. This flag clears when you edit the program.
- **D** = indicates duplicate program. A program was loaded that has a number that duplicates one already in memory.
- **T** = temporary program. There isn't yet a program in memory with that number, but the number was chosen for multi load. This flag clears when you exit the load page.
- **R** = means the program is restricted; a program cannot be displayed or edited, renumbered or deleted unless you have access to the Program Protect function. A multi save operation that attempts to save all programs will not save restricted programs.
- **P** = means the program is protected; a program can be displayed, but it can't be edited, renumbered or deleted.

Load/Save Page Softkeys

The softkeys on the load/save page have the following meanings:

- [SAVE]-- Press the [SAVE] softkey on the load/save page to access the save page, as shown in figure 11.2. The name of the save device is shown just to the left of the word "SAVE".

The features of this page are explained in section 11.3.

- **[VERIFY]**-- Press the [VERIFY] softkey on the load/save page to call up the verify page, as shown in figure 11.4. The features of this page are explained in section 11.5.

- **[LOAD]**-- Press the [LOAD] softkey on the load/save page to call up the load page. Figure 11.3 shows the load page. Select the load function to load programs from the selected input device. The name of the load device is shown just to the left of the word "LOAD". The features of this page are explained in section 11.4.

- **[SAVE SELECT]** -- Press the [SAVE SELECT] softkey to advance to the next SAVE device. The name of the device appears just to the left of "SAVE" labels the first softkey.

  The device, including its name, is defined under the support menu on the I/O configuration screen. Devices that are not configured for SAVE are skipped.

- **[LOAD SELECT]** – Press the [LOAD SELECT] softkey to advance to the next LOAD device. The name of the device is displayed just to the left of "LOAD" that labels the third softkey.

  The device, including its name, is defined under the support menu I/O configuration screen. Devices that are not configured for LOAD are skipped.

11.3 Save Page

Press the [SAVE] softkey on the load/save page to access the save page as shown in figure 11.2.

Important: Before you press any of the keys on this page:

- make sure you have chosen the appropriate save device by pressing [SAVE SELECT] on the load/save page. The device name appears next to [SAVE] on the load/save page and at the upper left of the SAVE page.

- make sure the proper cable is connected to the I/O port of the control.

- make sure the device is ON and properly configured for accepting output data from the control. See chapter 12 to properly configure the save device.

Figure 11.2 - Save Page

```
|   CASSETTE II SAVE       FIXTURES | *
| < operator messages>TOOL OFFSETS  | *
|                              AMP  | *
|  COMMENTX_DIM                     | -
|  Program      153  MULTIPLE PROG  | *
|  Counter        0    SINGLE PROG  | *
|  Size          22      NEXT PROG  | *
|  Flags          P           FEED  | *
| <  error messages  >   DIRECTORY  | *
```

Save Page Softkeys

Each of these features has a specific function as explained below.

- **[FIXTURES]**-- Press the [FIXTURES] softkey to save the fixture offset table on the selected output device. The Fixture Zero and Last Fixture Number fields are also saved.

- **[OFFSETS]**--Press the [OFFSETS] softkey to save the tool offset table to the selected output device.

- **[AMP]**-- Press the [AMP] softkey to save the AMP table on the selected output device. The axis calibration information is also saved.

- **[MULTIPROG]**-- Press the [MULTI PROG] softkey to go to the multi save page. When you press this key, you will see the screen shown in figure 11.5. The features of the page are explained in section 11.6. This page lets you save multiple programs as a single set, so you can reload them as a set rather than one by one. Note: programs saved under the MULTIPROG function cannot be loaded using the single program feature.

- **[SINGLE PROG]**-- Press the [SINGLE PROG] softkey on the save page to save the current program to the selected output device.

- **[NEXT PROG]**-- Press the [NEXT PROG] softkey to advance the current program number to the next one in storage.

- **[FEED]**-- Press the [FEED] softkey to output a length of leader or trailer, or a form feed character to the selected output device. The amount of leader and the character generated is determined by the I/O configuration page. See chapter 12 for additional information.

- **[DIRECTORY]**-- Press the [DIRECTORY] softkey to call up the directory page. You can enter a program number from the directory by pressing the number then [ENTER]. The control will automatically return to the save page.

11.4 Load Page

The [LOAD] key calls up the load screen, as shown in figure 11.3. Use the load function to load programs from the selected input device.

Figure 11.3 - Load Page

```
|   CASSETTE II LOAD        FIXTURES |  *
| < operator messages>TOOL OFFSETS   |  *
|                              AMP   |  *
|  COMMENTX_DIM           NEW PROG   |  *
|  Program      153  MULTIPLE PROG   |  *
|  Counter        0    SINGLE PROG   |  *
|  Available 143926    INCR PROG #   |  *
|  Flags               ACCEPT ERRS   |  *
|_<  error messages  >   DIRECTORY_  |  *
```

Important: Before you press any of the keys on this page, make sure you:

- have chosen the appropriate load device by pressing [LOAD SELECT] on the load/save page. The device name appears next to [LOAD] on the load/save page and at the upper left of the load page.
- make sure the proper cable is connected to the I/O port of the control.
- make sure the device is ON and properly configured for providing input data to the control. See chapter 12 to properly configure the device.

Load Page Softkeys

The softkeys on this page have the following functions:

- **[FIXTURES]**-- Press the [FIXTURES] softkey to initiate the loading of the fixture offset table from the selected input device.
- **[TOOL OFFSETS]**-- Press the [TOOL OFFSETS] softkey to load the tool offsets table from the selected input device.
- **[AMP]**-- Press the [AMP] softkey to load the AMP table from the selected input device. This also loads the axis calibration information

Important: You can only load AMP if the drives are off, and the logged on user has access to AMP via the access control page.

- **[MULTIPLE PROG]**-- Press the [MULTIPLE PROG] softkey to get the screen shown in figure 11.5 and access the multi load function. This lets you load programs as a set rather than one by one.

  Multi load loads only selected program (those with "S" flags set) or, if none are selected, every program is loaded. Every program from the device data format must conform to multi-program format. See chapter 12 for additional information.

Important: programs saved under the MULTIPROG function cannot be loaded using the single program feature.

- **[SINGLE PROG]**-- Press the [SINGLE PROG] softkey to load a program from the selected input device into the program number shown on the load page.

- **[INCR PROG #]** -- Press the [INCR PROG #] softkey to increment program numbers to the number for the program you want to load. When an available number (unused in the directory) appears, "NEW PROG" appears in the program name space. Single program load inputs a program under the number shown on the load page. Multi load always loads back into the same program numbers selected for save.

- **[ACCEPT ERRS]**-- Press the [ACCEPT ERRS] softkey to toggle the mode on (indicated by reverse video) and off. This function allows you to load a program even if it has errors.

  When [ACCEPT ERRS] is on, errors are ignored and the program is loaded, any error messages will still appear, and the L and C flags may appear. If the PROGRAM SPACE FULL error message appears, the portion of the program that sucessfully loaded will be in memory.

  If [ACCEPT ERRS] is off, detected errors will prevent you from loading the program, the load operation is terminated, and appropriate error messages will appear.

  Important: If there are errors in AMP, tool offsets, or fixture offsets, the control will abort the operation and error messages will appear even if [ACCEPT ERRS] is on.

- **[DIRECTORY]**-- Press the [DIRECTORY] softkey to access a directory of all the programs stored in memory. Type in a number and press [ENTER] to make the program number the one shown on the load page. The control will automatically exit back to the load page.

11.5 Verify Page

Verify works similar to the load functions except that data is not loaded. Rather, the incoming data is tested for proper checksum or is compared against memory, depending on the storage format. This function verifies that an error-free copy has been saved to the selected output device.

Press the [VERIFY] softkey on the load/save page to access the page shown in figure 11.4.

Figure 11.4 - Verify Page

```
| CASSETTE II VERIFY       FIXTURES | *
| < operator messages>TOOL OFFSETS | *
|                              AMP | *
|  COMMENTX_DIM                    | -
|  Program      153  MULTIPLE PROG | *
|  Counter        0    SINGLE PROG | *
|  Available 143926    INCR PROG # | *
|  Flags                           | -
| <  error messages  >   DIRECTORY | *
```

Verify Page Softkeys

The softkeys on this page have the following functions:

- **[FIXTURES]**-- Press the [FIXTURES] softkey to initiate verifying the fixture offset table from the selected input device.

- **[TOOL OFFSETS]**-- Press the [TOOL OFFSETS] softkey to verify the tool offsets table from the selected input device.

- **[AMP]**-- Press the [AMP] softkey to verify the AMP table from the selected input device.

- [MULTIPLE PROG]-- Press the [MULTIPLE PROG] softkey to initiate a checksum verification of the selected program from the input device.

  This checks the checksum of only those programs in multi-program format. See chapter 12 for additional information on multi-program format. Programs do not need to be in memory to use multiple program verify.

- [SINGLE PROG]-- Press the [SINGLE PROG] softkey to verify a program from the selected input device.

- [INCR PROG #] -- Press the [INCR PROG #] softkey to increment program numbers and reach the number you want.

- [DIRECTORY]-- Press the [DIRECTORY] softkey to access a directory of all the programs stored in memory. Type in a number and press [ENTER] to make the program number the one shown on the verify page. The control will automatically exit back to the verify page.

11.6 Multi-Program Save

When you press the [MULTI PROG] softkey on the save page, you will see more options. These are explained in the following paragraphs. Use multiprogram save when you want to save a series of programs that are stored in memory (those with "S" flags set). You can also use multi save to save a single program. When you use multi save, the program number of each program is also saved so that the program will automatically load with that number.

On this page, you can type in the program number you want to save, or you can press [NEXT PROG] to step through the directory or you can select the directory page and enter the program number there..

The programs are saved in the order they are in memory, not necessarily in numerical order, or in the order they were selected.

Figure 11.5 shows the multi save page as it appears on the screen.

Figure 11.5 - Multiple Program Save Page

```
|   CASSETTE II MULTI SAVE   START | *
| < operator messages>             | -
|                                  | -
|  COMMENTX_DIM         NEXT PROG  | *
|  Program      153   SELECT PROG  | *
|  Counter        0 NEXT SELECTED  | *
|  Size          22   DELETE PROG  | *
|  Flags         SP    DELETE ALL  | *
|_<  error messages  >  DIRECTORY  | *
```

Multiple Program Save Page Softkeys

These functions are explained below:

- **[START]**-- Press the [START] softkey on the multi save page to save all of the programs you have selected. If you have not selected any programs, all non-restricted programs are saved.

- **[NEXT PROG]**-- Press the [NEXT PROG] softkey on the multi save page to advance the current program number to the next program in numerical order.

- **[SELECT PROG]**-- Press the [SELECT PROG] softkey to select or deselect the current program number. The "S" flag appears in the flags field when the program is selected.

- **[NEXT SELECTED]**-- Press the [NEXT SELECTED] softkey to advance the program number to the next selected program in the directory. Use this function to determine if the desired programs have all been selected.

- **[DELETE PROG]**-- Press the [DELETE PROG] softkey to delete the current program. After you press this key, a screen will appear. Press the [YES] softkey if you want to delete the program. Press [NO] if you don't want to delete the currently active program. The control will not let you delete a protected program ("P" flag set).

- **[DELETE ALL]**-- Press the [DELETE ALL] softkey to delete all unprotected programs. A message will appear asking you whether want to delete all the programs in the directory. Press the [YES] softkey if you want to delete all programs. Press [NO] if you don't want to delete all programs You cannot delete protected programs.

- **[DIRECTORY]**-- Press the [DIRECTORY] softkey to access a directory of all the programs stored in memory. Type in a number and press [ENTER] to make that number the be the current one on the multi save page. The control will automatically exit back to the multi save page.

11.7 Multi-Program Load

Multi-program load works similar to the load function, except that you are loading a series of selected programs that have been saved with multi save operations. The functions of multi load are explained below. Figure 11.6 shows the multi load page as it appears on the screen.

Figure 11.6 - Multi-Program Load Page

```
|   CASSETTE II MULTI LOAD   START | *
| < operator messages> ACCEPT ERRS | *
|                                  | -
|  * NEW PROG *                    | -
|  Program       153   SELECT PROG | *
|  Counter         0 NEXT SELECTED | *
|  Available 143926    DELETE PROG | *
|  Flags          SL    DELETE ALL | *
| <  error messages  >   DIRECTORY | *
```

Multiple Program Load Page Softkeys

The Multi-Program Load Page functions are explained below:

- **[START]**-- Press the [START] softkey to load all of the "selected" programs from the input device. If there are no selected programs, all programs on the input device will be loaded.

  If a program in memory has a number that is the same as a program loaded with multi load, two programs with that number will exist after loading. The "D" flag will appear in the flags field to indicate the duplication. With this condition, execution is not possible.

- **[ACCEPT ERRS]**-- Press the [ACCEPT ERRS] softkey to toggle the mode on (indicated by flashing reverse video) and off. This function allows you to load a program even if it has errors.

  When [ACCEPT ERRS] is flashing, errors are ignored and the program is loaded, any error messages will still appear, and the L and C flags may appear in the flags field.

  If [ACCEPT ERRS] is off, detected errors will prevent you from loading the program, the load operation is terminated, and appropriate error messages will appear.

- **[SELECTPROG]**-- Press the [SELECT PROG] softkey to select or deselect the current program number. The "S" flag appears in the flags field when the program is selected. Using [SELECT PROG] for multi load sets the number of a selected program for input from the load device. You cannot select a program number that already exists in memory. The "T" and the "S" flag will appear in the flags field to indicate this. "T" indicates that the program is "temporary" and doesn't actually exist yet in memory. The "T" flag will clear when you exit load/save if the program was not loaded.

The other functions shown (next selected, delete prog, delete all, and directory) are the same as described in the previous section.

11.8 Chapter Summary

In this chapter you read about loading and saving a program or other data with an external device, such as a cassette or tape reader. In the next chapter we will tell you about the support page.

Chapter 12 – Support Page

12.0 Chapter Overview

This chapter describes the support page.
You access the support page by pressing the [SUPPORT] softkey on the main menu display. After reading this chapter you'll know:

- the functions you can perform from the support page
- the procedures for using these functions
- how to configure the control for control parameters and serial I/O

Important: If you are a machine operator, you may not be able to use some of the functions described in this chapter. This is due to the Access Control function, which assigns control functions to different users. If you are not assigned a function under Access Control, the message "FUNCTION NOT ALLOWED" will appear when you try to use the function. Refer to section 12.2 for more information.

12.1 Support Page Functions

To access the support page, press [SUPPORT] on the Main Menu. You'll see the Support Page on the screen (figure 12.1).

Figure 12.1 - Support Page

```
| SUPPORT            PROGRAM PROTECT | *
| < operator messages>               | -
|                        DIAGNOSTICS | *
|                     ACCESS CONTROL | *
|                                AMP | *
|                                PAL | *
|                                    | -
|                                    | -
| <  error messages  >         LOGON | *
```

Here is a brief overview of what you can do with Support Page functions. The following sections describe the functions in detail.

[PROGRAM PROTECT] – lets you restrict or protect programs. To help you perform the restrict/protect task, the program protect function lets you display programs and look at the program directory. See section 12.5.

[DIAGNOSTICS] – lets you examine the Diagnostics pages, which gives the status of PAL variables. See section 12.6.

[ACCESS CONTROL] – lets you determine which users of the control have access to which functions. See section 12.2.

[AMP] – lets you program Adjustable Machine Parameters for such things as backlash compensation, control parameters, and serial I/O configurations. See section 12.4.

[PAL] – lets you control the activation of a pal program from software or firmware. See section 12.7

[LOGON] – lets a different user log onto the control by entering his password. See section 12.3

12.2 Access Control

Access control lets the installers of your system assign different functions of the control to different users.

To display the Access Control page, press [ACCESS CONTROL] on the Support Page. Figure 12.2 shows the access control page.

Important: You can't access this page unless you have access to AMP (flag 2 on the Access Control page is set for you as a user).

Figure 12.2 - Access Control Page

```
| ACCESS CONTROL  1 BASKLASH COMP                                 USER 1 |  *
|                 2 CONTROL PARAMETERS & I/O CONFIGURATION               |
|  USER 8         3 SPECIAL AUX CONTROLS                          USER 2 |  *
| ACCESS CODE:    4 MID PROGRAM START                                    |
| >****           5 INCH/METRIC;OP STOP;BLOCK DELETE;H/Z/L        USER 3 |  *
|                 6 TOOL/FIXTURE OFFSETS                                 |
|                 7 CHECK OUT                                     USER 4 |  *
|                 8 JOG & MDI                                            |
|                 9 PROGRAM LOAD/DELETE                           USER 5 |  *
|                10 PROGRAM SAVE/DISPLAY                                 |
|                11 PROGRAM SELECT;I/O DEVICE SELECT              USER 6 |  *
|                12 PROGRAM EDIT/TEACH                                   |
|                13 DIAGNOSTICS                                   USER 7 |  *
|                14 PROGRAM PROTECT                                      |
|                15 JOB DATA                                      USER 8 |  *
|                16 OPERATOR MACHINE HOME                                |
|                                                                        |  -
|________________________________________________________________________|
```

12.2.1 Control Functions

Sixteen control functions can be assigned under access control. Each function has a corresponding flag. Flags are numbered 1 through 16.

To give access to a function, cursor to the flag and press [ENTER]. The flag appears in reverse video when the function is allowed. Press [ENTER] again to disallow the function.

If a user tries to access a function that is not assigned to him, the message "FUNCTION NOT ALLOWED" appears on the CRT.

Here are brief descriptions of the sixteen control functions:

1. BACKLASH COMP -- lets you access the backlash compensation function on the AMP page. This gives access to backlash compensation values for the machine axes, but not to control parameters.

2. CONTROL PARAMETERS & I/O CONFIGURATION -- lets you access the control parameters and I/O configuration pages under the AMP page and lets you access the Access Control page.

3. SPECIAL AUX CONTROLS -- not implemented

4. MID PROG START -- lets you use the control's mid program start function.

5. INCH/METRIC; OP STOP; BLOCK DELETE; H/Z/L -- lets you use the inch/metric change, optional stop, and block delete features on the job setup page. H/Z/L is not implemented.

6. TOOL/FIXTURE OFFSETS -- lets you use the tool and fixture offsets pages of the control.

7. CHECK OUT -- lets you use the check out functions of the control, such as quick check, feed check, dry run, etc.

8. JOG & MDI -- lets you use the jog and MDI functions of the control.

9. PROGRAM LOAD/DELETE -- lets you load programs (input into the control) or delete programs already in memory.

10. PROGRAM SAVE/DISPLAY -- lets you save (output to a storage device) or display programs.

11. PROGRAM SELECT; I/O DEVICE SELECT -- allows you to change the current program number, lets you select a program for save (set the S flag), and choose a load or save device on the load/save page.

12. PROGRAM EDIT/TEACH -- lets you edit a part program or use the teach functions to develop portions of part programs.

13. DIAGNOSTICS -- lets you view the diagnostics (discrete I/O) pages of the control.

14. PROGRAM PROTECT -- lets you use the program protect page of the control.

15. JOB DATA -- not implemented

16. OPERATOR MACHINE HOME -- lets you home the machine using the manual operate page.

12.2.2 Users and User Numbers

The installer can define 8 users. User 1 has highest priority; user 8 has lowest priority. User 1, who is usually the system installer, can use all of the control's functions. User 1 defines the functions that are available to other users, and assigns them their original passwords.

Higher numbered users (those with lower priority than user 1) can assign functions to users that have yet higher numbers than they do. That is, user A can assign functions to user B as long as A < B, and the functions that B has are a proper subset of A's allowed functions.

12.2.3 Assigning Passwords

A password (or access code) is what you enter on the Logon page to allow you access to the control. Here are the steps user 1 performs to assign a user password. (If you are not user 1, you can use these steps to change your password, and you can change the password of any other user that you can access.):

1. Press the softkey that corresponds to the user you want to assign. The user number next to the softkey appears in reverse video, and also appears in the upper left part of the screen. The access code (password) will have 4 ":" marks that indicate the password has not been assigned.

   The control is shipped from the factory with no passwords assigned to users 2 through 7. The password for user 1 is predefined. (Check with Allen-Bradley if you do not know this password). User 8 has a password of 4 asterisks, which means that user 8 is automatically logged on. You should use the steps in this section to change the password for user 8 if you do not want user 8 to be automatically logged on.

2. Press the keys for a password up to 4 characters long, then press [ENTER]. The control stores the new password for the user you selected.

Important: If you are user 1, you have a special password, provided by Allen-Bradley, that lets you have access to all functions of the control. You can change this password using the above steps. If you do change your user 1 password, however, do not forget it. The special password will no longer work. A service call will be required to let you back into the system if you forget your new password.

12.2.4 Automatic Log On

If you define a user's password as 4 asterisks (****), that user is automatically logged on when the control is turned on. (If more than one user has 4 asterisks as his password, the user with highest priority -- lowest number -- is automatically logged on.) In this case, the control does not prompt for an access code upon coming out of Standby, and will only allow use of the functions that are set for this user. To log on a different user, use the Logon function on the Support Page to select another user with another password.

12.2.5 Assigning Functions

You can assign functions to another user only if:

- you have a lower user number

  AND

- you have access to AMP

  AND

- you can access all the functions the other user can (his functions are a subset of yours)

You cannot change your own functions, and you cannot assign any functions that you can not access.

Assigning Functions

To assign functions:

1. Use the softkeys to select the user you want.
2. Move the cursor to the left of the function you want to assign to this user, then press [ENTER] to toggle the enable/disable of the function for the selected user. The number of the function appears in reverse video when it is enabled.

No user can change what he has been assigned. No user can display another user who has functions he doesn't have. However, any user can display higher-numbered users who have functions that are a subset of his functions.

Also, higher priority users (those with lower numbers) can assign new functions to lower priority users (those with higher numbers). But, they can only assign those functions that they have been assigned.

12.3 Logon Page

The log on function lets you log onto the control under the password of a different user. Use this page to:

- change the logged on user without going to the main menu, setting standby, and relogging on
- change the logged on user if automatic logon is being used

Press [LOGON] on the Support Page, and you'll see the log on screen as shown in figure 12.3.

Figure 12.3 - Log On Page

```
 _______________________________
|                       STANDBY | *
|                               | -
|                               | -
|                               | -
|         ACCESS CODE:          | -
|           >****               | -
|                               | -
|                               | -
|_______________________________| -
```

The control prompts you to enter the 4 character access code assigned to the user whose functions you want to access. Type in the code you want and press [ENTER]. If the code corresponds to the password of one of the assigned users, you will have access to the functions of that user. If the code does not match an assigned user password, the control will ignore your input.

Note that you can enter Standby mode from this page by pressing [STANDBY]. The Standby mode powers down the drives and makes the screen go blank. The red [CYCLE STOP] lamp will be on. To remove Standby, press [EXIT].

12.4 AMP Page

Adjustable Machine Parameters (AMP) lets you customize the control's operation for a particular application. Programming and adjusting AMP parameters is usually a task that the installer performs. As such, generally only high priority users have access to AMP functions.

Pressing [AMP] on the Support Page brings the AMP page to the screen. See figure 12.4.

Figure 12.4 - AMP Page

```
 ____________________________________
| AMP                                | -
| < operator messages>      AXIS CAL | *
|                      BACKLASH COMP | *
|                 CONTROL PARAMETERS | *
|                                    | -
|                      CONFIGURE I/O | *
|                                    | -
|                                    | -
|_<  error messages  >_______________| -
```

The AMP page gives you these functions:

AMP Page Softkeys

[AXIS CAL] -- lets you adjust for certain inaccuracies in the feedback positioning caused by mechanical deficiencies in a machine. See section 12.4.4.

[BACKLASH COMP] -- lets you specify the amount of backlash that the control compensates for when an axis changes direction. You can specify individual amounts for each axis. This function gives you independent access to the backlash parameter that is under control parameters. Use of this function is governed by the state of access control flag 1. See section 12.4.3.

[CONTROL PARAMETERS] -- lets you enter the parameters that affect how the control monitors and directs the performance of the axis drives, and more. See section 12.4.1.

[I/O] -- lets you enter the parameters that affect the performance of the serial I/O port, with selected devices. See section 12.4.2.

Each of these functions are covered in this section.

12.4.1 Control Parameters

Pressing [CONTROL PARAMETERS] opens the drive enable relay and brings this page to the screen. See figure 12.5.

Important: Figure 12.5 does not show the power-up values assumed by the control. Do not use the values shown to establish your control parameters. The system installer must program the correct control parameters for your machine application.

Figure 12.5 - Control Parameters Page

```
| LDEC    4                RAPID      300.0  + LIMIT   50.0000   X | *
| fine     dps ROTARY      DRY        200.0  - LIMIT  -50.0000     |
| inch    >CHE  0.0002     DEFAULT    250.0  HOM POS   0.0000   V | *
| slaved   not homed=x     JOG_INC    150.0  MAX RATE    500.0     |
| rot DRO  not homed=V     JOG_RTN    100.0  RAMP/.1S     25.0  W | *
| rotary   not homed=W     MX_TOOL   0.5000  VEL STEP     25.0     |
| lin DRO  not homed=X     MASK           0  use ACC/DEC        X | *
| disabled not homed=Y     HANDWH_MUL     5  MIN FE    0.0200     |
| linear   1st homed=Z     HANDWH_DIV    20  MUL FE        15   Y | *
|          revision        SPINDL_MUL 15625  TIM CON        3     |
| DEFAULT PP VER           GAIN BP 1   1024  BACKLASH  0.0000   Z | *
|  ALLEN-BRADLEY           GAIN BP 2   2048  + FIRST TO HOME       |
|   REV F1-003             A-B GRAPHIC MILL  HOME %#        5     | -
|    GRAPHICS              SKEW_MX   0.1000  HOME TOL  0.0050     |
|  MILL CONTROL            SKEW_GAIN      8  BLNC TOL  0.0005     | -
|                                 GAIN   -2  INPS TOL  0.0025     |
| < operator messages>     METRIC MUL   254  INCH     MUL   1     | -
| <  error messages  >            DIV   100           DIV   1     |
```

CAUTION: When the handwheel is being used while the feedrate override switch is set to 10%, 20%, or 30%, and the control is configured for 'coarse' resolution, small axis movements can be made that do not appear on the axis displays. This occurs because the handwheel move increment (as controlled by the feedrate override switch setting) is smaller than the display resolution.

For example, if your control is configured for:

```
LDEC  =  4
resolution  =  coarse
active mode =  Handwheel Jog
feedrate override = 30%
```

then the control will display 0.000 for linear axes, but the axis will move 0.0005 with each pulse of the handwheel. Every pulse from the handwheel will result in a move, but only every other move will result in a change in the axis display.

This same situation, but with different amounts of motion, will result if LDEC is changed to 3, 4 or 5, or if the feedrate override switch is 10%, 20% or 30%.

Important: The handwheel relies on a changing magnetic field that creates pulses as it is rotated. If it is rotated very slowly, the amplitude of the pulses will be reduced, and the axis may not move.

Default Control Parameters

When the control is powered up for the first time, the control parameters will have default values; values that are always present as part of the firmware. We set these values to provide a limited amount of axis travel and a low rapid traverse rate.

The default control parameters are for a 3 axis mill application. Programming and display is in inch mode with 0.0001 inch resolution. There is a 50 ipm rapid traverse rate. All axes are set for using a 500 line encoder coupled directly to a 5 pitch leadscrew. All axis homing is disabled.

This allows you to test the motion of the the machine before you attempt a machine home operation. Axis travel limits are set to $\pm$5.0 inches in the X axis, $\pm$3.0 inches in the Y axis, and $\pm$1.0 inch in the Z axis. We mention these and other default values in the detailed explanations that follow.

Important: If the control detects a condition which may indicate loss of integrity of control parameters, it will automatically reset all of them to the default conditions and it will display a message "AMP WAS DEFAULTED". The drives can not be turned on until this condition is cleared. To clear the message, it is necessary to log on as User 1 and enter the AMP - CONTROL PARAMETERS page. Note that the User 1 password is also defaulted.

Data Entry and Programming Procedure

There are 2 ways to enter data in the control parameters page. Which method you use depends on the parameter:

- Move the cursor to the left of the parameter. (Note that the cursor is to the left of CHE in figure 12.5.) Key-in a numeric or alphanumeric entry you want, then press [ENTER]. If you move the cursor away from a parameter before you press [ENTER], the control restores the previous value.

- Move the cursor to the left of the parameter, and press [ENTER] to "toggle" (sequence through) parameters that offer a choice among different operating modes.

CAUTION: When you enter data, the control does not accept values that are out range for a given parameter. Note that this does not guarantee that the data you enter is correct. You must perform sufficient tests to make sure that the operation of your control is safe and correct.

Global Parameters

You enter "global" and "per axis" parameters. "Global" parameters, as the name suggests, are those that determine the major portion of the control's operation of the machine. They are not subject to change on a per axis basis.

For example, they define:

- the control's programming resolution
- which axes are present and their mode of operation
- system feedrates for all axes
- system wide parameters and messages

Global parameters are generally those in the two left columns of the control parameters page. When you program control parameters for a particular system, you should enter most of the global parameters first. The values of some global parameters affect the "per axis" parameters.

Per Axis Parameters

"Per axis" parameters are those that apply to each machine axis, individually. They define characteristics of the drives and positioning. To program per axis parameters correctly, you should know the characteristics of the axes in your system. Things to consider are the lead and pitch of the leadscrews, the length and travel limits of the axes, where Machine Zero is relative to Machine Home, the type and resolution of the feedback device, reversal compensation, and homing rate and direction.

Per axis parameters are generally those in the right column of the control parameters page. To distinguish between a global and per axis parameter, you can press one of the softkeys for the axis you want. The page will show the per axis parameters for that axis.

In general, you enter per axis parameters after you choose global parameters.

Axis Integration Notes

The main goal of programming control parameters is to make the axes run with optimum performance. For making the axes run at their best, here is a general guide on how to proceed.

1. Select a value for the LDEC parameter based on the feedback device characteristics. See axis scaling parameters on page 37 in this section.

2. Perform feedback scaling calculations for the MUL and DIV parameters based on the axis (motor and drive system) characteristics using the formulas starting on page 25 and examples on page 39 in this section.

3. Determine rapid rates according to the values obtained from steps 1 and 2. If you set the LDEC too high you will limit the rapid rate because the control can only handle up to 333,333.3 pulses per second.

   2000 IPM x 10,000 pulses per inch = 20 million pulses

   $$\frac{\text{20 million pulses}}{\text{60 seconds}} = \text{333,333. pulses per second}$$

   The frequency of the square wave output from the resolver is 83,000 pulses per second at resolution of .0001.

4. Determine the system gain you want (small "g" in the formulas following Table 12.A).

5. Determine the value for the gain factor on the control parameters page (large G in the following formulas). You should select the maximum G such that the equation is true:

inch mode: $$2^G \le \frac{2{,}047{,}000 \times g}{10^P \times R}$$

metric mode: $$2^G \le \frac{5{,}199{,}380 \times g}{10^P \times R}$$

rotary mode: $$2^G \le \frac{2{,}047{,}000 \times g}{1000 \times R}$$

where:

g = the overall system gain you want for speed from 0 to maximum contouring speed

P = the number of programming places to the right of the decimal point. For example, with LDEC = 4 and fine programming, P = 4 for inch and P = 3 for metric.

R = the rapid axis speed that the machine and the drives were designed for.

The gain factor (G) must be an integer value from -3 to 3. The purpose of the gain factor is to match the output of the control with the performance level of the machine. The calculated gain factor (G) will be different than g, the overall system gain.

The overall system gain is a function of the control, machine, drives and axis servo motors. For example, many machines are sized with drives and motors so that the axes can run at a system gain of 1 (0.001 in. of following error for each 1 ipm of feedrate) up to the machine's specified maximum cutting speed.

Typical values for system gain range between 0.5 for "loose" systems (large, heavy axes with long acceleration and deceleration times) to 1.5 for "tight" systems (small, light systems with short acceleration and deceleration times).

Table 12.A
System Gain and Following Error

System Gain (g)	Following Error per ipm of Velocity
0.25	0.0040 inch
0.50	0.0020 inch
1.00	0.0010 inch
2.00	0.0005 inch

In general,

inch mode:
$$\frac{\text{following error}}{\text{ipm}} = \frac{0.001 \text{ in.}}{g}$$

metric mode:
$$\frac{\text{following error}}{\text{mmpm}} = \frac{0.001 \text{ mm}}{g}$$

rotary mode:
$$\frac{\text{following error}}{\text{dps divided by 60}} = \frac{0.001\,^{\circ}}{g}$$

Using the correct gain factor (G) parameter insures that the range of the command voltage output will be as close as possible to the maximum range of $\pm 10V$.

Calculating Gain Break Points

6. If the machine is designed to operate axes at speeds greater than the maximum cutting speed, then a gain break point should be calculated. Use the following formulas:

inch mode: $$GBPT = \frac{ipm_{gb} \times 10^P \times 2^G}{1000 \times g}$$

metric mode: $$GBPT = \frac{mmpm_{gb} \times 10^P \times 2^G}{2540 \times g}$$

rotary mode: $$GBPT = \frac{rpm_{gb} \times 10^P \times 2^G}{2.78 \times g}$$

where:

ipm_{gb}, $mmpm_{gb}$ or rpm_{gb} = the maximum cutting speed for the machine

g = the overall system gain

P = the number of programming decimal places

G = the calculated gain factor (G)

The same formulas are used for GB1 and GB2.

The initial system gain will be multiplied by the factor of 1/2 for all axis speeds above the first gain break point. The new system gain will equal

$$\frac{\text{initial system gain}}{2}$$

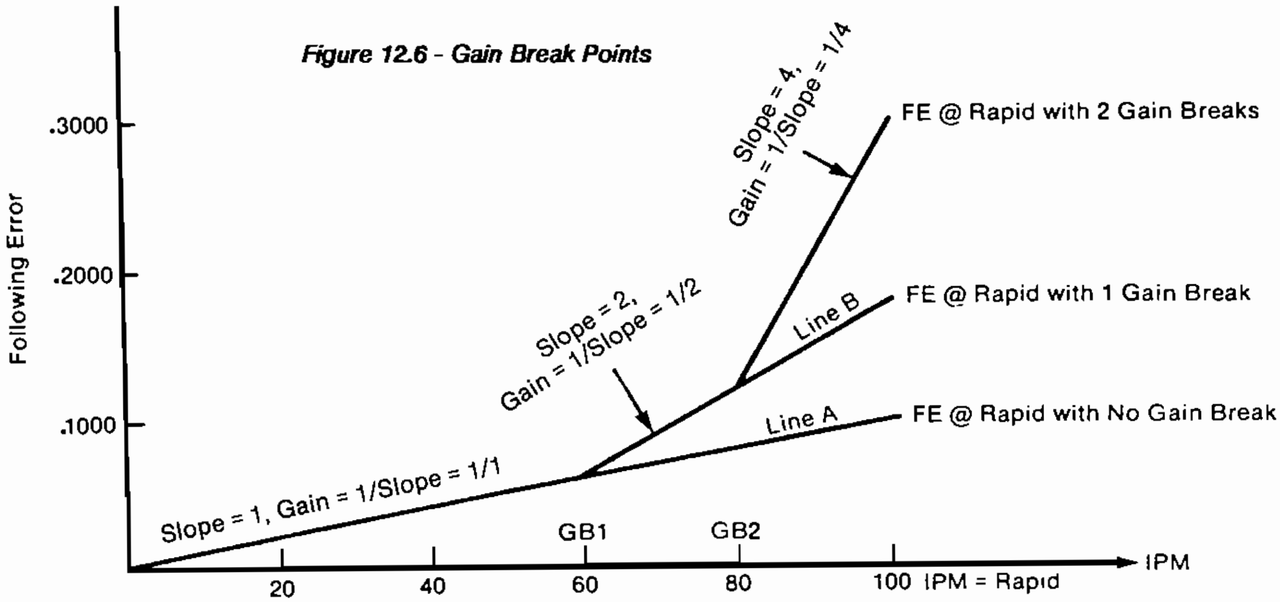
active to rapid speed, or the second gain break point.

After the second gain break point, initial system gain will be multiplied by a factor of 1/4. The new system gain after the second gain break will equal

$$\frac{\text{initial system gain}}{4}$$

up to rapid speed. See figure 12.6

Figure 12.6 - Gain Break Points



For line A, a GB1 = 2048 means no gain break. Axes run at initial system gain up to rapid speed.

For line A or B, a GB2 = 2048 means no second gain break.

7. Calculate the command voltage output of the control at rapid with the following formulas:

inch mode: volts@ rapid $= \dfrac{10^P \times 2^G \times R}{g \times 204{,}700}$

metric mode: volts@ rapid $= \dfrac{10^P \times 2^G \times R}{g \times 519{,}938}$

rotary mode: volts@ rapid $= \dfrac{1000 \times 2^G \times R}{g \times 2{,}047{,}000}$

g = the overall system gain you want for speed from 0 to maximum contouring speed

P = the number of programming places to the right of the decimal point. For example, with LDEC = 4 and fine programming, P = 4 for inch and P = 3 for metric.

R = the rapid axis speed designed for the machine and the drives

8. Use the following error (lag) display to tune and check the settings on the drives. Jog an axis at a speed below any gain break and monitor the following error. After the lag error has reached steady state, note the value. The value, (or accumulated lag error) at that speed should equal:

$$\text{inch mode:}\quad \text{lag error} = \frac{0.001\ \text{in.} \times \text{speed checked}}{g}$$

$$\text{metric mode:}\quad \text{lag error} = \frac{0.001\ \text{mm} \times \text{speed checked}}{g}$$

$$\text{rotary mode:}\quad \text{lag error} = \frac{0.001\ \text{deg.} \times \begin{array}{c}\text{speed checked}\\ \text{(in deg/min)}\end{array}}{g}$$

Perform this check in both directions and adjust the drives until the proper lag error is obtained.

You can also perform a voltage check to see if the drives are set correctly. Measure the command voltage at the drives when the axis is running at rapid. Your measured value should be approximately equal to the value calculated using the formula above (step 5).

Definitions for Global Parameters

This section describes each global parameter. Note in this section that we say "toggle," "numeric entry," or "alphanumeric" for each parameter. When you toggle a parameter, you cursor to the parameter and press [ENTER] to change its meaning. When you enter a value, you press alphanumeric keys then press [ENTER] to set the value.

1. inch/metric (toggle)

You should set this parameter to the normal programming units you want for the control. When "inch" is set, the power-up programming units of the control are inches. Setting "metric" establishes millimeters as the control's programming units. Of course, the operator can make a change between inch and millimeter programming units at the front end of the control to meet the requirements of a particular program or operation.

The default operating mode is inch programming. Note that switching between inch and metric causes a mathematical conversion to occur to certain other parameters on the control parameters page. Inch or metric should be set first and any remaining numeric entries on the control parameters page should correspond to the units you choose.

2. LDEC (numeric entry)

The linear decimal (LDEC) parameter sets the number of internal counts the control produces for each inch or millimeter of programmed axis travel. The LDEC parameter is important to system scaling and also affects the programmable range of axis travel. LDEC does not affect rotary axes.

You can set LDEC for 3, 4, or 5. These values correspond to powers of ten: 10^3, 10^4, or 10^5 counts per inch or centimeter of travel. The control will not accept an entry less than 3 or more than 5. The default value is 4.

Table 12.B summarizes the relationship between LDEC, the number of internal counts per inch or millimeter of axis travel, and the resulting programmable range of travel.

Table 12.B
LDEC and Programming Units

LDEC	Inch		Metric	
	# counts/inch	prog. range	# counts/mm	prog. range
5	10^5 = 100,000	±99.99999in.	10^4 = 10,000	±999.9999mm
4	10^4 = 10,000	±999.9999in.	10^3 = 1,000	±9999.999mm
3	10^3 = 1,000	±9999.999in.	10^2 = 100	±99999.99mm

3. fine/coarse (toggle)

This parameter sets fine or coarse programming. It is used along with the number of internal counts set with LDEC. When fine programming is selected, programming decimal places and the internal counts set with LDEC are the same. When coarse programming is chosen, programmed and displayed decimal places are one less than LDEC decimal places. The table 12.C shows the relationships.

CAUTION: When the handwheel is being used while the feedrate override switch is set to 10%, 20%, or 30%, and the control is configured for 'coarse' resolution, small axis movements can be made that do not appear on the axis displays. This occurs because the handwheel move increment (as controlled by the feedrate override switch setting) is smaller than the display resolution.

For example, if your control is configured for:

```
LDEC  =  4                 resolution  =  coarse
feedrate override = 30%    active mode =  Handwheel Jog
```

then the control will display 0.000 for linear axes, but the axis will move 0.0005 with each pulse of the handwheel. Every pulse from the handwheel will result in a move, but only every other move will result in a change in the axis display.

This same situation, but with different amounts of motion, will result if LDEC is changed to 3, 4 or 5, or if the feedrate override switch is 10%, 20% or 30%.

Table 12.C
Fine/Coarse Programming Decimal Places

	FINE		COARSE	
LDEC	inch	metric (mm)	inch	metric (mm)
5	0.00001	0.0001	0.0001	0.001
4	0.0001	0.001	0.001	0.01
3	0.001	0.01	0.01	0.1

The default mode is fine positioning. You can use coarse positioning in some situations where repeatability is more important than programming resolution. For example, with LDEC = 4 and inch/coarse mode there is 0.0001 inch repeatability (because there are 10,000 counts per inch) with a 0.001 inch resolution system. The machine that is controlled will affect actual repeatability and resolution to a greater extent.

4. CHE (numeric entry)

The CHE global parameter sets the maximum "chordal error" allowed during circular interpolation of the axes. The control uses a series of chords, or line segments, to approximate an arc. "Chordal error" refers to the amount that the chord approximation differs from the actual arc. It is the maximum distance between two parallel lines; one being the chord, and the other, a line tangent to the arc.

The value you select for CHE affects the speed (feedrate) as well as the precision of circular contouring moves. It affects feedrate because the control calculates the end point of a chord in each servo sample period. If there are many chords in the arc, the accuracy is finer but the maximum rate of execution is slower.

Table 12.D gives an idea of how the value for chordal error affects feedrate and chord length. For example, with chordal error at 0.0002 and an arc radius in the range 6.5536 to 26.2143 inches, there are 805 chords to a full circle, chord length ranges from 0.0512 to .2047 inch and maximum execution feedrate ranges from 383 to 1535 inches per minute.

Table 12.D
Effects of Different CHE

Chordal Error (CHE) = 0.0002

	ARC RADIUS		CHORD LENGTH		MAX EXECUTION FEEDRATE	
# Chords	MIN	MAX	MIN	MAX	MIN	MAX
13	0.0001	0.0063	0.00005	0.00315	0.4	23.6
25	0.0064	0.0255	0.0016	0.00638	11.0	47.8
51	0.0256	0.1023	0.0032	0.01279	23.0	95.9
101	0.1024	0.4095	0.0064	0.02559	47.0	191.0
202	0.4096	1.6383	0.0128	0.0511	95.0	383.0
403	1.6384	6.5535	0.0256	0.1023	191.0	767.0
805	6.5536	26.2143	0.0512	0.2047	383.0	1535.0
1609	26.2144	104.8575	0.1024	0.4095	767.0	*3071.0
3217	104.8576	419.4303	0.2048	0.8191	1535.0	*3276.0
6434	419.4304	999.9999	0.4096	1.6383	*3071.0	*3276.0

Table 12.D
Effects of Different CHE (continued)

Chordal Error = 0.0008

# Chords	ARC RADIUS MIN	ARC RADIUS MAX	CHORD LENGTH MIN	CHORD LENGTH MAX	MAX FEEDRATE MIN	MAX FEEDRATE MAX
13	0.001	0.0255	0.00005	0.01275	0.4	95.6
25	0.0256	0.1023	0.0064	0.02558	47.0	191.8
51	0.1024	0.4095	0.0128	0.05119	95.0	383.9
101	0.4096	1.6383	0.0256	0.10239	191.0	767.0
202	1.6384	6.5535	0.0512	0.2047	383.0	1535.0
403	6.5536	26.2143	0.1024	0.4095	767.0	*3071.0
805	26.2144	104.8575	0.2048	0.8191	1535.0	*3276.0
1609	104.8576	419.4303	0.4096	1.6383	*3071.0	*3276.0
3217	419.4304	999.9999	0.8192	3.2767	*3276.0	*3276.0

Chordal Error = 0.0032

# Chords	ARC RADIUS MIN	ARC RADIUS MAX	CHORD LENGTH MIN	CHORD LENGTH MAX	MAX FEEDRATE MIN	MAX FEEDRATE MAX
13	0.0001	0.1023	0.00005	0.00515	0.4	383.6
25	0.1024	0.4096	0.0256	0.10238	191.0	767.8
51	0.4096	1.6383	0.0512	0.20479	383.0	1535.9
101	1.6384	6.5535	0.1024	0.40959	767.0	3071.0
202	6.5536	26.2143	0.2048	0.8191	1535.0	*3276.0
403	26.2144	104.8575	0.4096	1.6383	*3071.0	*3276.0
805	104.8576	419.4303	0.8192	3.2767	*3276.0	*3276.0
1609	419.4304	999.9999	1.6384	6.5535	*3276.0	*3276.0

* This is a calculated maximum feedrate. Consult Allen-Bradley if you intend to have axis feedrates greater than 2000 IPM.

The table above shows relationships for inch programming units with four decimal place accuracy. Similar tables could be made for metric systems and varying decimal places. The important thing to notice is how chordal error affects accuracy and maximum execution feedrate. A large chordal error gives more speed but also produces longer chords for a given radius resulting in less accuracy.

The value entered here must be greater than zero but less than .025. It can be entered with a positive or negative sign. The sign affects the way a the final chord in an arc move is calculated.

- Positive (or unsigned) CHE Parameter - The control will calculate a "Short Length" for the last chord.

  Advantage -- Actual chordal error throughout the programmed arc will never exceed the value entered as the CHE AMP Parameter.

  Disadvantage -- If circular contouring is done at high speed (generally over 200 IPM) the feedrate for the short chord may be slower than the programmed feedrate for the arc. This is due to the way the control executes servo updates. This slowing in feedrate may cause axis hesitation.

- Negative CHE Parameter -- The control will calculate a "Long Length" for the last chord.

  Advantage -- The feedrate for the last chord will always be at a speed that is, at a minimum, equal to the feedrate used for the previous chords (no axis hesitation). A negative CHE parameter will produce smoother axis motion when circular contouring at high speed (generally over 200 IPM).

  Disadvantage -- Since the last chord is longer than the previous chord, the actual chordal error for this last section of the arc may exceed the CHE AMP Parameter.

For either a positive or negative CHE Parameter, the following is true:

- Arc end point will always be reached accurately.

- Actual feedrate in circular contouring is governed by the absolute value or the CHE Parameter. See table 12.D.

Important: Be aware of the affect of the chordal error on feedrate and precision. If you enter a value here that allows arcs to be made at higher feedrates (CHE greater than .0002) the result will be a rougher arc made up of larger chords that may be noticeable on the part.

If you enter a value here that forces smooth, precise arcs, (CHE less than .0002) the result will be a reduced allowable feedrate while executing that arc. If you are commanding a higher feedrate than allowed here, the control will automatically reduce the feedrate. There is no error message and speed will resume after the arc is complete.

5. dps/dpm/rpm ROTARY (toggle)

This entry toggles the programming units for rotary feedrates. You have a choice of degrees per second (dps), degrees per minute (dpm), and revolutions per minute (rpm). The control sequences through these choices each time you press [ENTER]. The default is dps programming units.

6. disabled/ rotary/linear/ DRO/Slaved

These entries configure each machine axis. When an axis is disabled, the control ignores any attempt to move the axis during operation. A disabled axis is not shown on most of the pages of the control. (PAL can disable an axis, too, but this does not affect the display of the axis.) Programmed movements of disabled axes give the error message "CODE NOT KNOWN" and the axis cannot be jogged.

When an axis is rotary it responds to normal operation and is programmed in degrees with rotary feedrates (see the dps/dpm/rpm parameter). When an axis is linear it is programmed in inches or millimeters with linear feedrates.

On the Control Parameters Page shown in Figure 12.5, there are two DRO modes. These are 'rot DRO' and 'lin DRO', which correspond to Rotary Digital Readout and Linear Digital Readout. The X,Y, and Z axes have only one DRO mode, 'lin DRO'. The U,V, and W axes can be configured to either DRO mode.

DRO Operation

Axes selected as DRO axes:

- Will have zero volts output on the servo output
- Will give '____ CODE NOT KNOWN' if you attempt to execute a motion command for that axis
- Will not display the 'In Position' (I), 'Mirror' (M), or 'Homed' (H) flags on the status page.
- Will not display the 'Homed' (*) flag on the Machine Home page
- Will not be prompted for in motion blocks under the Prompt Editor
- Cannot be selected for Jog but are displayed on those pages

Presetting DRO Axes

Axes selected as DRO Axes can be preset using a G92 command. Similarly, the preset can be released with a G99 to return the reference to machine zero.

The Jog Handwheel and the Jog Continuous pages display DRO axis with no sign indicator before the axis letter.

Split Axes(Slaved)

The split axes feature allows the use of mechanically linked but individually controlled axes in a gantry type configuration.

Characteristics

The master to slave relationships are fixed and are as follows: X to U, Y to V, and Z to W. The slaved axis will be displayed similar to a DRO axis. To configure an axis as slaved, the operator need only bring up the 'CONTROL PARAMETERS' page and choose the 'slaved' axis mode. If an axis is in slaved mode, the axis letter of its master axis will be used in lower case for all displays related to the slaved axis. For example, if 'W' is selected as 'slaved', all pages providing axis information will display 'W' as 'x', with the lower case letter indicating slaved mode. If an axis is in slave mode, it will display skew in place of following error, with the exception of the axis calibration page which shows actual following error for the slave axis.

There are two parameters that must be defined for an axis that is to be used as a master, 'SKEW__MX and SKEW__GAIN. Skew is defined as the difference in following error between the master axis and the slave axis. SKEW__MX is the maximum amount of skew allowed for a master slave relationship. SKEW__GAIN is a numeric value between 1 and 24 that is used to adjust the rate of skew compensation. The following formulas show the application of SKEW__GAIN in the calculation of master and slave following error.

```
Master FE = Master FE' + (SKEW/2)*(SKEW_GAIN/8)

Slave FE = Slave FE' - (SKEW/2)*(SKEW_GAIN/8)
```

The SKEW__MX and SKEW__GAIN entries on the 'CONTROL PARAMETERS' page for a master slave pair must be entered for the master axis. Any values entered for a non-master axis SKEW__MX and SKEW__GAIN will be ignored.

All per axis parameters, with the exception of BACKLASH, will be taken from the master axis parameters.

Split Axes Operation

All master axis motion is mimicked by the slave axis. All request for motion on a slave axis are ignored and a '__CODE NOT KNOWN' error message will result.

If either master or slave axis contacts a hard, soft, or safe limit, motion is stopped for both axes. Also, if either master or slave axis is clamped, motion is inhibited for both axes.

The homing sequence for a master slave pair is as follows. Both axes move toward their home switch with no detection of skew. The first axis to contact the home switch stops and remains stopped until the next axis contacts the switch. Then both master and slave move off the switch. The first axis to move off of the switch will stop until the other axis is also off the home switch. Next, both axes move until their encoder mark is found and as before, the first axis to find the mark stops until the second axis has found its' mark. When both master and slave have found their respective marks, the pair is then considered homed.

If at any time during operation of a master slave combination, the skew error exceeds the SKEW_MX parameter, the axis will decelerate until the skew is reduced to an amount determined acceptable by the SKEW_MX parameter.

When the drives are turned off, the position of the slave axis is taken from the master axis information. If there is a great deal of skew and the drives are turned on, the slaved axis may jump or cause a 'LAG ERROR' message.

The teach feature will only teach the master axis. Also, if a G92 preset is performed on the master axis, the slave axis is also preset to the same program zero value. The PAL may not disable a slave axis, but if PAL disables a master axis, the slave is automatically disabled also. Safe areas for the master axis are used for the slave axis.

To select the mode of each axis, position the cursor to the left of the line for the axis and press [ENTER]. The entry will sequence through disabled, rotary,slaved,DRO and linear each time you press [ENTER].

The default modes for the axes are:

disabled = U
disabled = V
disabled = W
disabled = X
disabled = Y
disabled = Z

The U, V, and W axes can be disabled, rotary or linear. X, Y, and Z can be either disabled or linear.

7. Homing Sequence (toggle)

This parameter sets the homing sequence of machine axes. Axes will either be "not homed" or homed in a specific order that you define. You determine the order of homing by moving the cursor to the axis that should home first, and then pressing [ENTER]. "1st homed" automatically appears. Cursor to the 2nd homed axis and press [ENTER], etc., until you have set the homing sequence for all active axes.

The default homing sequence is:

not homed = U
not homed = V
not homed = W
not homed = X
not homed = Y
not homed = Z

The control requires you to do a Machine Home operation before running a program or using the MDI mode even though the axis defaults are "not homed."
In this case, when you press the green [CYCLE START] button to begin the Machine Home sequence, and the axes are set to "not homed," no axis motion will occur and the position register will not change (will not zero).

8. revision/ message (toggle)

This parameter, found immediately below the Z axis definition, determines whether the control shows the current firmware revision, or a customer defined message in the second line of the Main Menu page's identity field. (The identity field is shown at the lower left corner of the control parameters page, figure 12.5). The default is "revision." When you select "message," you can substitute a 14 character alphanumeric entry in the identity field.

9. AMP Version (alphanumeric entry)

This entry is a 14 character alphanumeric message that you can use to identify the control's AMP version that you program. The control also displays this message on the Diagnostics page. The default message is DEFAULT PP VERS.

10. Identity Messages (alphanumeric entry)

These areas in the identity field of the Mode Select page allow the entry of 14 character messages. The default is:

```
ALLEN-BRADLEY
REV XX - XXX
  GRAPHICS
MILL CONTROL
```

The Emergency Stop message field is 16 characters long and is immediately above the SKEW_MX parameter. Any text entered in this location will appear on the bottom line of the Emergency Stop screen.

11. Feedrates (numeric entries)

These global parameters set maximum and minimum feedrates for the system. Remember that maximum feedrate is a function of the machine's drive components: the ballscrew, motor, servo amplifier, etc.

RAPID - sets the maximum rapid traverse (G00) feedrate for the system. The default is 50 inches per minute. The maximum is 2000.0 ipm, minimum is 0.1 ipm. If your application requires feedrates faster or slower than these, please consult Allen-Bradley.

DRY - sets the minimum feedrate for DRY RUN execution. The default is 50 inches per minute. The maximum is the RAPID rate, minimum is 0.1 ipm.

DEFAULT - sets the feedrate for G01 moves when no feedrate is programmed. The default is 10 inches per minute. The maximum is the RAPID rate, minimum is 0.1 ipm.

JOG INC - sets the feedrate for JOG INCREMENTAL moves when the feedrate override switch is set at 100%. The default is 10 inches per minute.

JOG RTN - sets the feedrate for and JOG-AND-RETURN moves (Z infeed only) with the feedrate override switch at 100%. The default is 10 inches per minute.

12. MX TOOL (numeric entry)

This entry determines the longest tool length offset that you can enter on the Tool Offset pages. You should set this value to the largest value for length offset that you anticipate in the application. The default value is 0.5000 inch. The maximum is 999.9999 in., the minimum is 0.0001 inch.

13. MASK (numeric entry)

The MASK parameter is a numeric value (the decimal representation of a 16 bit binary field, ranging from -32768 to 32767) that the control passes from the control parameters page to programmable applications logic (PAL). The MASK value informs PAL of various requirements of the ladder program.

MASK Values with Standard Ladder

Parameters that you can select with the MASK value in the standard ladder available from Allen-Bradley include:

- whether to use the limits or separate switches for homing
- whether to check the W axis limits (linear W) or not (rotary W)
- whether to check the V axis limits (linear V) or not (rotary V)
- whether to check the U axis limits (linear W) or not (rotary U)
- whether spindle speed control is analog output or Quickdraw RPM Changer
- whether the analog output is positive only, negative only, or bipolar
- which value to use for the maximum S word entry

To use table 12.E: Choose 1 option value from each group in table 12.E and add all the values together to form the complete MASK value.

Table 12.E
Base Numbers for Revision D1 Ladder

Group	Value	Function
1	0 thru 511	Maximum S word. This value (0 through 511) selects the maximum S value allowed in the ladder. If RPM Changer mode is selected, this number is 1 less than the actual maximum number of positions on the RPM Changer mechanism (position S0 is valid). Selecting 0 forces the ladder to use the default value of 25. If the analog mode is selected, this number is multiplied by 10 to get the actual maximum S value. For example, selecting 200 for this value allows a maximum S value of 2000. Selecting 0 forces the ladder to use the default value of 10, that is, S100 is the maximum S value. S0 causes 0V DC on the analog output, and Smax causes 10V DC on the analog output.
2	*0*	*Selects positive only analog output.*
	512	*Selects negative only analog output.*
	1024	*Selects bipolar analog output.*
	1536	*Selects RPM Changer spindle speed control.*
3	*0*	*Don't check the U axis limits (U rotary or not implemented).*
	2048	*Check U axis limits (U linear).*
4	*0*	*Don't check the V axis limits (V rotary or not implemented).*
	4096	*Check V axis limits (V linear).*
5	*0*	*Don't check the W axis limits (W rotary or not implemented).*
	8192	*Check W axis limits (W linear).*
6	*0*	*Use the axis limit switches for home switches.*
	16384	*Use separate switches for home position.*

Standard ladder MASK example 1:

500	Max spindle speed number = S5000
1024	Select bipolar analog output
4096	Check V limits (don't check U or W)
16384	Use separate home switches
22004	MASK value you enter.

Standard ladder MASK example 2:

21	Max spindle number = S21 (22 RPM positions)
1536	Select RPM Changer mode
8192	Check W limits (don't check U or V)
0	Use limit switches for homing
9746	MASK value you enter.

14. HANDWH_MUL and DIV (numeric entry)

These parameters establish a scaling factor for the Jog Handwheel on the front panel. The default values are shown in figure 12.5.

Important: Leave these values as they are set in the default control parameters. With HANDWH_MUL set to 5 and DIV set to 20, the scaling factor is 5/20 = 1/4. This means that the pulses generated by the handwheel mechanism are divided by 4 before they are processed by the control to produce axis motion. The mechanism currently being shipped from the factory requires this scaling factor. Do not change this factor unless directed by the factory to do so.

15. SPINDL_MUL (numeric entry)

This is a scaling factor for the pulses received from a spindle coupled optical encoder. The spindle rotation is fed back to the control for direct display of RPM on the Status page. Be aware that the encoder signal must be directed to a specific input port in the Logic Module, and there are special PAL ladder considerations to make this feature work.

The formula for determining the SPINDL_MUL factor is:

$$\text{SPINDL\_MUL} = \frac{18{,}750{,}000}{N \times E}$$

where:

N = the number of encoder turns per spindle revolution

E = the number of encoder lines (pulses) per encoder revolution

The default value for SPINDL_MUL is 15625, which corresponds to a 1200 line encoder directly coupled to the spindle.

16. GAIN BP 1 and GAIN BP 2 -- Gain Break Points (numeric entry)

These parameters determine the gain break points for velocity control. To understand the concept of gain break, you should refer to the axis integration notes earlier in this section.

17.EMERGENCY STOP Message (alphanumeric entry)

This area allows you to specify a message up to 14 characters long that will appear on the bottom of the Emergency Stop page of the control.

Definitions for Per Axis Parameters

Per axis parameters apply to each axis individually and define characteristics of one axis at a time in the system.

You select the axis you want to program by pressing one of the soft keys. When a soft key is pressed, its axis letter will appear in reverse video and the parameters for that axis will be called to the screen. You can then program the parameters for the axis as you did for the global parameters.

18. ± LIMITS and HOME POS (numeric entries)

These parameters determine the soft travel limits and the position of Machine Home relative to Machine Zero for the axis. +LIMIT and -LIMIT set the soft travel limits of the axis referenced to Machine Zero on the axis. HOME POS is the value the axis assumes, referenced to Machine Zero, after the axis is homed.

The default values for all the axes are:

	X	Y	Z	W	V	U
+ LIMIT	+5.0000	+3.0000	+1.0000	0.0000	0.0000	0.0000
- LIMIT	-5.0000	-3.0000	-1.0000	0.0000	0.0000	0.0000
HOME POS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

You can see that it is not necessary to have the home position equal to Machine Zero . An example configuration for the X axis might be:

```
 + LIMIT =  15.0500
 - LIMIT = -15.0500
HOME POS =  15.0000
```

This sets Machine Zero for the X axis in the middle of axis travel with Machine Home at the positive limit of travel. You can locate Machine Zero and Machine Home at any point in the travel of the axis, as the Home position is within the axis travel limits.

CAUTION: Any values can be entered for +LIMIT and -LIMIT. The only requirement for linear axes is that the +LIMIT value be more positive than the -LIMIT. However, for rotary axes the limits will not be detected if:

$$+LIMIT - (-LIMIT) > 359.999^{\circ}$$

Failure to note these requirements will render this important protection feature useless.

---

19. Max Rate

Max Rate is the maximum feedrate for a specific axis. The control uses the most restrictive rate of either the MAX RATE or the RAPID rate for the axis, to limit the feedrate during a move.

20. Following Error Adjustments (numeric entry)

These parameters set the way the control responds to changes in following error.

MIN FE - is the maximum deviation from zero following error allowed by the control at zero desired speed. If you tried to turn the axis by hand beyond this amount, the axis would tell the control to go into an "AXIS EXCESS LAG" condition (the axis will "drop out" -- that is, the drives will shut down). During initial tuning, if your axis "drops out" before it even starts to move, the MIN FE value should be increased slightly, and you should try again. The default value is 0.0020.

MUL FE - is a multiplying factor of the current programmed rate that determines an upper limit to the following error during a move. During initial tuning, if the axis drops out during acceleration or during the move, you should increase MUL FE slightly and try again. The default value is 15. The range is 1 to 99.

TIM CON - is a factor the control uses to track following error during deceleration. It determines the rate of decay of the maximum allowable following error. During initial tuning, if the axis drops out on deceleration, increase the TIM CON value by an increment of one and try again. The default value is 3. The range is 1 to 5.

CAUTION: Care should be taken so that these values monitor the following error closely, that is they are not too loose with respect to following error. Failure to do so could affect the control's ability to quickly detect a loss of feedback.

Important: Once the axis is responding well and has gone through complete tuning, it is sometimes desirable to go back and restore the default values of these parameters or reduce the amount of each. This will "tighten" the monitor of following error.

21. BACKLASH (numeric entry)

This parameter specifies a value for Reversal Error Compensation for the axis. Backlash is the amount of slack apparent when an axis changes direction. The method of compensation used by the control takes up the slack just before the move starts when the axis is commanded to change direction. The amount of slack that is taken up is the amount specified with this parameter. The default value for all axes is 0.0000, in other words, no Backlash Take-Up.

Note that you can specify the amount of backlash separately from this parameter using [BACKLASH COMP] on the AMP page. When you use [BACLASH COMP], it changes the corresponding values in the control parameters page. It is intended for independent access to backlash values for those users that do not have access to control parameters.

22. + or - FIRST TO HOME (toggle)

This parameter is toggled between + and - each time [ENTER] is pressed. It determines the initial direction the axis will move when it seeks the home switch during the Machine Home operation. The default direction of movement is +.

There are two situations that could occur:

- The home switch is at the positive or negative limits of axis travel.

  if the home switch is at the positive limit of travel, then you should set +. If it is at the negative limit of travel, then you should set -. Remember that with standard ladders, you have to make a MASK parameter entry that sets homing to the limit switches.

- The home switch is at some point in the middle of axis travel.

  The switch must be held closed on one side of the home point, and it must be held open on the other side of the home point. If the axis must travel in a positive direction to close the switch, then you should select +. If it must travel in a negative direction to close the switch, then you should select -. Remember that with standard ladders, you must make a choice for the MASK parameter that lets the control home to switches that are separate from the travel limit switches.

23. HOME % # (numeric entry)

This entry determines the initial homing speed of the axis.

A numeric entry between 2 and 15 can be specified. The control automatically limits entries to this range.

The Home % # you choose clamps the initial home speed at some percent of the rapid speed, based on the following equation:

$$\text{initial homing speed} = \frac{R \times N}{15}$$

where: **R** = the rate entered for RAPID

N = the HOME % # entry

EXAMPLE: RAPID set to 400 ipm, HOME % # set to 3

$$\text{homing speed} = \frac{400 \times 3}{15} = 80 \text{ ipm}$$

The initial homing speed, in this case, is 80 IPM. If another axis has a different RAPID rate, it will home at a different speed based on the above equation. This is also affected by the the setting of the Feedrate Override switch, but only when it is used to <u>reduce</u> the initial homing speed.

Also, if the Feedrate Override switch is set to 20% or 10%, The initial home feedrate will be forced to 5 or .5 IPM respectively. when the Feedrate Override Switch is set to 100%.

CAUTION: Care should be used when setting the initial homing speed to a high feedrate. If the home point is at the travel limits, the control may not detect the home switch in time to decelerate before reaching the hard stop.

This initial homing speed is used as the axis first moves toward the limit switch. Once the limit switch is made the axis will reverse direction and back off the limit switch at the lower speed of either:

$$\frac{20}{150}(\text{rapid}) \text{ IPM} \quad \text{or} \quad 5 \text{ IPM}$$

After coming off the limit switch, the axis continues in the reverse direction at .5 IPM until it detects a marker pulse. It then reverses direction again at .5 IPM until it reaches the edge of the marker. This is home position. If you are homing at Metric feedrates the last two speeds will be 50 MMPM and .5MMPM.

Axis D/A output is based on axis integration parameters, See section 12.

24. Position Tolerances (numeric entry)

These parameters establish "in position" tolerances for the axis for various modes of operation.

HOME TOL - sets the farthest distance the axis can be from actual home to be considered "homed". This does not affect accuracy of the home position, or positioning accuracy. It simply determines when the "H" indicator is shown in reverse video on the Status page, or the asterisk (*) is shown in reverse video on the machine home page. It also determines when the PAL home flags will be set, and therefore may affect PAL operation. The default value for each axis is 0.0050 inch.

BLNC TOL - sets the amount for the servo autobalance circuitry. The control can compensate for small inherent imbalances in the servo mechanism. This parameter establishes a window in which the autobalance function is disabled. The default value for each axis is 0.0005 inch. Whenever actual position of an axis at rest is farther away from its programmed position than this distance, the control will output a servo command to reduce the difference.

INPS TOL - sets the farthest distance the axis can be from its programmed position to be considered in position for G00 (rapid point-to-point) and G73 (feedrate point-to-point). This does not affect the final positioning accuracy of the system. It simply determines how close to the endpoint of one move the axis must be before the next move can begin. It also determines when the PAL in-position flags will be set, and therefore may affect PAL operation. The "I" indicator on the Status page will appear in reverse video when in-position is achieved. The default value is 0.0025 inch.

25. Inch/Metric Axis Scaling Parameters (numeric entry)

Each axis has a multiplier (MUL) parameter and a divider parameter (DIV) that are used to match the encoder feedback pulses to the control's resolution. MUL is used to multiply the feedback pulses so the control counts more pulses than the feedback device actually produces.

DIV is used to divide the feedback pulses so the control counts fewer pulses than the feedback device actually produces. Together, MUL and DIV allow one encoder to work in many applications. For example, using MUL = 3 and DIV = 2 would make a 500 line encoder look like a 750 line encoder.

MUL and DIV must be entered as whole numbers and each can range from 1 to 9999. The default values for MUL and DIV are 1 and 1 which match the feedback from a 500 line encoder coupled directly to a 5 pitch leadscrew to a control resolution of 10,000 pulses per inch programmed (LDEC = 4). Once values for MUL and DIV are determined, a metric equivalent must be calculated and entered separately (this allows the internal scaling routine to be simpler and to be executed more quickly).

If the axis being scaled has an inch leadscrew or a feedback device that produces feedback as pulses per inch of motion, then MUL and DIV are derived as inch quantities and the metric equivalent is calculated by multiplying the fraction MUL/DIV by 254/100 (a whole number fraction equal to 2.54). Thus, the metric equivalent of the default values MUL = 1 and DIV = 1 are MUL = 254 and DIV = 100 (MUL = 127 and DIV = 50 would also work properly).

If the axis being scaled has a metric leadscrew or a feedback device that produces feedback as pulses per millimeter of motion, then MUL and DIV are derived as metric quantities and the inch equivalent is calculated by multiplying the fraction MUL/DIV by 100/254 (a whole number fraction equal to 1/2.54). The following sections on inch, metric, and rotary mode scaling contain formulas that are solved to obtain values for the fraction MUL/DIV. This fraction should be reduced as much as possible and the numerator used as the MUL value and the denominator as the DIV value.

Inch Mode Scaling

The selection for LDEC affects the control resolution. The values for inch mode are:

For LDEC =	Control Resolution =
3	1,000 pulses per inch
4	10,000
5	100,000

Linear Axis with Rotary Encoder

To scale a linear axis with a rotary encoder:

$$\frac{MUL}{DIV} = \frac{CR}{(4E)\ (N)\ (P)}$$

where: CR = the control resolution from LDEC in pulses per inch

E = the encoder lines per encoder revolution

N = the ratio of encoder turns to leadscrew turns

P = leadscrew pitch in turns per inch

EXAMPLE: 4 pitch leadscrew, 1000 lines per revolution, encoder mounted on motor, 2:1 speed reduction from motor to leadscrew, LDEC set to 4

$$\frac{MUL}{DIV} = \frac{10{,}000}{(4)(1000)(2)(4)} = \frac{10{,}000}{32{,}000}$$

This can be reduced to 5/16 so MUL = 5 and DIV = 16.

The metric equivalent would be:

(5/16)(254/100) = 1270/1600.

This can be reduced to 127/160 so metric MUL = 127 and metric DIV = 160.

Linear Axis with Linear Scale

To scale a linear axis with a linear encoder (glass scale):

$$\frac{MUL}{DIV} = \frac{CR}{4E}$$

where: CR = the control resolution from LDEC

E = the encoder lines per inch

EXAMPLE: 500 line per inch glass scale, LDEC = 3

$$\frac{MUL}{DIV} = \frac{1000}{(4)(500)} = \frac{1000}{2000}$$

This can be reduced to 1/2 so MUL = 1 and DIV = 2.

The metric equivalent would be:

(1/2)(254/100) = 254/200

This can be reduced to 127/100 so the metric MUL = 127 and the metric DIV = 100

Metric Mode Scaling

The values for the effect of LDEC on control resolution in metric mode are:

For LDEC =	Control Resolution =
3	100 pulses per mm
4	1000
5	10000

Linear Axis with Rotary Encoder

To scale a linear axis with a rotary encoder:

$$\frac{MUL}{DIV} = \frac{(CR)(L)}{(4E)(N)}$$

where: **CR** = the control resolution from LDEC

L = the lead of the leadscrew in mm per turn

E = the encoder lines per encoder revolution

N = the ratio of encoder turns to screw turns

EXAMPLE: 5mm leadscrew, 500 line encoder mounted on the motor, 1:1 ratio between motor and leadscrew, LDEC set to 4

$$\frac{MUL}{DIV} = \frac{(1000)(5)}{(4)(500)(1)} = \frac{5000}{2000}$$

This can be reduced to 5/2 so MUL = 5 and DIV = 2.

The inch equivalent would be:

$$(5/2)(100/254) = 500/508.$$

This can be reduced to 125/127 so inch MUL = 125 and inch DIV = 127.

Linear Axis with Linear Scale

To scale a linear axis with a linear encoder (glass scale):

$$\frac{MUL}{DIV} = \frac{CR}{4E}$$

where: CR = the control resolution from LDEC

E = the encoder lines per mm

EXAMPLE: 100 line per mm glass scale, LDEC set to 3

$$\frac{MUL}{DIV} = \frac{100}{(4)(100)} = \frac{100}{400}$$

This can be reduced to 1/4 so MUL = 1 and DIV = 4.

The inch equivalent would be:

$$(1/4)(100/254) = 100/1016.$$

This can be reduced to 25/254 so inch MUL = 25 and inch DIV = 254.

Rotary Mode Scaling

The choice for LDEC does not affect control resolution for a rotary axis. There are always 1000 pulses per degree.

Rotary Axis with Rotary Encoder

To scale a rotary axis with a rotary encoder:

$$\frac{MUL}{DIV} = \frac{360{,}000}{(4E)(R)}$$

where: **E** = the encoder lines per encoder revolution

R = the gear reduction of the rotary table

EXAMPLE: 500 line encoder mounted on the motor, 90:1 rotary table

$$\frac{MUL}{DIV} = \frac{360{,}000}{(4)(500)(90)} = \frac{360{,}000}{180{,}000}$$

This can be reduced to 2/1 so MUL = 2 and DIV = 1.

A rotary axis does not require the calculation of a metric equivalent, simply put 2 in both MUL locations and 1 in both DIV locations. (Yes, it is necessary to enter both locations.)

Rotary Axis with Linear Type Scale

To scale a rotary table with a linear type encoder (rotary glass scale):

$$\frac{MUL}{DIV} = \frac{360{,}000}{4E}$$

where: **E** = the encoder lines per table turn

EXAMPLE: rotary table with 18,000 lines per turn glass scale

$$\frac{MUL}{DIV} = \frac{360{,}000}{(4)(18{,}000)} = \frac{360{,}000}{72{,}000}$$

This can be reduced to 5/1 so MUL = 5 and DIV = 1.

Again, a rotary axis does not require the calculation of metric equivalents, simply put 5 in both MUL locations and 1 in both DIV locations.

26. GAIN (numeric entry)

This is the gain factor (G) as described in the axis integration notes earlier in this section.

It is a signed numeric entry with a default value of -2. The number is used internally by the control in the system gain calculations but it is not the actual system gain.

In general, more negative gain factor numbers here result in an overly "loose" system with long acceleration/ deceleration times. More positive gain factor numbers result in overly "stiff" performance with oscillations on acceleration and deceleration. In general, systems with greater resolution (LDEC = 5) will require more negative gain factor numbers, systems with less resolution (LDEC = 3) will require more positive gain numbers. See the axis integration notes earlier in this section.

27. Acceleration /Deceleration Parameters

The Acc/Dec feature is provided to protect mechanical machine components from damage caused by sudden changes in velocity. Acc/Dec allows the operator to set acceleration-deceleration rates for each axis individually. The rate of Acc/Dec is defined by the operator as an AMP parameter, as is the maximum step in velocity the machine can accept. Acc/Dec can be enabled/disabled on the CONTROL PARAMETERS page on a per axis basis.

Acc/Dec adds three new parameters to the CONTROL PARAMETERS page, **RAMP/.1S**, **VEL STEP** and **ACC/DEC**.

RAMP/.1S specifies the acceleration/deceleration rate at which the control will operate. The units for RAMP/.1S entries are axis feed units/.1 sec. For example: 50 inches/min./.1 sec. The values of the entries have the same ranges as the MAX RATE parameter entry.

VEL STEP specifies the maximum step in velocity the axis should make at the start or end of a move. This value is a design limitation of the mechanical components of the machine tool. The values of the entries have the same ranges as the MAX RATE parameter entry. If a change in velocity exceeds the VEL STEP value, the axis will be ramped to speed at the specified RAMP/.1S rate.

ACC/DEC parameter determines if accel/decel is enabled or disabled on a per axis basis. The parameter toggles between 'use' and 'no' when the enter key is pressed.

ACC/DEC Operation

ACC/DEC requires the control to ramp its velocity up and down except on contact with hard limits, soft limits, inhibited zones, or E-Stop conditions. If a part program is currently running and is interrupted, the control may need to start executing the next block before it is able to finish ramping down to a stop.

This could be caused by FEED HOLD, BLK/BLK, CYCLE STOP, ABORT, CLAMPS, Hard Limit, Soft Limit, or Zone Limit. Any move must start at a rate which it can decelerate to (VEL STEP)/2 by the end of that move. Short moves may have their start rate limited to accomplish this. If short moves are not limited, they may not be able to run at the programmed feed rate.

Acc/Dec can be inhibited during program execution by executing a G97 command. If the operator wishes to disable Acc/Dec for the X and Z axis, the appropriate syntax would be G97 X Z. Following this command, if the operator wishes to enable Acc/Dec for X and Z and disable Y, the proper command would be G97 Y.

Notice the fact that X and Z do not appear in the G97 is an indication to the control to enable Acc/Dec for these axes. Any axis that requires Acc/Dec must be enabled on the CONTROL PARAMETERS page. Acc/Dec cannot be inhibited by a G97 command for Jog continuous, Jog handwheel, Jog incremental, Return from jog, the Machine Home sequence, and Axis Calibration.

If motion is requested that requires velocity changes on more than one axis with Acc/Dec enabled, the control will use the lowest MAX RATE of all motion axes. It will also limit the RAMP/.1S and VEL STEP to the lowest of all motion Acc/Dec axes.

Ending Control Parameter Programming

When all parameters have been entered for control parameters, you can exit the control parameters page and enable the drives.

Make thorough tests of system performance after programming control parameters and modify them accordingly. Once control parameters have been programmed, save AMP to a storage device, and copy the parameters on paper. Keep them for your records.

12.4.2 Input/Output

The serial I/O configuration page lets you define the serial I/O operation of the control for up to 8 peripheral devices. Through the I/O page you can:

- assign a unique name to a device
- specify whether a device is used for input and/or output
- specify input and output baud rates for a device
- set up the control for RS-491 Level II protocol

To access the I/O Page, press [CONFIGURE I/O] on the AMP page. The I/O page will appear. If this is the first time you have accessed the I/O page, it will look like figure 12.7

Figure 12.7 - I/O Configuration Page - Initial Settings

```
| SERIAL I/O CONFIGURATION      < operator messages>  TAPE READER | *
| Name      DEVICE 5....        <  error messages  >               |
| Mode     >BOTH           Soft Controls  OFF          TAPE PUNCH | *
| Baud In   4800            Punch Starts  ON                       |
| Baud Out  4800            Reader Starts ON              COMPUTER | *
| Baud Clk  INT            Leader Count    100                     |
| List Mode OFF            Characters                  CASSETTE I | *
| Parity    EVEN            RDR Start DC1   17                     |
| Word Size 7               RDR Stop  DC3   19        DEVICE 5.... | *
| Stop Bits 1               PUN Start DC2   18                     |
| Outbound Controls         PUN Stop  DC4   20        DEVICE 6.... | *
|  RTS(4)   ALWAYS HI       Abort     DLE   16                     |
|  DTR(20)  ALWAYS HI       End       EOT    4        DEVICE 7.... | *
| Inbound Controls          Leader           0                     |
|  CTS(5)   NONE                                    DEVICE 8.... | *
|  DCD(8)   NONE      System Start Up Mode SAVE                    |
| I/O PORT  232-P4          Default Load  CASSETTE II              | -
|                           Default Save  CASSETTE II              |
```

Choosing a Device Name

Notice that DEVICE 5...., next to the soft key, is in reverse video. This device name also appears in the Name space. Press another soft key, and that device name will highlight and appear in the name space. When you have chosen the device number you want to assign, cursor to the name space, type in the name up to 12 characters long, and press [ENTER]. The name you entered will appear in place of the device number next to the soft key. This is the name that will appear before SAVE or LOAD on the Load/Save page when that device is selected.

System Start up Mode (toggle)

This selects LOAD or SAVE as the power-up I/O mode for the control. This option tells the control how to configure the serial I/O port on power-up based on the currently selected load device or save device. To change between load and save, cursor to the right of Mode and press [ENTER].

The remaining options on the page configure the I/O port for a particular device.

Mode (toggle)

This lets you set a device for LOAD, SAVE, BOTH, or NONE.

- LOAD -- the device can only be used to load data into the control. On the Load/Save page, you can only select LOAD with Load Select, not with Save Select.
- SAVE -- the device can only be used to save data from the control. On the Load/Save page, you can only select SAVE with Save Select, not with Load Select.
- BOTH -- the device can be used to load and save data. On the Load/Save page, you can select BOTH with either Load Select or Save Select.
- NONE -- the device cannot be used for load or save, this is the setting for an unassigned device.

Baud In (toggle)

This selects the baud rate used for LOAD, VERIFY, and EXECUTE FROM I/O. This baud rate is used by both the receive and transmit channels during input. This does not need to be the same rate as Baud Out.

You can select one of the following baud rates:

75, 110, 300, 600, 1200, 2400, 4800, 9600

Cursor to the space, and press [ENTER] for different rates.

Baud Out (toggle)

This selects the baud rate used for SAVE. This baud rate is used by both the transmit and receive channels during output. This does not need to be the same rate as Baud In.

You can select one of the following baud rates:

75, 110, 300, 600, 1200, 2400, 4800, 9600

Cursor to the space and press [ENTER] to select different rates.

Baud Clock (toggle)

The baud clock toggles between INT (internal) and EXT (external) clock source. This feature defaults to INT.

List Mode (toggle)

When List Mode is ON, the control assumes the output device is a printer. A SINGLE program SAVE to this device will have a line number at the beginning of each line, and a # at the end of each block.

For example,

```
00001 ;CIRCLE#
00002 G91#
00003 G2XYIJ-.5#
00004 M2#
```

The line number output cannot be loaded. Its purpose is simply to provide a listing for program documentation or to make editing programs easier.

List mode should be OFF if the device is not a printer.

Cursor to the space and press [ENTER] to change between ON and OFF.

Parity (toggle)

This selection lets you specify parity as ODD, EVEN, or NONE.

During output, the control generates code with the specified parity. During input, the control checks for the specified parity.

Parity is generally used with 7 bit words only.

Cursor to the parity space and press [ENTER] to toggle between ODD, EVEN, and NONE. Set parity to EVEN for ASCII. Set it to NONE for EIA. (The control cannot output EIA code. On input of EIA code, you must be set up for 8 bits, no parity.) The NONE option does not generate parity.

Word Size (toggle)

You can specify the size of character words, <u>not including parity</u>, as 7 or 8 bits.

Cursor to this space and press [ENTER] to toggle between 7 and 8 bits.

Stop Bits

This selects the number of stop bits generated during a SAVE (send) operation as 1 or 2 bits.

The number of stop bits recognized during LOAD (receive) operations is always 1. We recommend, however, that the device be set up to send 2 stop bits to the control. This allows a small amount of time between characters for synchronization of data.

Cursor to this selection and press [ENTER] to toggle between 1 and 2.

Outbound Controls RTS and DTR (toggle)

This area lets you configure the RTS and/or DTR output lines for handshaking. These lines let the device know when the control is ready for data.

The selections you can make for each line are:

- ALWAYS HI -- the line will always be high (+12V)
- ALWAYS LO -- the line will always be low (-12V)
- BUSY HI -- the line will be high when the control cannot accept input data, low when the control can accept input data
- BUSY LO -- the line will be low when the control cannot accept input data, high when the control can accept input data
- INPUT HI -- the line will be high when the control is in input mode, low when the control is in output mode (same as BUSY LO)
- INPUT LO -- the line will be low when the control is in input mode, high when the control is in output mode (same as BUSY HI)
- OUTPUT HI -- the line will be high when the control is in output mode, low when the control is in input mode
- OUTPUT LO -- the line will be low when the control is in output mode, high when the control is in input mode

Cursor to the line and press [ENTER] to sequence through these choices.

Inbound Controls CTS and DCD (toggle)

The CTS and DCD input lines can be used to let a device guide the transmission of data from the control. You can configure either or both of these lines as a handshake line to signal when the device is ready to accept data.

The choices you can make for each line are:

- NONE -- the state of the line (low or high) has no meaning
- BUSY HI -- the line is high when the device is not ready to accept data, low when the device is ready to accept data
- BUSY LO -- the line is low when the device is not ready to accept data, high when the device is ready to accept data

Cursor to the space and press [ENTER] to sequence through these options.

I/O Port (toggle)

This option determines whether the input/output is done through the RS232 or the RS422 port. These are physically two different connectors on the control: P4 and P8, respectively. The field entry will toggle between "232-P4" and "422-P8".

Soft Controls (toggle)

This option determines whether or not character codes are used to control the serial data flow and the device. You can set this option to either ON or OFF. When ON, it activates a software control mode that can be configured similar or equivalent to EIA RS-491 Level II protocol.

Cursor to this selection and press [ENTER] to choose between ON and OFF.

Punch Starts (toggle)

When Soft Controls is ON, this selection determines whether the control will wait for a response after sending the initial PUN Start character.

If the punch does not need to be commanded on, set Punch Starts to ON. In this case, the control does not wait after sending the initial PUN Start character, but starts sending data immediately.

If the punch must be commanded on before it can punch data, set Punch Starts to OFF. In this case, the control sends a PUN Start character every three seconds until the punch responds with the RDR Start character to indicate it is ready to receive data.

Cursor to this selection and press [ENTER] to choose between ON and OFF.

Reader Starts (toggle)

When Soft Controls is ON, this selection determines whether or not the control sends a RDR Start character to start the reader during a Load operation.

If the reader does not have to be commanded to turn on, set Reader Starts to ON. In this case, the reader reads and sends data when the outbound hardware controls (if used) are set in the ready mode, without RDR Start being sent.

If the reader must be commanded to turn on, set Reader Starts to OFF. In this case, the control sends a RDR Start character every three seconds until it receives input.

Cursor to this selection and press [ENTER] to choose between ON and OFF.

Leader Count (numeric)

This option determines how many leader and trailer characters are generated. It also determines how many such characters are generated when FEED is pressed on the SAVE page. The character used is defined with the Leader selection under Characters.

With a device such as a tape punch, there are typically 10 characters per inch in leaders, and the character is defined as a null.

With a device such as a printer, there is typically one leader character, and the character is a form feed instead of a null.

Cursor to the Leader Count selection, type in the number you want, and press [ENTER] to specify the number.

Characters (numeric)

This area lets you define the software control characters used with the input or output device.

Cursor to the selection you want, type in a decimal number that is equivalent to the control character you want to specify, and press [ENTER].

- RDR Start DC1 -- defines the character:
  - sent to the load device (reader) to turn it on and to begin sending data before a load.
  - sent by the save device (punch) to tell the control to send data when Soft Controls is ON.
  - expected in response to a PUN Start character to indicate that the save device is turned on

    If you specify 0, no character is sent. To send a 0 (with even parity and a word size of 7 bits), enter 128.

- RDR Stop DC3 -- defines the character:
  - sent to the load device (reader) to tell it to stop sending data, typically because the control's buffer is full. The RDR Start character is used to resume transmission.
  - the save device (punch) will send to the control to tell the control to suspend data transmission, typically because its buffer is full. The RDR Start character is sent to the control to resume operation.

  If you specify 0, no character is sent. To send a 0 (with even parity and a word size of 7 bits), enter 128.

- PUN Start DC2 -- defines the character sent to the save device (punch) to turn it on for a save operation. If you specify 0, no character is sent. To send a 0 (with even parity and a word size of 7 bits) enter 128.

- PUN Stop DC4 -- defines the character sent to the save device (punch)to turn it off after a save operation. If you specify 0, no character is sent. To send a 0 (with even parity and a word size of 7 bits) enter 128.

- Abort DLE -- defines the character generated by the control if the operator aborts the I/O operation. This character is sent even if it is zero.

  If this character is received during a LOAD, the load operation is aborted and a message is displayed.

- End EOT -- defines the character generated at the end of output from a SAVE operation. If it is defined as null (0), no character is generated.

- Leader -- defines the character generated for leader or trailer on tape or form feed character for a printer, and the character generated by the FEED soft key on the SAVE page. You would normally use a null (0) for devices such as tape punches, or a form feed character for printers.

System Start Up Mode (toggles)

This determines whether the control will initialize the I/O ports for Load or Save conditions at power up. Some I/O devices may require this to assure acceptable conditions on power up.

Default Load (toggles)

This determines which of the defined input devices is selected at power up.

Default Save (toggles)

This determines which of the defined output devices is selected at power up.

Example SAVE with Soft Controls ON

The following sequence takes place when Soft Controls is ON.

1. If Punch Starts is OFF, and PUN Start is not 0, the control sends the PUN Start character.
2. If Punch Starts is OFF, the control waits for a RDR Start character from the punch indicating that it is ready to receive data. If the character is not received in 3 seconds, the control sends the PUN Start character and waits again.
3. The control sends the Leader character, Leader Count times.
4. The control sends % crlf (carriage return and linefeed) as a rewind stop character.
5. The control sends data.
6. The control sends the Leader character, Leader Count times.
7. The control sends the End EOT character, if it is not 0, to mark the end of transmission.
8. The control sends the PUN Stop character, if it is not 0.
9. The control receives a RDR Stop character indicating that the punch has stopped (optional).

At any point in the above sequence:

- If you abort the SAVE operation , the control sends the Abort character to the punch.
- If the control receives the RDR Stop character, output from the control stops and the control waits for the RDR Start character.

Example LOAD with Soft Controls ON

The following sequence occurs when Soft Controls is ON, and a LOAD operation is performed.

1. If Reader Starts is OFF, or if the RDR Start character is not 0, the control sends the RDR Start character.

2. The control waits for input. If none is received within 3 seconds, the control sends the RDR Start character and waits again.

3. If the control's input buffer fills, it sends a RDR Stop character to suspend input. The control sends a RDR Start character when it is ready to accept more data.

4. When the control has received enough data, it sends the RDR Stop character.

Note that if you abort the LOAD at any point in the above sequence, the control sends the Abort character.

Typical I/O Configurations

The following figures show some standard configurations on the I/O page for selected devices.

The control maintains the configuration for the 215-100 06 CASSETTE II interface as a default device. You cannot view or alter this configuration, but you can select it on the Load/Save page.

Kaypro II

This configuration is for the Kaypro II computer used as a terminal. (This is not for the downloading of PAL.) You will need a communications program for the Kaypro II, like COMMX or XMODEM.

Figure 12.8 - Configuration for Kaypro II as a Terminal

```
| SERIAL I/O CONFIGURATION      < operator messages>   TAPE READER | *
| Name      COMPUTER ...        <  error messages  >               |
| Mode     >BOTH             Soft Controls  ON          TAPE PUNCH | *
| Baud In   1200              Punch Starts  ON                     |
| Baud Out  1200              Reader Starts ON            COMPUTER | *
| Baud Clk  INT              Leader Count    100                   |
| List Mode OFF              Characters                 CASSETTE I | *
| Parity    EVEN              RDR Start DC1   17                   |
| Word Size 7                 RDR Stop  DC3   19      DEVICE 5.... | *
| Stop Bits 2                 PUN Start DC2   18                   |
| Outbound Controls           PUN Stop  DC4   20      DEVICE 6.... | *
|  RTS(4)   ALWAYS HI         Abort     DLE   16                   |
|  DTR(20)  ALWAYS HI         End       EOT    4      DEVICE 7.... | *
| Inbound Controls            Leader           0                   |
|  CTS(5)   NONE                                      DEVICE 8.... | *
|  DCD(8)   NONE      System Start Up Mode SAVE                    |
| I/O PORT  232-P4            Default Load  CASSETTE II            | -
|                             Default Save  CASSETTE II            |
```

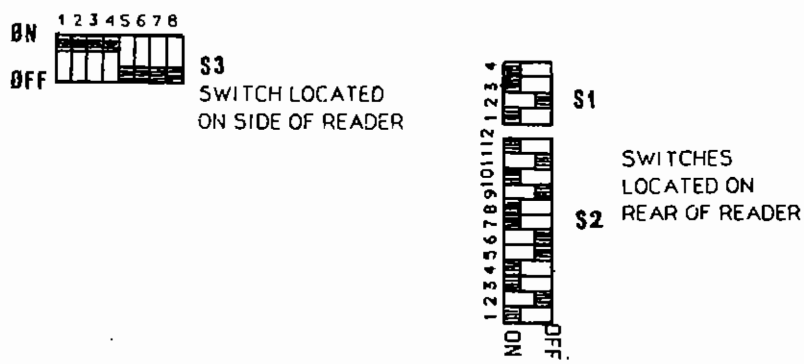
Decitek Tape Reader

This is the required configuration for the Decitek tape reader Allen-Bradley Catalog No. (8000XRDR or 8000XRDRP), when used with cable catalog No. 8410-CBTR The control will accept input with this configuration in ASCII tape code only (See Figure 12.9).

Figure 12.9 - Required Configuration for Decitek Tape Reader

```
| SERIAL I/O CONFIGURATION      < operator messages>  TAPE READER | *
| Name       TAPE READER        <  error messages  >              |
| Mode     >LOAD           Soft Controls  ON           TAPE PUNCH | *
| Baud In   2400            Punch Starts  OFF                     |
| Baud Out  2400            Reader Starts ON             COMPUTER | *
| Baud Clk  INT            Leader Count    100                    |
| List Mode OFF            Characters                 CASSETTE I  | *
| Parity    EVEN            RDR Start DC1   17                    |
| Word Size 7               RDR Stop  DC3   19       DEVICE 5.... | *
| Stop Bits 1               PUN Start DC2   18                    |
| Outbound Controls         PUN Stop  DC4   20       DEVICE 6.... | *
|  RTS(4)   BUSY LO         Abort     DLE   16                    |
|  DTR(20)  ALWAYS HI       End       EOT    4       DEVICE 7.... | *
| Inbound Controls          Leader           0                    |
|  CTS(5)   NONE                                   DEVICE 8.... | *
|  DCD(8)   NONE      System Start Up Mode SAVE                   |
| I/O PORT  232-P4          Default Load  CASSETTE II             | -
|                           Default Save  CASSETTE II             |
```

Decitek Switch Settings for Model 762B9-008, AB cat. no. 8000-XRDR and 8000-XRDRP.



Decitek Switch Settings for Model 762B9-008, A-B cat. no. 8000-XRDR and 8000-XRDRP. continued

SWITCH #	POSITION	FUNCTION
S1 -1	ON	
S1 -2	OFF	SETS RS232
S1 -3	ON	MODE
S1 -4	ON	
S2-1	ON	1 STOP BIT
S2-2	OFF	"
S2-3	ON	PARITY OFF
S2-4	ON	"
S2-5	OFF	NO FUNCTION
S2-6	OFF	"
S2-7	ON	LEVEL II PROTOCOL
S2-8	ON	NO CODE CONVERSION
S2-9	OFF	SET BAUD TO
S2-10	ON	2400
S2-11	OFF	
S2-12	ON	
S3-1	ON	DCD INT. PULLUP
S3-2	ON	CTS " "
S3-3	ON	DSR " "
S3-4	ON	RS232 MODE
S3-5	OFF	NO FUNCTION
S3-6	OFF	"
S3-7	OFF	PASSIVE LOOP
S3-8	OFF	"

DSI Tape Punch

This is the configuration for a DSI Tape Punch. When used with Allen-Bradley cable catalog No. 8410-CBTR

Figure 12.10 - Configuration for DSI Tape Punch

```
| SERIAL I/O CONFIGURATION      < operator messages>   TAPE READER | *
| Name         TAPE PUNCH       <  error messages  >               |
| Mode     >SAVE            Soft Controls  OFF          TAPE PUNCH | *
| Baud In   600              Punch Starts  ON                      |
| Baud Out  600              Reader Starts ON             COMPUTER | *
| Baud Clk  INT             Leader Count    100                    |
| List Mode OFF             Characters                 CASSETTE I  | *
| Parity    EVEN             RDR Start DC1   17                    |
| Word Size 7                RDR Stop  DC3   19        DEVICE 5....| *
| Stop Bits 2                PUN Start DC2   18                    |
| Outbound Controls          PUN Stop  DC4   20        DEVICE 6....| *
|  RTS(4)   ALWAYS HI        Abort     DLE   16                    |
|  DTR(20)  ALWAYS HI        End       EOT    4        DEVICE 7....| *
| Inbound Controls           Leader           0                    |
|  CTS(5)   NONE                                       DEVICE 8....| *
|  DCD(8)   BUSY  LO System Start Up Mode SAVE                     |
| I/O PORT  232-P4           Default Load  CASSETTE II             | -
|                            Default Save  CASSETTE II             |
```

I/O Formats for External Tape Preparation

The following examples show how the control formats programs for a SAVE operation. You can use these examples to generate programs "off-line" on tape for loading into the control.

Important: These formats do not apply to AMP, Fixtures Offsets, or Tool Offsets. The format for these data transfers is Intel Hex compatible. However, the data from a control having one revision of firmware is not necessarily compatible with that from a control having a different revision of firmware.

Format for SINGLE LOAD/SAVE

This format is used when you LOAD or SAVE programs one at a time using the SINGLE option.

Data	Notes
leader	Optional on input
% crlf	crlf = carriage return and line feed. Only the line feed is required for input.
block crlf	The cr is not required on input. Note that the # is not included with the block.
block crlf . . .	More data blocks.
block crlf	This block is normally M2 or M30.
leader	Optional on input.

Format for Multiple

This format is used when you "string" programs together for input or output using the MULTIPLE option.

Data	Notes
leader	Optional on input
% crlf	crlf = carriage return and line feed. Only the line feed is required for input.
:Pnnbbbb crlf	Where nn is the program number in hexadecimal, always 2 digits. And bbbb is the program length in characters (or bytes) in hexadecimal, always 4 digits. The crlf is optional on input.
block # crlf	Note that the # is included. The crlf is optional on input.

block # crlf	More blocks in the first program.
.	
.	
.	
block # crlf	Usually an M2 or M30, the last block in the first program.
:Essss crlf	Where ssss is the checksum (see Note 1, below) for the the first program in hexadecimal, always 4 digits. The crlf is optional on input.
:Pnnbbbb crlf	Next program.
.	
.	
.	
:Essss crlf	Checksum for this program.
.	
.	
.	
:Pnnbbbb crlf	Last program
.	
.	
.	
:Essss crlf	Checksum for last program.
:X crlf	Stops input. (This was not implemented in rev C). The crlf is optional on input.
leader	Optional on input.

Important: The checksum is generated by beginning with a 16 bit word that has a value of HEX 11. Then each 7 bit character in the program, including the # but not the crlf, is added and the result rotated one bit left (by multiplying by the hex value of two) after each add. If the resultant sum exceeds 4 hex digits the 5th digit is added to the 1st digit, then the 5th digit is dropped. Only the characters of the program are included in the checksum, not the %, :P, or :E lines or the crlf.

12.4.3 Backlash Comp

Pressing [BACKLASH COMP] on the AMP page brings the Backlash Comp page to the screen. Access to this page is controlled by access control flag 1. see figure 12.11.

Figure 12.11 - Backlash Comp Page

```
| BACKLASH COMP                                       U | *
|                                                       |
|                                                     V | *
|                                                       |
|                                                     W | *
|                                                       |
|                                                     X | *
|                              >BACKLASH   0.0000       |
|                                                     Y | *
|                                                       |
|                                                     Z | *
|                                                       |
|                                                       | -
|                                                       |
|                                                       | -
|                                                       |
|                                                       | -
|_______________________________________________________|
```

The Backlash Comp page gives you the ability to change backlash values for each axis without accessing the Control Parameters page.

Backlash

Backlash is the amount of mechanical slack in an axis. If your machine has mechanical slack, and you don't compensate for it, there will be inaccuracies in movements when the axes change direction.

Backlash compensation lets you automatically compensate for mechanical slack. You specify the amount of slack on the Backlash Comp page, and each time the axis changes direction the control will take up the slack by the amount you specify. The control then goes on to complete the move you programmed.

Entering Backlash Comp Values

To enter Backlash Comp values:

1. Choose the axis you want to modify and press its corresponding softkey. The selected axis will appear in reverse video. The control will display the value that is currently entered on the Control Parameters page for that axis.

2. Type in the value you want for compensation for the axis and complete the entry by pressing [ENTER].

3. Select the next axis and continue until you have entered all the values you need.

   When you finish entering backlash comp values, press [EXIT] to return to the AMP page. Any values you change will automatically be changed on the AMP Control Parameters page.

12.4.4 Axis Calibration

This feature provides a means to adjust for certain inaccuracies in feedback positioning caused by mechanical deficiencies in a machine. A table of correction factors is created for each axis which add or subtract from the indicated feedback position to provide a corrected position.

Entering Axis Calibration Parameters

Axis calibration is set up using the AXIS CAL Selection under the amp selections on the support page.

An axis must be `homed' before axis calibration is attempted. Otherwise, a **MACHINE HOME REQ'D** error message will result. The figure below shows the AXIS CAL screen.

Figure 12.12 - Axis CAL

```
| AXIS CALIBRATION        < operator messages>        NEXT POINT | *
| -IDLE                   <  error messages  >                   |
|                                                        UPDATE  | *
|                         >DISABLED    COUNTS                    |
|        CAL TABLE START    2.0000     20000             CLEAR   | *
|        INTERVAL DIST.    -0.8192     -8192                     |
|        NUMBER OF POINTS        0                           X   | *
|        CAL TABLE END      2.0000     20000                     |
|        APPROACH DIST.     0.0000         0                 V   | *
|                                                                |
|        CURRENT POSITION  -0.0002                           W   | *
|        LAG ERROR          0.0000                               |
|                                                            X   | *
|        CURRENT POINT           1                               |
|        DESIRED LOCATION   2.0000     20000                 Y   | *
|        ACTUAL LOCATION    2.0000     20000                     |
|        CORRECTION DIST.   0.0000         0                 Z   | *
|________________________________________________________________|
```

Axis Selection

The axis that is to be calibrated is selected by pressing the appropriate soft key labeled with the desired axis letter. When an axis is selected, the letter corresponding to the selected axis will appear in reverse video, and all axis related fields on the page will display data pertaining to the selected axis. If the axis calibration page is exited, the currently selected axis will be selected on return to the page.

Enabled/Disabled

A selected axis may have axis calibration either ENABLED or DISABLED. To change the state of a selected axis, position the cursor on the ENABLED/DISABLED field of the page and press the [ENTER] key. Each time the ENTER key is pressed, the field will toggle.

Important: axis calibration will not be applied when on the axis calibration page regardless of ENABLE/DISABLE selection.

Cal Table Start

The starting location of the axis calibration section for a selected axis is entered by positioning the cursor on the field labeled CAL TABLE START. The position is entered in program units (Inch/Metric).

Interval Distance

The distance from one calibration point to the next is the Interval Distance. The Interval Distance is calculated by the control using the AMP page MLT and DIV entries. These Interval Distances are based on 2^5 - 2^{14} encoder counts multiplied by the DIV entry and divided by the MLT entry.

There are 20 values for Interval Distance (-16384 * DIV)/MLT through (16384 * DIV)/MLT incremented by 2^n. The sign of the interval indicates the direction of calibration. For example, an interval distance of -000.0512 would indicate on Interval Distance of .0512 and a calibration direction from positive to negative.

To select the Interval Distance, position the cursor on the field labeled INTERVAL DIST. and press the [ENTER] key. The field will step to the next larger interval value. If the largest value is currently displayed, the field will wrap around to the smallest value. Interval distances are selected on a per axis basis for both size and direction. The interval size is displayed in both (inch/metric) and encoder counts.

[Update] -- When the soft key labeled [UPDATE] is pressed, the current axis calibration parameters are defined as an axis calibration point. The correction distance is calculated and displayed. The number of points for the selected axis and the cal table end value are changed if the updated point is a new point. An existing point may be recalibrated at any time by adjusting the actual location amount and pressing the update soft key.

[Clear] -- Pressing the soft key labeled [CLEAR] will cause the control to bring up a verification page to confirm that the user would like to clear all the axis calibration points for the selected axis.

If the user presses the soft key labeled [NO], the control will return to the axis calibration page and the selected axis parameters will remain unchanged. If the user presses the soft key labeled [YES], the selected axis will have all the axis calibration points cleared and will return to the axis calibration page. None of the axis calibration parameters are cleared when clearing the calibration points.

[Axis Letters] -- For the keys labeled [U], [V], [W], [X], [Y], and [Z], see the previous section titled Axis Selection.

ROTARY AXIS

A rotary axis may be calibrated over a partial section or over the full 360°. The axis may also be calibrated in the positive or the negative direction. If the full 360° are being calibrated, the interval size need not divide evenly into the number of encoder counts per revolution.

When the point nearest the full wrap location is calibrated, the control will automatically calculate the correction amount for the distance from the last point back to the first point. The user will notice the number of points display will increase by two rather than the normal one point increase.

If the user tries to move to the next point the control will issue an error message. If the last point is recalibrated the wrap interval will automatically have its correction value recalculated. Note if the calibration interval is less than 360° and the axis is to be used over the full 360°, the correction amount of the last point should match the correction amount of the first point to avoid a jump on entry to the calibrated section.

CYCLE START

When the [CYCLE START] key is pressed, the table will move to the location displayed in the field labeled DESIRED LOCATION + the amount displayed in the field labeled APPROACH DISTANCE, and then move to the location displayed in the DESIRED LOCATION field. This allows the user to remove any backlash in the table before coming to the calibration point.

FEATURES

Number of Calibration Points

There are 1000 Calibration Points.These points may be distributed over six axes in any proportions as long as the total number of points of all axes combined, does not exceed 1000.

Ease Of Data Entry

The calibration data may be entered either by typing in the number and pressing the enter key or using the hand wheel to increase or decrease the amount of the actual location displayed.

Cal Point to Cal Point Interpolation

The amount of the error correction for a position falling somewhere between two calibration points is interpolated during execution of the part program.

How to Calibrate an Axis

1. Go to the Axis Cal page by pressing the [AXIS CAL] soft key on the AMP page

2. Turn the drives on.

You will notice on entry and exit to the Axis Cal page that the drives will automatically shut off. This is done because on entry to the Axis Cal page the calibration cannot be applied to an axis, and on [exit], any axis that is enabled for calibration will have that calibration applied.

Important: when the drives are turned on, both on entry and exit from the to the Axis Cal page, the operator may see some movement on a calibrated axis. The position of least calibration is the axis home position. This is the best location for the machine on entry to the Axis Cal page.

3. Select the axis for calibration.

4. Define the calibration table start location.

5. Select the desired interval distance.

6. Define the approach distance if needed.

WARNING: performing step 7. will cause the currently selected axis to position to the CAL TABLE START location.

---

7. Press the [CYCLE START] pushbutton.

8. Either turn the handwheel to adjust the ACTUAL LOCATION field or enter the desired ACTUAL LOCATION value.

9. Press the [UPDATE] softkey to define the calibration point. Notice that the NUMBER OF POINTS field will increase by 1. The CORRECTION DIST. field will display the amount of calibration and the CAL TABLE END field will be adjusted.

10. Press the [NEXT POINT] softkey and the DESIRED LOCATION field, ACTUAL LOCATION field, and the CURRENT POINT field will all be updated.

11. Repeat steps 7 through 10 until the total span of the calibration table is completed.

12. Select the next axis to be calibrated and repeat steps 4 through 10.

13. Enable axis calibration for selected axis.

14. When calibration is complete [EXIT] the Axis Calibration page. When the drives are turned on, the calibration will be applied.

ERRORS

The following actions will cause error messages to be displayed

- Correction amount greater than maximum amount allowed.
- Next point soft key was pressed when current point was beyond existing end of cal table.
- Next point soft key was pressed and the total number of calibration points exceeded 1000.
- Next point soft key was pressed when the next calibration point interval would exceed 360 .

ERROR MESSAGES

- CALIBRATION TOO LARGE
- BEYOND END OF TABLE
- TOO MANY POINTS
- BEYOND 360 DEGREES

12.5 Program Protect

Press [PROGRAM PROTECT] on the Support page to bring the page shown in figure 12.13 to the screen. Access to this page requires access control flag 14 being set. Access to the restrict control function also requires access to Control Parameters as controlled by access control flag 2.

Figure 12.13 - Program Protect Page

```
| PROGRAM PROTECT              DISPLAY | *
| < operator messages>                 | -
|                             RESTRICT | *
|  COMMENTX_DIM                PROTECT | *
|  Program       153         NEXT PROG | *
|  Flags                      RENUMBER | *
|  Size           22       DELETE PROG | *
|  Available 143926                    | -
| <  error messages   >      DIRECTORY | *
```

The Program Protect page gives several standard functions on support of controlling program protect or restrict flags. See chapter 7 for descriptions of the features not covered in this section.

The purpose of the Program Protect page is to let you toggle the state of the R -- restricted and P -- protected flags for a program. Basically Protect is intended to protect against inadvertent program modification, Restrict is for programs that have been added to the control by an OEM or a user and are essentially considered enhancements to the control.

- **R** -- restricted. The program can not be displayed, edited, saved, renumbered, or deleted. The individual blocks will still display as the program is executed.

- **P** -- protected. The program can be displayed or saved, but it cannot be edited, renumbered or deleted.

See chapter 11 for definitions of the other flags on this page.

Use the [RESTRICTED] and [PROTECTED] soft keys to toggle these flags. When the flag is active the corresponding letter will appear in the Flags field on this page and other similar pages. The R and P flags will also appear in the program directory when these flags are active.

12.6 Diagnostics

Press the [DIAGNOSTICS] soft key on the Support page to bring the Diagnostics page to the screen. Access to this page requires access control flag 13 being set. See figure 12.14.

The main diagnostics screen can be selected from the SUPPORT page by selecting the DIAGNOSTICS soft key (3rd softkey down). The main diagnostics screen can also be selected from the AUTO page or any of the four CHECKOUT pages by pressing the fourth soft key from the top (just below MDI). Note **this soft key is not labeled** on those pages. This is intentional, so that the operator not involved in PAL development is not confused by a DIAGNOSTICS soft key label. The access from these screens is necessary if you wish to display the status of PAL parameters while a program is running.

Access

Access to the Diagnostic screens requires that Access code 13 be enabled for the logged on operator. Otherwise a FUNCTION NOT ALLOWED error message will be displayed when access is attempted.

Parameters Displayed

The main Diagnostic screen displays the B, H, N, R, and V PAL parameters. It also provides soft keys to access four additional screens. These are called A FLAGS, L FLAGS, G FLAGS, and C T M D. The first three display the A, L, and G parameters, respectively. The C T M D screen displays the C,T,M, and D parameters.

Parameter Categories

Screen	Type	Function
Main	B	Boolean Bits, Scratch
Main	H	Boolean Bits, CNC to PAL
Main	N	Boolean Bits, PAL to CNC
Main	R	Boolean Bits, Input/Output
Main	V	Arithmetic Variables Input/Output
A Flags	A	Arithmetic Variables, Scratch
L Flags	L	Arithmetic Variables, PAL to CNC
G Flags	G	Arithmetic Variables, CNC to PAL
C T M D	C	Counters
C T M D	T	Timers
C T M D	M	Message Enable Boolean Bits
C T M D	D	Arithmetic Variables, Double Precision

All of the 'Boolean Bits" (B, H, N, R, and M) are displayed in groups of 8, in increasing order, with the left most bit corresponding to the label. For example, the field labeled B009 displays bits 9 through 16 with bit 9 on the left. The bits are displayed as '-' if they are false, or as 'T' if they are true.

The single precision arithmetic variables (A, L, G, C, and T) are displayed as unsigned decimal numbers of up to 5 digits (0 to 65535).

The double precision arithmetic variables (D) are displayed in decimal form with up to 10 digits.

Refer to the PAL manual and your PAL ladder documentation for the specific meaning of the various parameters.

Pages

The following five Figures show the layout of the five Diagnostic pages.

Figure 12.14 - Main Diag. Page

```
INPUT / OUTPUT  DISPLAY F1-003 01JAN87                        A FLAGS
< operator message >    DEFAULT PP VER
< error message    >    LDDR 215-102-90 C4                    L FLAGS
B001-------- H001-TTTTTT- N001-------- R001--------
B009-------- H009-------- N009-------- R009--------           G FLAGS
B017-------- H017-------- N017-------- R017--------
B025-------- H025-T---T-- N025-----T-- R025--------          ·C T M D
B033-------- H033-----T-- N033----T--- R033--------
B041-------- H041-------- N041---TTTTT R041--------
B049-------- H049-------- N049T---T--- R049--------
B057-------- H057-------- N057T----TT- R057-------- V001        0
B065-------- H065-------- N065-------- R065-T------ V002        0
B073-------- H073-------- N073-------- R073---T---- V003        0
             H081-------- N081-------- R081-------- V004        0
             H089-------- N089-------- R089-------- V005        0
             H097-------- N097-------- R097-------- V006        0
                                       R105-------- V007        0
                                       R113-------- V008        0
```

The firmware revision number is shown in the top center of the page.

Immediately below the firmware revision is the message that identifies the AMP version. This message is defined in control parameters.

Below the AMP version message is the identity field for the PAL program that is implemented in the control.

Figure 12.15 - A Flags Page

```
 A ARITHMETIC PC LOCAL VARIABLES   < operator message >
                                   < error message    >
A001=      0    A011=       0    A021=       0    A031=       0
A002=    100    A012=       0    A022=       0    A032=       0
A003=      0    A013=       0    A023=       0    A033=       0
A004=      0    A014=       0    A024=       0    A034=       0
A005=      0    A015=       0    A025=       0    A035=       0
A006=      0    A016=       0    A026=       0    A036=       0
A007=      0    A017=       0    A027=       0    A037=       0
A008=      0    A018=       0    A028=       0    A038=       0
A009=      0    A019=       0    A029=       0    A039=       0
A010=      0    A020=       0    A030=       0    A040=       0
```

Figure 12.16 - L Flags Page

L ARITHMETIC PC TO CNC VARIABLES < operator message >
< error message >

L001=	0	L011=	0	L021=	0	L031=	0
L002=	0	L012=	0	L022=	0	L032=	0
L003=	100	L013=	0	L023=	0	L033=	0
L004=	0	L014=	0	L024=	0	L034=	0
L005=	24	L015=	0	L025=	0	L035=	0
L006=	0	L016=	0	L026=	0	L036=	0
L007=	0	L017=	0	L027=	0	L037=	0
L008=	0	L018=	0	L028=	0	L038=	0
L009=	0	L019=	0	L029=	0	L039=	0
L010=	0	L020=	0	L030=	0	L040=	0

Figure 12.17 G Flags Page

G ARITHMETIC CNC TO PC VARIABLES < operator message >
< error message >

G001=	0	G011=	0	G021=	0	G031=	0
G002=	0	G012=	0	G022=	0	G032=	0
G003=	0	G013=	0	G023=	0	G033=	0
G004=	0	G014=	0	G024=	0	G034=	0
G005=	0	G015=	0	G025=	0	G035=	0
G006=	0	G016=	0	G026=	0	G036=	0
G007=	0	G017=	0	G027=	0	G037=	0
G008=	12	G018=	0	G028=	0	G038=	0
G009=	0	G019=	0	G029=	0	G039=	0
G010=	204	G020=	0	G030=	0	G040=	0

Figure 12.18 - C T M D Page

```
| C T M D VARIABLES  < operator message >< error message    > |
|  COUNTER         TIMER          MESG BOOLEANS    DOUBLE WORD ARTH |
| C001=       0   T001=       0   M001=---------   D001=          0 |
| C002=       0   T002=       0   M009=---------   D002=          0 |
| C003=       0   T003=       0   M017=---------   D003=          0 |
| C004=       0   T004=       0   M025=---------   D004=          0 |
| C005=       0   T005=       0   M033=---------   D005=          0 |
| C006=       0   T006=       0                    D006=          0 |
| C007=       0   T007=       0                    D007=          0 |
| C008=       0   T008=       0                    D008=          0 |
| C009=       0   T009=       0                    D009=          0 |
| C010=       0   T010=       0                    D010=          0 |
| C011=       0   T011=       0                                     |
| C012=       0   T012=       0                                     |
| C013=       0   T013=       0                                     |
| C014=       0   T014=       0                                     |
| C015=       0   T015=       0                                     |
| C016=       0   T016=       0                                     |
```

12.7 PAL

Press the [PAL] softkey on the Support page to bring the PAL page to the screen. See figure 12.19.

Figure 12.19 - PAL Page

```
| PAL SUPPORT      ENABLE PROM PAL | *
| < operator messages>             | -
| LDDR 215-102-90 C4               | -
|                                  | -
|                                  | -
|                                  | -
|                                  | -
|                                  | -
| <  error messages  >             | -
```

PAL is normally executed from EPROM memory. During development of a PAL it is sometimes desirable to download a copy of the PAL code from the PAL programming device into the control's user memory for debugging purposes. Once the PAL is proven, it is burned into EPROMs and installed in the control.

PAL Ladders can be loaded into RAM only while on this page. After they are loaded, the control will execute from the PAL in RAM and will display a message "EXECUTING RAM PAL". If the first soft key, [ENABLE PROM PAL], is pressed, the control will return control to the PAL stored in EPROM.

Exiting from this page while a PAL load is in process will abort the load. The PAL loading is through I/O Port P3.

The complete procedure for loading and editing PAL is in the PAL Users Manual, publication number 8400 4.2. See Appendix B

12.8 Chapter Summary

This chapter discussed the Support page and the functions that it gives the installer and the operator. You should now know how to establish access control and AMP for the control.

This chapter completes our discussion of the operation of the control. The remaining chapters of this manual cover the programming language that the control uses. The next chapter gives you an introduction to this programming language and its format.

Chapter 13 -- Introduction to Programming

13.0 Chapter Overview

This chapter gives you a general overview of the programming language used by the control.

We give the structure and requirements of the programming language in this chapter. How to use specific programming features is covered in detail in later chapters.

13.1 Data Blocks

The basic building block in a program is a data block. A data block is made up of:

- characters -- A character is a number, a letter, or a symbol that means something specific to the control. For example, 1, G, %, # are characters the control recognizes as meaningful information.
- addresses -- An address is a letter that defines the instruction for the control. Examples of addresses are: G, X, Z, F.
- words -- A word consists of an address followed by a numeric value. Examples of words are: G1, X10.5, F50., M2. Each word requires a specific format for its numeric part. We give you these formats later in this chapter.

A data block is made up of a series of words. The words identify the operations the control must carry out. A program is made up of a series of blocks. For example:

Data Block	Comment
G91 #	-- sets incremental positioning
G1 X-1. Y2. F10. #	-- move 1 in. in the negative X direction and 2 in. in the positive Y direction at 10 inches per minute. This is called linear interpolation.
X1. Y-2. #	-- move 1 in. in positive X direction and 2 in. in the negative Y direction, which returns to the starting point of the previous block. Linear interpolation is still in effect since the G1 code is modal.
M2 #	-- identifies the end of a program

The example above is a series of 4 valid data blocks. The words in the data blocks mean specific things to the control. By telling the control the correct words in the correct order, you can have it perform precise, automatic operations.

End-of-Block, #

Each data block must have an end-of-block character in the last character position. The control displays the end-of-block character as #.

Use the [EOB] key to specify # in your data blocks.

Comment Block, ;

The semicolon character (;), or comment code, lets you define a comment in your program.

The comment code can appear anywhere in a data block. The characters that come after it become the comment, and are ignored by the control.

If a comment block is the first block of a program, it denotes the program name. The program name appears on many of the pages of the control when a program is active. The program name is limited to 12 characters in length. See the following example:

Data Block	Comment
;CW CIRCLE #	-- comment block that appears as the program name
G91 #	-- sets incremental mode
G2 XYIJ-.5 ;CIRCLE #	-- executes a 0.5 in. radius circle; the comment CIRCLE is ignored by the control but is stored as part of this data block.
M2 #	-- end of program

Block Delete, /

A block delete character (/) is used to allow the control to ignore any characters that appear after it in a data block. It should be entered only as the first character of the block. The operator determines whether the control ignores that block. The operator must select [BLOCK DELETE] on the Job Setup page to make the control ignore the block.

For example:

Data Block	Comment
G91 #	-- sets incremental mode
G1 X1. Y1. F10. #	-- move 1 in. X positive and 1 in. Y positive at 10 ipm.
/G X-1. Y-1. #	-- rapid move of 1 in. X negative and 1 in. Y negative. If [BLOCK DELETE] is active, the control ignores this data block
M2 #	-- end of program

Checking for Errors

If the control detects an error in a block during execution of a program, it will halt with that block active. Several pages of the control including the Status and Graphics pages will show the active block. Examine the data block, and correct it using Program Edit.

You can check programs for errors using Quick Check. All programs should be tested thoroughly using the Check Out modes before you run them in Auto Operate.

Illegal Characters, Addresses and Words

Characters, addresses and words that are not recognized by the control cause program execution to stop. The control displays the message "_____ CODE NOT KNOWN" on the CRT. (The "_____" displays the illegal coding.)

Illegal Data Blocks

A data block cannot contain more than one word with the same address. For example, if you program two G words in the same block, the control displays "MORE THAN ONE G CODE" when the program is executed. Execution will stop at that point.

13.2 Programs

A series of data blocks arranged in the right order and stored in the control, or externally, make up a program. The examples shown in section 13.1 are short programs.

Programs that are stored in the control are listed on the Directory pages. The control can hold up to 65 programs and subprograms. (Subprograms are those that are called for execution by other programs.) Individual programs are limited to a little over 64,000 characters. The total memory space for all programs varies with system configuration.

Each program and subprogram must have a P number designation. A P number identifies programs uniquely. You specify the P number of a program when you create it using Program Edit, or load it using Load/Save. The control allows P numbers from 1 to 250.

Each program and subprogram must end with a data block that contains an M2 or M30 code. If the control detects that the M2 or M30 is missing, the error message "PROGRAM _____ NO M2" occurs when the program is run.

13.3 Numeric Values

Most words in the data blocks of your programs require some sort of numeric value. This section describes the important considerations for programming numeric values.

Decimal Point Numeric Format

In this manual, we represent the decimal point format of a word by giving the number of digits allowed to the left of the decimal point, the decimal point, and then the number digits allowed to the right of the decimal point. For example, a decimal point format such as:

```
X 3.4
```

means that the X address allows 3 digits to the left of the decimal point and 4 digits to the right of the decimal point.

Important: Decimal point formats may vary from control to control. The installer of your system has set values in AMP (Adjustable Machine Parameters) that affect decimal point format, programming resolution, and positioning resolution for some words. Check with your installer for the decimal point format of your control.

Inch/Metric Numeric Formats

The control can operate with either inch or metric units. The number of numeric digits associated with each word remains the same for inch and metric operations.

However, the position of a decimal point for some words may change when the operating mode is changed between inch and metric units.

For example:

inch	metric	
X125.0125 in.	X4256.975 mm	
3 . 4	4 . 3	= 7 digits for both units

Number of Allowed Digits

The number of digits allowed for an address may change if the meaning of the word changes. The particular meaning of a word depends on the G word data block in which it is used, or to which it applies.

For example, consider the I word:

In a G2 circle block	In a G25 step-and-repeat block
I132.2506 in.	I115 points
3 . 4 = 7 digits	3 digits

Important: The control editor will let you enter any number of digits, but to avoid errors during program execution you must use no more than the number of allowed non-zero digits.

Implied Decimal Point Programming

For words that allow the decimal point, you can program them without the decimal point, and let the control interpret the value. The control determines the position of the implied decimal point according to the following rules:

- leading zeroes are not required

  Leading zeroes are not needed in any numeric value. The control regards G01 and G1 as the same value; even G00 and G are the same.

- trailing zeroes are required

  Trailing zeroes are needed when you don't program the decimal point, and the address normally uses one. Trailing zeroes are not needed when you do use the decimal point. For example, the following words are the same:

  X1.209 = X12090 -- assuming 3.4 format

- The number of digits to the right of the implied decimal point are determined according to the "current format" of the word, assuming the right most digit is the least significant.

  "Current format" is the decimal point format for the word at that point in the program. The "right-most" digit is the digit that appears farthest to the right in the word after you enter the number.

For example,

X1 = X.0001 ---assuming 3.4 format

13.4 Word Formats and Functions

The following table shows, in alphabetical order, the addresses for words that are recognized by the control, their formats, and their general meanings.

Important: Later sections and chapters discuss these words in more detail and their particular meanings that are associated with certain G codes.

Table 13.A
Word Formats and Descriptions

Address	Format Inch	Format Metric	Meaning
A	4.3	4.3	Angle of next move with vertex at the present location. Measured from the positive X axis in absolute (G90), or from the previous move in incremental (G91).
B	3.3	3.3	Angle increment between points in the bolt circle autoroutine (G24)
C	3.3	3.3	Angle of next move with vertex at a center point. Measured from the positive X axis in absolute (G90), or from the previous move in incremental (G91). Rotary axis feedrate in DPM(3.1), RPM(4), or DPS(3.1) when used in a G94 block.
D	3	3	Divide a move into a number of segments. Maximum is 255. Number of rough cuts in X and Y for pocket (G26) and post (G27) milling. Dwells in autocycles Perform the profile at the first point in the path in the cavity autocycle (G75).
E	4	4	Ending sequence number in a program branch instruction.

Table 13.A
Word Formats and Descriptions *(continued)*

Address	Format Inch	Format Metric	Meaning
F	4.1	5	Feedrate definition in motion blocks step-and-repeat (G25), pocket milling (G28), post milling (G27) Infeed rate from R-plane to depth in (G81-G85) autocycles.
	3.2	3.2	Dwell time in seconds in dwell (G04) blocks.
G	2	2	Preparatory function that defines mode of operation for the control.
H	4	4	Jump to starting sequence number in a subroutine branch instruction.
	4.1	5	Finish feedrate for pocket (G26) and post (G27) milling.
I	3.4	3.4	X distance to circle center. Measured from Program Zero in absolute (G90), or measured from present location in incremental (G91). X axis lead per revolution in helical interpolation (G22, G23) X finish cut for pocket (G26) and post (G27) milling. Peck drill (G83) initial peck increment.
	3	3	Number of points along X axis in step-and-repeat (G25).

Table 13.A
***Word Formats and Descriptions** (continued)*

Address	Format Inch	Format Metric	Meaning
J	3.4	3.4	Y distance to circle center. Measured from Program Zero in absolute (G90), or measured from present location in incremental (G91).
			Y axis lead per revolution in helical interpolation (G22, G23).
			Y finish cut for pocket (G26) and post (G27) milling.
			Peck drill (G83) peck difference.
	3	3	Number of points along Y axis in step-and-repeat (G25).
K	3.4	3.4	Z distance to circle center. Measured from Program Zero in absolute (G90), or measured from present location in incremental (G91).
			Z axis lead per revolution in helical interpolation (G22, G23)
			Z finish cut for pocket (G26) and post (G27) milling.
			Peck drill (G83) minimum peck increment.
L	3	3	Block execution counter, allows you to repeat a block L times (0 to 255).
			Number of rough cuts in Z for pocket (G26) and post (G27) milling.

Table 13.A
***Word Formats and Descriptions** (continued)*

Address	Format Inch	Format Metric	Meaning
M	2	2	Miscellaneous function and block conditional.
N	4	4	Sequence number
O	2	2	Offset select
P	3	3	Subprogram call in a branch instruction. Retract code for autocycles (G81-G85)
Q	3.4	3.4	Radius for blended move (QUICK PATH). Also radius for G05 arc center programming.
	na	na	Grid toggle in graphics control (G66)
R	3.4	4.3	Radius in motion blocks. Measured from the center in absolute (G90), or measured from the present location in incremental (G91). Rapid plane definition in autocycles.
	na	na	Rapid display toggle in graphics control (G66).
S	4	4	Spindle speed definition.
T	2	2	Tool select.

Table 13.A
***Word Formats and Descriptions** (continued)*

Address	Format Inch	Format Metric	Meaning
U	3.4	4.3	Secondary axis parallel to X
	3.3	3.3	Secondary axis rotary about X
	2.4	2.4	Scaling (G72), Scale Factor for U axis.
	n/a	n/a	Axis Mirror G30, G31
V	3.4	4.3	Secondary axis parallel to Y
	3.3	3.3	Secondary axis rotary about Y
	4.1	5	Plunge feedrate definition for pocket (G26) or post (G27) milling.
			Counterbore retract feedrate for boring autocycle (G85).
	2.4	2.4	Scale factor for V axis.
W	3.4	4.3	Secondary axis parallel to Z
			Rough cut in X and Y for pocket (G26) and post (27) milling.
			Chip break retract for peck drill autocycle (G83).
	3.3	3.3	Secondary axis rotary about Z
	3	3	Number of points for bolt circle autoroutine (G24).
	na	na	Erase inhibit for graphics control (G66).
	2.4	2.4	Scale factor for W axis.

Table 13.A
***Word Formats and Descriptions** (continued)*

Address	Format Inch	Format Metric	Meaning
X	3.4	4.3	Move in the main X axis. Measured from Program Zero in absolute (G90), or measured from the present location in incremental (G91).
			Start position in X for bolt circle autoroutine (G24).
			Increment between points in X for step-and-repeat (G25).
			Position of opposite corner in X for pocket (G26) and post (G27) milling.
	2.4	2.4	Scale factor for X axis, the value entered affects the I center word by the same amount.
Y	3.4	4.3	Move in the main Y axis. Measured from Program Zero in absolute (G90), or measured from the present location in incremental (G91).
			Start position in Y for bolt circle autoroutine (G24).
			Increment between points in Y step-and-repeat (G25).
			Position of opposite corner in Y for pocket (G26) and post (G27).
	2.4	2.4	Scale factor for Y axis, the value entered affects the J center word by the same amount.
	na	na	Axis Mirror G30, G31

Table 13.A
Word Formats and Descriptions (continued)

Address	Format Inch	Format Metric	Meaning
Z	3.4	4.3	Move in the main Z axis. Measured from Program Zero in absolute (G90), or measured from the present location in incremental (G91).
	2.4	2.4	Scale factor for Z axis, the value entered affects the K center word by the same amount.

13.5 Word Descriptions

This section describes general features of the words used in your programs. Later chapters in this manual describe, in detail, how to use these words.

13.5.1 G Word Data Blocks

G words are preparatory functions. They define the modes of program operation for the control. The G word allows a 2 digit numeric value.

G words are either modal or non-modal:

- A modal G word establishes a mode of operation that remains in effect for data blocks that come after it. The mode remains until a G word that cancels it is executed.
- A non-modal G word establishes a mode that remains active only for the data block in which it is programmed.

However, when the program is entered or displayed in prompt edit, the G word will appear as the first word in the block, after an initial character and sequence number, if present.

If more than one G word is present in a block, the error message, "MORE THAN ONE G CODE" is displayed on the CRT when the block is encountered. Execution is halted, and you should correct the problem using Program Edit.

G Word Groups

For ease in programming, the G codes are divided into groups according to their function. These groups are given on the G CODES prompt pages of the control.

Figure 13.1 - G Words on Page 1 of G Code Prompt Page

```
AUTOCYCLES          AUTOROUTINES        POSITIONING/DWELL
G75 CAVITY          G22 HELIX CW        G00 POINT-POINT
G79 PROGRAMMABLE    G23 HELIX CCW       G01 LINEAR
G80 CANCEL          G24 BOLT CIRCLE     G02 ARC CW
G81 DRILL           G25 STEP/REPEAT     G03 ARC CCW
G82 COUNTER BORE    G26 POCKET MILL     G04 DWELL
G83 PECKDRILL       G27 POST MILL       G05 ARC TANGENT
G84 TAP             G29 EXEC AUTOCYCLE
G85 BORE            G39 PARA SUBPROG
G89 RESTORE                             VARIABLES
                                        G39 LOCAL
                                        G59 PAL
                                        G79 GLOBAL



                                                  PAGE 1 OF 2
                                        > G   0     NEXT PAGE
```

Figure 13.2 - G Words on Page 2 of G Code Prompt Page

```
COORDINATE CONTROL  AXES MODES          CNC CONTROL
G70 INCH            G17 XY PLANE        G60 ZONE INHIBIT
G71 METRIC          G18 ZX PLANE        G62 OVERRIDE INHIBIT
G72 SCALING         G19 YZ PLANE        G66 GRAPHICS CONTROL
G74 ROTATION        G30 CANCEL MIRROR
G90 ABSOLUTE        G31 AXES MIRROR
G91 INCREMENTAL     G40 CANCEL CTR COMP
G92 PRESET PROG 0   G41 CUTTER LEFT
G94 PER TIME FEED   G42 CUTTER RIGHT
G99 CANCEL PRESETS  G45 FIXTURE OFFSET




                                                  PAGE 2 OF 2
                                        > G   0     NEXT PAGE
```

Page 1 G Word Groups

Autocycles --
G words in the autocycles group program operations that occur automatically after rapid moves. Once they are programmed, they remain in effect until they are canceled.

Autoroutines --
G words in the autoroutines group program special positioning operations, such as pocket milling, or a bolt circle.

Positioning/Dwell --
The G words in the positioning group are the most basic and most widely used in your program. They define how the control performs positioning, and when pauses (dwells) occur in the program.

Page 2 G Word Groups

Coordinate Control --
The coordinate control group sets conditions that affect the coordinate system of the machine, and so, the type of programming you use for positioning. G words from this group will generally be among the first in your programs.

Axes Modes --
The axes modes group sets conditions that affect how and which axes are used in your positioning data blocks. They are also generally among the first G words used in your programs.

CNC Control --
These G words affect specific operations of the control, such as erasing the Graphics display from the program, or preventing the operator from adjusting certain controls on the front panel. They may be used any where in your program.

The operation of each of these group is described in the chapters that follow.

Legal Addresses in G Word Data Blocks

Certain addresses may not be allowed in some G word data blocks. The following table gives the allowed addresses in each G word data block. Be sure to read about these data blocks in the chapters that follow.

Table 13.B
Legal Addresses in G Word Blocks

Point-Point	- G00	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Linear	- G01	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Arc CW	- G02	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Arc CCW	- G03	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Dwell	- G04	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Arc Tangent	- G05	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
XY Plane	- G17	*N*
XZ Plane	- G18	*N*
YZ Plane	- G19	*N*
Helix CW	- G22	*I J K N X Y Z*
Helix CCW	- G23	*I J K N X Y Z*
Bolt Circle	- G24	*A B C E H I J L N P R W X Y*
Step/Repeat	- G25	*E F H I J N P X Y*
Pocket Mill	- G26	*D F H I J K L N O Q R U V W X Y Z*
Post Mill	- G27	*D F H I J K L N O Q R U V W X Y Z*
Exec Autocycle	- G29	*L N*
Cancel Mirror	- G30	*N U V W X Y Z*
Axes Mirror	- G31	*N U V W X Y Z*
Local Variable	- G39	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Ctr Comp Cancel	- G40	*N X Y Z*
Cutter Left	- G41	*N X Y Z*
Cutter Right	- G42	*N X Y Z*
Fixture Offset	- G45	*E H N O P*
PAL Variable	- G59	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Zone Inhibit	- G60	*A B C I J K N R U V W X Y Z*
Override Inhibit	- G62	*F N Q S*
Graphics Control	- G66	*N Q R W*
Inch	- G70	*N*
Metric	- G71	*N*
Scaling	- G72	*E H L N P U V W X Y Z*
P-P Interpolate	- G73	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Rotation	- G74	*A C E H I J L N P R X Y Z*
Cavities	- G75	*D E H N P X Z*
Global Variable	- G79	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*
Cancel	- G80	*N*
Drill	- G81	*D F N P R V X Y Z*
Counter Bore	- G82	*D F N P R V X Y Z*
Peckdrill	- G83	*D F I J K N P R U W X Y Z*
Tap	- G84	*D F N P R X Y Z*
Bore	- G85	*D F N P R V X Y Z*
Restore	- G89	*N*
Absolute	- G90	*A B C I J K N R U V W X Y Z*
Incremental	- G91	*A B C I J K N R U V W X Y Z*
Preset Prog 0	- G92	*N U V W X Y Z*
Per Time Feed	- G94	*C F N*
RTH/PRST Cancel	- G99	*A B C D E F H I J K L M N O P Q R S T U V W X Y Z*

13.5.2 Words in Positioning Blocks

The following words may appear in a positioning block, and they program the movement of axes, either directly or in conjunction with each other:

- X Y Z U V W I J K A B C R

Each of these letters may be in absolute or incremental. Basically, absolute dimensions are distances from program zero or angles from the positive X axis, incremental dimensions are distances from the current location or angles from a line from a center point through the current location.

The mode of each letter can be set to absolute with a G90 or set to incremental with a G91 (See Sections 14.6 and 14.7).

Alternatively, the mode of any letter that is currently absolute can be forced to incremental for the current block by using a lower case letter.

See chapter 16 for specifics on how to use these words in each of the positioning blocks.

Linear Axes Words X, Y, Z, U, V, W

In general, the words X, Y, Z refer to machine axes that move back and forth along linear paths. These will be the most common axes found in most applications, and they are called linear axes. Linear axis moves are programmed in units of inches or millimeters.

The installer may have provided you with up to 3 additional machine axes. These axes are represented by the U, V, and W words, and they may be linear or rotary (in any combination) as determined by the installer.

When these axes are linear, they are generally set up to be parallel to the X, Y, and Z linear axes. In this manual, we assume:

- the U axis is parallel to X
- the V axis is parallel to Y
- the W axis is parallel to Z.

These axes, if present, cannot be used to produce paths along an arc. They are not allowed in G2 or G3 blocks.

In the absolute mode (set by G90 and referenced with upper case letters), the moves of linear axes are programmed from Program Zero (the established zero position on each axis). Linear moves in absolute are moves to specified coordinates in the established coordinate system.

In the incremental mode (set by G91 or reference by lower case letters), the moves of linear axes are programmed from the current position of the axes. They give the direction and distance of moves along each axis.

Rotary Axes Words U, V, W

A rotary axis is one that rotates in a circular path, like the table of a record player. Rotary axes are usually set up for moving about the X, Y or Z linear axes.

This manual assumes that when the U, V, or W axes are rotary,

- the U axis is rotary about X
- the V axis is rotary about Y
- the W axis is rotary about Z

If you program a positive direction for a rotary axis, it is counterclockwise. A negative direction is clockwise. This assumes you are looking toward the negative direction of the linear axis that is associated with the rotary axis.

Rotary moves are programmed in units of degrees (format 3.3). Rotary axes "rollover" at 360 degrees, that is, the 0 position and the 360 position are the same. So the normal programming range of rotary axes is 0 to 359.999 degrees.

When rotary moves are made in the absolute mode (G90), they define a degree position from 0 degrees. Rotary moves in absolute go in the direction of the sign of the angle programmed up to 360^{0}. A move of -90. is a move to 90^{0} in the minus direction and a move of +645. is a move to 285^{0} in the plus direction. If no sign (+or-) precedes the number of degrees, plus is assumed.

Important: To go to 0^{o} in the minus direction, you must program -360.000 since -0.000 will be interpreted the same as +0.000.

When rotary moves are made in the incremental mode (G91) they define the direction and amount of movement in degrees from the current position. In incremental mode, the rotary axis can move more than 360 degrees in any direction.

Feedrates for rotary axis moves are defined with a G94 data block. The C word in this data block tells the control the rate for the rotary axis in revolutions per minute, degrees per second, or degrees per minute as determined on the Control Parameters page.

If multiple rotary axes are in the block, the feedrate applies to the axis traveling the furthest.

Rotary and Linear Axis Moves in the Same Data Block

The CNC calculates the length of time to complete the programmed move for each axis in a program block using the programmed feedrate (not the maximum speed of the axis). The axis performing the move which takes the longest time to complete is the limiting axis. The feedrate and feedrate type for the limiting axis is displayed on the Status page in the upper right hand corner. The limiting axis may be IPM (linear), or RPM, DPM, DPS (rotary). See figure 10-3.

The CNC commands the limiting axis to move at a rate as close to 100% of the programmed feedrate as possible without pushing any other axis over its maximum speed or 150% of its programmed speed, whichever is less.

When the limiting axis is linear, the rotary axis will not move any faster than its global rapid equivalent. For example, if the global rapid is 200 IPM, at 10,000 counts per inch, you would have 2,000,000 counts per minute. This equals about 33,300 counts per second. Rotary axis resolution on the 8400MP is always 1000 counts per degree. So with a global rapid rate of 200 IPM, the rotary maximum speed in linear/rotary combined moves is limited to about 33.3 degrees per second. This happens even if rotary max is greater than 33.3 DPS.

Example: 1

```
G94 F200 C1.69 #
G00 X0.0 W0.0  #
G01 X1.0 W45.0 #
```

Time at programmed rate:
(X axis .3 sec.) (W axis 26.946 sec.)

'W' limits 'X'

$$\frac{1\ \text{inch}}{26.946\ \text{sec}} \times 60\ \text{sec} = 2.22\ \text{IPM}$$

X runs at 2.22 IPM
W runs at 1.67 DPS

Example: 2

```
G94 F200 C1.67 #
G00 X0.0 W40.0 #
G01 X20.0 W45.0 #
```

Time at programmed rate:
(X axis 6.0 sec.) (W axis 3.0 sec.)

'X' limits 'W'

$$\frac{5\ \text{degrees}}{6\ \text{sec}} = .83333\ \text{DPS}$$

X runs at 200.0 IPM
W runs at .833 DPS

Important: No axis may exceed the max rate set in AMP, even if the programmed feedrate is higher than the max rate.

Center Position Words – I, J, K

The words I, J, and K in positioning blocks define location of a center. This center position may be the center of an arc, or the center of polar positioning. See chapter 16 for more details on how these words are used.

Polar Positioning Words – A, B, C, R

The words A, B, C, and R define angles and radial lengths in positioning blocks that use polar programming. (The B word is used only in the G24 - Bolt Circle Autoroutine. The R word also defines a rapid plane in the Autocycle G word group.) See chapter 15 for more details on how these are used in polar

13.5.3 Feedrates – F Word

An F word in a positioning data block programs a linear axis feedrate. The F word may have units of:

- inches per minute (ipm, format F4.1)
- millimeters per minute (mmpm, format F5)

depending on whether the inch (G70) or metric (G71) programming mode is in effect.

The feedrate that you program with the F word is modal. It applies to every positioning block that comes after the block in which it appears. You can program a subsequent block with another F word to set a new feedrate.

The feedrate for linear moves is the rate along the orthogonal distance of all of the axes programmed. If a single axis move is programmed, then the feedrate will apply directly to that axis. If multiple linear axes are programmed to move simultaneously, then the time to execute the move is the distance corresponding to the square root of the sum of the squares of all of the programmed axes distances divided by the feedrate.

You should know how to calculate the feedrate for your specific operations. Program the F word assuming the Feedrate Override Switch is set to 100%. You can vary the programmed feedrate, and adjust for specific cutting conditions, by adjusting this switch when you run the program.

The system installer sets a feedrate that the control will use if none is programmed. You can see this feedrate on the Status page of the control before executing any part program.

13.5.4 Block Divide – D Word

The D word (format 3) in a positioning data block divides that block into a number of equal length segments. The D word can have a numeric value from 0 to 255.

Important: The D word of 0 cancels the fixed cycle for the block it is in.

The D word can be used to divide both linear and circular moves, but not rotary. If the move is circular, the control moves in straight lines to the divide points. It does not follow the contour of the circle.

The points that separate the segments of the move divided with the D word may be positions where you want a specific operation to take place; perhaps an autocycle that drills a hole, or cuts the profile of a cavity. Rather than programming each of these points separately, you can let the D word locate these points.

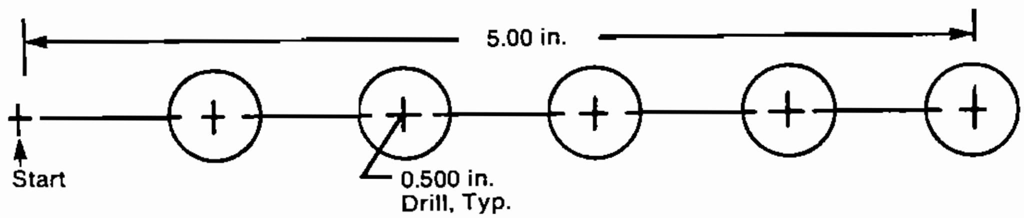
For example, if you program

Data Block	Comment
G70#	-- sets inch mode programming
G91 XZ#	-- sets X and Z to incremental
G81 Z-1. F10.#	-- a drill autocycle that drills 1 inch below the current position of the Z axis, infeed rate is 10 ipm.
G X5. D5#	-- a rapid move that is 5 inches in the positive X direction, divided into 5 equal segments. The drill autocycle will be performed at the end of each segment.
G80 #	-- cancels the drill autocycle
G X-5.#	-- rapid return to the start of the move
M2#	-- end of program

The program given above uses some programming features that we haven't discussed yet. See the chapters that follow for descriptions of these features.

Figure 13.3 shows the result of this example program.

Figure 13.3 - Example Program for Block Divide



If you wanted the autocycle performed at the start point in this example, you could use one of 3 techniques:

- program an X0 in the G81 autocycle block
- program G0 X0 # (remember incremental) immediately after the G81 block
- program a G29 # immediately after the G81 block.

13.5.5 M Word – Miscellaneous Functions

M words (format M2) define miscellaneous functions. They are used to define actions taken by PAL (Programmable Applications Logic). PAL is a separate foreground program that the control runs, which controls such things as tool changes, coolant on or off, and the spindle.

The installer of your control has programmed PAL to take specific action in response to an M word in a data block. Because many of these M functions are unique to your particular system, it is impossible to define the specific operation of M functions in this manual.

A few M words are predefined for the control, and do specific things internally. These are outlined below. The specific action these words produce in PAL, however, is determined by the installer.

M00 – Program Stop

The M00 word causes a program stop. When the data block containing M00 is performed, execution is halted. The messages, "HALTED" and "AWAITING START CYCLE" are displayed on the CRT.

Press the green [START CYCLE] button, or the yellow [BLK/BLK] button, to resume program execution with the next data block. Depending on the programming of PAL, you may have to perform certain functions before the control will let you continue operation.

The M00 code is ignored by the control during Quick Check.

M01 – Optional PROGRAM STOP

The M01 word produces the same effect as the M00 word providing that you select the OPTIONAL STOP function On the Job Set Up page.

If OPTIONAL STOP is not selected, M01 is ignored by the control. In either case, M01 is ignored by the control during Quick Check.

M02 – Program End

The M02 word is used to end every program and subprogram. You should program the M02 in a block by itself, and make it the last data block in your programs.

If the control detects the absence of the M02 in a program, execution halts, and the message "PROGRAM ______ NO M02" appears on the CRT.

Any other PAL related functions associated with M02 are the responsibility of the installer.

Both M02 and M30 terminate a program load, if they are immediately followed by the # character.

M06 – Program Stop, Manual Tool Change

The M06 word produces the same effect as M00, but it is used to initiate a manual tool change.

When M06 is programmed in a data block with axis movement, the M06 turns the spindle off before the move is executed. (This requires proper PAL programming -- the responsibility of the installer.) It is assumed that the move returns the axes to a position that is convenient for you to make a manual tool change. After you make the tool change, the next block in the program should return the axes to the machining position and start the spindle.

In the standard PAL, if the spindle was on then the [START] pushbutton will start the spindle.

M30 – Program End

The M30 word operates in the same way as the M02 word.

M-Function – Conditional Branching

Data blocks that include an M word with an H, E, or P word are conditional blocks. The control waits for PAL to complete the M function.

If a "conditional flag" is returned from PAL in a false (reset) state, branching is executed according to the H, E, and P words. If the conditional flag is true (set), branching is not performed and execution resumes with the next data block in the program.

Important: Conditional branching using M-Functions requires proper PAL programming. Check with your system installer for operation and programming details.

Other M Words - Standard PAL

The [M CODES] prompt page of the control lists the number and name of M functions programmed in PAL. If your control has the standard PAL program offered by Allen-Bradley, the [M CODES] page will look like figure 13.4.

Check with the installer of your system on the function and use of these M words.

Figure 13.4 - Standard PAL M Words on the [M CODES] Prompt Page

```
| SPINDLE CONTROL       PROGRAM CONTROL      TOOL CHANGER        | -
| M03 SPINDLE CW        M00 PROGRAM STOP     M06 PROG STOP MAN TC |
| M04 SPINDLE CCW       M01 OPT PROGRAM STOP M20 TOOL OUT   -ATC- | -
| M05 SPINDLE OFF       M02 PROGRAM END      M21 TURRET CW  -ATC- |
|                       M30 PROGRAM END/REW  M22 TURRET CCW -ATC- | -
| COOLANT CONTROL                            M23 TOOL IN    -ATC- |
| M07 COOLANT ON-MIST                        M27 TURRET HOME-ATC- | -
| M08 COOLANT ON-FLOOD  CLAMPS                                    |
| M09 COOLANT OFF       M10 CLAMPS ON        RPM CHANGER          | -
|                       M11 CLAMPS OFF       M24 RPM UP           |
| SPINDLE W/FLOOD COOL                       M25 RPM DOWN         | -
| M13 SPINDLE CW                             M28 LOW GEAR         |
| M14 SPINDLE CCW                            M29 HIGH GEAR        | -
|                                                                 |
|                                                                 | -
|                                                                 |
|                                                  PAGE 1 OF 1    | *
|                                          > M   0      NEXT PAGE |
```

13.5.6 Branching -- N, H, E, L, and P Words

This section describes the 5 words used in program branching. You can use program branching to alter and control the execution of your programs.

Program branching transfers execution from one part of a program to another part (a subroutine), or transfers It to another program (a subprogram). When the control finishes running the part you specify, it returns to the data block that specified the branch (except for an H word only unconditional branch).

N Word - Sequence Number

An N word (format N4) programs a sequence number. It labels the data block in which it appears. You do not need to program sequence numbers in every data block.

Sequence numbers are used to make programs more readable. More importantly, they are also used with H word and E word program jump instructions.

You can assign sequence numbers in any order, but we recommend that you use an ascending order throughout your programs. It is convenient to use numbers in intervals of 5 or 10. This lets you insert additional N word blocks later without having to reassign the sequence numbers of already programmed blocks or give up the sequential order.

P Word – Subprogram Branch

Every program or subprogram stored in the control must be identified with a P number in the range P1 to P250. You specify the P number of a program or subprogram using Program Edit.

There is no structural difference between a program and a subprogram. The only distinction is that a branch instruction calls a subprogram for execution from another program.

A P word (format P3) in a data block specifies a branch to subprogram stored in the control except for retract codes for autocycles (G81-G85). Therefore, a subprogram must exist and have a number corresponding to the number of the P word. The branch occurs after any move programmed in the data block.

If you program a P word that is not currently in storage, the error, "P*** NOT LOADED (***)" is displayed on the CRT, and execution stops. The number shown with P is the subprogram number you programmed. The number in parentheses is the number of the program that called the subprogram.

Subprogram execution begins with the first block of the subprogram (unless an H word is programmed with P; see the H word below, and the examples).

Subprogram execution ends with the M2 block of the subprogram (unless an E word is programmed with P, see the E word below, and the examples).

Subprogram branches may be nested up to 5 levels deep. For example:

```
Level 0    P1 calls
Level 1          P2 calls
Level 2                 P3 calls
Level 3                        P4 calls
Level 4                               P5 calls
Level 5                                      P6 calls
```

At the end of subprogram P6, execution eventually returns to the main program, P1. If you try to program beyond 5 levels, the error message "P NESTING TOO DEEP" appears on the CRT.

The control checks for errors in subprogram branching, such as an infinite loop of execution. When the control detects such an error, the message "P NESTING TOO DEEP" is also displayed on the CRT and program execution stops. Run all programs using Quick Check to test them before you run them in Auto Operate. This will let you detect such errors.

The following parameters, if changed in a subprogram, are restored to their original status in the calling program after a subprogram finishes executing:

- absolute (G90)/incremental (G91) mode
- circular plane (G17, G18, G19)
- mirror status (G30, G31)
- autocycle (G75 - G85)
- autoroutine (G22 - G29)
- feedrate (F word for axis moves)
- remainder of current block being executed
- dwell amount (F word for G04)
- interpolation mode (rapid or feed)

Important: You can renumber programs stored in the control using Program Edit. If you renumber a subprogram, the control does not automatically update P words in your programs. In this case, you should use Program Edit to correct your subprogram branch instructions.

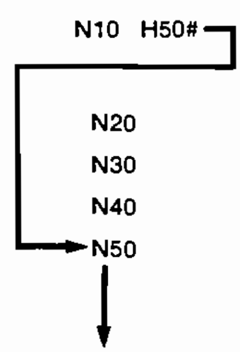
H Word – Jump/Subroutine Call

An H word (format H4) calls for a jump to a block that has a sequence number (N word) that is equal to the H word.

In most cases, this will be the first block in a subroutine. The control executes blocks beginning with the specified N word block, and continues until it reaches an M2 or the last block in the subroutine.

H Word Only Branch

Branch to block labeled with N50 and continue.

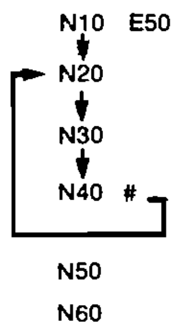


E Word – Ending Sequence Number

An E word (format E4) specifies an ending sequence number in a subroutine. When the E word is used, the last block executed in a subroutine is the one that comes just before the block that has an N word equal to the E word. If the E word has a value of zero, then the subroutine will be executed to the end of the program before returning to the block after the E word.

E Word Only Branch

Execute from current block to the block just before N50 and return to block following the current block.



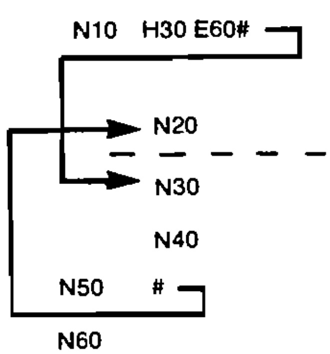
The control executes blocks from the beginning of the subroutine up to the block with the N word.

Branching Examples

The following examples further illustrate how to use branching in your programs.

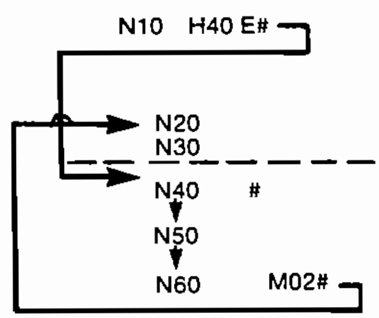
H Word and E Word Subroutine Call

Branch to block labeled with N30 and execute to the block just before N60, then return to the block just after the calling block.



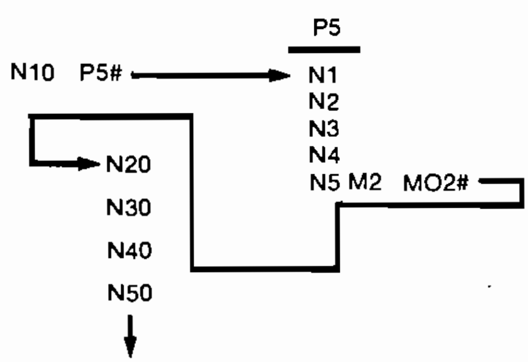
H Word and EO Subroutine Call

Branch to block labeled with N40 and execute to the end of the program, then return to the block just after the calling block.



P Word Subprogram Call

Branch to program in storage whose number is 5 and execute until the end of that program, then return to the block following the calling block.

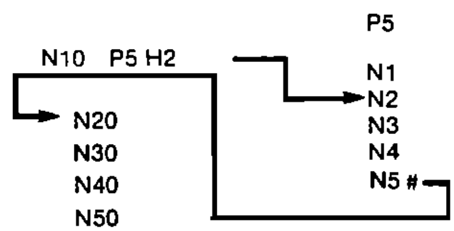


P word call with conditional M word

Upon execution of the M(GB) P73, The PAL program (specifically written to assign a value for GB) performs some operation and/or calculation and then assigns either a one or a zero to the variable GB. If it assigns a zero, then the subprogram will not be called and program execution will continue with the block labeled N20. If the variable GB was assigned a one, then the subprogram branch will be executed as in the previous example.

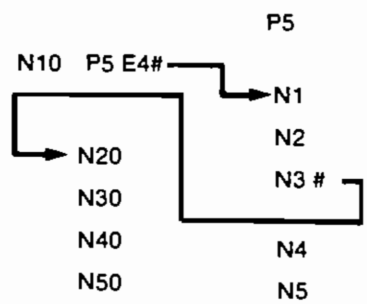
P Word and H Word Subroutine Call

Branch to subprogram 5 and execute from N2 to N5, then return to the block following the calling block.



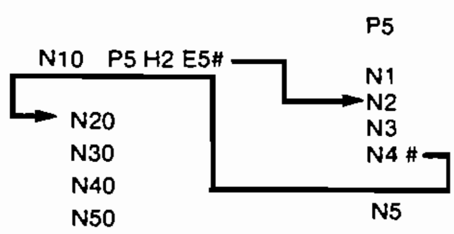
P Word and E Word Subroutine Call

Branch to subprogram 5 and execute from the beginning of the subprogram until the block just before N4, then return to the block just after the calling block.



P Word, H Word, and E Word Subroutine Call

Branch to subprogram 5 and execute from N2 to the block just before N5, then return to the block just after the calling block.



L Word – Block Repeat

An L word (format L3) in a data block causes that data block to execute L times. The L word can have a value from 0 to 255, and can be nested to 5 levels.

- To repeat an axis move:

```
G1 X-1. L4#  -- assuming incremental mode (G91),
this executes a 1 inch move in the negative X
direction a total of 4 times
```

- To repeat a subprogram:

```
P5 L4#  -- execute program 5 a total of 4 times
```

- To repeat a subroutine:

```
H100 E200 L4#  -- execute subroutine from N100 to
N200 a total of 4 times
```

If axis moves are included in the block that programs a subprogram or subroutine repeat, they are executed first, followed by the subprogram or subroutine.

Important: The following special rules for subroutine calls that use the L word:

Blocks with an E word and no H word:

- with no L in the block: always loop back
- with L0 in the block: jump to N word = E word
- with L1 in the block: continue with next block (the E word is not executed)

Blocks with an H word and no E word:

- with no L in the block: always branch
- with L0 in the block: not branch (defaults to L1)
- with L1 in the block: not branch (it is executed at once)

13.5.7 Spindle Speed – S Word

The S word defines the spindle rotation speed, typically in RPM, or as a percent of maximum spindle speed.

Controlling the machine's spindle is a function of PAL. Check with your system installer for details of operation and programming. This section assumes you are using the standard PAL available from Allen-Bradley.

The value of the S word is modal. Once a value is programmed, it remains in effect until it is changed.

When the spindle is turned on in a program (using an M3 - spindle forward, or M4 - spindle reverse), the current value of the S word sets the spindle's rotation speed.

The S word must be programmed or the spindle is kept off. If the spindle is starting from a rest point, or is changing in speed, the control waits for the programmed speed to be achieved before continuing with program execution.

You can modify the programmed spindle speed using the Spindle Override Switch on the front panel. This switch will normally vary the speed from 50% to 120% of programmed speed. You can see the actual spindle speed value on the Status page of the control.

13.5.8 Tool and Fixture Offsets Select -- O Word

The O word (format O2), when it is used in a block other than G45, selects a tool offset.

To select a tool offset, you program the number (1 to 48) of an offset that is entered on the Tool Offsets pages of the control. For example,

03#

This selects a length value and a diameter value entered in position number 3 on the Tool Offsets pages. The length value is used to offset Z axis moves from Program Zero. The diameter value is automatically halved to give a radius value that is used in Cutter Radius Compensation (G40-G42) programming. No motion takes place when the O word is executed.

Instead, the offset values are combined with the next programmed axis moves, and become active at that point. For example, to activate the length offset, you program a Z axis move after the block that contains the O word. For the diameter (radius) value to be fully employed, "moves" should be programmed for the axes that use the compensation.

The tool offset number that is currently active in a program is shown on the Status page of the control in the space next to "O."

To cancel tool offsets, program:

```
O0 #
```

The offset is removed with the next programmed moves of the relevant axes.

When the O word is used together with a G45 it selects a fixture offset number (1 to 12) from the Fixture Offsets page of the control. For example, a G45 O1 # calls for fixture offset value number 1.

Fixture offsets are offsets to programmed moves for the X, Y, Z, and W axes. They move the program zero reference point in a program by amounts that you specify on the Fixture Offsets page.

Again, no motion takes place when the O word for fixture offsets is executed. The offsets are combined with the next programmed moves on the affected axes, and become active at that point in the program.

To cancel fixture offsets, program a data block with:

```
G45 O0 #
```

The current fixture offset number that is active in a program is shown on the Status page of the control in the space next to "FX."

13.5.9 Tool Select – T Word

The T word (format T2) allows for tool selection from an automatic tool changer. The 2 digit number of the T word identifies the position of the tool in the tool changer. The upper limit on the T is determined by the PAL controlling the tool changer. The maximum T-word is T99. The standard PAL allows tool numbers from 1 to 24 to operate the Allen Bradley ATC.

The control requires proper PAL programming to let it control the tool changer. Check with your installer for details on programming and operating.

When the T word is programmed, the tool changer performs automatic operations that remove the current tool and load the desired tool onto the machine.

The length and diameter values associated with each tool are entered on the Tool Offset pages of the control. These values are programmed with an O word.

For example, to load tool number 10 from the tool changer and apply offsets number 10 from the Tool Offsets page of the control, you would program:

```
T10 010 #
```

Other operations may need to be programmed to support the use of the tool changer. This could mean moving to a clear location for the tool changer to work, for example. Know the operation requirements of your machine and program accordingly.

When you program the T word, its value appears on Status page of the control next to the "T" space.

13.6 Chapter Summary

This chapter defined the programming language of the control in general terms. You should now know about the structure of the programming language, what data blocks are, and about the words that can be used in your programs.

The chapters that follow describe each of the G word groups in detail. Examples illustrate how each G word works in your programs.

The next chapter describes the coordinate control G word group. These G words define characteristics of the coordinate system of the control.

Chapter 14 - Coordinate Control

14.0 Chapter Overview

This chapter covers the G word group called coordinate control. These G words will be among the first in your program because they define how the control treats the coordinate system of the machine.

After reading this chapter, you will know how to:

- set inch or metric units for positioning
- set absolute or incremental positioning
- set a Program Zero point for absolute positioning
- scale the coordinate system
- rotate the coordinate system
- cancel the Program Zero point

14.1 Setting Inch Mode -- G70

A G70 block at the beginning of a program tells the control to use inch values for axis moves, feedrates, clearance planes and offsets.

Important: The control does not convert metric dimensions to inch or vice versa. You cannot mix inch and metric operations in any one program. Make sure all dimensional entries correspond to the dimensional mode you select. This includes the contents of the offset tables if they are to be used.

The control assumes inch mode (G70) or metric mode (G71) at power-up, as determined in AMP by the system installer.

The G70 word must be programmed in a data block by itself, and should be programmed before any moves in the part program. G70 is modal and cancels G71.

A program command of G70 will will test the selected Inch/Metric mode. If it is not properly selected, the program will halt with the following error message displayed: INCH/METRIC CHANGE. The machine will now be in Inch Mode. No rehoming is required.

14.2 Setting Metric (mm) Mode -- G71

A G71 block at the beginning of your program selects metric units for all axis moves, feedrates, clearance planes and offsets.

Important: The control does not convert metric dimensions to inch or vice versa. You cannot mix inch and metric operations in any one program. Make sure all dimensional entries correspond to the dimensional mode you select. This includes the contents of the offset tables if they are to be used.

The control assumes inch mode (G70) or metric mode (G71) at power-up, as determined in AMP by the system installer.

The G71 word must be programmed in a data block by itself, and should be programmed before any moves in the part program. G71 is modal and cancels G70.

A program command of G71 will test the selected Inch/Metric mode. If it is not properly selected, the program will halt with an error message displayed stating: INCH/METRIC CHANGE. The machine will now be in Metric Mode. No rehoming is required.

14.3 Scaling - G72

A G72 block reduces or enlarges subsequently programmed movements by a multiplying factor in the range 0.0500 to 20.0000.

Jog moves, tool length and diameter offsets and fixture offsets are not affected by scaling.

14.3.1 Prompts

The table below describes the prompts that the control gives for G72.

Table 14.A

Prompt	Function
SCALING------G72	
X FACTOR-----(X)	scale factor for X axis, the value entered affects the I center word by the same amount
Y FACTOR-----(Y)	scale factor for Y axis, the value entered affects the J center word by the same amount
Z FACTOR-----(Z)	scale factor for Z axis, the value entered affects the K center word by the same amount
W FACTOR-----(W)	scale factor for W axis
V FACTOR-----(V)	scale factor for V axis
U FACTOR-----(U)	scale factor for U axis

Each axis entry can have a different scaling factor, as long as you stay inside the programmable scaling range, 0.0500 to 20.0000.

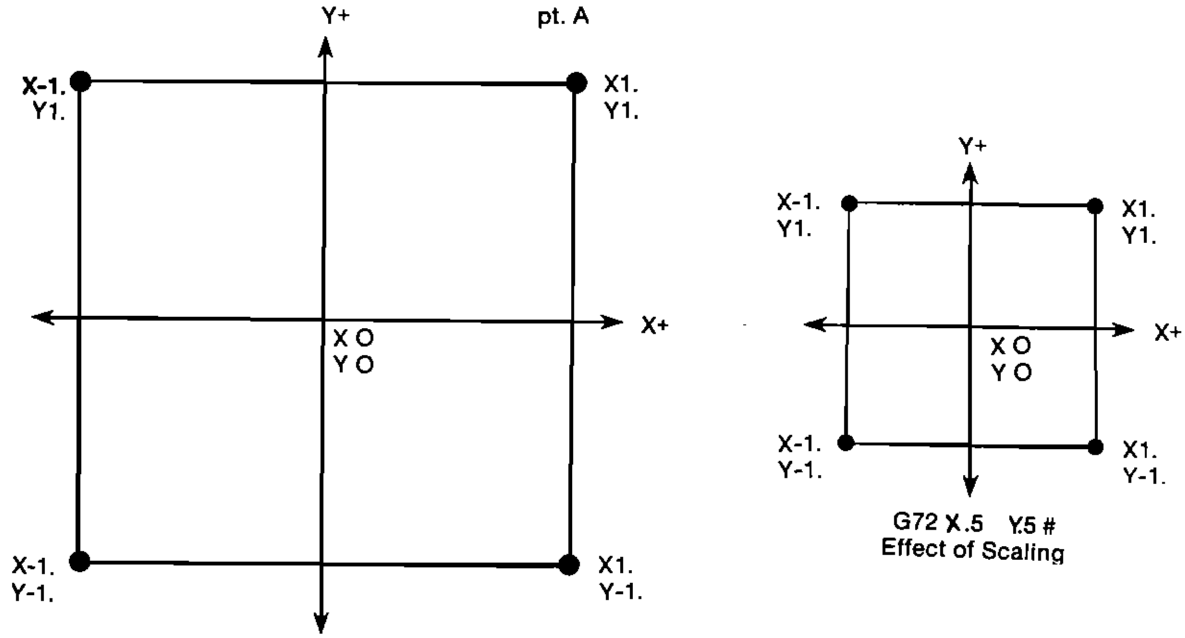
14.3.2 Scaling Example

Assume that you want to scale the moves that produce a square pattern with the X and Y axes. The following program will make the square smaller by a factor of 0.5.

Data Block	Comment
;SCALE SQR #	-- program name
G99 #	-- removes any program zero
G XYZ #	-- rapid to machine zero, assumed in the middle of each axis travel
G91 #	-- set incremental mode for all axes
G72 X.5 Y.5 #	-- set scaling factor of 0.5 on X and 0.5 on Y, the factor will affect X and Y moves following this block.
G1 X2. F20. #	-- this move and the next 3 produce the square, it will be 1 inch on each side instead of 2 inches because of the 0.5 scale factor.
Y-2. #	
X-2. #	
Y2. #	
G72 #	-- cancels scaling by restoring a scale factor of 1 to all axes
M2 #	-- end of program

See figure 14.1 for an illustration of this program example.

Figure 14.1 - Scaling Incremental Moves



14.3.3
Canceling Scaling

To remove the effects of scaling, program a data block that contains only a G72. This sets a scale factor of 1 to all axes, and so, removes any scaling.

14.3.4
Scaling Arc Entries

If you scale arc moves (G02/03 - circular interpolation, or G22/23 - helical interpolation) both axes in the circular interpolation plane must be scaled by the same amount.

For example, if you try to scale an arc in the XY plane (G17) by different amounts in X and Y, an error message "IMPROPER G02/3 ARC" will appear on the CRT.

14.3.5
Scaling Absolute Moves

If you scale absolute (G90) moves, scaling is applied directly to the coordinate system around the zero point. For example, assume that the current absolute position of the axes is X1. Y1. and you program:

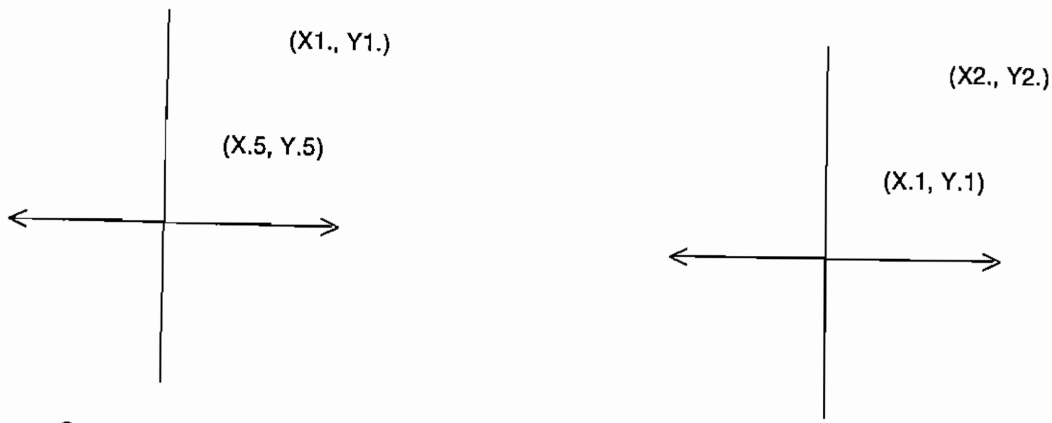
```
G72 X.5 Y.5 #
```

The position of the tool in the scaled coordinate system becomes X2.Y2. If you now program

```
G1 X1.Y1. #
```

the axes will move to that point in the new, scaled coordinate system. See figure 14.2 for an illustration.

Figure 14.2 - Scaling Absolute Moves



Coordinate System
Before Absolute Scaling

Coordinate System
After Absolute Scaling
G72 X.5 X.5

14.3.6 Multiple Scaling

You can produce scaling that is the multiple of scaling factors. For example,

Data Block	Comment
.	
.	
G72 X.5 #	-- sets a scale factor of 0.5 for X axis moves
.	
.	
G72 X.5 #	-- programs a scale factor of 0.5 for X axis moves, but the resulting scaling is 0.5 x 0.5 = 0.25
.	
.	

If you use multiple scaling and the resulting factor exceeds the programmable range, 0.0500 to 20.0000, the control will give the error message "____ CODE OUT-OF-RANGE" where the ____ shows the axis whose scale factor is out of range.

14.3.7 Scaling with Subprograms and Subroutines

Scaling factors used in main programs can be altered in subprograms or subroutines. If you use scaling in a subprogram or subroutine, it is good programming practice to restore the original scaling factors before returning to the main program. There may be an instance where you do not want to restore the original scaling factors to allow the main program to run with the same factors as the subprogram. Unless this is the case, you should always restore the original scaling factor when returning to the main program.

You do this by programming a G72 block that sets the multiple scaling factor in the subprogram to 1. For example, if you use a G72 during a subprogram to set a scaling factor to 0.25, then just prior to returning to the main program a G72 with a factor of 4.0 should be programmed to restore the subprogram scale factor to 1 (0.25 x 4.0 = 1). the programming example on the following pages illustrates this concept.

You should not cancel scaling in a subprogram or subroutine by programming a G72 by itself if the main program is using scaling. Doing this cancels scaling in the main program in addition to the subroutine or subprogram.

For example:

Data Block	Comment
;MAIN PROG#	-- program name of the main program
.	
.	
G72 Y2. #	-- sets scaling factor in main program to 2.0 on the Y axis
.	
.	
P2 ;SUB PROG #	-- calls subprogram number 2, execution jumps to the program in memory whose number is 2, the comment in this block is ignored

MAIN PROGRAM BEGINS

(example continues)

(example continues)

Data Block	Comment
;SUB PROG #	-- program name of program 2 in memory
.	
.	
G72 Y.25 #	-- scale factor in this subprogram is 0.25 on Y. The resulting scaling on Y is now 2.0 x 0.25 = 0.5
.	
.	
G72 Y4. #	-- sets the multiple of scaling in the subprogram to 1 since 0.25 x 4.0 = 1, note that the scaling that is active before returning to the main program is 0.5 x 4.0 = 2.0 on the Y axis
M2 #	-- end of subprogram, return to main program

SUB PROGRAM BRANCH

G Y1. #	-- the main program resumes at this point with a scaling factor of 2.0 on the Y axis
.	
.	
G72 #	-- cancels all scaling by setting factor to 1 for all axes
M2 #	-- end of main program

MAIN PROGRAM RESUMES

14.4 Point to Point Interpolation -- G73

A G73 word defines a linear positioning block that removes corner rounding effects when contouring consecutive linear moves around a part. The G73 word programs a linear interpolation move similar to G01 except that the control waits for an "In Position" signal at the end of each move. There is a slight pause at the start of consecutive linear moves until the control senses an "In Position."

G73 is modal and it cancels G00 and G01.

The prompts that the control gives for G73 are identical to those for G00 and G01. See section 15.4 for more on G73.

14.5 Parts Rotation – G74

A G74 block lets you rotate the coordinate system in the XY plane of subsequently programmed movements. Rotation is at a specified point at a specified angle. Rotation is used to rotate the coordinate system, and may be used to produce parts that have radial symmetry.

14.5.1 Prompts

The table below describes the prompts that the control gives for G74.

Table 14.B

Prompt	Function
ROTATION----G74	
X ROT CENTER--(I)	"I" specifies the X center of rotation. In absolute mode (G90), the value of I is the X coordinate of the center measured from Program Zero. In incremental, I is the X direction and distance from the current location of the axes to the center. If you don't program an I, the last programmed I is assumed to the the X center of rotation. If there is no last I, the control assumes a value of 0 for I.
Y ROT CENTER--(J)	"J" specifies the Y center of rotation. In absolute mode (G90), the value of J is the Y coordinate of the center measured from Program Zero. In incremental, J is the Y direction and distance from the current location of the axes to the center. If you don't program a J, the last programmed J is assumed to the the X center of rotation. If there is no last J, the control assumes a value of 0 for J.
ROTATE ANGLE--(C)	"C" specifies angle of rotation in degrees. A counterclockwise direction is positive, clockwise is negative. The angle C has its vertex at the center of rotation specified by I and J. When C is absolute, it is measured from the positive X axis that extends from the center. When C is incremental, it is measured from an imaginary line that extends from the center to the current location of the axes. If C is not programmed in this block, the last programmed C is assumed by the the control. If there is none, a value of 0 is assumed for C.

Important: The control does not prompt for the following words, but they can be used in the G74 block.

- X -- programs an absolute or incremental X move that is made after the coordinate system has been rotated according to I, J, and C.
- Y -- programs an absolute or incremental Y move that is made after the coordinate system has been rotated according to I, J, and C.
- P -- programs the number of a subprogram that is performed after rotation and any XY moves have occurred.
- H -- programs the starting sequence number of a subroutine that is performed after rotation and any XY moves have occurred.
- E -- programs the ending sequence number of a subroutine that is performed after rotation and any XY moves have occurred.
- L -- programs a repeat execution count for the entire G74 block.

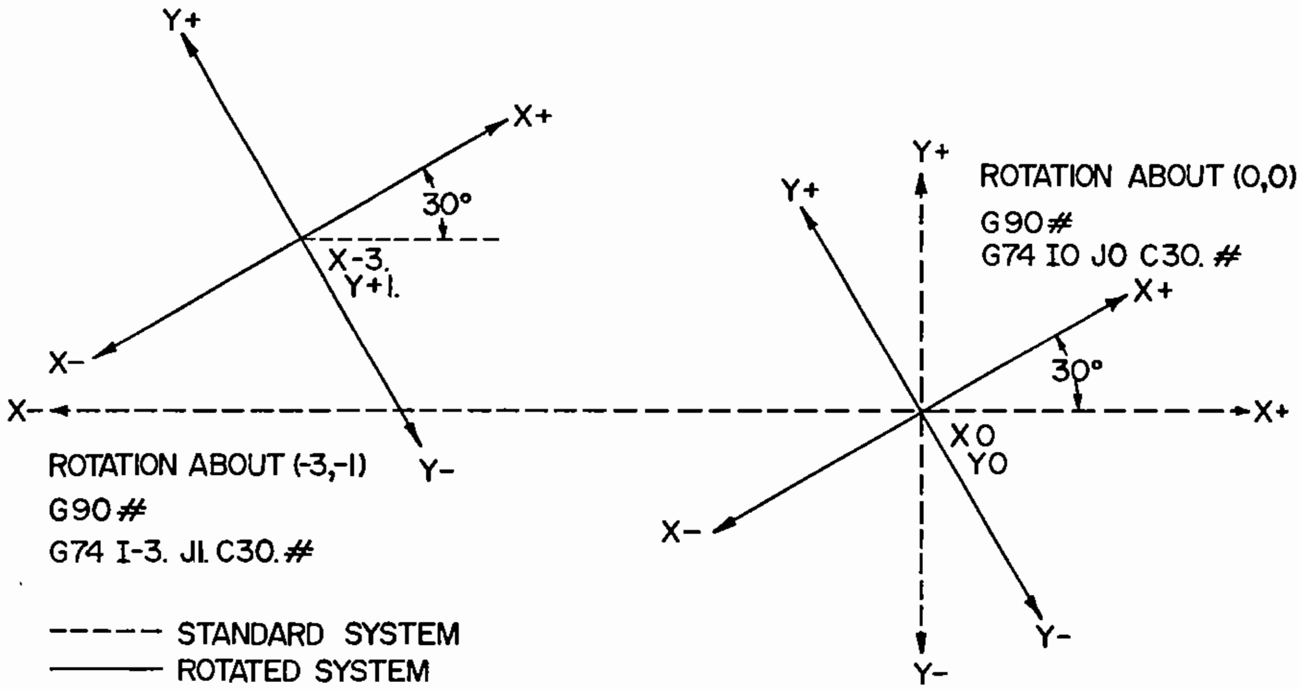
14.5.2 Canceling Rotation

To cancel the effects of rotation, program a G74 word in a block by itself. This removes the rotation angle C and restores the offset to the Program Zero, its non-rotated position.

14.5.3 Rotation About Zero and Non-Zero Coordinates

Figure 14.3 shows the effects of rotating the coordinate system around a Program Zero point (I0 J0 in absolute), and around a non-zero point (I1. J2. in absolute). Note how the Program Zero point has been shifted and the point rotation has not in the non-zero case.

Figure 14.3 - Rotation About Zero and Non-Zero Coordinates



14.5.4 X and Y Moves in the G74 Block

When X and Y moves are included in the G74 block, they specify moves to locations within the rotated coordinate system. You can use this to precisely locate the start point of operations in the rotated system.

When X and Y are absolute, they program moves to coordinates measured from the current (and perhaps, rotated) Program Zero location.

When X and Y are incremental they program a rotated move from the current location of the axes to a new location according the the rotation angle, C.

14.5.5 Repeating a G74 Block - Effect of Absolute/Incremental I, J, and C

When angle C is absolute in the G74 block, the coordinate system will rotate to a single angular position. If you program an L word to repeat this G74 block, the rotation will only occur once.

However, if C is incremental in the G74 block and you repeat it, the rotation will occur each time the G74 block is executed.

When I and J are absolute in the G74 block, they specify rotation only around one point. If you repeat that block, the rotation center does not change. When you repeat a G74 block that has an incremental I and J, the rotation center will change each time the block is executed.

14.5.6 Subprograms and Subroutines in the G74 Block

When you program a subprogram or subroutine by using P, H, or E in the G74 block, and rotation is about a non-zero position, the moves in the subprogram or subroutine should be:

- all absolute moves, or
- a closed path of incremental moves
  A "closed path" is one that starts and ends at the same point.

If you choose to program incremental moves and you cannot program a closed path, you should program absolute X and Y moves in the G74 block to properly locate the start point for the subprogram or subroutine. This is especially the case if you want to repeat the G74 block that calls the subprogram or subroutine.

14.5.7 Examples

The following example program produces the pattern shown in figure 14.4

MAIN PROGRAM BEGINS

Data Block	Comment
;ROTATE DRILL #	-- main program name
G99 #	-- cancel any previous Program Zero and set Machine Zero reference
G90 #	-- set absolute mode for all axes
G92 X10. Y5. Z0 #	-- set current location to this coordinate and thus set Program Zero
G81 Z-1. R0 P0 #	-- establish drill autocycle
G00 X2. Y1. P2 ;SUB PROG #	-- rapid to X2. Y1. from Program Zero and perform subprogram in storage whose number is 2, the comment in this block is ignored

SUB PROGRAM BRANCH

Data Block	Comment
;SUB PROG #	-- name of program 2 in storage, the first block in the subprogram
G91 X #	-- set incremental mode on X axis only
X.5 L3 #	-- execute 0.5 in. rapid move in X a total of 3 times
M2 #	-- end of subprogram

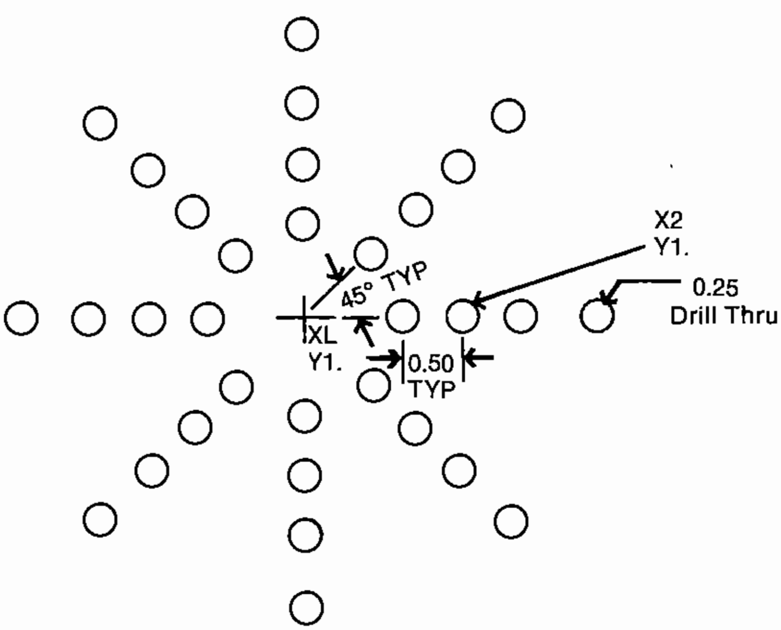
example continues

example continues

MAIN PROGRAM RESUMES

```
G91 C #                -- set C to incremental, remember
                          that X is still absolute in the
                          main program

G74 I1. J1. C45.       -- rotation center is absolute
    X2. Y1. P2 L7 #       I1. J1., angle of rotation is
                          incremental C45., reference
                          position is absolute X2. Y1. in
                          the rotated system, call
                          subprogram 2, and perform
                          entire block 7 times

G74 #                  -- cancel rotation

G80 #                  -- cancel G81 drill autocycle.

G99 XYZ #              -- cancel position preset and
                          return at rapid to Machine Zero

M2 #                   -- end of main program
```

Figure 14.4 - Rotation Example 1



The following program produces the path shown in figure 14.5.

MAIN PROGRAM BEGINS

Data Block	Comment
;ROTATE PATH #	-- main program name
G99 #	-- cancel any previous Program Zero
G90 #	-- establish absolute mode for all axes
G X-10. Y-4. Z-.5 #	-- rapid to start of pattern from Machine Zero
G92 XYZ #	-- set Program Zero at this point
G91 C #	-- set C to incremental
G74 IJ C60. X.6392 Y1.357 P2 L6 #	-- rotation about I0 J0, incremental C60., reference position X.6392 Y1.357 in the rotated system, execute program in storage whose number is 2, execute entire block 6 times

SUB PROGRAM BRANCH

Data Block	Comment
;SUB PROG #	-- name of program 2 in storage
G91 #	-- set incremental mode for all move in this subprogram
G1 Z-1. F10. #	-- feed Z to depth
G3 X-1.2784 Y I-.6392 J-1.357 #	-- arc move from A to B
X-.7513 Y1.3013 #	-- move B to C
G2 X2.781 Y I1.3905 I1.3905 J-2.6583 #	-- arc move C to D
X-.7513 Y-1.3013 #	-- move D to A
G Z1. #	-- rapid Z up
M2 #	-- end of subprogram

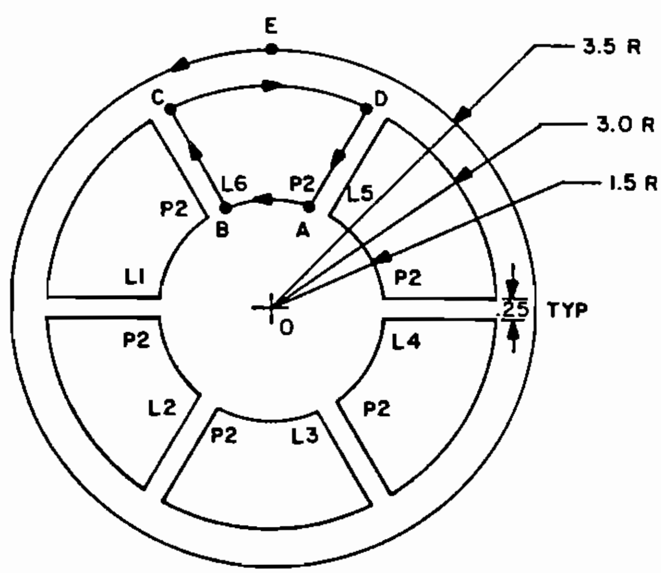
(example continues)

(example continues)

MAIN PROGRAM RESUMES

```
G74 #               -- cancel rotation
X Y3.5 #            -- rapid to perimeter start E
G1 Z-1. F10. #      -- feed Z to depth
G2 X Y3.5 IJ #      -- execute full circle with 3.5
                       in. radius, center at Program
                       Zero
G Z #               -- rapid Z up to Program Zero
G99 XYZ #           -- cancel Program Zero and rapid
                       back to Machine Zero
M2 #                -- end of main program
```

Figure 14.5 - Rotation Example 2



14.6 Absolute Programming Mode -- G90

In the absolute programming mode, axis locations are measured from the Machine Zero position or the established Program Zero position. (Program Zero is established in a G92 data block.)

The alternative to absolute mode is the incremental mode (G91). In incremental mode, all axis moves are measured from the current location of the axes.

The control assumes the absolute mode for all axes and positioning words at power-up, and at the start of any main program.

The words that use the absolute mode are:

X Y Z U V W I J K A B C R

You can select which words are absolute and which are incremental. Programming G90 by itself in a data block sets the absolute mode for all positioning words. I,J,K,A,B,C,and R are not axes but are affected by G90. Programming G90 and only selected words sets absolute mode only for those words. For example,

Data Block	Comment
.	
.	
G90 #	-- sets absolute mode for all positioning words
.	
.	
G91 #	-- sets incremental mode for all positioning words
G90 X Y #	-- sets absolute mode for only the X and Y words, the rest are incremental due to the previous block
.	
.	

You can see which axes and positioning words are absolute or incremental by looking at the Status page of the control. Words that appear in uppercase letters are currently absolute, and words that are lowercase are currently incremental.

A G90, once set for a word, is modal and cancels G91 for the axes involved.
However, absolute mode words can be forced to incremental for the current block by referencing them with a lower case letter.

For example:

Data Block	Comment
•	
•	
G90 #	-- sets absolute mode for all positioning words
GO X1. y.5 #	-- Moves X to absolute coordinate of 1.0; Moves Y (incremental) by 0.5 from its current location
XO YO #	-- Moves X and Y to absolute coordinates of 0.0.
•	
•	

To program lower case letters, hold the [SHIFT] key down and press the desired letter.

14.7 Incremental Programming Mode -- G91

In incremental mode, the control measures programmed axis locations from the current location of the axes. Positioning words in this mode give the distance and direction from the current location to the end of the move. The axes to which an incremental mode may be applied are:

X Y Z U V W I J K A B C R

Programming a G91 in a block alone sets the incremental mode for all positioning words. Programming a G91 block with specific positioning words sets incremental mode only for those words.

For example:

Data Block	Comment
•	
•	
G91 #	-- set incremental mode for all positioning words
•	
•	
G90 #	-- sets absolute mode for all positioning words
G91 X Y #	-- sets incremental only for the X and Y words, the rest are absolute due to the previous block
•	
•	

You can see which positioning words are incremental and which are absolute by looking at the Status page of the control. Words that are shown with lowercase letters are incremental, those shown with uppercase letters are absolute.

A G91, once it is set for a word, is modal and it cancels G90 for the words specified.

Words that are set to incremental will behave the same, regardless whether the letter is upper or lower case.

14.8 Preset Program Zero -- G92

A G92 block lets you define the origin of absolute coordinates in your program. It lets you set a Program Zero point that is different from the Machine Zero point (the machine's origin of absolute coordinates) that the installer sets.

To set a Program Zero, program a G92 block that calls out the current position from Program Zero.

For example:

Data Block	Comment
•	
•	
G92 X1. Y2. Z.5 #	-- sets the current location of the axes at this coordinate referenced to Program Zero, it essentially sets Program Zero at a point that is 1 in. in negative X, 2 in. in negative Y, and 0.5 in. in negative Z away from the current location.
•	
•	

Or you can move to a location from Machine Zero, then set this location as Program Zero. For example:

Data Block	Comment
•	
•	
G99 #	-- cancels Program Zero and sets Machine Zero reference.
G X-5. Y-3. Z-1. #	-- rapid to coordinate from Machine Zero
G92 XYZ #	-- sets this location as Program Zero for subsequent absolute moves
•	
•	

You can set the Program Zero for the U, V, and W axes as well, by including them in the G92 block.

The Program Zero point set with G92 is modal. It is canceled by a G99 data block. The G99 code returns the origin of coordinates to the Machine Zero position.

14.9 Per Time Feedrate -- G94

A G94 data block lets you define feedrates for linear and rotary moves.

14.9.1 Prompts

The following table describes the prompts that the control gives for G94.

Table 14.C

Prompt	Function
PER TIME FEED--G94	
LINEAR FEED---(F)	"F" specifies the feedrate for linear moves that come after it in the program. Units for linear axis feedrate are inches per minute (ipm, format F4.1), or millimeters per minute (mmpm, format F5). This rate remains in effect until a new F word is programmed, either in another G94 block, or in a motion block.
ROTARY FEED--(C)	"C" specifies the feedrate for rotary moves that come after it in the program. This rate remains in effect until a new C word is programmed in another G94 block. Note that the system installer sets rotary feedrate programming in AMP as revolutions per minute (rpm), degrees per minute (dpm), or degrees per second (dps).

14.10 Cancel Program Zero -- G99

A G99 block removes the effect of a previous G92 data block. It defines the origin of absolute coordinates as the Machine Zero position. Unlike the G92 block in which you can preset an individual axis without affecting the reset, the G99 cancels all presets.

Any programmed axis words in the G99 block will cause the axis to move. These moves will be to absolute positions from Machine Zero if the current mode is absolute (G90). Or they will be to incremental positions from the current location if the mode is incremental. These move can also be made at rapid (G00) or at feedrate (G01) depending on the current interpolation mode when the G99 block is executed.

For example:

Data Block	Comment
•	
•	
G99 #	-- cancels any previous Program Zero set with a G92 block, and sets the origin of absolute coordinates to Machine Zero
G XYZ #	-- rapid move to Machine Zero for the X, Y, and Z axes
•	
•	
G X-10. Y-5. Z-1. #	-- rapid to this coordinate from Machine Zero
G92 XYZ #	-- set this point as Program Zero for subsequent absolute moves
•	
•	
G #	-- set rapid mode without moving
G99 XYZ #	-- rapid move back to Machine Zero
•	
•	

The G99 word is modal and it cancels the effect of any previously programmed G92.

14.11 Chapter Summary

This chapter described the programming of G word data blocks in the coordinate control group. A thorough understanding of this group will make your programming easier and less complex by letting you control the coordinate system of the machine at will.

The next chapter describes G word data blocks in the positioning/dwell group. These words define the most basic and most widely use positioning operations.

Chapter 15 - Positioning/ Dwell

15.0 Chapter Overview

This chapter covers the group of G words that define positioning/dwell data blocks. After reading this chapter, you will know where to find out:

- what linear and circular interpolation are
- what words you can use in positioning blocks
- how to program positioning blocks
- how to use polar positioning
- how to divide a move into a number of segments
- how to program a dwell time

15.1 Positioning G Words

There are five G words you can use to define data blocks that position the machine axes. Other G words may define data blocks that produce movement, but these words produce the basic kinds of positioning that you use most often.

15.1.1 Linear Interpolation G Words

The following G words define blocks that move the machine axes along linear paths. This is called linear interpolation since the control coordinates the movement of separate axes to produce moves from point to point.

The G words for moving along a linear path are (see section 15.4):

- G00 -- rapid positioning mode, makes the axes move at their maximum rate that is set by the installer.
- G01 -- linear positioning mode, makes the axes move at a programmed feedrate along a continuous path.
- G73 -- point to point positioning, makes the axes move at a programmed feedrate along a straight line, point to point. The control waits for an "In Position" signal before proceeding with the next programmed move.

Important: G73 is listed in chapter 14, Coordinate Control, but the programming concepts for G73 are found in this chapter.

15.1.2 Circular Interpolation G Words

The control uses 2 G words for producing circular paths, or arcs, with 2 axes at a time. This is called circular interpolation because the control coordinates the separate movements of the axes to produce an arc. The arc is produced by making a series of very small linear paths between points along the arc.

The G words that define circular interpolation are (see section 15.5):

- G02 -- clockwise circular interpolation, makes 2 axes move along an arc in a clockwise direction.
- G03 -- counterclockwise circular interpolation, makes 2 axes move along an arc in counterclockwise direction.

15.1.3 Axis Words in Positioning Blocks

The following words may appear in positioning blocks to define motion. They program the movement of axes, either directly or in conjunction with each other. Other words may also be used to determine other conditions and functions.

```
X Y Z U V W I J K A C R
```

Linear Axes Words

The words X, Y, and Z represent the standard linear axes. They program positions along linear paths. The control can coordinate the movement of any 2 of these axes to produce paths along an arc. Linear axis words are programmed in units of inches (G70) or millimeters (G71).

Secondary Axes Words -Linear or Rotary

The installer of your system may have provided you with up to 3 additional axes: U, V, and W. These axes may be linear, in which case they are usually parallel in any combination to the X, Y, and Z axes, respectively. These axes cannot be used to produce programmed paths along an arc. They cannot be included in G02/G03 blocks. When these axes are linear, they are programmed in units of inches (G70) or millimeters (G71)

The U, V, or W axes may also be rotary, in other words, they rotate like the table of a record player. When these axes are rotary, they are programmed in units of degrees.

Center of Movement Words

The words I, J, and K in positioning blocks program the center of movement. They are used to locate center points for the X, Y, and Z axes, respectively. These words are used for circular interpolation and polar positioning.

Polar Positioning Words

The words A, C, and R are used in polar positioning blocks. The A word programs an angle whose vertex is at the current axes position. The C word programs an angle whose vertex is at the center of movement. The R word programs a radial length. See the specific descriptions that follow for how to use these words.

15.2 Feedrates

There are two methods you can use to establish feedrates for positioning data blocks: one for feedrates applied to linear axes, and one for feedrates applied to rotary axes.

15.2.1 Linear Feedrates – F Word

An F word in a positioning data block programs a feedrate for the movements of linear axes. F is programmed in units of:

- inches per minute (ipm, format F4.1) in G70 mode
- millimeters per minute (mmpm, format F5) in G71 mode

Feedrates for linear and circular interpolation are "vector" feedrates. That is, all axes move simultaneously at individual rates so that the rate along the effective path is equal to the programmed F rate.

A feedrate, once programmed, is modal. It applies to every positioning block that comes after it in the program unless you program a new F word value.

15.2.2 Rotary Feedrates – G94 C Word

Feedrates for rotary axes are programmed with a C word in a G94 block. The C word programs rotary feedrates in units of:

- the C word revolutions per minute (rpm) format is C4
- the C word degrees per minute (dpm) format is C3.1
- the C word degrees per second (dps) format is C3.1

Which format is used for rotary feedrates depends on the programming of control parameters in AMP.

The rotary feedrate, once established, is modal for rotary moves made independently from the linear axes. It remains in effect until another G94 C block is programmed.

You can program linear and rotary moves in the same positioning data block. When you combine linear and rotary moves, the CNC figures out the correct feedrate to use.

The CNC calculates the length of time to complete the programmed move for each axis in the program block using the programmed feedrate (not the maximum speed of the axis). The axis performing the move which takes the longest time to complete is the limiting axis. The feedrate and feedrate type for the limiting axis is displayed on the Status page in the upper right hand corner. This limiting axis may be IPM (linear), or RPM, DPM, DPS (rotary). See figure 10-3.

The CNC commands the limiting axis to move at a rate as close to 100% of the programmed feedrate as possible without pushing any other axis over its maximum speed or 150% of its programmed speed, whichever is less.

When the limiting axis is linear, the rotary axis will not move any faster than its global rapid equivalent. For example, if the global rapid is 200 IPM, at 10,000 counts per inch, you would have 2,000,000 counts per minute. This equals about 33,300 counts per second. Rotary axis resolution on the 8400MP is always 1000 counts per degree. So with a global rapid rate of 200 IPM, the rotary maximum speed in linear/rotary combined moves is limited to about 33.3 degrees per second. This happens even if rotary max is greater than 33.3 DPS.

```
Example: 1

G94 F 200 C 1.69 #
G00 X 0.0 W 0.0  #
G01 X 1.0 W 45.0 #

Time at programmed rate:
(X axis .3 sec.) (W axis 26.946 sec.)

'W' limits 'X'
```

$$\frac{1 \text{ inch}}{26.946 \text{ sec}} \times 60 \text{ sec} = 2.22 \text{ IPM}$$

```
X runs at 2.22 IPM
W runs at 1.67 DPS

Example: 2

G94 F 200 C 1.67 #
G00 X 0.0 W 40.0 #
G01 X 20.0 W 45.0 #

Time at programmed rate:
(X axis 6.0 sec.) (W axis 3.0 sec.)

'X' limits 'W'
```

$$\frac{5 \text{ degrees}}{6 \text{ sec}} = .83333 \text{ DPS}$$

```
X runs at 200.0 IPM
W runs at .833 DPS
```

Important: No axis may exceed the max rate set in AMP even if the programmed feedrate is higher than the max rate.

15.3 Block Divide -- D Word

A D word (format D3) in a positioning block divides the move into a number of equal length segments. The D word can have a numeric value from 0 to 255.

The end point of each segment of the move is a position where a specific operation should take place: an autocycle, subprogram or subroutine, for example. The block divide point always looks for an autocycle, but by using a G79 code the autocycle can be a subprogram or subroutine. You can program the D word in any positioning block, and the automatic operation you select will be performed at the end of each segment.

When you use the D word to divide a circular interpolation move, the movement between points that define the segments is by the shortest linear path, that is, the moves between points does not follow the arc.

For more on block divide see section 13.5.4, and chapter 17, Autocycles.

15.4 Linear Interpolation Modes -- G00, G01, and G73

The G words that define data blocks for linear interpolation move one or more axes from point to point along linear paths. The programmed move begins at the current position of the axes and ends at the destination programmed in the block. The axes start and stop moving at the same time.

- G00 -- causes the axes to move along a straight line to their programmed destination points at a maximum (rapid) feedrate. When the axes reach their destinations, the servo following error must be within predefined limits ("In Position") before the next move is started. The G00 word is modal and cancels G01 and G73. The G00 mode will be automatically cancelled if a G02/G03 circular interpolation move is executed. The control sets the G01 mode for subsequent blocks.

- G01 -- causes the axes to move along a straight line at a constant velocity specified by the programmed feedrate. You can modify the actual feedrate by using the Feedrate Override Switch. When a rotary axis move is programmed in the block, it starts and stops simultaneously with any linear axis moves programmed in the same block. The G01 word is modal and cancels G00 and G73. The control assumes the G01 mode at power-up.

- G73 -- causes the axes to move along a straight line at a constant velocity specified by the programmed feedrate. It is similar to the G01 mode except that the control waits for the "In Position" signal before proceeding with the next move. This eliminates "corner rounding" when sharp corners are required for consecutive moves around the perimeter of a part. The G73 word is modal and cancels G00 and G01. The G73 mode will be automatically cancelled if a G02/G03 circular interpolation move is executed. The control sets the G01 mode for subsequent blocks.

15.4.1 Linear Interpolation Prompts

The following table explains the prompts that the control gives for G00, G01, and G73. Only G00 is shown, but the prompts are identical for the other linear interpolation words. Note that only those axes that are enabled in the control are prompted.

When in the absolute mode (G90), incremental moves can be programmed by typing in lower case letters for the axis.

Table 15.A

Prompt	Function
RAPID------G00	
X END POINT--(X)	"X" programs the end point of movement for the linear X axis. The X word may be absolute (G90) or incremental (G91). When X is absolute, it programs the coordinate of the end point for the X axis measured from Program Zero. When X is incremental, it programs the distance and direction along the X axis from the current location to the end point.
Y END POINT--(Y)	"Y" programs the end point of movement for the linear Y axis. The Y word may be absolute (G90) or incremental (G91). When Y is absolute, it programs the coordinate of the end point for the Y axis measured from Program Zero. When Y is incremental, it programs the distance and direction along the Y axis from the current location to the end point.
Z END POINT--(Z)	"Z" programs the end point of movement for the linear Z axis. The Z word may be absolute (G90) or incremental (G91). When Z is absolute, it programs the coordinate of the end point for the Z axis measured from Program Zero. When Z is incremental, it programs the distance and direction along the Z axis from the current location to the end point.
W END POINT--(W)	"W" programs the end point of movement for the linear or rotary W axis. The W word may be absolute (G90) or incremental (G91). When W is absolute, it programs the coordinate of the end point for the W axis measured from Program Zero. When W is incremental, it programs the distance and direction along the W axis from the current location to the end point.

U END POINT--(U)	"U" programs the end point of movement for the linear or rotary U axis. The U word may be absolute (G90) or incremental (G91). When U is absolute, it programs the coordinate of the end point for the U axis measured from Program Zero. When U is incremental, it programs the distance and direction along the U axis from the current location to the end point.
V END POINT--(V)	"V" programs the end point of movement for the linear or rotary V axis. The V word may be absolute (G90) or incremental (G91). When V is absolute, it programs the coordinate of the end point for the V axis measured from Program Zero. When V is incremental, it programs the distance and direction along the V axis from the current location to the end point.
POLAR LENGTH-(R)	"R" specifies a polar length, and is used only for polar positioning blocks. The R word can be absolute or incremental. When R is absolute, it programs a polar length measured from the center of polar movement to the desired end point. When R is incremental, it programs a polar length measured from the current location of the axes (or its polar projection) to the desired end point.
START ANGLE--(A)	"A" specifies the angle of the next move in polar positioning blocks. The vertex of angle A is the current location of the axes, and a positive angle is counterclockwise. The A word can be absolute or incremental. When A is absolute, it is measured from the positive X axis that extends from the current location. When A is incremental, it is measured from an imaginary extension of the previous move.
CENTER ANGLE-(C)	"C" specifies an angle whose vertex is at a center point in polar positioning blocks. A positive C angle is counterclockwise. The C word can be absolute or incremental. When C is absolute, it is measured from the positive X axis that extends from the center point. When C is incremental, it is measured from an imaginary line that extends from the center point to the current tool position.
X POLAR CENTR-(I)	"I" programs the center of polar positioning for the X axis. The I word can be absolute or incremental. When I is absolute, it is measured from Program Zero. When I is incremental, it programs the distance and direction from the current position to the center along the X axis.

Y POLAR CENTR-(J)

"J" programs the center of polar positioning for the Y axis. The J word can be absolute or incremental. When J is absolute, it is measured from Program Zero. When J is incremental, it programs the distance and direction from the current position to the center along the Y axis.

15.4.2 Standard Linear Interpolation

Standard linear interpolation combines the movement of up to 6 axes. All axes specified in the data block start and stop at the same time, and move at a feedrate specified with the F word (unless G00 is programmed, then the rate is rapid). Feedrates for rotary axes are specified with G94 and a C word.

As an aid to understanding linear interpolation, study the following example program.

Data Block	Comment
;POSITION #	-- comment block that gives the program name
G99 #	-- cancel any previous position preset, establish machine zero as reference
G90 #	-- establish absolute positioning mode
G00 X0 Y0 Z0 W0 #	-- rapid mode set with G00, rapid to machine zero (we assume, for the purposes of our examples, that machine zero is somewhere in the middle of machine travel). The control will wait for "in position" before proceeding to the next move.
X-1. Y-2. #	-- rapid (since G00 is modal) to X-1. Y-2. from machine zero.
G92 X0 Y0 Z0 W0 #	-- set the current position as program zero, W (rotary in this example) is set to 0
G94 C10. #	-- set feedrate for rotary axis (10 dps in this example)

(example continues)

```
G01 X2. F30. #        -- set linear mode (G01), and move
                         to X2. from program zero at 30
                         ipm (modal linear feedrate)

W20. #                -- move W axis 20° (positive =
                         CCW) from program zero at 10
                         dps, mode is G01 as set by
                         previous command

Y-2. Z-1. #           -- move Y-2. Z-1. from program
                         zero at 30 ipm, mode is G01

G73 X-2. Y2. Z-1. F50.# change mode to G73, feedrate is
                         now 50 ipm, move to X-2. Y2. Z-
                         1. from program zero and wait
                         for "in position"

G91 #                 -- change mode to incremental for
                         all axes

Y-2. W20. #           -- simultaneous incremental move
                         for the Y an W axes, both axes
                         start and stop at the same
                         time, the rate for Y axis is 50
                         ipm, and the rate for W
                         (rotary) axis is such that it
                         keeps up with the Y axis.  If
                         this rate will exceed rapid for
                         W, the Y axis rate is scaled
                         down so that W does not exceed
                         rapid, mode is still G73

G00 #                 -- set mode to rapid

G90 #                 -- set mode to absolute

G99 X0 Y0 Z0 W0 #     -- cancel program zero and return
                         the axes at rapid to machine
                         zero

M02 #                 -- end of program
```

15.4.3 Polar Linear Positioning

You can program linear movements (G00, G01, and G73) using polar positioning regardless of plane selection.

Polar linear moves use the words R, A, or C in several combinations.

- R -- specifies a polar length. When R is absolute, it programs a polar length measured from the center of polar movement to the desired end point. When R is incremental, it programs a polar length measured from the current location of the axes (or its polar projection) to the desired end point.

- A -- specifies the starting angle of the next move in polar positioning blocks. For linear moves it is equivalent to the angle to the end point. The vertex of angle A is the current location of the axes, and a positive angle is counterclockwise. When A is absolute, it is measured from the positive X axis that extends from the current location. When A is incremental, it is measured from an imaginary extension of the previous move.

- C -- specifies the angle to the end point whose vertex is at a center point in polar positioning blocks. A positive C angle is counterclockwise. When C is absolute, it is measured from the positive X axis that extends from the center point. When C is incremental, it is measured from an imaginary line that extends from the center point to the current tool position.

There are 7 combinations of words that form legal polar positioning blocks for linear moves. They are:

Legal Linear Polar Block Formats

- R
- RA
- RIJ
- RCIJ
- C
- RC
- CIJ

Each of these is described below.

R Only Block Format

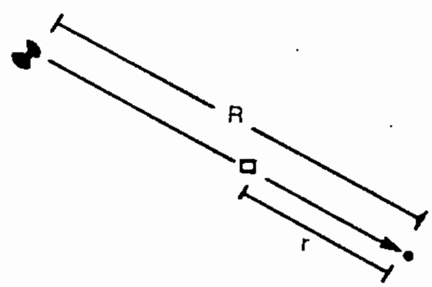
When R is the only word defining motion in a linear interpolation block, it commands a move of length R along a line that extends from program zero through the current location of the axes.

R can be absolute, in which case it programs the distance from program zero to the end of the move. When R is incremental, it programs the distance from the current location of the axes to the desired end point.

Figure 15.1 gives a concept view of the R only polar block format.

Figure 15.1 - R Only Block Format Concept View

upper case letter = absolute
lower case letter = incremental
= program zero
□ = current location
● = end point



C Only Block Format

When C is the only word defining motion programmed in a polar linear interpolation block, it commands a move from the current location of the axes to an end point at angle C from program zero and the same distance from program zero as the current position.

The length of this line at angle C is:

$$\sqrt{X^2 + Y^2}$$

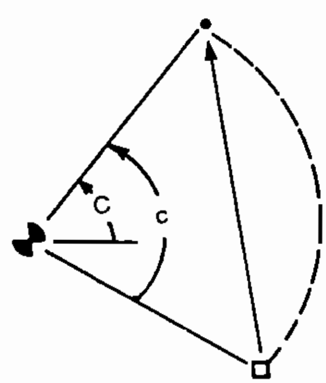
as measured from program zero to the current location of the axes.

The angle C has its vertex at the last polar center defined. If not defined then the vertex is at program zero. Angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis that extends from program zero. When C is incremental, it is measured from the line that extends from program zero to the current location of the axes.

Figure 15.2 gives a concept view of the C only block format for polar positioning.

Figure 15.2 - C Only Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
= program zero
□ = current location
● = end point



RA Block Format

When R and A are the only words defining motion in a polar positioning block, they program a move from the current location of the axes to an end point that is R distance away at an angle A.

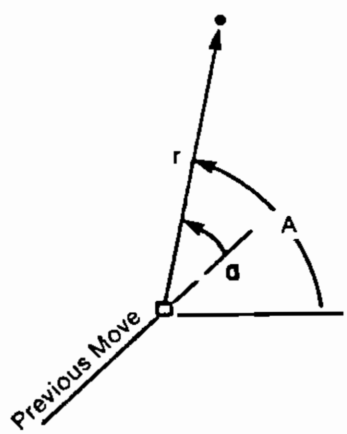
R can only be incremental in this format and a move must precede the RA block.

Angle A has its vertex at the current location of the axes. Angle A can be absolute or incremental. When A is absolute, it is measured from the positive X axis that extends from the current location of the axes. When A is incremental, it is measured from an imaginary extension of the move that preceded the RA block.

Figure 15.3 gives the concept view of the RA polar command.

Figure 15.3 - RA Block Format — Concept View

upper case letters = absolute
lower case letters = incremental
□ = *current location*
● = *end point*



RC Block Format

When R and C are the only words defining motion that appear in a polar linear interpolation block, they program a move from the current location of the axes to an end point that lies on a line at angle C.

R can be absolute or incremental. When R is absolute, the length of the line at angle C is the length of R. When R is incremental, the length of the line at angle C is:

$$R + \sqrt{X^2 + Y^2}$$

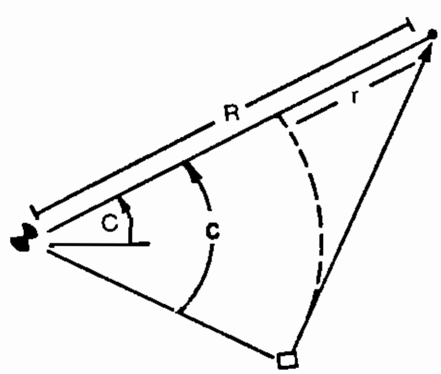
where X and Y are the coordinates of the current location of the axes measured from current arc center.

Note that $\sqrt{X^2 + Y^2}$ is the distance the current location is from current arc center.

Angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis that extends from current arc center. When C is incremental, it is measured from a line that extends from current arc center to the current location of the axes.

Figure 15.4 - RC Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
✦= arc center point
□ = current location
● = end point



RIJ Block Format

When R, I, and J are the only words defining motion in the polar linear positioning block, they program a move from the current location of the axes to an end point on a line. This line extends from a center point defined by I and J through the current location of the axes.

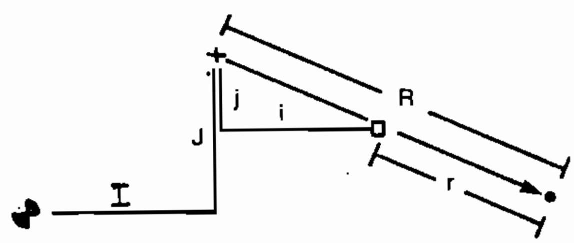
I and J can be absolute or incremental. When they are absolute they define the X and Y coordinate of the center point measured from program zero. When they are incremental, they define the distance and direction on the X and Y axes from the current location of the axes to the center point.

R can be absolute or incremental. When R is absolute, it is the distance along the line extending from the center point to the end point. When R is incremental, is the distance along the line extending from the center point, from the current location of the axes to the end point.

Figure 15.5 gives the concept view of the RIJ polar positioning command.

Figure 15.5 - RIJ Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
= program zero
+ = center point
□ = current location
● = end point



CIJ Block Format

When C, I, and J are the only words defining motion in a polar linear interpolation block, they program a move from the current location of the axes to an end point on a line at angle C and at the same distance from a point defined by I and J as the current location.

I and J can be absolute or incremental. When they are absolute they define the X and Y coordinates of the center point measured from program zero. When they are incremental, they define the distance and direction on the X and Y axes from the current location of the axes to the center point.

Angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis that extends from the center point. When C is incremental, it is measured from a line that extends from the center point to the current location of the axes.

The length of the line at angle C is:

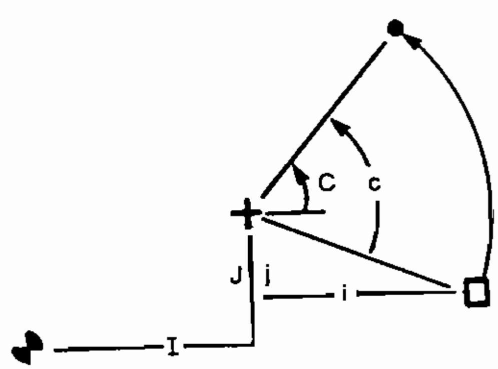
$$\sqrt{X^2 + Y^2}$$

where X and Y are the distances from the center point to the current location of the axes. That is, it is the same distance from the point defined by I and J as the current location.

Figure 15.6 gives the concept view of the CIJ polar command.

Figure 15.6 - CIJ Block Format -- Concept View

upper case letters = absolute
lower case letters = incremental
= program zero
+ = center point
□ = current location
● = end point



RCIJ Block Format

When R, C, I, and J are the only words defining motion used in a polar linear interpolation block, they program a move from the current location of the axes to an end point on a line. The line is at angle C and extends from a center point defined by I and J for a distance defined by R.

I and J can be absolute or incremental. When they are absolute they define the X and Y coordinate of the center point measured from program zero. When they are incremental, they define the distance and direction on the X and Y axes from the current location of the axes to the center point.

R can be absolute or incremental. When R is absolute, the length of the line at angle C is the length of R.

When R is incremental, the length of the line at angle C is R more than the distance from the center point to the current axes location:

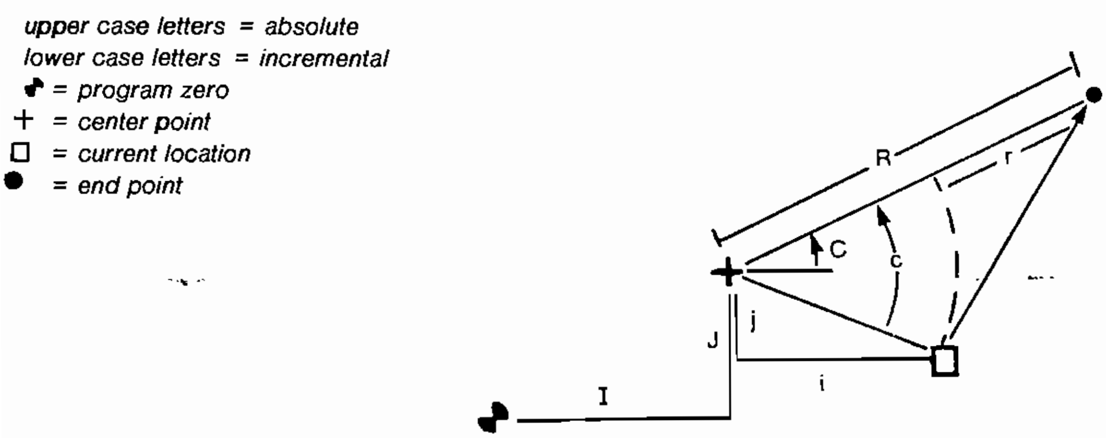
$$R + \sqrt{X^2 + Y^2}$$

where X and Y are the coordinates of the current location of the axes measured from the center point.

Angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis that extends from the center point. When C is incremental, it is measured from a line that extends from the center point to the current location of the axes.

Figure 15.7 gives the concept view of the RCIJ polar command.

Figure 15.7 - RCIJ Block Format – Concept View



15.5 Circular Interpolation – G02 and G03

A G02 block generates a clockwise circular arc. A G03 block produces a counterclockwise arc. The arc is made with simultaneous movements of 2 axes in a plane.

Movement is clockwise or counterclockwise as you look in the negative direction of the axis that is not in the plane of the axes that produce the arc.

Each G02 or G03 block can define an arc of any length. The arc crosses quadrant boundaries continuously.

Before executing a block containing circular interpolation, the control automatically checks the arc end point and center for correct dimensions. If a geometrically impossible arc is programmed, the control halts, and displays "IMPROPER G02/3 ARC." Programs can be checked quickly for such errors using Quick Check.

G02 and G03 are not modal. However, they cancel the G00 and G73 mode and set the G01 mode for subsequent blocks.

15.5.1 Prompts

The following table describes the prompts that the control gives for G2 Clockwise and G3 Counterclockwise Circular Interpolation. G2 is shown, but G3 has the same prompts.

Table 15.B

Prompt	Function
ARC CW -- G2	
X END POINT--(X)	"X" specifies the end point of the arc on the X axis. It is used only for non-polar arcs, and only for arcs in the XY (G17) or XZ (G18) planes. X can be absolute or incremental. When X is absolute, the X value is the X coordinate of the end point measured from Program Zero. When X is incremental, it programs the distance and direction from the start point of the arc to the end point along the X axis.

Y END POINT--(Y)

"Y" specifies the end point of the arc on the Y axis. It is used only for non-polar arcs, and only for arcs in the XY (G17) or YZ (G19) planes. Y can be absolute or incremental. When Y is absolute, the Y value is the Y coordinate of the end point measured from Program Zero. When Y is incremental, it programs the distance and direction from the start point of the arc to the endpoint along the Y axis.

Z END POINT--(Z)

"Z" specifies the end point of the arc on the Z axis. It is used only for non-polar arcs, and only for arcs in the XZ (G18) or YZ (G19) planes. Z can be absolute or incremental. When Z is absolute, the Z value is the Z coordinate of the end point measured from Program Zero. When Z is incremental, it programs the distance and direction from the start point of the arc to the endpoint along the Z axis.

X ARC CENTER--(I)

"I" specifies the location of arc center on the X axis. It is used for non-polar arcs in the XY (G17) and XZ (G18) planes, and for the CIJ polar arc format (XY plane only). It can be absolute or incremental. When I is absolute, it specifies the X coordinate of the center measured from Program Zero. When I is incremental, it specifies the distance and direction from the start point of the arc to the center along the X axis.

Y ARC CENTER--(J)

"J" specifies the location of the arc center on the Y axis. It is used for non-polar arcs in the XY (G17) and YZ (G19) planes, and for the CIJ polar arc format (XY plane only). It can be absolute or incremental. When J is absolute, it specifies the Y coordinate of the center measured from Program Zero. When J is incremental, it specifies the distance and direction from the start point of the arc to the center along the Y axis.

Z ARC CENTER--(K)

"K" specifies the location of arc center on the Z axis. It is used only for non-polar arcs in the XZ (G18) and YZ (G19) planes. It can be absolute or incremental. When K is absolute, it specifies the Z coordinate of the center measured from Program Zero. When K is incremental, it specifies the distance and direction from the start point of the arc to the center along the Z axis.

POLAR ANGLE---(C)	"C" specifies the length of an arc in degrees. The angle of the arc has its vertex at the center of the arc. C is used only for polar arcs (XY plane only). It can be absolute or incremental. When C is absolute, the angle is measured from the positive X axis that extends from the center. When C is incremental, it is measured form a line that extends from the start point of the arc to the center.
POLAR LENGTH--(R)	"R" specifies the radius of a polar arc. It is used only for polar arcs (XY plane only), and must be programmed as absolute. A move must precede the polar arc block in which R is used. R is measured from the end point of this move.
START ANGLE---(A)	"A" specifies the angle of an imaginary line that a polar arc is tangent to. It is used only for polar arcs (XY plane only). A move must precede the polar arc block in which A is used. Angle A has its vertex at the end point of this move. Angle A can be absolute or incremental. When A is absolute, the angle is measured from the positive X axis extending from the end point of the previous move. When A is incremental, it is the angle between a line tangent to the end of the previous move and a line tangent to the beginning of the move being programmed.

15.5.2 Selecting the Plane

Before you program the block that executes the arc, program a block that selects the plane for the 2 axes that produce the arc. Use:

- G17 -- X and Y axes produce the arc. This plane is selected by the control at power-up and at the beginning of a main program. This plane must be used with any arc that is programmed with polar positioning (using R, A, or C words).
- G18 -- X and Z axes produce the arc.
- G19 -- Y and Z axes produce the arc.

15.5.3 Programming the Arc

To produce an arc, the control needs to know, or be able to calculate the:

- center of the arc
- end point of the arc

The start point of the arc is always the current position of the axes.

There are a number of ways to program a block that produces an arc. We illustrate each of the ways in the following sections.

15.5.4 Standard Circular Interpolation

The block formats in this section produce arcs in each of the 3 planes. They do not use polar positioning (R, A, or C words).

You have 2 choices with standard circular interpolation:

- program the center and the end point

  This produces an arc of any length up to a full circle. By programming the end point, the control knows to stop at a certain position. To program a full circle with this format, the start point (current position of the axes) and the end point of the arc must have the same value.

- program the center only

  This always produces a full circle. By not programming the end point of the arc, the control assumes the end point is the same as the start point, and a full circle results.

Center and End Point Arcs

In absolute mode, positions are referenced to Program Zero. So, you can program the center and the end point of the arc directly with their coordinate values.

In incremental mode, positions are referenced to the current position of the axes. So, program the center as the distance and direction from the start point to the center. Program the end point as the distance and direction from the start point to the end point.

You can mix absolute and incremental modes by using the appropriate G90 and G91 codes.

Remember, the start point of an arc is always the current position of the axes.

Table 15.C shows which words you use to program arcs in each of the 3 planes.

Table 15.C
Block Formats for Standard Arcs in All Planes

If the plane is:	program the end point with:	program the center with:	for example:
XY (G17)	X and Y	I and J	G02 XYIJ #
XZ (G18)	X and Z	I and K	G02 XZIK #
YZ (G19)	Y and Z	J and K	G02 YZJK #

Figure 15.8 shows a concept view of how the words program the center and the end point in absolute and incremental modes. This figure is for the XY plane only, but the concepts apply to the other planes as well.

Figure 15.8 - Center and End Point Arcs — Concept View

upper case letters = absolute
lower case letters = incremental
= program zero
+ = center point
□ = current location
● = end point

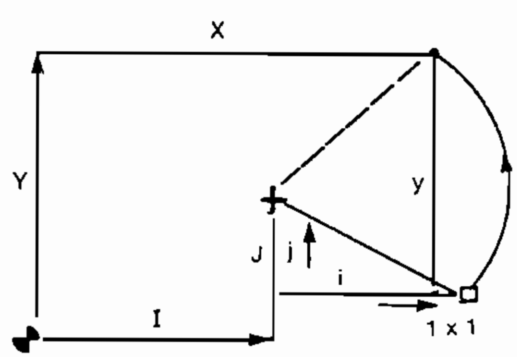
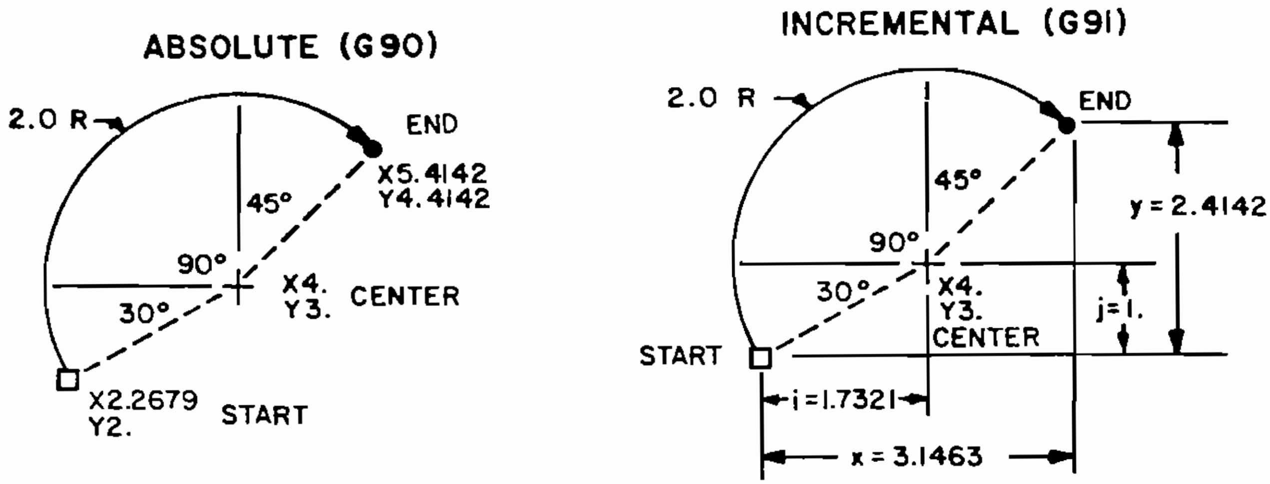


Figure 15.9 gives examples for programming center and end point arcs.

Figure 15.9 - Center and End Point Arc Examples



Center Only Circles

When you program only the center of an arc, the control generates a full circle. The control assumes the end point is the same as the start point when the end point is not programmed.

Table 15.D show the words you use to program the center in each of the 3 planes.

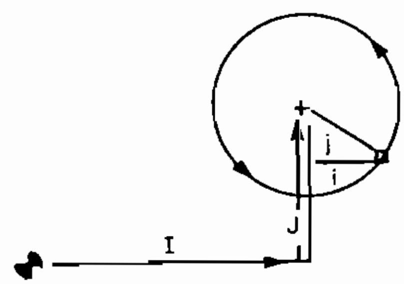
Table 15.D
Center Only Circle Words for the 3 Planes

If the plane is:	program the center with:	for example:
XY (G17)	I and J	G02 IJ #
XZ (G18)	I and K	G02 IK #
YZ (G19)	J and K	G02 JK #

Figure 15.10 gives the concept view of programming a center only circle. This view is for the XY plane only, but the concepts apply to the other planes as well.

Figure 15.10 - Center Only Circles – Concept View

upper case letters = absolute
lower case letters = incremental
= program zero
+ = center point
□ = current location
□ = end point



15.5.5 Polar Arcs

You can program arcs using polar positioning, but only in the XY plane (G17). If you try to execute a polar arc with another plane set, the control will halt and show the message, "IMPROPER CIRCLE AXIS." Make sure the G17 plane is set before programming a polar arc.

Polar arcs use the words R, A, or C in several combinations.

- R -- programs the radius of an arc that begins at the end point of the previous move and is tangent to the line or arc created by that move. R must be set as an absolute word before it can be used in polar arcs.
- A -- programs an angle that originates at the current position of the axes. When it is absolute, A is measured from the positive X axis. When it is incremental, A is measured from the move that just precedes the block in which it is used.

- C -- programs an angle that originates at a center point. When it is absolute, C is measured from the positive X axis. When it is incremental, C is measured from an imaginary line from the center through the current location.

There are 6 combinations of words that form legal polar positioning blocks for arcs. They are:

Legal Polar Arc Block Formats

- R
- RA
- RAC
- RC
- C
- CIJ

Each of these is described below.

R Only Block Format

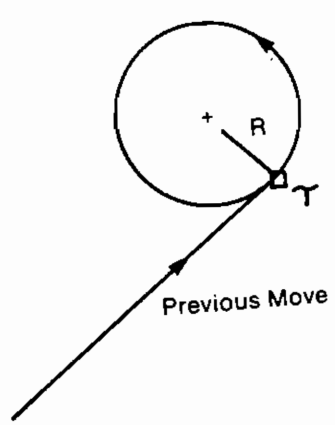
When R is the only motion defining word in an arc block, it specifies a full circle. The circle has a radius of R, and begins at the end point of the previous move and is tangent to the line or arc created by that move.

R must be set to absolute before the block is executed. A non-zero move must precede the block, or the error message, "REFERENCE LENGTH = 0" will occur.

Figure 15.11 shows the concept view for the R only polar arc format.

Figure 15.11 - R Only Block Format -- Concept View

upper case letters = absolute
+ = center point
□ = current location
□ = end point
ϒ = point of tangency



RA Block Format

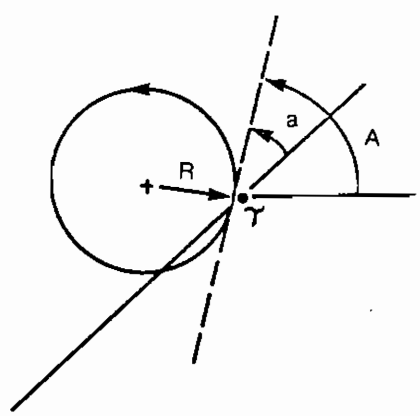
When R and A are the only motion defining words used in the arc block, the result is a full circle. The circle has a radius of R, and the circle begins at the end point of the last programmed move. The circle is tangent to a line specified with the angle A.

R must be absolute. A non-zero move must precede the arc block. The a word can be absolute or incremental. When A is absolute, the angle of the line to which the circle is tangent is measured from the positive X axis. When A is incremental, the angle is measured from an imaginary extension of the move that precedes the arc block. The vertex of A is at the end point of the previous move.

Figure 15.12 shows the concept view for the RA polar arc format.

Figure 15.12 - RA Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
+ = center point
● = current location
● = end point
𝒯 = point of tangency



RAC Block Format

When R, A, and C are the only motion defining words in the polar arc block, they specify an arc of any angle up to a full circle.

The arc has radius R, and it starts at the end point of the last programmed move. The arc is tangent to a line specified by the angle A. The size of the arc is specified by C.

R must be absolute, and a non-zero move must precede the arc block.

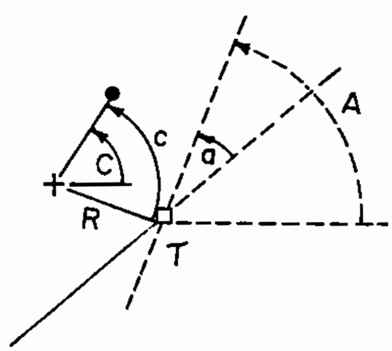
The angle of the line to which the arc is tangent is programmed with A. The vertex of this angle is the end point of the move that precedes the arc block. Angle A can be absolute or incremental. When A is absolute, it is measured from the positive X axis. When A is incremental, it is measured from a line tangent to the end point of the move that precedes the arc block.

The length of the arc, angle C, has its vertex at the center of the arc. Angle C can also be absolute or incremental. When C is absolute, it is measured from the positive X axis extending from the center. When C is incremental, it is measured from a line that extends from the start point of the arc to the center.

Figure 15.13 gives the concept view of the RAC polar arc format.

Figure 15.13 - RAC Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
+ = *center point*
□ = *current location*
● = *end point*
$\mathcal{T}$ = *point of tangency*



RC Block Format

When R and C words are used for polar arcs, an arc of any size up to a full circle is the result.

The arc has radius R, and it is C degrees in length.

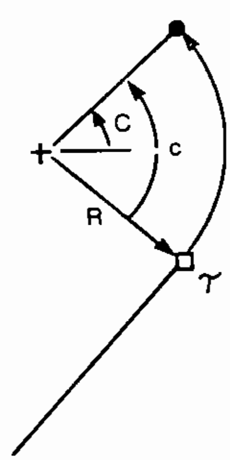
R must be absolute, and a non-zero move must precede the arc block. The arc is tangent to this move at its end point.

The length of the arc, angle C, has its vertex at the center. Angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis extending from the center. When C is incremental, it is measured from the line that extends from the center to the end point of the previous move.

Figure 15.14 shows the concept view for the RC polar arc format.

Figure 15.14 - RC Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
+ = *center point*
□ = *current location*
● = *end point*
Υ = *point of tangency*



C Only Block Format

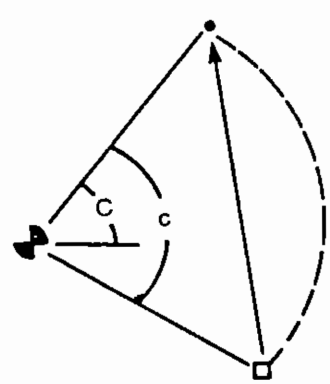
When C is the only motion defining word programmed in an arc block, it produces an arc that is C degrees in length. The center of the arc is always the last Polar center defined. If not defined then the vertex is at Program Zero. Therefore, the radius of the arc is the length of a line from the current position of the axes to Program Zero.

The angle C can be absolute or incremental. When C is absolute, it is measured from the positive X axis extending from Program Zero. When C is incremental, it is measured from the line that extends from the current position of the axes to Program Zero.

Figure 15.15 gives the concept view of the C only polar arc format.

Figure 15.15 - C Only Block Format – Concept View

upper case letters = absolute
lower case letters = incremental
+ = *program zero*
□ = *current location*
● = *end point*



CIJ Block Format

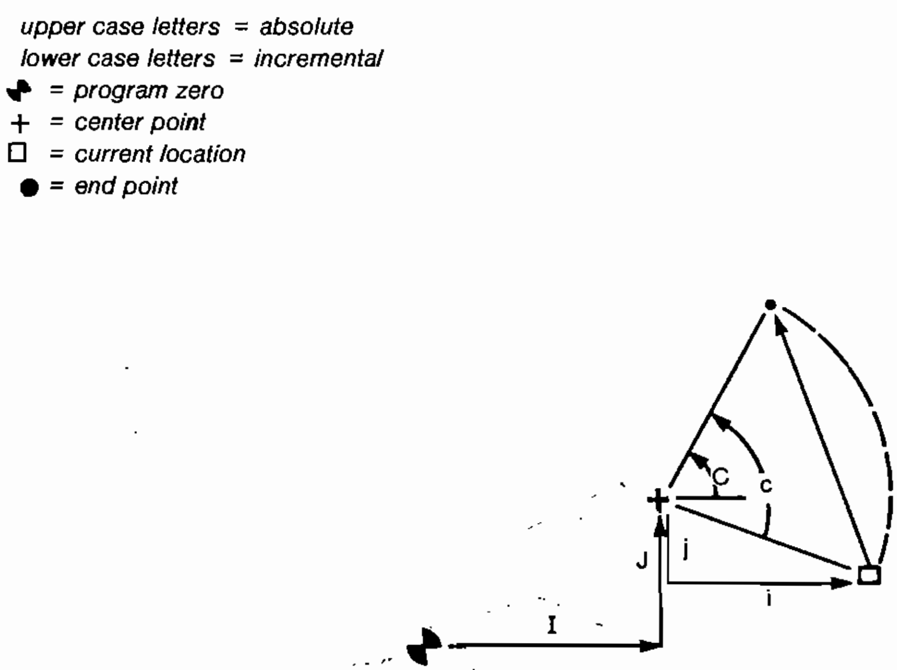
When C, I, and J are used for the polar arc block, an arc of C degrees is the result. The radius of the arc is the distance from the current position of the axes, to the center point as defined by I and J.

I and J program the center point of the arc. They can be absolute or incremental. When they are absolute, the center is a point measured from Program Zero. When they are incremental, they give the distance and direction from the start point of the arc to the center point.

The length of the arc, angle C, has its vertex at the center. It can also be absolute or incremental. When C is absolute, it is measured from the positive X axis extending from the center. When C is incremental, it is measured from a line that extends from the start point of the arc to the center.

Figure 15.16 give the concept view for the CIJ polar arc format.

Figure 15.16 - CIJ Block Format – Concept View



15.6 Dwell Time -- G04

A G04 block programs a dwell time. The length of the dwell in seconds is defined with F word (format 3.2) that appears in the G04 block. This dwell time is modal. It is used for any subsequent G04 blocks in the program that do not contain another F word.

If you wish to change the dwell time, you can specify a new F word in another G04 block.

For example:

```
.
.
G4 F10.#    -- executes a dwell of 10 seconds
.
.
G4#         -- executes a dwell of 10 seconds
.
.
G4 F5.#     -- executes a dwell of 5 seconds
.
.
G4#         -- executes a dwell of 5 seconds
.
.
```

While the G04 block is executing, the dwell time counts down from its initial value to zero. You can see the countdown on the Status page of the control.

A dwell may be necessary in some machining operations, or it can be used to give the operator time to read a part program comment on the Status page. If this is the case, the comment block should immediately follow the G04 block.

The G04 does not change the status or function of the control except to insert a pause of the specified time. After the dwell time is complete, the control continues execution with the next block.

15.6.1 Prompts

The table below explains the prompts that the control gives for G04.

Table 15.E

Prompt	Function
G4--DWELL	
SECONDS--(F)	"F" specifies the amount of time to dwell in seconds. The format for F is 3.2. This amount remains in effect for G04 blocks that come after it, unless a new dwell is programmed.

15.7 Chapter Summary

This chapter covered the programming features of G word data blocks that produce axis positioning along straight lines or along curved paths. Programming dwell times was also covered.

The next chapter covers G word data blocks that program axis modes: such things as axis mirrors, and cutter radius compensation.

Chapter 16 -- Axes Modes

16.0 Chapter Overview

This chapter describes the G word data blocks of the Axes Modes group. After reading this chapter you will know:

- how to select the plane for circular interpolation and cutter compensation
- how to program and cancel axis mirroring
- how to program cutter radius compensation
- how to select fixture offsets

16.1 Plane Selection G17, G18, G19

The words G17, G18, and G19 let you select the plane in which circular interpolation (including the arc move of helical interpolation) and cutter radius compensation occur.

- G17 -- selects the XY plane
- G18 -- selects the XZ plane
- G19 -- selects the YZ plane

The control assumes G17 (XY plane) at power-up and at the end of a part program (after the M02 or M30 in a main program).

Each word is modal. Once you establish a word from this group, it remains in effect until the control executes another word from this group, or executes the M02 or M30 in a main program to set G17 (XY plane) again.

You can program an N word in the data block with G17, G18, or G19. Words other than N are not allowed, and produce the error message " ____ CODE NOT KNOWN" when encountered by the control. Execution will halt.

If the control encounters a circular interpolation data block that is not in the selected plane, it halts execution and displays the error message "IMPROPER G02/G03 PLANE."

Once you establish the plane for cutter radius compensation and turn compensation on through G41 or G42, simultaneous linear interpolation moves are limited to the selected plane. For example, if the G17 plane is active, and you program an XZ or YZ move (or an X or Y move with a U, V, or W) in a block, an error message "IMPROPER CRC PLANE" will occur and execution will halt.

16.2 Axis Mirror G30, G31

The words G30 and G31 are used to cancel and establish "mirroring" of axis moves. By mirroring, we mean that the direction you program for an axis is the reverse of the normal operating direction when the control executes a move. For example, if you program mirroring for the X axis, then program a positive X move, the resulting move is in the negative X direction. Note that with G31 active, the control mirrors programmed axes motions about the program zero point.

- G30 -- cancels mirroring

  The control assumes G30 on power-up and after the M02 or M30 in a main program. It restores the normal operating directions for the axes and cancels G31. A G30 without axis words in the same data block cancels all mirroring for all axes. A G30 with axis words cancels only the mirroring for the axes you specify. For example, G30 X # cancels mirroring for the X axis only.

- G31 -- axis mirror

  A G31 data block reverses the sense of direction for the axes specified in the block. For example, G31 X Y # sets mirroring for the X and Y axes only, and the remaining axes retain their normal operating directions. While mirroring of an axis is active, the Status page shows this with "M" (Mirror) in reverse video to the right of the axis position display. G31 is modal, and remains in effect until cancelled with G30, or until the M02 or M30 of a main program is executed.

16.3 Cutter Radius Compensation G40, G41, G42

Cutter radius compensation lets you offset part dimensions by tool radius values. The tool radius value can be an increment of tool diameter to compensate for an oversized or undersized cutter in the program. It can also be the measured diameter of the tool. In this case, you program the part dimensions and let the control directly compensate for the tool radius.

You enter tool diameter and length values under an offset number (1 to 48) on the Tool Offsets page of the control. The length and diameter values of the tool are applied in a program with an O word. The numeric value of the O word corresponds to the entry on the Tool Offsets page. The control halves the diameter value to derive the cutter radius compensation value.

If the diameter value from the Tool Offsets page is negative (-), the control compensates inside the originally programmed tool path. If the value is positive, the control compensates outside the original path. Thus, if you program a part for a tool of a specific diameter, the path can be adjusted for oversized tools (positive entry), undersized tools (negative entry), or for tool wear (negative entry).

Compensation is applied only to movements made at feedrate (G01, G02, G03, G73). Compensation is automatically removed for rapid traverse (G00) moves. When rapid traverse moves conclude, the control restores compensation by interpolating the radius offset into the next programmed feedrate move in the axes of the selected plane.

16.3.1 Cancel Radius Compensation – G40

The word G40 cancels G41 and G42 (cutter radius compensation left and right, respectively). G40 is modal and is the mode assumed by the control at power-up and at the end of a program.

A G40 in a block by itself will cancel compensation, and the active compensation value will be moved out during the first axis move (or dummy move) that follows the G40 block. This move (that removes compensation) must occur before any G word data block that is not a movement instruction, or before an O word data block.

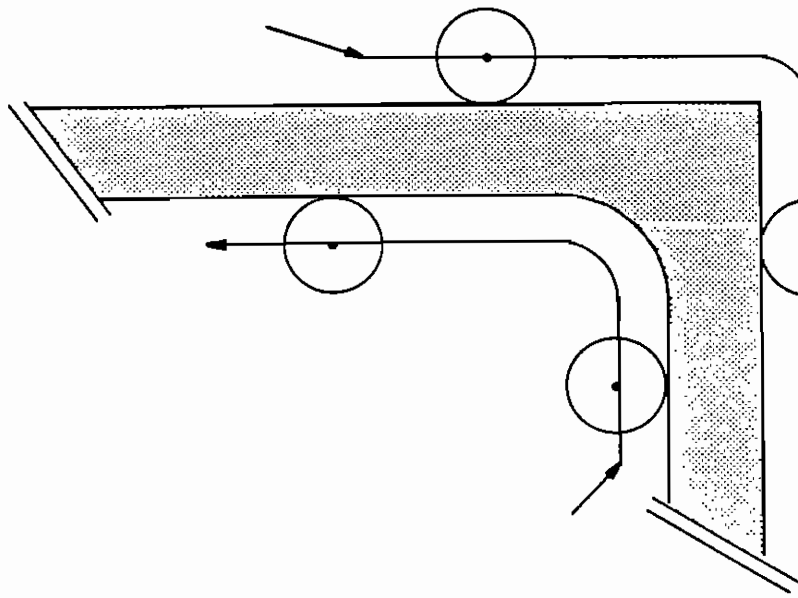
A G40 block may include axis entries that specify a finishing move along the compensated cutter path. This finishing move is a look ahead entry to which initial compensation is correctly applied (as if the move were to be executed), but compensation is cancelled prior to the moves actual execution. The primary intent of this finishing move is to provide the closing compensation move from the end of the final programmed move to the beginning of the initial programmed move of a closed path.

See the examples in section 16.3.6.

16.3.2 Cutter Left -- G41

The G41 word selects cutter radius compensation with the tool assumed to be left of the part being machined, as viewed in the direction that the tool is traveling.

Figure 16.1 - G41--Cutter Left



The G41 data block may include axis entries that specify a position from which you assume the cutter moved to reach its current position. You choose the coordinates of this assumed position. This simplifies the process of applying compensation to subsequently programmed moves. Compensation is applied with appropriate movements of the axes in the selected plane. Thus, following execution of the G41, the cutter is in the position it would be in if it had made a compensated move from the point specified in the G41 block. Programming G41 and an assumed start point like this greatly simplifies setup for entry into a pocket.

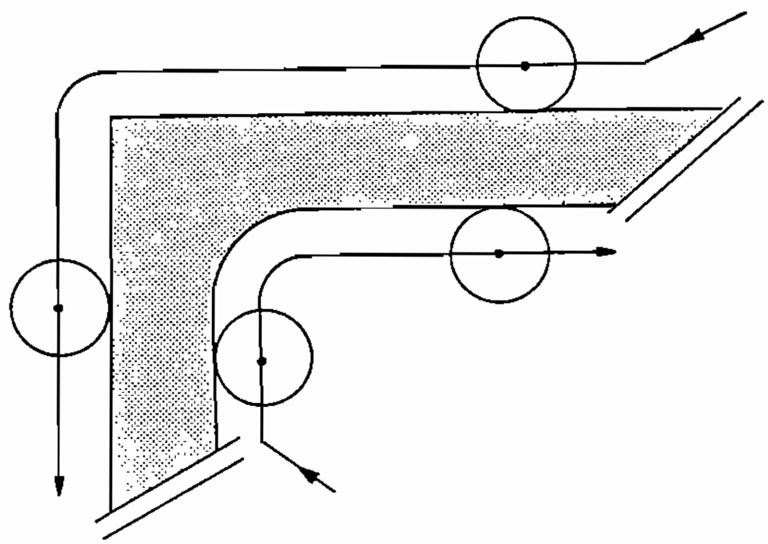
When G41 is programmed in a block by itself, compensation is interpolated into the next programmed axis move. Therefore, we recommend that this move occur while not cutting metal.

G41 is modal and cancels G40 and G42. See the examples in section 16.3.6

16.3.3 Cutter Right – G42

The G42 word selects cutter radius compensation with the tool assumed to be right of the part being machined, as viewed in the direction that the tool is traveling.

Figure 16.2 - G42 – Cutter Right



The G42 data block may include axis entries that specify a position from which you assume the cutter moved to reach its current position. You choose the coordinates of this assumed position. This simplifies the process of applying compensation to subsequently programmed moves. Compensation is applied with appropriate movements of the axes in the selected plane. Thus, following execution of the G42, the cutter is in the position it would be in if it had made a compensated move from the point specified in the G42 block. Programming G42 and an assumed start point like this greatly simplifies setup for entry into a pocket.

When G42 is programmed in a block by itself, compensation is interpolated into the next programmed axis move. Therefore, we recommend that this move occur while not cutting metal.

G42 is modal and cancels G40 and G41. See the examples in section 16.3.6

16.3.4 Plane Selectable Compensation

The control lets you program cutter radius compensation for moves in any of 3 planes. You should program the plane in which compensation occurs (using G17, G18, or G19) before activating the compensation (wth G41 or G42).

- G17 -- compensation in the XY plane

  This is the power up mode of the control and does not need to be programmed if you have not programmed a different plane since power-up or the last executed M02 or M30 of a main program. G17 cancels G18 and G19.

- G18 -- compensation in the XZ plane

- G19 -- compensation in the YZ plane

When G41 or G42 is active, moves of any and all axes can be programmed. However, only those axes within the selected plane have their moves compensated. Axes that are not part of the selected plane respond directly to their programmed moves.

While compensation is active, you cannot combine the moves of axes in the selected plane with those not in the plane in the same data block. For example, if G17 (XY plane) is active, a data block such as:

```
G1 X1. Z1. #
```

is not allowed while cutter compensation is in effect. The control will generate an error when it encounters such a data block.

16.3.5 Compensation Restrictions

Once the control executes a G41 or G42 data block, certain G words are not allowed:

- G17, 18, G19 -- plane selection

- G22, G23 -- helical interpolation

- G30, G31 -- axis mirror

- G92 -- preset program zero

- G99 -- preset cancel

If the control encounters these G words, an error message "______ IN CUTTER COMP" occurs and execution halts. Use G40 to cancel compensation before programming these operations.

You must also cancel compensation with a G40 before changing between G41 and G42, or calling for a new O word tool offset.

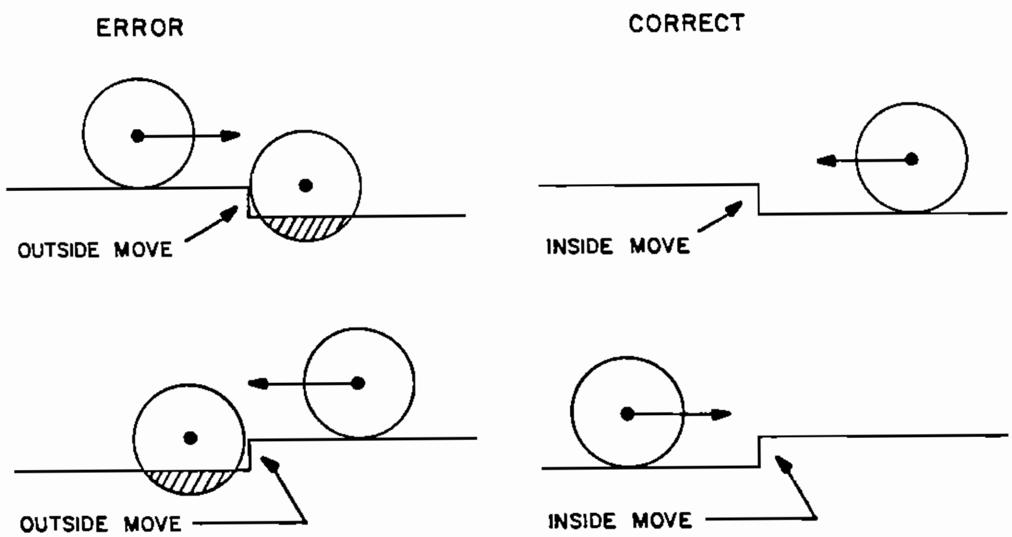
If you program a move in the selected plane that results in no net movement (i.e., a zero length move), the control generates the error message "BAD CRC PLANE AXIS" and halts execution.

If you need to perform any of the functions discussed above, first cancel compensation with a G40, then re-establish the cutter compensation mode you want with G41 or G42.

A final consideration in cutter radius compensation is the look ahead requirements of the control. The control examines the data block that is one ahead of the currently executing block. It checks to see if the tool will be tangent to the programmed path, and whether the tool is too large to produce the desired contour. In general, the control will not compensate an outside move that is less than the cutter radius if this move is followed by an inside move that is greater than the cutter radius.

Important: An "inside" move is one in which the tool turns toward its position relative to the part. For example, if the tool is traveling left of the part (G41) and turns left, it has made an inside move. An "outside" move is one in which the tool turns away from its position relative to the part. For example, if the tool is traveling left of the part (G41) and turns right, it has made an outside move. The control will compensate an inside move that is followed by an outside move.

Figure 16.3 - Look Ahead Restrictions



16.3.6 Compensation Examples

To properly set up, or remove, cutter compensation you can use one of 3 techniques:

- entrance move -- a programmed move of one or both axes in the selected plane that is equal to or more than the cutter radius (half the diameter value specified on the Tool Offsets page) that brings the tool to the start of the path.

- dummy move -- a programmed location of one or both axes in the selected plane that you use only to invoke compensation. The dummy move produces movement only because it invokes or removes compensation. When you invoke compensation, the dummy move tells the control where the tool is, uncompensated. By looking ahead to the next move, the control calculates how to properly set up compensation. When you remove compensation, the dummy move gives the control a look ahead move (to prevent notching the part, for example).

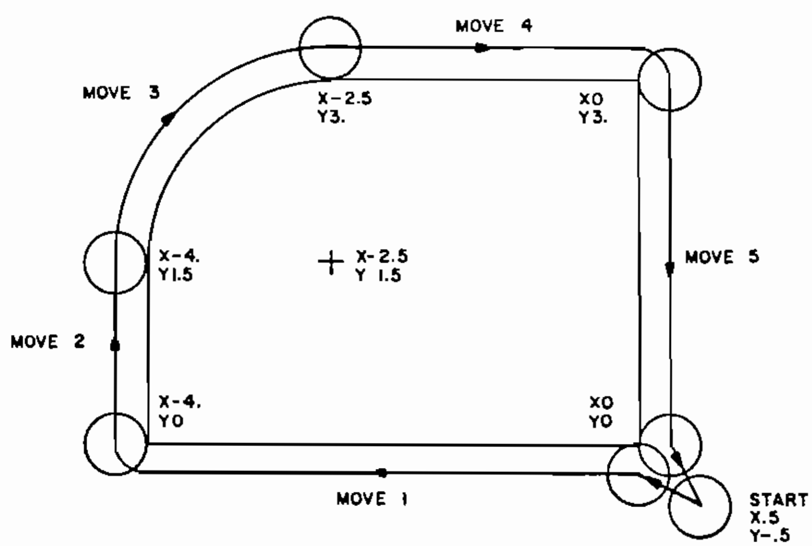
- look ahead start move -- a programmed location of one or both axes in the selected plane. You specify this location in the G41 or G42 block to tell the control where the tool might have come from to be properly positioned for the next move. In the G40 block, the look ahead start move tells the control how to finish the contour if compensation were to remain in effect.

The following examples illustrate how to use each of these techniques.

Entrance Move

Figure 16.4 and the following example program illustrate using an entrance move to invoke compensation.

Figure 16.4 - Entrance Move Compensation Example



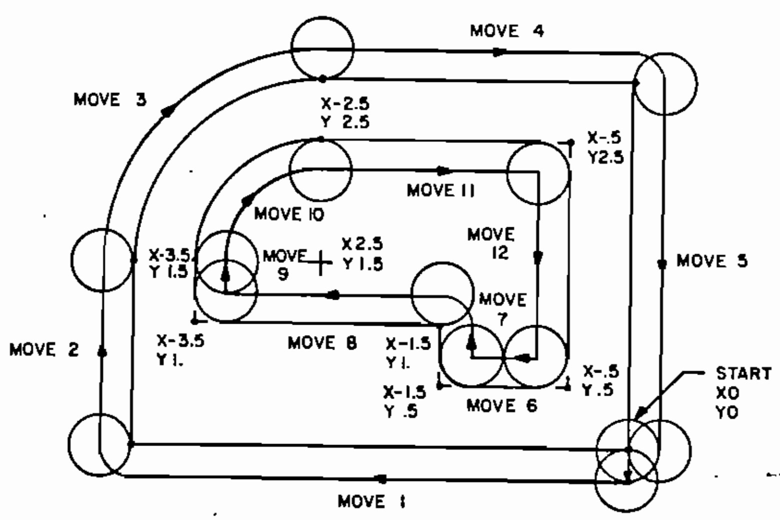
Data Block	Comment
;ENTER MOVE#	Program name comment block
G99#	Cancel any previus position preset
G90#	Establish absolute mode
G00 X-10. Y-5. Z-.5 #	Rapid to start point
G92 X.5 Y-.5 Z0 #	Set program zero
O1 #	Call tool offset 1 from the Tool Offsets page, 0.5000 in. diameter.
G41 #	Call for cutter compensation with the tool traveling to the left of the part path. Compensation is assumed to be in the XY plane (G17).
G01 Z-1. F10. #	Lower tool to working depth at 10 ipm; compensation is ignored.
X0 Y0 F20. #	Entrance move to the part contour that is greater than the tool radius, compensation is interpolated into this move; feedrate is 20 ipm.
X-4. #	Move 1
Y1.5 #	Move 2
G02 X-2.5 Y3. I-2.5 J1.5 #	Move 3
X0 #	Move 4
Y0 #	Move 5
G00 Z0 #	Rapid Z up from work.
G40 #	Cancel compensation; the compensation value is removed with the next XY move.

X.5 Y-.5 #	Tool returns to the start position at rapid and compenstaion is remove with this command.
G99 X0 Y0 Z0 #	Cancel position preset and return to Machine Zero location at rapid.
M02 #	End of program

Important: A special situation exists for rapid traverse moves. The tool position at the end of a rapid traverse move is the uncompensated programmed position unless the next move is a feed move. If the next move is a feed move, the tool goes directly, at rapid traverse, to the proper compensated position for the start of the feed move. This is true whether the tool is currently at either a compensated or uncompensated location.

Dummy Move The following example and figure 16.5 illustrate using a dummy move to invoke and remove cutter compensation.

Figure 16.5 - Dummy Move Compensation Example



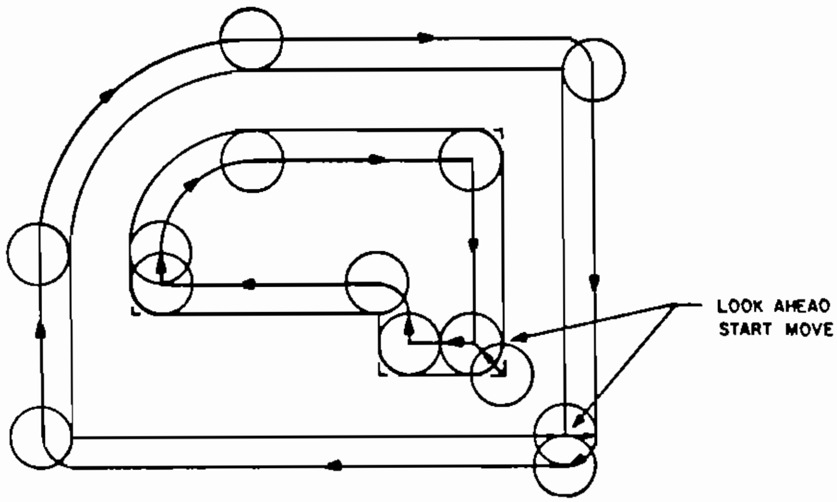
Data Block	Comment
;DUMMY MOVE#	Program name comment block
G99#	Cancel any previus position preset
G90#	Establish absolute mode
G00 X-10. Y-5. Z-.5 #	Rapid to start point
G92 X.5 Y-.5 Z0 #	Set program zero
01 #	Call tool offset 1 from the Tool Offsets page, 0.5000 in. diameter.
G41 #	Call for cutter compensation with the tool traveling to the left of the part path. Compensation is assumed to be in the XY plane (G17).
G01 Y0 #	"Dummy move" to invoke compensation for the outside contour of the part. The tool moves to Y-.25 with this command in preparation for the next programmed X move. A dummy move must be programmed at feedrate.
Z-1. F10. #	Lower tool to working depth at 10 ipm; compensation is ignored.
X-4. F20. #	Move 1, 20 ipm
Y1.5 #	Move 2
G02 X-2.5 Y3. I-2.5 J1.5 #	Move 3
X0 #	Move 4
Y0 #	Move 5
G00 Z0	Rapid Z up from work.

G40 #	Cancel compensation; the compensation value is removed with the next XY move.
X-.75 Y.75 #	Rapid to start point of pocket.
G42 #	Call for compensation with the cutter traveling to the right of the programmed path.
G01 Z-1. F10. #	Lower tool to working depth at 10 ipm.
X-1.5 F20. #	Move 6, 20 ipm
Y1. #	Move 7
X-3.5 #	Move 8
Y1.5 #	Move 9
G02 X-2.5 Y2.5 I-2.5 J1.5 #	Move 10
X-.5 #	Move 11
Y.5 #	Move 12
X-.75 #	Dummy move to prevent the tool from notching the part. This move is equal to the tool radius and is a look ahead move that stops the cutter before it completely executes the full length of move 12, above.
G00 Z0 #	Rapid Z up from work.
G40 #	Cancel compensation; the compensation value is removed with the next X and Y moves.
X0 Y0 #	Rapid to start point.
G99 X0 Y0 Z0 #	Cancel position preset and return at rapid to Machine Zero.
M02 #	End of program

Look Ahead Start

Figure 16.6 and the following example illustrate using look ahead start moves to invoke and cancel cutter compensation.

Figure 16.6 - Look Ahead Start Compensation Example



Data Block	Comment
;LOOK AHEAD #	Program name comment block
G99#	Cancel any previus position preset
G90#	Establish absolute mode
G00 X-10. Y-5. Z-.5 #	Rapid to start point
G92 X.5 Y-.5 Z0 #	Set program zero
01 #	Call tool offset 1 from the Tool Offsets page, 0.5000 in. diameter.

G41 Y3. #	Call for cutter compensation with the tool traveling to the left of the part path and specify a look ahead start move of Y3. This entry describes the previous location of the tool. The control assumes this move to have occurred with compensation active, although this entry actually invokes compensation. This entry gives the coordinate of the last assumed position in absolute (G90) or the dimension to the last position in incremental (G91). It could have any length desired, since it simply tells the control how to enter compensation properly for the next move.
G01 Z-1. F10. #	Lower tool to working depth at 10 ipm; compensation is ignored.
X-4. F20. #	Move 1, 20 ipm
Y1.5 #	Move 2
G02 X-2.5 Y3. I-2.5 J1.5 #	Move 3
X0 #	Move 4
Y0 #	Move 5
G40 X-4. #	Cancel and remove compensation value with a look ahead move to an assumed compensated position. The move is not actually performed but compensation is appled initially to simplify the finshing of the compensated path.
G00 Z0 #	Rapid Z up from work.
X-.5 Y.5 #	Rapid to start point of pocket.

G42 Y2.5 #	Call for compensation with the cutter traveling to the right of the programmed path and specify a look ahead start move to enter compensation for the pocket. (If G91 mode where active, the axis entry would be Y2., for example.)
G01 Z-1. F10. #	Lower tool to working depth at 10 ipm.
X-1.5 F20. #	Move 6, 20 ipm
Y1. #	Move 7
X-3.5 #	Move 8
Y1.5 #	Move 9
G02 X-2.5 Y2.5 I-2.5 J1.5 #	Move 10
X-.5 #	Move 11
Y.5 #	Move 12
G40 X-1.5 #	Cancel compenstion and remove compensation with a look ahead move to which compensation is applied. This entry prevents the tool from notching the part and correctly finishes the pocket.
G00 Z0 #	Rapid Z up from work.
G99 X0 Y0 Z0 #	Cancel position preset and return at rapid to Machine Zero.
M02 #	End of program

16.4 Fixture Offsets -- G45

A G45 block lets you select a fixture offset from the Fixture Offsets page of the control. Fixture offsets are offsets from machine zero to program zero in the X, Y, Z, or W axes. They can be used to shift the program zero by predetermined amounts.

There are up to 12 fixture offset values (numbered from 1 to 12 on the Fixture Offset page).

Multiple fixture may represent different fixtures for different jobs or they may represent multiple copies of the same job.

The program can explicitly specify a fixture number, or it can sequence through a sequential set of fixtures allowing the operator the ability to determine how many fixture locations are used.

With suitable programming, it is easy to have the machine execute a number of cuts using different offsets with a given tool before indexing to the next tool. This can improve speed and efficiencyby making cuts in several locations before changing tools for each offset. A program written in this manner can still be executed with none, one, or several fixture location without modifying the program.

To explicitly invoke a fixture offset, program G45 followed by O and the number of the fixture offset value. For example, if you program:

```
G45 01 #
```

fixture offset 1 is selected.

Fixture offsets are interpolated into the next programmed movements of the axes. They are fully active when all axes specified in the fixture offset have been moved.

If Last Fixture is selected to 0 by the operator, Fixture Offsets are off and this command has no effect.

To cancel fixture offsets, program:

```
G45 00 #
```

The fixture offsets are removed with the next programmed movements of the axes affected by the fixture offset.

G45# increments to the next Fixture number. If the Fixture number was equal to or greater than Last Fixture (as entered by the operator), the Fixture number goes back to number 1. If the Fixture number was 0, it will remain 0.

G45 with a subroutine call specified by H, E, and/or P except for E only (H only is not a subroutine call) will execute the subroutine first. Then it will execute the equivalent of G45# (incrementing the fixture number if it isn't zero). If the control just incremented the fixture number back to 1 or if the fixture number is 0, it will proceed to the next block, otherwise, the subroutine is called again and the sequence repeats.

G45 with E and no H or P is not a valid command.

If an O is included in a G45 with a subroutine call, the O specifies the beginning fixture number. This has no effect if Last Fixture is set to 0.

G45 with an H and no E or P parameter executes like G45 and then it will execute the branch specified by the H unless the control just incremented the fixture number back to 1 or the fixture number is 0 in which case it boes not branch but instead proceeds with the next block. An O parameter is not allowed in this format.

For example:

```
G4501# (optional, if program wants to force starting on
fixture number one)

N100 Program
          •
          •
          •
G45H100#
M2
```

If the operator wants to run with no fixtures, he can set "Last Fixture" to 0 under Job Setup. If he wants to run only one part in fixture number one, he can set "Last Fixture" to 1 under Job Setup. If he wants to run five parts in fixtures one through five, he can set "Last Fixture" to 5. In this case, the program will loop back to N100 from the G45 until it has done all five fixtures and then it will ignore the branch and go to the next block (M2#).

In any case, no change is required to the program.

This provides a further capability to minimize tool changing on multiple fixture parts. By programming as follows:

```
Using H branching:
      G4501#
      T1
      N100 Program for tool #1
                    .
                    .
      G45H100
      T2
      N200 Program for tool #2
                    .
                    .
      G45H200
      T3
      etc.


Using P,H,E Subroutine Calls:
      ;;Main Program
      G45 01 P5 H100 E200#
      G45 P5 H200 E300#
       .
       .
      etc.

      ;;P5 Subprogram
      N100 T1  Pattern for tool #1
                      .
                      .
      N200 T2  Pattern for tool #2
                      .
                      .
      N300   etc.
```

The control will select tool number 1, execute the portion of the program for tool number 1 on all fixtures up the "Last Fixture" selected by the operator, then change to tool number 2, and execute the portion of the program for tool number 2 on all fixtures, etc. Note that even here the program needs no change for the operator to run with none, one, or several fixtures.

16.5 Chapter Summary

This chapter discussed the Axes Modes group of G word data blocks. You should now know how to use plane selection, axis mirrors, cutter radius compensation, and fixture offsets.

The next chapter covers the G words that program autocycles; repetitive machining cycles that simplify machining tasks.

Chapter 17 -- Autocycles

17.0 Chapter Overview

This chapter covers the G word data blocks in the autocycles group. Autocycles (sometimes referred to as canned cycles) establish repetitive machining operations. You add numeric and parameter information to specify the actions of the cycle.

After reading this chapter, you will know how to:

- program a cavity
- program your own autocycle
- program drilling, counter-boring, and boring autocycles
- program a peck drilling autocycle
- program a tapping autocycle
- cancel autocycles
- execute or restore autocycles even if they have been canceled

17.1 Introduction to Autocycles

The control provides the following autocycles and autocycle control words:

- G29 -- execute last autocycle (see section 18.6). Calls for an execution of the last autocycle that was defined in a program even if it has been canceled with a G80 block.
- G75 -- cavity autocycle (see section 17.2). Defines a profile in the XZ plane, or scales the defined profile along a path. By applying the profile at points along an XY path, you produce an "into work" and/or "out of work" 3 dimensional pattern ("cavity").
- G79 -- programmable autocycle (see section 17.3). Lets you define a subprogram or subroutine to work as an autocycle, that is, executed after rapid XY moves.
- G80 -- cancel autocycle (see section 17.4). Removes the last defined autocycle from active status.

- G81 -- drill autocycle (see section 17.5). Defines simple drilling operations.
- G82 -- counter-bore autocycle (see section 17.6). Defines a drill operation with a dwell time at hole bottom.
- G83 -- peck-drill autocycle (see section 17.7). Defines a deep hole drill cycle with chip break dwell or retract.
- G84 -- tap autocycle (see section 17.8). Defines a right-hand threading cycle. You must provide a floating tap holder if you use this cycle.
- G85 -- bore autocycle (see section 17.9). Defines a drill cycle with feedrate retract.
- G89 -- restore last autocycle (see section 17.10). Restore the last autocycle that was canceled with G80 to active status

Autocycles in the G81 to G85 series define Z axis infeed operations.

Once you define an autocycle in a program, it is automatically performed after rapid (G00) XY moves, or moves divided with a D word. Autocycles are modal and remain active until they are cancelled with a G80 block.

17.2 Cavity Milling - G75

The cavity autocycle produces a convex (out of work) or concave (into work) cavity pattern by repeating a profile along a path. A profile is a sequence of moves that you define in the XZ plane. A path is a sequence of moves in the XY plane.

The cavity is produced by executing the programmed profile, moving along the path, then repeating the profile. The control performs this sequence until the end of the path is reached. Although the profile is programmed in X and Z, it is executed orthogonally to the path in XY.

The profile can be scaled up or down, linearly, along the path to vary the size of the cavity.

17.2.1 Prompts

The table below shows the prompts that the control gives for G75 - Cavity Milling, and a brief explanation of each parameter allowed in G75.

Table 17.A

Prompt	Function
CAVITY---------G75	
X FACTOR-------(X)	"X" programs the scaling factor applied to the X axis in the profile. This scaling factor is applied linearly along the path according to the increments programmed with D in the path. When the path reaches its final destination, this factor will have been fully employed. The resulting scaling will be a mulitiple of any initial scaling in the profile.
Z FACTOR-------(Z)	"Z" programs the scaling factor applied to the Z axis in the profile. This scaling factor is applied linearly along the path according to the increments programmed with D in the path. When the path reaches its final destination, this factor will have been fully employed. The resulting scaling will be a multiple of any initial scaling in the profile.
FIRST POINT----(D)	"D" tells the control to perform the profile at the first point in the path. If D is not programmed, the first point in the path is skipped. In either case, the profile is performed at the end of each D segment programmed in the path, and at the end of the path.
SUBPROGRAM-----(P)	"P" programs the number of the subprogram that contains the profile. If a subprogram is not used, then a subroutine must be specified with H and/or E.
SUB START NO.--(H)	"H" programs the starting sequence number of the subroutine that contains the profile.
SUB END NO.----(E)	"E" programs the ending sequence number of the subroutine that contains the profile.

17.2.2 Planning the Profile

You must program the data blocks that produce the profile in a subprogram or subroutine. The G75 will call the subprogram or subroutine and produce the profile.

Moves that do cutting in the profile should be programmed in the XZ plane. The profile can include any feedrate and scaling information (although scaling won't normally be required in a profile).

The profile should include:

- feedrates for the moves in the profile
  These feedrates don't affect the feedrates used along the path.

- moves in the profile
  These moves should be defined in the XZ plane.

- offset and cutter compensation
  You should program the tool radius offset you want to use in the profile if it differs from the one used in the main program. This compensation value is generally the radius of the end of a ball end mill.

  Compensation for the XZ plane (G19) is automatically assumed.

- initial scaling (not normally needed)
  The profile can include any initial scaling value, programmed with a G72 block, at the beginning of the profile. This scaling is for the X and Z axes. Remember that if the profile includes arc moves, the scaling amount for both axes should be the same. Note that the scaling programmed with the G75 cavity autocycle will be multiplied to the scaling value in the profile.

17.2.3
Ending the Profile

The G75 performs the following each time the profile execution ends:

- returns the Z axis to its start position in the profile
- turns off cutter compensation
- returns the X axis to its start position in the profile at the current feedrate in the calling program

17.2.4
Planning the Path

The path along which the profile is performed can be any valid motion block, or series of motion blocks.

If you program a D word in a motion block of a path, the block is divided into D number of segments. The profile is orthogonal to the path of those points defined by these segments.

If no D is programmed in a motion block of a path, the profile is performed at the end point of the motion block.

Note that profiles are not automatically performed at the first point in motion blocks. You must program a D in the G75 block to perform the profile at the first point in a motion block. Having a profile at the first point is not generally desired unless the cavity starts out of the work or a closed path must be programmed. In these cases, the last point in the last move of the path will perform the profile at the first point of the first move

17.2.5
Planning the Cavity

The G75 block is used to:

- establish the profile

  The G75 block can use the branch words P, H , and E, or any valid combination of these, to call the profile. For example:

  - G75 P3#

  establishes program 3, which contains the moves of the profile.

- G75 H100 E200#

establishes the subroutine that begins with N100 and ends with the block just before N200. This subroutine contains the moves of the profile.

- set scaling applied to the profile along the path

  X and Z words programmed in the G75 block, are scaling factors applied to X and Z words of the profile. The scaling called for in G75 is applied linearly along the path. The scaling starts from some initial value and ends up at a final value programmed with X and Z in G75. A G75, X, and Z block with P, H, or E specifies the initial scaling. A G75, X, and Z block without a P, H, or E specifies the final scaling.

  For example:

  G75 X.2 Z.2 # -- calls for .2 scaling to X and Z moves in the profile applied along the programmed path.

  G1 X5. Y2. # -- the programmed path, scaling starts from the inital value at the first point in this path and ends with the .2 factor fully applied.

  G75 X3. Z3. # -- scaling to be applied along the next motion block in the path.

  The rules for G75 scaling are the same as those for G72. If the profile contains scaling (G72), the result will be the multiple of the G75 and G72 factors. If your profile contains G2 or G3 arcs, be sure to scale both X and Z by the same amount.

- determine whether to execute the profile at the first point of the path

  The control will normally execute a profile at segment points along the path, and at the last point of any motion block. A D word programmed in G75 tells the control to execute the profile at the first point of a motion block that is before the first move segment, as well as at the segment points and last point.

17.2.6 Cancelling G75

G75 is canceled with a G80 block, just as any other autocycle. Program the G80 in a block by itself at the point in the program where you no longer want the profile performed. Autocycles canceled with a G80 can be restored with a G89 (See Section 17.10) or forced to execute with a G29 (See Section 18.6).

17.2.7 Example

The following example illustrates the use and programming of the G75 cavity autocycle. See figure 17.1

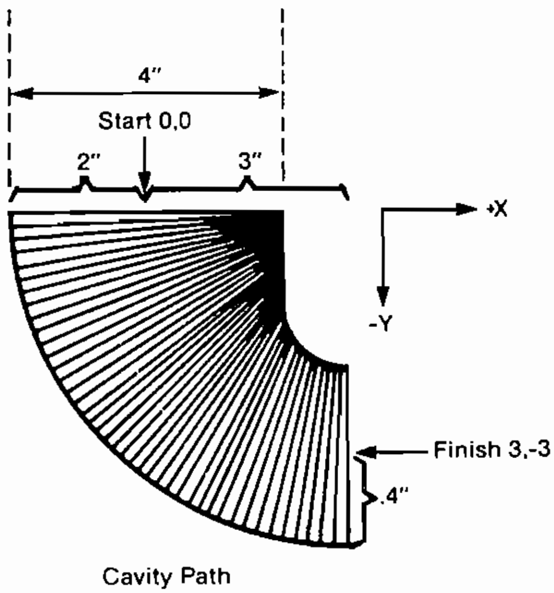
Data Block	Comment
;CAVITY PATH #	-- names the program. This program defines the path along which the profile will be applied.
G99#	-- cancels any previous position preset
GOO XYZ #	-- moves the axes to machine zero at rapid (assumed somewhere in the middle of machine travel and at the desired start point of the program)
G91 #	-- establish incremental mode
G75 P2 #	-- establishes cavity milling autocycle. The program in storage whose number is 2 is considered to be the subprogram that contains the moves of the profile. Initial scaling is 1.
G75 X.2 Z.2 #	-- calls for a 0.2 scaling factor to be applied to the X and Z moves in the profile. The scaling is applied linearly along the programmed path.

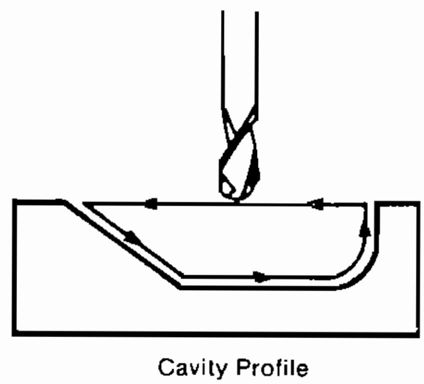
Data Block	Comment
G3X3.Y-3.I3.J0D40#	-- programs a counterclockwise 90° arc, the path. The value of D determines the number of equal segments along the path, in this case 40. The profile is executed at the end point of each of these segments.
G80 #	-- cancels G75
G90 #	-- establish absolute mode
G00 XYZ #	-- rapid back to machine zero, the start point of the path.
M2 #	-- end of program

Data Block	Comment
;PROFILE P2 #	-- names the subprogram that contains the profile. This is assumed to have a program number of 2 in storage.
G91 #	-- establishes incremental mode
G423 #	-- cutter comp right is ZX plane
G00 X-2. #	-- rapid move of X-2.
G1 Z-2.5 X2. F20.#	-- establishes linear interpolation. A move of Z-2.5 and X2. at 20 ipm.
X.5 F10.#	-- a move of X .5 at 10 ipm.
G2 X1.5 Z1.5 I0 K1.5	-- performs a clockwise 90° arc
F10. #	-- at 10 ipm.
Z1. F35. #	-- a move of Z 1. at 35 ipm.
G00 X-2. #	-- rapid move of X-2. This is rapid return to the start point of the profile.

Data Block	Comment
M2 #	-- end of the profile subprogram. cutter comp (G40) automatically done by cavity

Figure 17.1 - Cavity Example





17.3 Programmable Autocycle - G79

A G79 block specifies a subprogram or subroutine to act as an autocycle. It is executed in response to rapid (G00) moves or blocks with the D parameter within a program. The specific actions of the G79 autocycle are determined by the codes used within the subprogram or subroutine of the cycle.

17.3.1 Prompts

Table 17.B describes the prompts that the control gives for G79.

Table 17.B

Prompt	Function
PROGRAMMED ---- G79	
SUBPROGRAM--(P)	"P" specifies the number of a program (1 to 250) in storage. The program contains the blocks that perform the autocycle action. If you don't specify P, you must specify H and/or E.
SUBSTART NO.--(H)	"H" specifies the starting sequence number (N word) of a subroutine.
SUBEND NO.--(E)	"E" specifies the ending sequence number of a subroutine.

All the rules of programming subprogram and subroutine calls apply to the G79 autocycle. Nesting (up to 5) is allowed. That is, the subprogram or subroutine called by the G79 can in turn call another autocycle, even another G79. See chapter 13.

17.3.2 *Example*

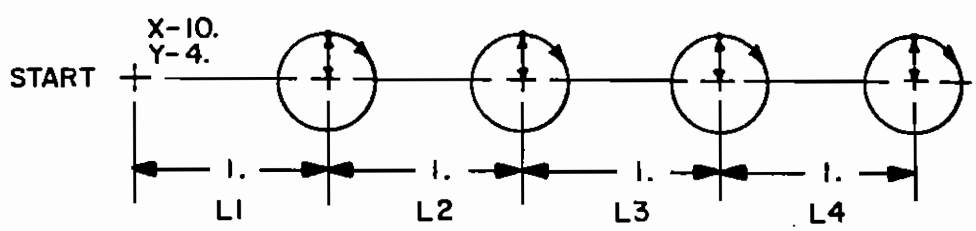
The following example illustrates how to use the G79 autocycle. Also see figure 17.2.

In this example, the main program (;MAIN #) rapids from machine zero to X-10. Y-4. Z-.5. This point is established as program zero. The G79 autocycle is programmed and a subprogram (;CYCLE # -- program number 2 in storage) contains the moves that will perform the action of the autocycle at subsequent rapid positions in the main program. The action of the subprogram autocycle is to feed to depth, move to the perimeter of a 0.25 inch circle and then return at rapid to the start position of the cycle. This action is produced 4 times along the X axis (in the main program). The cycle then is cancelled and the axes return to machine zero.

Data Block	Comment
;MAIN #	-- comment block specifying the name of the main program
G99 #	-- cancel any previous position preset
G90 #	-- establish absolute mode for all axes
G00 X-10. Y-4. Z-.5 #	-- rapid to X-10. Y-4. Z-.5 from machine zero
G92 X0 Y0 Z0 #	-- establish position preset X0 Y0 Z0 - program zero
G79 P2 #	-- programmable autocycle performed after rapid moves in the main program; action of the cycle is contained in program 2 in storage
G91 X#	-- establish X axis as incremental for the main program
X-1. L4 #	-- rapid X-1. four times; the autocycle will be performed after each rapid move
G80 #	-- cancel autocycle
G90 #	-- establish absolute mode for all axes
G99 X0 Y0 Z0 #	-- cancel position preset and return at rapid to machine zero
M02 #	-- end of program

Data Block	Comment
;CYCLE #	-- comment block with program name for program 2 in storage - the subprogram used for the G79 autocycle
G91 #	-- establish incremental mode for moves in the subprogram
G01 Z-.6 F5. #	-- feed to depth at 5 ipm
Y.25#	-- move to edge of circle from center
G02 X0 Y0 I0 J-.25	-- full circle center after completing circle
G00 Z.6 #	-- rapid Z up from work
Y-.25 #	-- rapid to circle center, the start point of the subprogram
M02 #	-- end of subprogram

Figure 17.2 - G79 Example



17.4 Cancel Autocycles - G80

A G80 block cancels a currently active autocycle. Program this code when you do not want an autocycle performed at the end of rapid moves, or blocks that include the D word.

Autocycles are automatically cancelled at the end of the program in which they are invoked.

When you cancel an autocycle, details of its action are retained within the control so that it can be executed in response to a G29 word (section 18.5), or restored to active status with a G89 word (section 17.10).

17.5 Drill Autocycle - G81

A G81 block can be used for simple drilling operations. The tool rapids to the R-plane, drills at feedrate to depth, performs an optional dwell time, and rapids or feeds up to the R-plane or initial tool position.

G81, and G82 all have the same words at their disposal for programming in their blocks. They are provided as separate codes in order to conform more closely to EIA RS-274 specifications, but their specific actions can be identical.

As with all autocycles, G81 is modal. It remains in effect until canceled with a G80 data block, or superceded by another autocycle, and is automatically performed after subsequent rapid (G00) moves or moves divided with the D word.

17.5.1 Prompts

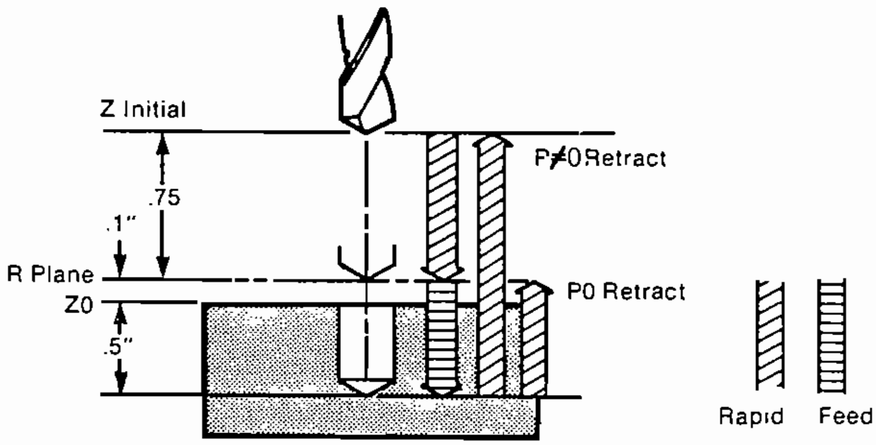
The following table describes the prompts that the control gives for the G81 autocycle.

Table 17.C

Prompt	Function
DRILL----G81	
DEPTH--Z	"Z" is required for the G81 autocycle. It programs the Z coordinate of maximum infeed (negative Z movement) when Z is absolute (G90). It programs the distance below the R-plane when Z is incremental (G91).
DWELL SECONDS--(D)	"D" is optional. It specifies the dwell time amount in seconds (0.01 to 99.99) at hole bottom.
RAPID PLANE--(R)	"R" is optional. It specifies the Z position of the R-plane. R is always an absolute Z axis position. The metal cutting operation of the autocycle occurs from the R-plane position to the depth specified with Z. All autocycles rapid down for the position of the R-plane from the initial tool position when the autocycle is performed. If you do not specify R in the autocycle, the control uses the current Z position.
INFEEDRATE--(F)	"F" programs a feedrate for infeed from the R-plane to depth. If not programmed, the last used F word is assumed.
RETRACT CODE--(P)	"P" programs the retract position of the tool between executions of the autocycle. If P is not programmed, or is programmed as zero, the tool retracts to the R-plane during XY moves that invoke the cycle. If P is programmed as non-zero (1 to 250), the tool retracts to the position it had at the time the autocycle was defined.
OUTFEED RATE--(V)	"V" programs the feedrate used for retract from total depth. If V is not programmed, the control assumes that retract should occur at the rapid rate.
X PRE-MOVE--(X)	"X" programs a move in X to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.
Y PRE-MOVE--(Y)	"Y" programs a move in Y to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.

Figure 17.3 - G81 Example

Absolute	***Incremental***
G92 Z.85 #	*G92 Z.85 #*
G90#	*G91*
G81 Z-.5 R.1 P0 #	*G81 Z-.6 R.1 P0 #*



17.6 Counter-Bore Autocycle -- G82

The G82 cycle is provided for counter-boring or drilling operations where a dwell is desired in the workpiece. Its operation and programming are identical to the G81 cycle described in section 17.5. The tool rapids to the R-plane, drills at feedrate to depth, performs an optional dwell time, and rapids or feeds up to the R-plane or initial tool position.

G82 and G81 all have the same words at their disposal for programming in their blocks. They are provided as separate codes in order to conform more closely to EIA RS-274 specifications, but their specific actions can be identical.

As with all autocycles, G82 is modal. It remains in effect until canceled with a G80 data block, or superceded by another autocycle, and is automatically performed after subsequent rapid (G00) moves or moves divided with the D word.

17.6.1 Prompts

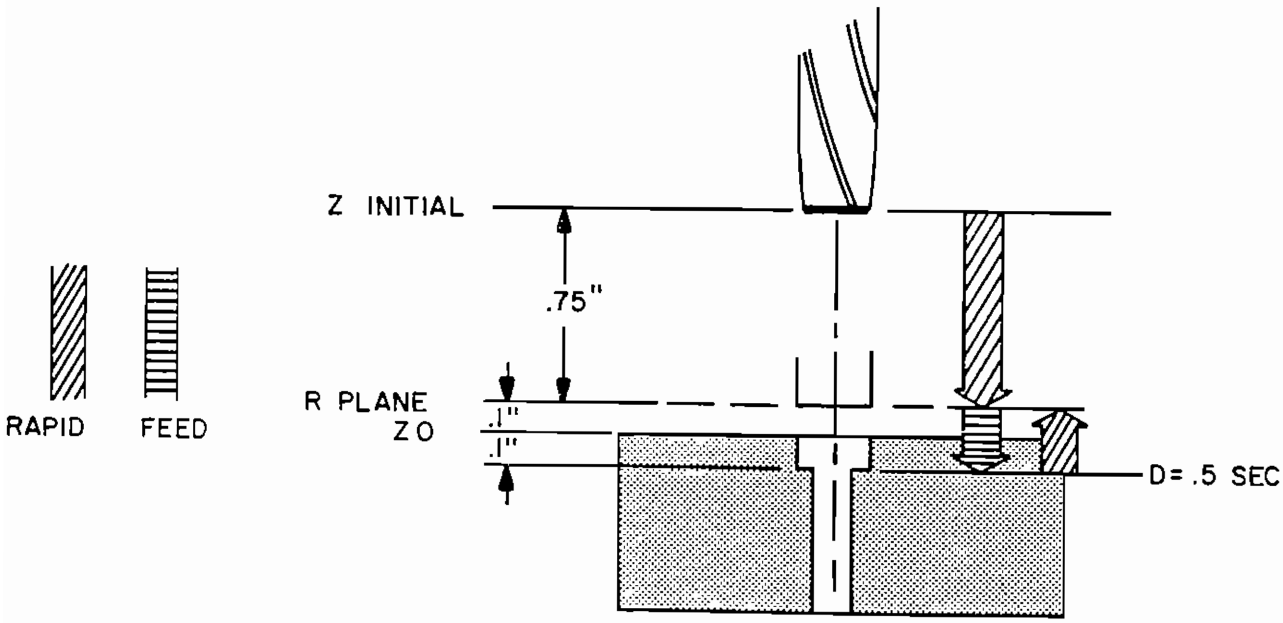
The following table describes the prompts that the control gives for the G82 autocycle.

Table 17.D

Prompt	Function
COUNTER BORE----G82	
DEPTH--Z	"Z" is required for the G82 autocycle. It programs the Z coordinate of maximum infeed (negative Z movement) when Z is absolute (G90). It programs the distance below the R-plane when Z is incremental (G91).
DWELL SECONDS--(D)	"D" is optional. It specifies the dwell time amount in seconds (0.01 to 99.99) at hole bottom.
RAPID PLANE--(R)	"R" is optional. It specifies the Z position of the R-plane. R is always an absolute Z axis position. The metal cutting operation of the autocycle occurs from the R-plane position to the depth specified with Z. All autocycles rapid down for the position of the R-plane from the initial tool position when the autocycle is performed. If you do not specify R in the autocycle, the control uses the current Z position.
INFEEDRATE--(F)	"F" programs a feedrate for infeed from the R-plane to depth. If not programmed, the last used F word is assumed.
RETRACT CODE--(P)	"P" programs the retract position of the tool between executions of the autocycle. If P is not programmed, or is programmed as zero, the tool retracts to the R-plane during XY moves that invoke the cycle. If P is programmed as non-zero (1 to 250), the tool retracts to the position it had at the time the autocycle was defined.
OUTFEED RATE--(V)	"V" programs the feedrate used for retract from total depth. If V is not programmed, the control assumes that retract should occur at the rapid rate.
X PRE-MOVE--(X)	"X" programs a move in X to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.
Y PRE-MOVE--(Y)	"Y" programs a move in Y to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.

Figure 17.4 - G82 Example

Absolute	*Incremental*
G92 Z.85 #	*G92 Z.85 #*
G90#	*G91 #*
G82 Z-.1 R.1 D.5 P0 #	*G82 Z-.2 R.1 D.5 P0 #*



17.7 Peck Drilling Autocycle - G83

A G83 block performs a drilling operation that includes a retract move or dwell for chip breaking or chip removal. You may specify a dwell and/or a Z positive incremental movement as the chip break action after each peck.

When full retraction is programmed for chip break, the tool may retract, depending on programming:

- after each peck
- at successive programmed depths
- or only at the end of the cycle

When a dwell is programmed for chip break, the autocycle may be programmed to perform an optional retract to the R-plane after each dwell.

The G83 autocycle rapids to the R-plane from which the peck and chip break action occurs until total depth is reached. The tool retracts at rapid either to the R-plane or initial tool position prior to the next execution of the cycle.

As with all autocycles, G83 is modal. It remains in effect until canceled with a G80 data block, or superceded by another autocycle, and is automatically performed after subsequent rapid (G00) moves or moves divided with the D word.

17.7.1 Prompts

The following table describes the prompts that the control gives for G83.

Table 17.E

Prompt	Function
PECK DRILL----G83	
DEPTH--Z	"Z" is required for the G83 autocycle. It programs the Z coordinate of maximum infeed (negative Z movement) when Z is absolute (G90). It programs the distance below the R-plane when Z is incremental (G91).
DWELL SECONDS--(D)	"D" programs a dwell time in seconds (0.01 to 99.99) that occurs after each infeed increment. If D is not programmed, there will be no dwell. If D is programmed and W is missing from the cycle, the control assumes a value for W = 0.1 in. (1.0 mm).
RAPID PLANE--(R)	"R" is optional. It specifies the Z position of the R-plane. R is always an absolute Z axis position. The metal cutting operation of the autocycle occurs from the R-plane position to the depth specified with Z. All autocycles rapid down for the position of the R-plane from the initial tool position while the autocycle is performed. If you do not specify R in the autocycle, the control uses the current Z position.
INFEEDRATE--(F)	"F" programs a feedrate for infeed from the R-plane todepth. If not programmed, the last used F word is assumed.

Table 17.E (continued)

INITAL PECK--I	"I" is required in the G83 cycle. It programs the magnitude of the first peck that occurs below the R-plane. This value is always treated as an incremental Z negative movement regardless of the mode (G90 or G91) in effect or the sign of the entry. If J and K are not programmed, I is the constant peck value used in the cycle.
DELTA Z PECK--(J)	"J" specifies the incremental distance to subtract from the last peck to form the next peck distance. For example, starting with the initial I value peck, the peck distances will be: I - J = 2nd peck distance 2nd peck - J = 3rd peck distance and so on until the minimum peck value, K, is obtained. If J and K are not programmed, the peck distance is constantly the I value.
MIN Z PECK--(K)	"K" specifies the minimum peck distance. The K value is used when: last peck - J $\leq$ K The K value remains in effect after this point until total depth is reached. If K and J are not programmed, the peck distance is constantly the I value.
CHIP BREAK--(W)	"W" specifies a chip break retract increment. This isan incremental Z positive movement that occurs at rapid after the tool has made each peck, allowing chip removal to occur. If W is not programmed, and U is not programmed the control assumes W to be 0.1 in. or 1.0 mm depending on the current program dimensions.

Table 17.E (continued)

Z W/O RETR--(U)	"U" programs the magnitude of depth between rapid retracts to the R-plane. This value is always considered an incremental Z negative distance regardless of the mode (G90 or G91) in effect. Each time the current "depth of cut without retract" is equal to or greater than the U value, a rapid retract to the R-plane is performed. If U is not programmed, or is less than I, the tool will retract to the R-plane after each peck. If U is greater than Z minus R, there will be no retract to the R-plane until final depth is reached.
RETRACT CODE--(P)	"P" programs the retract position of the tool between executions of the autocycle. If P is not programmed, or is programmed as zero, the tool retracts to the R-plane during XY moves that invoke the cycle. If P is programmed as non-zero (1 to 250), the tool retracts to the position it had at the time the autocycle was defined.
X PRE-MOVE--(X)	"X" programs a move in X to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.
Y PRE-MOVE--(Y)	"Y" programs a move in Y to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.

17.7.2 Examples

The following figures give examples for programming various features of the G83 autocycle.

Figure 17.5 G83 - Example With Chip Breaker only

Absolute

G92 Z1.1 #
G83 Z-2. R.1 F10.
I.6 W.1 U3. P0 #

Incremental

G92 Z1.1 #
G83 Z-2.1 R.1 F10.
I.6 W.1 U3. P0 #

***NOTE:** U > Z-R, retract at final depth only.*

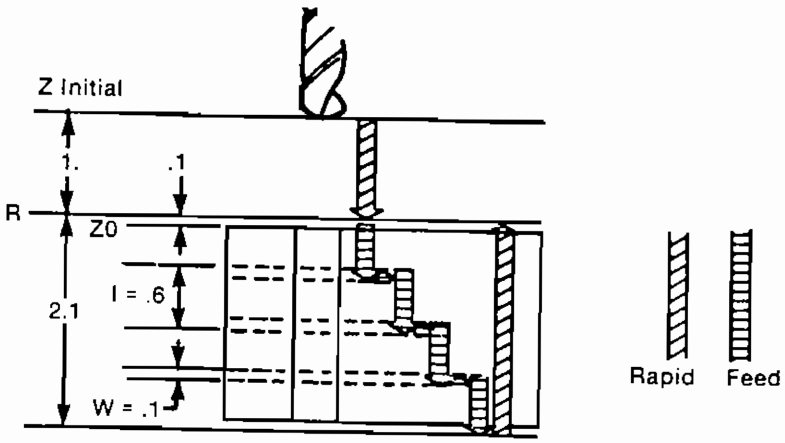


Figure 17.6 - G83 Example with Dwell and Retract only

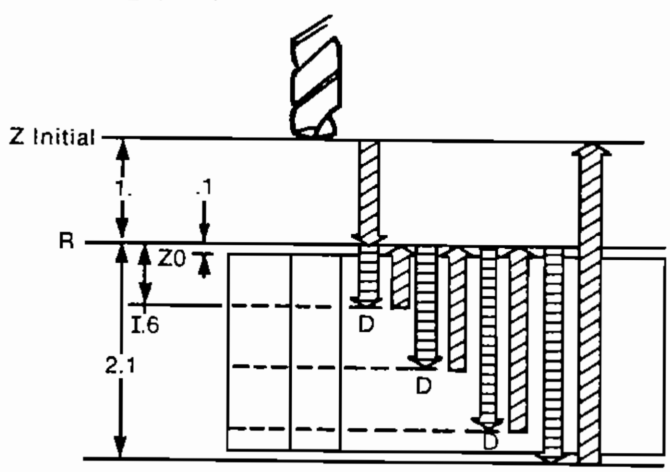
Absolute

G92 Z1.1 #
G83 Z-2. R.1 F10.
I.6 D1. P1 #

Incremental

G92 Z1.1 #
G83 Z-2.1 R.1 F10.
I.6 D1. P1 #

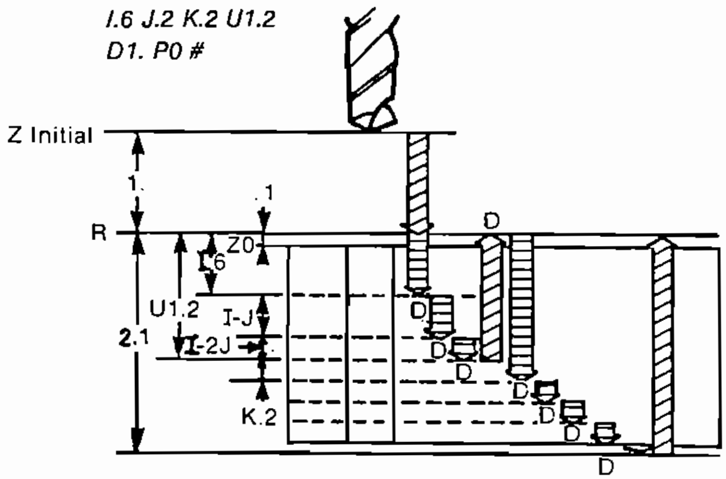
***NOTE:** U not programmed -- retract to R-plane each peck. P =/ 0 -- retract to Z initial at end of cycle.*



Important: After retract, the tool rapids back to depth less the default value of W. This situation is not depicted in the figure, above.

Figure 17.7 - G83 Example with Dwell, Chipbreak, and Z W/O Retract

Absolute	*Incremental*	*NOTE: Use of J, K and U; no W retract -- U =/ 0, D =/ 0*
G92 Z1.1 # *G83 Z-2. R.1 F10.* *I.6 J.2 K.2 U1.2* *D1. P0 #*	*G92 Z1.1 #* *G83 Z-2.1 R.1 F10.* *I.6 J.2 K.2 U1.2* *D1. P0 #*	



17.8 Tap Autocycle -- G84

A G84 block lets you tap right hand (RH) threaded holes and is intended for RH threaded holes only. You should create your own tapping routine and define it with a G79 block if you desire left hand (LH) threads.

CAUTION: To avoid damage to the tap, the workpiece, or the machine, use a floating tap holder for performing any tapping operations.

The G84 autocycle turns the spindle on CW at the start of the cycle (if interfaced via PAL). The operation rapids to the R-plane and feeds to depth. When the final depth is reached, the spindle is turned off then reverses (CCW) automatically. An optional dwell takes place and finally tool feeds back up to retract position.

Important: For safety reasons, the control does not allow you to adjust the feedrate and spindle speed override switches during the G84 autocycle. They assume a default value of 100% during G84 execution. Because of this, make sure that programmed feedrates and spindle speed values are correct.

Important: if you use the yellow [BLK/BLK] button for single cycle execution with G84, the control executes the entire cycle as if it were one block, without breaking it into move segments. Also, the [CYCLE STOP] button is functionally disabled during a G84 cycle.

As with all autocycles, G84 is modal. It remains in effect until canceled with a G80 data block, or superceded by another autocycle, and is automatically performed after subsequent rapid (G00) moves or moves divided with the D word.

17.8.1 Prompts

The table below describes the prompts that the control gives for G84.

Table 17.F

Prompt	Function
TAP--G84	
DEPTH--Z	"Z" is required for the G84 autocycle. It programs the Z coordinate of maximum infeed (negative Z movement) when Z is absolute (G90). It programs the distance below the R-plane when Z is incremental (G91).
DWELL SECONDS--(D)	"D" is optional. It specifies the dwell time amount in seconds (0.01 to 99.99) at hole bottom after the spindle reverses direction and before retract occurs.
RAPID PLANE--(R)	"R" is optional. It specifies the Z position of the R-plane. R is always an absolute Z axis position. The metal cutting operation of the autocycle occurs from the R-plane position to the depth specified with Z. All autocycles rapid down for the position of the R-plane from the initial tool position when the autocycle is performed. If you do not specify R in the autocycle, the control uses the current Z position.

Table 17.F (continued)

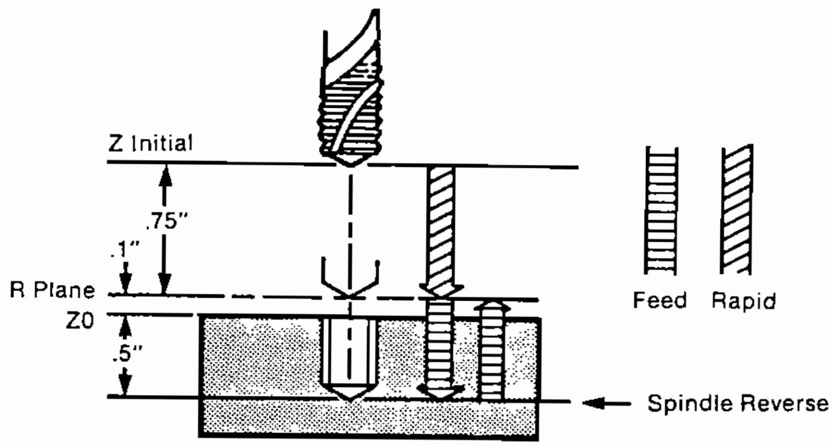
INFEEDRATE--(F)	"F" programs a feedrate for infeed from the R-plane to depth and outfeed from depth to retract position. The F word should be programmed such that: F = RPM x Lead of tap If not programmed, the last used F word is assumed.
RETRACT CODE--(P)	"P" programs the retract position of the tool between executions of the autocycle. If P is not programmed, or is programmed as zero, the tool retracts to the R-plane during XY moves that invoke the cycle. If P is programmed as non-zero (1 to 250), the tool retracts to the position it had at the time the autocycle was defined.
X PRE-MOVE--(X)	"X" programs a move in X to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.
Y PRE-MOVE--(Y)	"Y" programs a move in Y to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.

17.8.2 Example

Figure 17.8 gives an example of G84 programming.

Figure 17.8 - G84 Example

Absolute	*Incremental*	*Note:* RPM = 200,
		LEAD = 1/16,
G92 Z.85 #	G92 Z.85 #	F = 200 X 1/16 = 12.5
G84 Z-.5 R.1 F12.5 P0 #	G84 Z-.6 R.1 F12.5 P0 #	IPM



17.9 Bore Autocycle - G85

The G85 autocycle performs boring operations. The cycle rapids to the R-plane, feeds to depth, dwells and then feeds up to the retract position.

As with all autocycles, G85 is modal. It remains in effect until canceled with a G80 data block, or superceded by another autocycle, and is automatically performed after subsequent rapid (G00) moves or moves divided with the D word.

17.9.1 Prompts

The table below decribes the prompts that the control gives for G85.

Table 17.G

Prompt	Function
BORE--G85	
DEPTH--Z	"Z" is required for the G85 autocycle. It programs the Z coordinate of maximum infeed (negative Z movement) when Z is absolute (G90). It programs the distance below the R-plane when Z is incremental (G91).
DWELL SECONDS--(D)	"D" is optional. It specifies the dwell time amount in seconds (0.01 to 99.99) at hole bottom.
RAPID PLANE--(R)	"R" is optional. It specifies the Z position of the R-plane. R is always an absolute Z axis position. The metal cutting operation of the autocycle occurs from the R-plane position to the depth specified with Z. All autocycles rapid down for the position of the R-plane from the initial tool position when the autocycle is performed. If you do not specify R in the autocycle, the control uses the current Z position.
INFEEDRATE--(F)	"F" programs a feedrate for infeed from the R-plane to depth. If not programmed, the last used F word is assumed.
RETRACT CODE--(P)	"P" programs the retract position of the tool between executions of the autocycle. If P is not programmed, or is programmed as zero, the tool retracts to the R-plane during XY moves that invoke the cycle. If P is programmed as non-zero (1 to 250), the tool retracts to the position it had at the time the autocycle was defined.
OUTFEED RATE--(V)	"V" programs the feedrate used for retract from total depth. If V is not programmed, the control assumes that retract should occur at current F value.
X PRE-MOVE--(X)	"X" programs a move in X to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.
Y PRE-MOVE--(Y)	"Y" programs a move in Y to be performed prior to the first execution of the cycle. This move may be absolute/incemental or rapid/feedrate. If the mode is rapid the autocycle will be performed immediately after this move.

17.9.2 Example

Figure 17.9 gives an example of G85 programming.

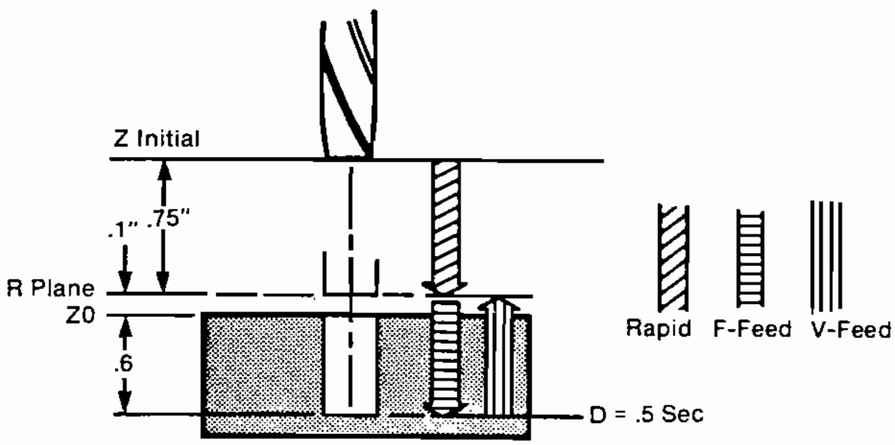
Figure 17.9 - G85 Example

Absolute

G92 Z.85 #
G85 Z-.6 R.1 F5.
V15. D.5 P0 #

Incremental

G92 Z.85 #
G85 Z-.7 R.1 F5.
V15. D.5 P0 #



17.10 Restore Last Cancelled Autocycle - G89

A G89 block re-enables or restores the last autocycle, cancelled with a G80 code, to active status within a program. The autocycle may have been cancelled to suspend the autocycle from acting upon certain rapid XY movements to move around obstructions such as clamps. Program a G89 as the only word within a block by itself. Using G89 can save the programming time required to re-enter a cancelled autocycle.

17.11 Chapter Summary

This chapter discussed the programming and operation of the Autocycle series of G word data blocks, which perform repetative machining operations at a series of locations.

The next chapter covers the G word data blocks in the Autoroutine group. This group simplifies the programming of complex movements that perform common machining operations, such as pocket milling or bolt patterns.

Chapter 18 -- Autoroutines

18.0 Chapter Overview

This chapter covers the Autoroutine group of G word data blocks. Autoroutines are G words that program complex machine movements with single data blocks. After reading this chapter, you will know how to program:

- G22 and G23 -- plane selectable helix moves in clockwise or counterclockwise directions. See section 18.1.
- G24 -- bolt circle patterns. See section 18.2.
- G25 -- step-and-repeat patterns. See section 18.3.
- G26 and G27 -- pocket and post milling routines. See section 18.4.
- G29 -- execution of the last active autocycle. See section 18.5.

18.1 Helical Interpolation G22 and G23

To make a helix, the control feeds two axes in an arc, and the third axis in a straight line. You can program a helical move with:

- G22 for a clockwise helix move
- G23 for a counterclockwise helix move

Cutter radius compensation is not allowed during a helix move. You must make sure that cutter radius compensation is cancelled with a G40 prior to executing a G22 or G23 helix move.

18.1.1 Prompts

The following table shows the prompts that the control gives for G22 - Clockwise Helical Interpolation. The prompts are identical for G23 - Counterclockwise Helical Interpolation.

Table 18.A

Prompt	Function
HELICAL CW----G22	
X CTR OR/REV--(I)	"I" programs the X axis arc center for G17 and G18 planes. It programs the linear move per rev when the plane is G19.
Y CTR OR/REV--(J)	"J" programs the Y axis arc center for G17 and G19 planes. It programs the linear move per rev when the plane is G18.
Z CTR OR/REV--(K)	"K" programs the Z axis arc center for G18 and G19 planes. It programs the linear move per rev when the plane is G17.
X END POINT--(X)	"X" programs the total linear move for the block when the plane is G19.
Y END POINT--(Y)	"Y" programs the total linear move for the block when the plane is G18.
Z END POINT--(Z)	"Z" programs the total linear move for the block when the plane is G17.

18.1.2 Helix Programming

To produce the helix, the control needs to know:

- plane of the arc

  Before you program the helix block, program a block that calls out the plane of the arc move. Use:

  - G17 for a XY plane arc move

  - G18 for a XZ plane arc move

  - G19 for a YZ plane arc move

  Note that your helix moves are plane selectable. Your helix block must have a format that is correct for the plane you choose, or you'll get an error, "IMPROPER CIRCLE AXIS," when you run the program.

- direction of the arc

  In the helix block, program:

  - G22 for clockwise movement in the selected plane

  - G23 for counterclockwise movement in the selected plane

- center of the arc move

  In the helix block, program the center of the arc with:

  - I and J for a XY plane (G17) arc

  - I and K for a XZ plane (G18) arc

  - J and K for a YZ plane (G19) arc

  Remember that the values for circle center are referenced to Program Zero in absolute (G90) mode, or to the present location of the axes in incremental (G91) mode.

- the linear move per full revolution of the arc

  In the helix block, program the move the linear axis (the one not in the arc plane) makes for each complete revolution of the arc axes. Program this value for a full circle even if the you won't make a full revolution with the helix block. Use these addresses:

  - K for a XY plane (G17) arc

  - J for a XZ plane (G18) arc

  - I for a YZ plane (G19) arc

  The value you program is always an increment. It needs no sign.

- the total length of the linear move for the block

  In the helix block, program the move the linear axis is to make for the block with:

  - Z for a XY plane (G17) arc

  - Y for a XZ plane (G18) arc

  - X for a YZ plane (G19) arc

  You can program the total linear move in absolute (G90) mode, or incremental (G91) mode. Be sure the distance and direction is correct for either case.

The arc move stops when the total linear move for the block is reached. This means that the arc axes could stop at any point in their revolution. The revolutions of the arc axes in the block are determined by:

$$\frac{\texttt{move per rev}}{\texttt{total linear move}} = \texttt{revolutions of the arc axes in the block}$$

Because of this, we recommend that you use absolute (G90) mode for axis moves to precise positions after the helix block.

The following table summarizes what you need to know to program a helix:

Table 18.B

If the circle plane is:	program the: center with:	move per rev with:	total line move with:
XY (G17)	I and J	K	Z
XZ (G18)	I and K	J	Y
YZ (G18)	J and K	I	X

18.1.3 Examples

Figure 18.1 shows a G22 clockwise helix produced in absolute mode and the XY arc plane. Note that the top of the part was set to Z.25 with a G92 block.

Figure 18.1 - Helical Interpolation-Absolute

```
G90 #
G01 X0. Y2. F10. #
G22 I1. J2. Z-1. K.25 #
X1. Y2.
```

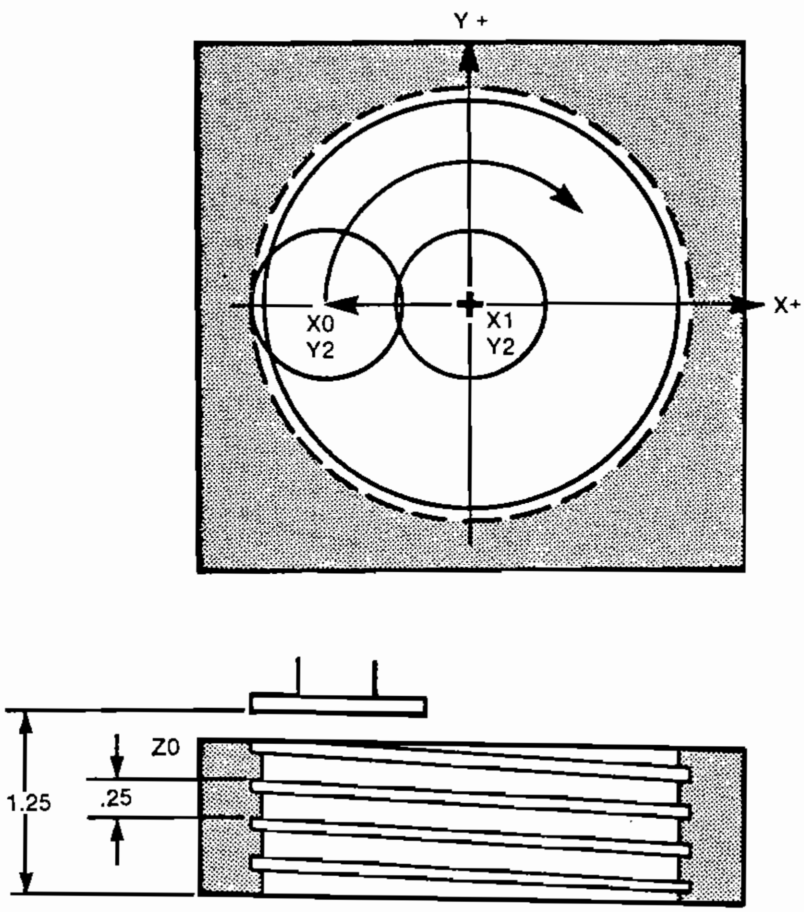
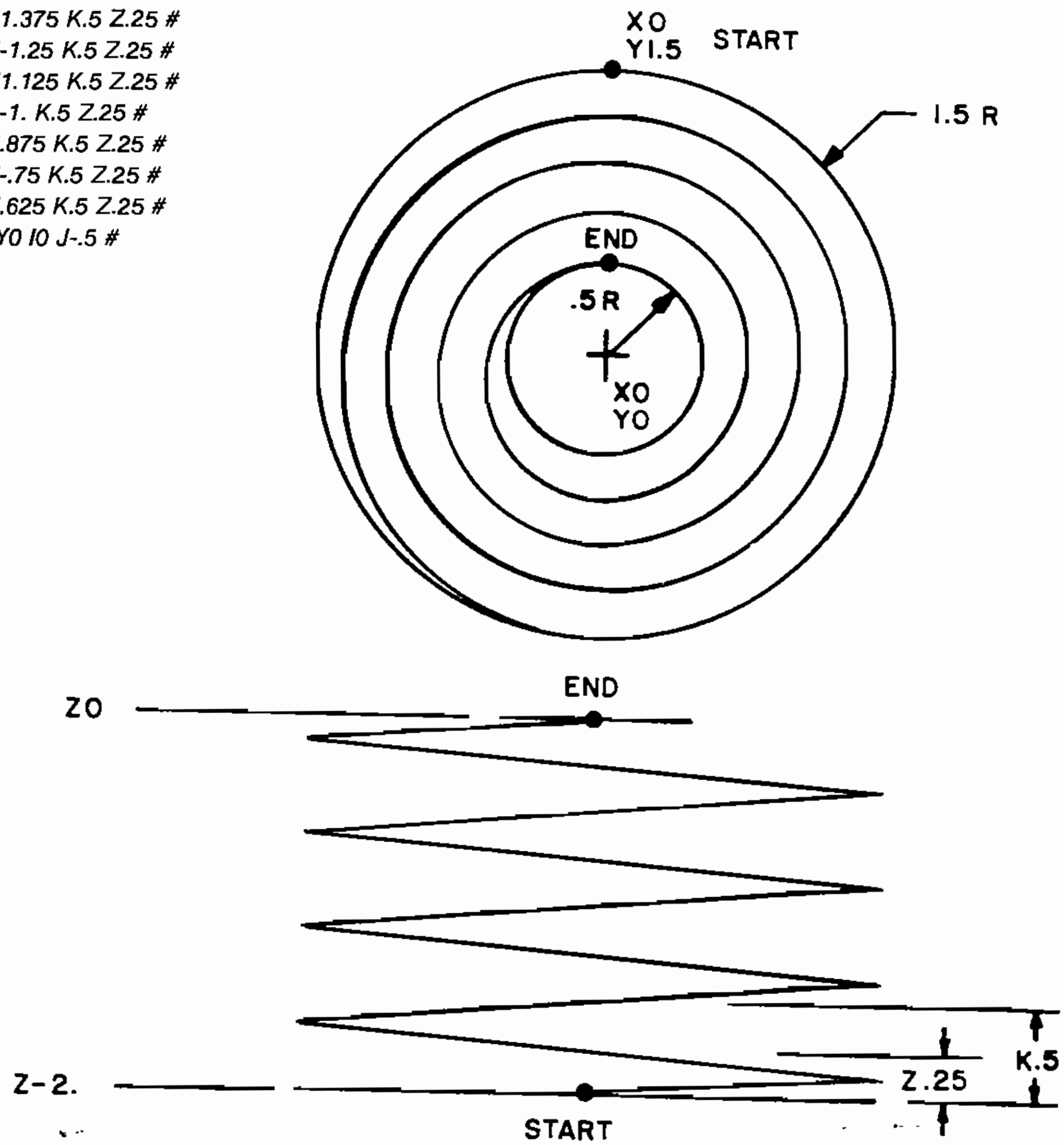


Figure 18.2 shows a G22 clockwise helix produced in incremental, and with the XY plane.

Figure 18.2 - Helical Interpolation-Incremental

```
G90 #
G00 X0 Y1.5
G01 Z-2. 2 #
G91 #
G02 X0 Y0 I0 J-1.5
G22 I0 J-1.5 K.5 Z.25 #
G22 I0 J1.375 K.5 Z.25 #
G22 I0 J-1.25 K.5 Z.25 #
G22 I0 J1.125 K.5 Z.25 #
G22 I0 J-1. K.5 Z.25 #
G22 I0 J.875 K.5 Z.25 #
G22 I0 J-.75 K.5 Z.25 #
G22 I0 J.625 K.5 Z.25 #
G02 X0 Y0 I0 J-.5 #
M02 #
```



The following example programs illustrate helical moves with other planes.

G18 XZ Plane Sample Program

Data Block	Comment
G99 #	-- cancel position preset (establish machine zero reference).
G90 #	-- establish absolute positioning.
G00 X-10. Y-4. Z-1.#	-- rapid to start of helix referenced to machine zero.
G92 X0 Y0 Z0 #	-- establish preset position (program zero).
G18 #	-- establish XZ circular interpolation plane.
G22 I0 K-.5 J.25 Y-1.#	make a clockwise helix move with the XZ circle plane, .5 inch radius, .25 inch per rev and move to Y-1. (4 revolutions).
G00 X0 Y0 Z0 #	-- rapid back to start of helix (at program zero).
M02 #	-- end of program.

G19 YZ Plane Sample Program

Data Block	Comment
G99 #	-- cancel preset (establish machine zero).
G90 #	-- establish absolute positioning.
G00 X-10. Y-4. Z-1.#	-- rapid to start of helix from machine zero.
G92 X0 Y0 Z0 #	-- establish program zero.
G19 #	-- establish YZ circular interpolation plane.
G22 J0 K-.5 I2. X1.#	-- make a clockwise helix move with the YZ circle plane, .5 inch radius, 2 inches per rev and move to X1. (one-half revolution).
G00 X0 Y0 Z0 #	-- rapid back to start of helix (at program zero).
M02 #	-- end of program.

18.2 Bolt Hole - G24

The G24 bolt hole autoroutine lets you automatically execute an autocycle (G79-G85), subprogram (P), or subroutine (P, H, E) at up to 255 equally spaced points in a circular pattern.

Moves between points in the pattern are made at rapid traverse. The control performs the autocycle, subprogram, or subroutine at a point, and then performs a move to the next point. When the pattern is complete, the position is the last point in the pattern.

18.2.1 Prompts

The following table describes the prompts the control gives for G24.

Table 18.C G24 Prompts

Prompt	Function
BOLT HOLE----G24	
X CIR CENTER--(I)	"I" defines the X center of the circle in absolute or incremental. If not programmed, the current X position is assumed to be the X center coordinate of the circle.
Y CIR CENTER--(J)	"J" defines the Y center of the circle in absolute or incremental. If not programmed, the current Y position is assumed to be the Y center coordinate of the circle.
X START POS--(X)	"X" defines the start position on the circle in absolute or incremental. If not programmed, the current X position is assumed to be the X start coordinate.
Y START POS--(Y)	"Y" defines the start position on the circle in absolute or incremental. If not programmed, the current Y position is assumed to be the Y start coordinate.
CIR RADIUS--(R)	"R" defines the radius of the circle.
STRT REF ANGL--(C)	"C" defines the angle of the start position on the circle. It is measured from the positive X axis in absolute or from the initial tool position in incremental.
ANGLE 1--(A)	"A" defines the angle in degrees from the start position on the circle to the first execution point. If you do not program A, the start position is assumed to be the first execution point. Note that in contrast to polar programming, the vertex for A is the center of the circle, not the current location.

Table 18.C (continued)

HOLES/CIRCLE--(W)	"W" specifies the total number of points in the complete circle and the clockwise (-W) or counterclockwise (W) direction to points. The maximum number is -255 clockwise and 254 couterclockwise. W or B must be programmed if L is greater than 1. When W is programmed, the point spacing is 360^{o}/W. If you do not program a W, the B value is used for point spacing.
HOLES DRIL'D--(L)	"L" specifies the actual number of points at which the autocycle, subroutine or subprogram will be performed. The maximum value for L is 255 and it can be less than the total number of points defined with W or B.
ANGLE/HOLE--(B)	"B" defines the angle between successive execution points around the circle. Program a B- to move to points clockwise, otherwise moves are counterclockwise. You must program a B or W if L is greater than 1. If B is not programmed, W is used to determine point spacing.
SUBPROGRAM--(P)	"P" specifies the number of a subprogram to be performed at the points of execution around the circle. If you do not program P, the currently active autocycle, or subroutine (HE) is assumed. If the subprogram or subroutine includes rapid moves, or moves with the D parameter, and an autocycle is active, the autocycle will be performed at the rapid points, or the D points in the subprogram, or subroutine.
SUB START NO--(H)	"H" specifies the starting sequence number of a subroutine.
SUB END NO--(E)	"E" specifies the ending sequence number of a subroutine.

18.2.2 Defining the Circle Center and Start

You can use a combination of IJ, XY, or RC to specify:

- center of the circular pattern
- start position on the circle

I and J define the center of the circle with absolute coordinates or incremental dimensions. If you don't program I and J, the control will assume the current position to be the center (unless R and C specify the center when used with X and Y defining the start).

X and Y define the start position of the circle with absolute coordinates or incremental dimensions. If you don't program X and Y, the control will assume the current position to be the start position (unless R and C specify start position versus the center).

R and C define the radius (R) of the circle and the angle (C) to the start position. These can also be absolute or incremental. You can program the RC combination by itself, with IJ, or with XY. The control does not allow the RC, IJ, and XY combination.

18.2.3 Valid Combinations for Start and Center

You can program the center and start position with the following combinations:

- IJ

  Defines the center at position IJ with the current position as the start position on the circle.

- XY

  Defines the current position as the center. A move will be made to XY, the start position on the circle.

- RC

  Defines the current position as the center. A move defined by RC (radius R at angle C) will be made to the start position on the circle.

- IJ XY

  The center is at IJ, the start position is at XY.

- IJ RC

  The center is at IJ, the start position is defined by radius R at angle C.

- XY RC

  The start position is defined by XY. R specifies the radius of the circle. C specifies the angle from the center to the start position. RC defines the center in this case.

18.2.4 Defining the Points on the Circle

You can define the number of points on the circle, how they are spaced, and the direction of movement (clockwise or counterclockwise) between these points.

Number of Points to Execute the Cycle

The L word defines the number of points you want to execute on the circle. It can be from 1 to 255. If you don't program L, the control assumes there is only 1 execution point. Note that L can be less than the total number of points on the circle, as defined by W or B.

Number and Spacing of Points

W or B is used to define the spacing of points on the circle. W and B cannot be used in the same block. W or B must be programmed if L is greater than 1 execution point.

Using W

W can be programmed from -255 to 254. The spacing of points is determined by:

$$\frac{360^{\circ}}{W} = \text{angle in degrees between points}$$

When W is negative, moves to execution points are made in a clockwise direction. A positive W means the movement between points is counterclockwise.

Note that W programs the total number of points on the circle. The number of execution points, programmed with L, may be less.

Using B

B defines the angle between execution points on the circle. Its word format is ± 3.3.

If B is negative, movement between execution points is clockwise. If B is positive, movement is counterclockwise.

Note that movement occurs B^{o} for each L execution point. This continues until every L point has been performed, and this can result in moving around the circle more than 1 time.

18.2.5 Defining the Angle to the First Point

The A word defines the angle to the first execution point from the circle's start position. Its word format is ± 3.3. Note that angle A is measured from the center of the circle.

If A is negative, movement to the first execution point is clockwise. If A is positive, movement is counterclockwise.

If A is not programmed, the control assumes the start position on the circle to be the first execution point.

18.2.6 Defining the Cycle

The cycle you use at execution points on the circle can be an autocycle, subprogram or subroutine.

If you plan to use an autocycle, program it in a block that comes before the G24 block.

If you plan to use a subprogram or subroutine, program this in the G24 block with a valid combination of the P, H and E words, or use a G79 autocycle.

If an autocycle is active, and the subprogram or subroutine includes rapid moves, or moves with D, the autocycle will be performed at the rapid or D points.

Important: Using block divide, D, provides an alternate method for defining a bolt circle rather than using G24.

18.2.7 Examples

Figure 18.3 graphically shows the use of words in the G24 block. Figure 18.4 shows two examples of G24 programming that produce the same result.

Figure 18.3 - G24 Parameters

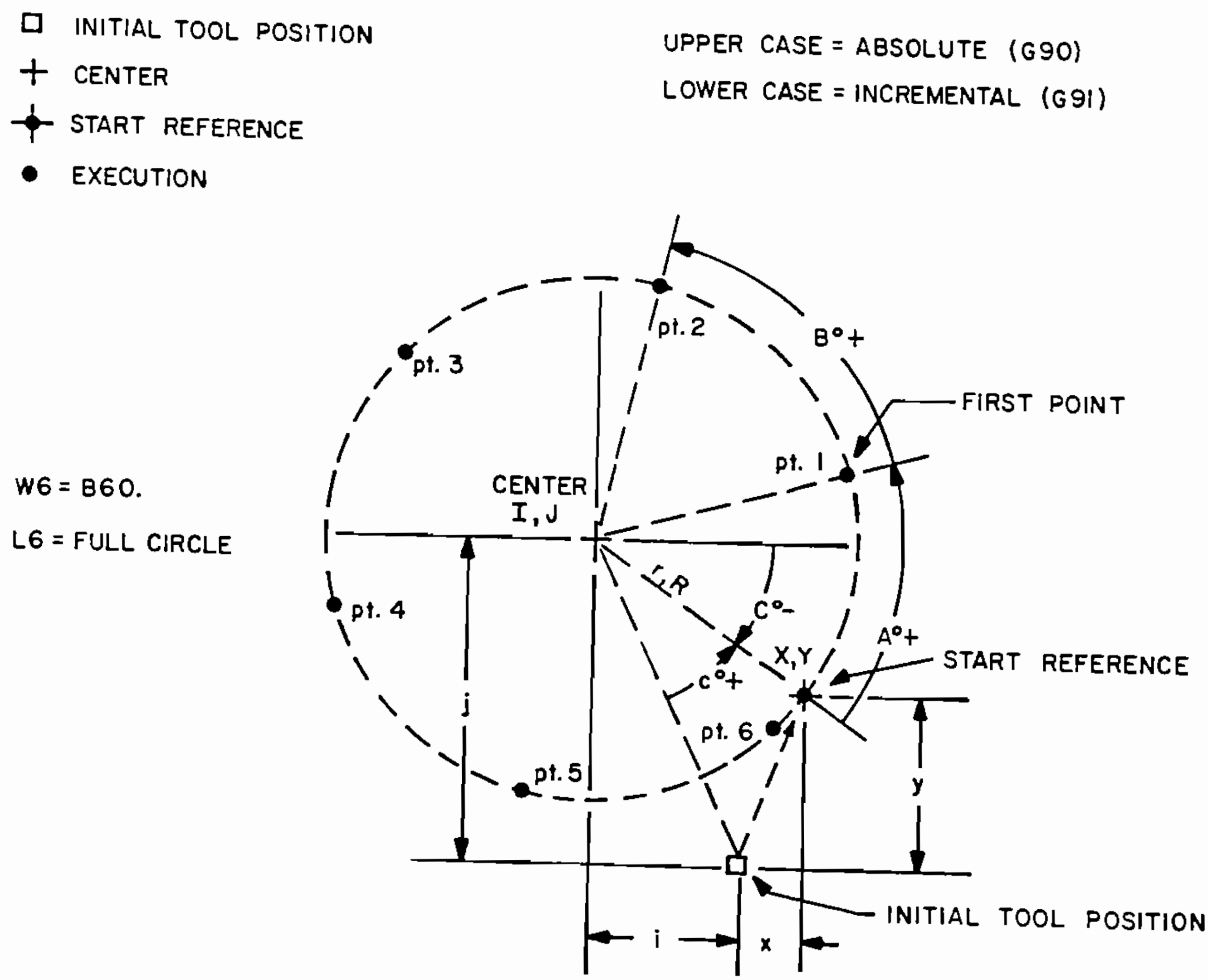


Figure 18.4 - G24 Example

Method 1

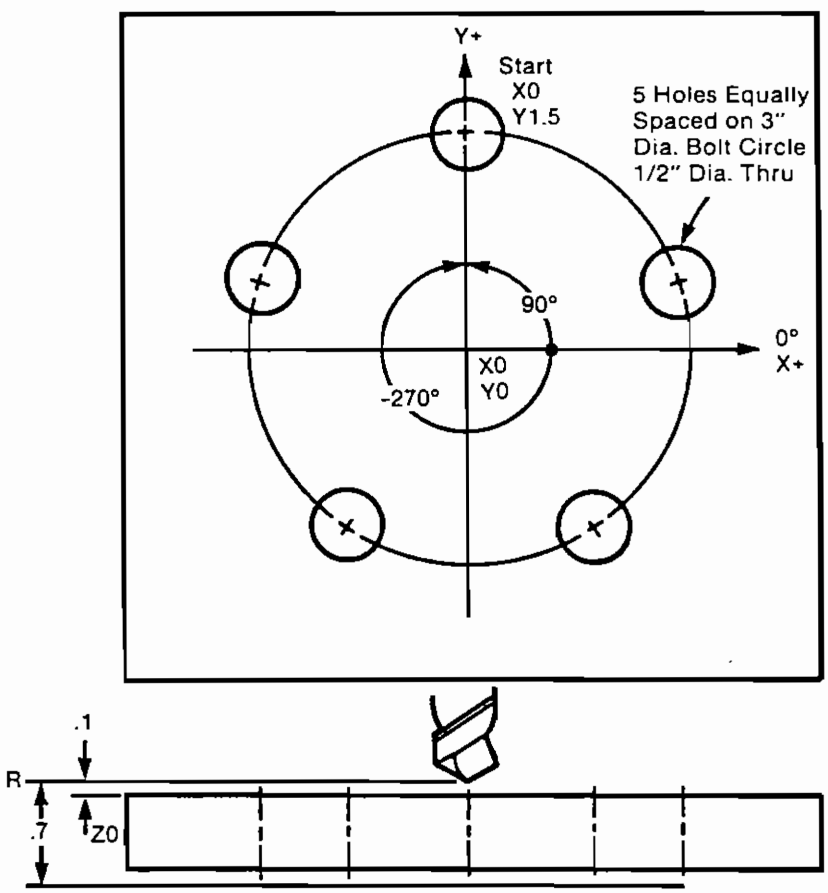
G92 Z.1 #
G90 #
G81 Z-.6 R.1 #
G24 I0 J0 X0 Y1.5
L5 W5 #
G80 #

Method 2

G92 Z.1 #
G90 #
G81 Z-.6 R.1 #
G24 X0 Y1.5 R1.5 C90.
L5 W5 #
G80 #

Method 3 (using D)

G92 Z.1 #
G90 #
G81 Z-.6 R.1 #
G3 I0 J0 D5 #
G80 #
(hole at start done last)



18.3 Step-and-Repeat G25

A G25 block lets you program a rectangular or square pattern of points. You can have the control automatically perform a subprogram (P), subroutine (H, E), and/or autocycle (G79-G85) at the points.

You can specify the spacing and number of points along the X and Y axes of the pattern. Moves between points can be made at rapid traverse, at a feedrate independent of the current F value, or the current F value. If moves between points are made at feedrate, the currently active autocycle is not performed (unless a specified subprogram contains rapid moves).

At completion of the G25 pattern, the control treats subsequent moves as originating from the start point of the pattern although actual position will be the last point in the pattern. This saves time when returning to the start point.

18.3.1 Prompts

The following table describes the prompts that the control gives for G25.

Table 18.D G25 Prompts

Prompt	Function
STEP/REPEAT--G25	
X INCREMENT--X	"X" specifies the the X incremental distance and direction between points of the rectangular pattern. This is always treated as an incremental direction and distance regardless of the mode (G90 or G91) in effect.
Y INCREMENT--Y	"Y" specifies the Y incremental distance and direction between points of the rectangular pattern. This is always treated as an incremental direction and distance regardless of the mode (G90 or G91) in effect.
NO. OF X PTS--I	"I" defines the number of points along the X axis of the rectangular pattern. The sign of this entry is ignored by the control.
NO. OF Y PTS--J	"J" specifies the number of points along the Y axis of the rectangular pattern. The sqign of this entry is ignored by the control.

Table 18.D G25 Prompts continued

FEEDRATE--(F)	"F" specifies the feedrate of moves between points in the rectangular pattern. This feedrate can be different than the current G01 feedrate, and will not affect the G01 feedrate. If F is not programmed, the feedrate defaults to rapid (G00) mode. If F0 is programmed, the current modal feedrate is used.
SUBPROGRAM--(P)	"P" specifies the number of a subprogram (1 to 255) in the control's memory.
SUB START NO--(H)	"H" specifies the starting number of a subroutine.
SUB END NO--(E)	"E" specifies the ending number of a subroutine.

18.3.2 Examples

Figures 18.5 and 18.6 give examples of G25 programming.

Figure 18.5 - G25 Example 1

```
G90 #
G92 Z1. #
G81 Z-.6 R.1 F10. P0 #
G25 X1. Y-.5 I6 J4 #
G80 #
M2 #
```

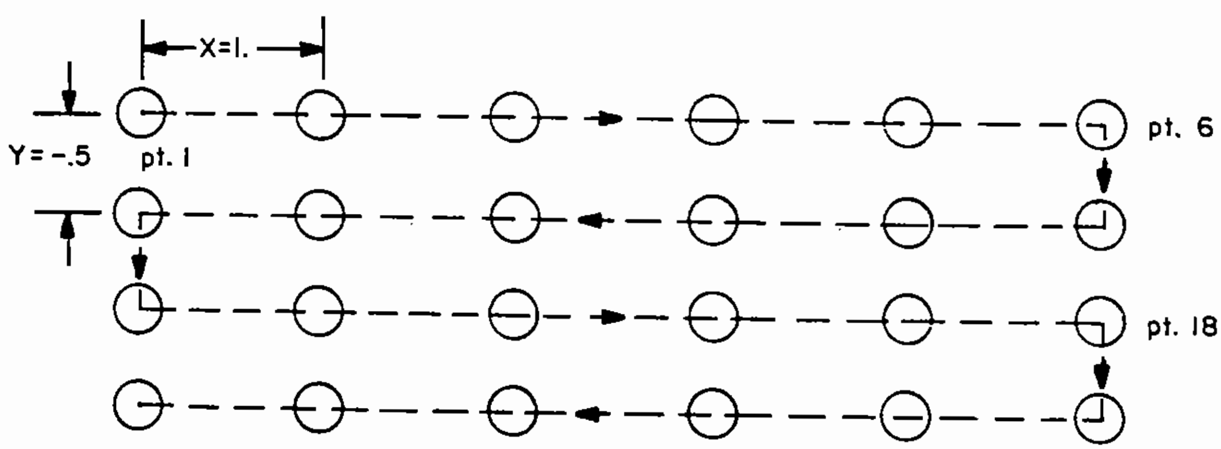
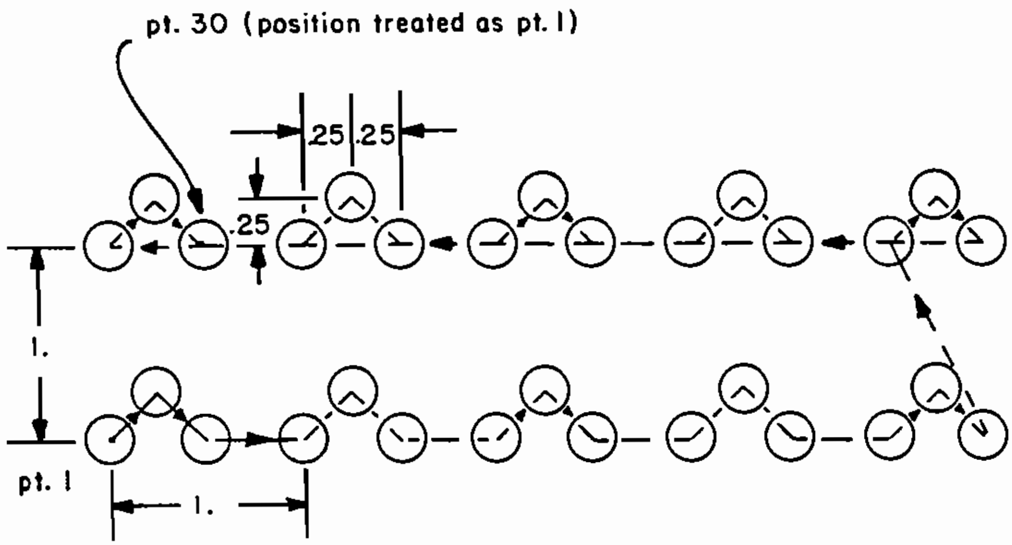


Figure 18.6 - G25 Example 2

```
;MAIN PROG #                 ;SUBPROG,P2#
G90 #                        G91 #
G92 Z1.                      G00 X0 Y0 #
G81 Z-.6 R.1 F10. P0 #       X.25 Y.25 #
G25 X1. Y1. I5 J2 P2         X.25 Y-.25 #
G80 #                        M2 #
M2#
```



18.4 Pocket Milling G26

A G26 block automatically roughs out and finishes a specified rectangular or circular pocket on both the sides and the floor. For rectangular pockets, you can specify a radius for corners in the pocket, using an R word, that is larger than the radius of the tool used to make the pocket. You can also use the G26 autoroutine to enlarge existing pockets.

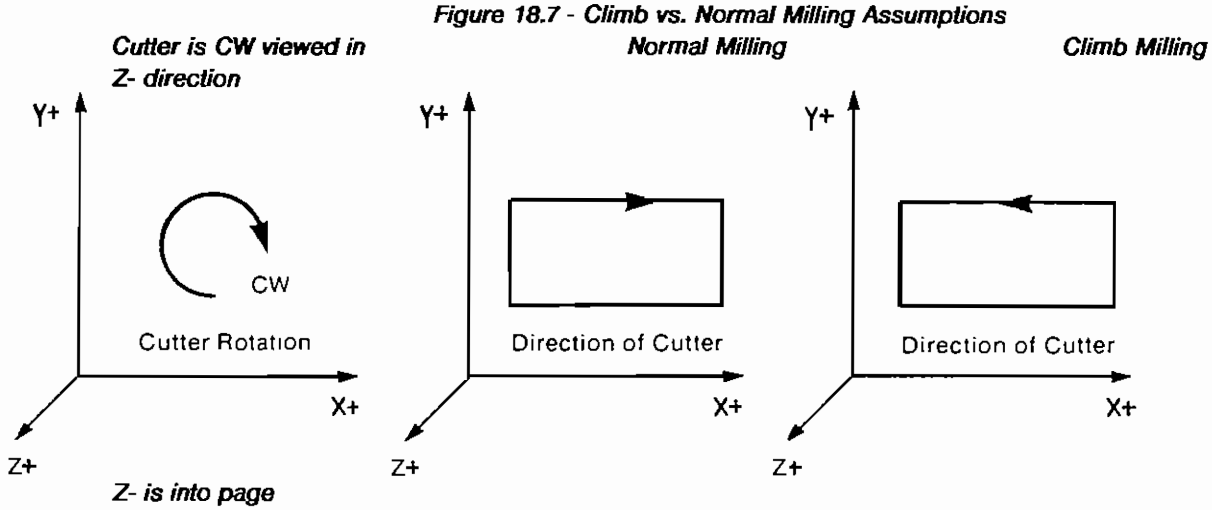
The G26 autoroutine supports an incremental or absolute definition of pocket dimensions, climb milling or normal milling control for both the roughing and finishing passes, with optionally programmed feedrates for plunge (V), rough (F), and finish (H).

The control assumes that the cutter is rotating clockwise and that you are using the coordinate system shown in figure 18.7 when it defines climb and normal milling.

The last move on a G26 cycle is a tangential pullout move. A 'Q' parameter has been added to limit the size of the pullout move.

On completion of the finish pass of a pocket, the control will continue around the corner to the next side and then to the center of that side. When the control reaches the center, it will swing a 90 degree arc away from the surface of the pocket. The radius of the arc will be 7/8 of the distance from the center of the pocket to the nearest cutter path of the finish pass. If a 'Q' parameter is entered, the pullout radius will not be greater than the 'Q' entry. Note: The cutter moves on the cutter path. The control will then retract the Z axis and return to the position it was in when the G26 began executing.

Figure 18.7 - Climb vs. Normal Milling Assumptions



The G26 autoroutine also lets you specify separate X/Y, and Z rough and finish thicknesses.

The G26 autoroutine automatically activates the cutter radius compensation mode.

18.4.1 Prompts

The following table describes the prompts that the control gives for G26.

Table 18.E

Prompt	Function
POCKET MILLING–G26	
LENGTH--------(X)	"X" programs the X axis position of the opposite corner of the pocket. When X is absolute, this is the X coordinate of the opposite corner measured from program zero. When X is incremental, it gives the distance and direction from the current position of the cutter to the opposite corner. If X and Y are not programmed, or are 0, a circular pocket is specified with radius R.
WIDTH---------(Y)	"Y" programs the Y axis position of the opposite corner of the pocket. When Y is absolute, this is the Y coordinate of the opposite corner measured from program zero. When Y is incremental, it gives the distance and direction from the current position of the cutter to the opposite corner. If X and Y are not programmed, or are 0, a circular pocket is specified with radius R.
DEPTH---------(Z)	"Z" specifies the position of the bottom of the pocket. When Z is absolute, it programs the Z coordinate of the bottom of the pocket. When Z is incremental, it programs the distance and direction from the clearance plane (position of Z when the autoroutine is defined) to the bottom of the pocket.
X FINISH CUT--(I)	"I" specifies the thickness of the finish cut along the X axis. This is always an incremental entry and its sign is ignored by the control. If I is greater than the cutter diameter, then a "CUT TOO LARGE" error is generated.
Y FINISH CUT--(J)	"J" specifies the thickness of the finish cut along the Y axis. This is always an incremental entry and its sign is ignored by the control. If J is greater than the cutter diameter, then a "CUT TOO LARGE" error is generated.
Z FINISH CUT--(K)	"K" specifies the thickness of the finish cut for the Z axis at the bottom of the pocket.
EXIT RADIUS--(Q)	"Q" specities the radius of the 90 degree exit arc.

Table 18.E (continued)

Z ROUGH CUT---(U)	"U" defines the thickness of the rough cut for the Z axis. U is always an incremental entry whose sign is ignored. If U is not programmed, the control uses the L word to determine the rough cut thickness.
NO. OF Z CUTS-(L)	"L" specifies the number of rough cuts to make along the Z axis. L is always a whole number entry whose sign is ignored. If U is not progammed, the rough cut thickness is: absolute: $\frac{Z_{clearance} - Z - K}{L}$ incremental: $\frac{Z - K}{L}$ If L is not programmed, the U word is used to determine the rough cut in Z.
XY ROUGH CUT--(W)	"W" specifies the thickness of the rough cut for moves in the X and Y axes. This is always an incremental entry, and its sign is ignored by the control. If W is not programmed, the rough cut for X Y will be determined by the control from the D parameter. If W alone, or D alone are programmed, the autoroutine will start at the center of the pocket. If both W and D are programmed, the autoroutine will start at the corresponding distance from the finish cut starting surface for pocket enlargement. If both W and D are programmed, and W x D > 1/2 the shortest side of the pocket, then D is ignored. If neither W nor D is programmed, then the rough cut for X and Y will be the current cutter radius. If W is greater than the cutter diameter, then a "CUT TOO LARGE" error is generated.
NO OF XY CUTS-(D)	"D" specifies the number of cuts performed by the XY axis. This is always a whole number entry whose sign is ignored by the control. See the description of the W word, above.
ROUGH RATE--(F)	"F" defines the roughing feedrate. If F is positive or zero, then normal milling is used for the roughing pass. If F is negative, or not programmed, then climb milling is used for the roughing pass. If F is zero or not programmed, the current modal feedrate is used for roughing.

Table 18.E (continued)

FINISH RATE--(H)	"H" defines the finishing feedrate. If H is positive or zero, then normal milling is used for the finishing pass. If H is negative or missing, then climb milling is used for the finishing pass. If H is zero or missing, then 2 times the roughing rate is used for the finishing rate.
PLUNGE RATE--(V)	"V" defines the plunging feedrate for the Z axis. If V is not programmed, then the roughing rate is used for plunging.
CORNER RADIUS--(R)	"R" programs the radius of corners in the pocket. If X and Y are not programmed or zero, R is the radius of a circular pocket. If is not programmed, but X and Y are programmed, the radius of the pocket corners will be the radius of the current tool.
OFFSET NUMBER--(O)	"O" specifies the offset number of an entry on the Tool Offsets page. This entry specifies the diameter of the tool used to mill the pocket. The control halves this value to determine the radius used for cutter radius compensation. If O is not programmed the control uses the current O word offset that is active in the program. If there is no active offset, the control will issue an error to this effect.

18.4.2 Restrictions

The following is a list of restrictions that you should note before programming a G26 autoroutine.

- O word offset values that have negative (-) cutter diameter values cannot be used with the G26 autoroutine.
- If scaling (G72) is used with the G26 autoroutine and a corner radius (R word) has been programmed in G26, the X and Y scaling factors must have the same values. If R is not programmed, unequal scaling values are allowed.
- The rough cut thickness for the X and Y axes, as defined by W or calculated based on D, cannot be greater than the diameter of the current tool offset.
- The diameter of the current tool offset cannot exceed the shortest side of a rectangular pocket.
- G18 or G19 cannot be active if R has been programmed in G26.

18.4.3 When You Omit Certain Words

The G26 autoroutine does not require that you enter all of the 15 words available to in the block. The table below describes the operation of the autoroutine if certain words are not programmed.

Table 18.F Omit Words

If you program:	*Then this occurs:*
no X	A slot is milled that has a length specified by Y and a width of 2R. If R is also not programmed, then the width of the slot will be equal to the cutter diameter.
no Y	A slot is milled that has a length specified by X and a width of 2R. If R is also not programmed, then the width of the slot will be equal to the cutter diameter.
no X and Y	A circular pocket will be formed with a diameter equal to 2 times R. The center of the pocket is the tool position when the G26 is defined in the program. **Important:** If you specify a circular pocket by programming R and no X and Y, then the finish cut in X (I) must equal the finish cut in Y (J), or the cutter will produce oval path rough cuts. The finish cut will properly finish the circular pocket.
no I	The finish cut specified by J will be used as the X finish cut.
no J	The finish cut specified by I will be used as the Y finish cut.
no I and J	All X and Y cuts will be rough cuts. The last cut will be executed at the finish feedrate.
no K	All Z cuts will be rough cuts.
no W	The width of the XY rough cuts will be determined by the control based on the D word.
no D	The width of the XY rough cuts will be determined by the control based on the W word.
no W and D	The rough cut for XY will be the cutter radius.

Table 18.F (continued)

no U	The rough cut thickness for the Z axis will be calculated by the control based on the L word.
no L	The number of rough cuts for the Z axis will be calculated by the control based on the U word.
no U and L	Only one Z cut will be executed. The depth of the cut will equal the distance from the clearance plane to the Z depth. The K word, if programmed, will be ignored. The X and Y moves will be executed at the finish rate once the Z axis has reached Z depth.
no H	Twice the roughing feedrate is used as the finish feedrate. Climb milling will be used for the finish pass.
no F	The modal feedrate (or the default feedrate) is used as the roughing feedrate. Climb milling is used for the roughing pass.
no V	The roughing feedrate will be used as the Z axis plunge feedrate.
no R	The pocket will have corners with radii equal to the cutter radius, as specified by the O word.
no O	The control will use the currently active tool offset.

18.4.4 When You Combine Certain Words

The table that follows tells what happens when you use certain combinations and values of words in the G26 autoroutine.

Table 18.G Combining Words

If you program these parameters:	*You get these results:*
L and U	case 1) If the number or rough Z cuts times the thickness of each cut is greater than the Z depth less the finish cut (L x U > Z - K), then the L word is ignored by the control and U is used to determine the number of roughing cuts for the Z axis. case 2) If the number of rough Z cuts times the thickness or each cut is less than the Z depth less the finish cut (L x U < Z - K), then the depth of the first rough cut will equal: $(Z_{depth} - K) - (L \times U) + U$ = depth of first cut The balance of the rough cuts will be U thick.
L	If L0 is programmed, only one Z cut is made (equal to Z depth) regardless of the programmed value of U and K, if any. The XY moves will be executed at the finished feedrate once the Z axis has reached Z depth. The maximum value for L is 255.
U and K	Certain combinations of the Z rough cut thickness and Z finish cut (U and K) will require a first rough cut that is less than U. For example, Z DEPTH -- Z = 2. Z ROUGH CUT -- U = 0.6 Z FINISH CUT - K = 0.25 results in 1 rough cut of 0.55 in. deep, 2 rough cuts of 0.60 in. deep, and 1 finish cut of 0.25 in. deep. (1 x 0.55) + (2 x 0.60) + (1 x 0.25) = 2.00 in. The short, or "shallow" cut will be executed first.
F	If you program F with a negative value, or if it is not programmed, climb milling is performed for roughing passes in X and Y. If F is programmed as positive, or unsigned, then normal milling will be performed for roughing passes.

Table 18.G (continued)

H	If you program H with a negative value, or leave H out of the block, climb milling is used for finish passes in X and Y. If H is programmed as positive, or unsigned, then normal milling will be used for finish passes.
D and W	If the number of XY cuts times the XY rough cut is greater than 1/2 of the shortest side of a rectangular pocket (D x W > 1/2 min X, Y), then D is ignored. The maximum value for D is 255.

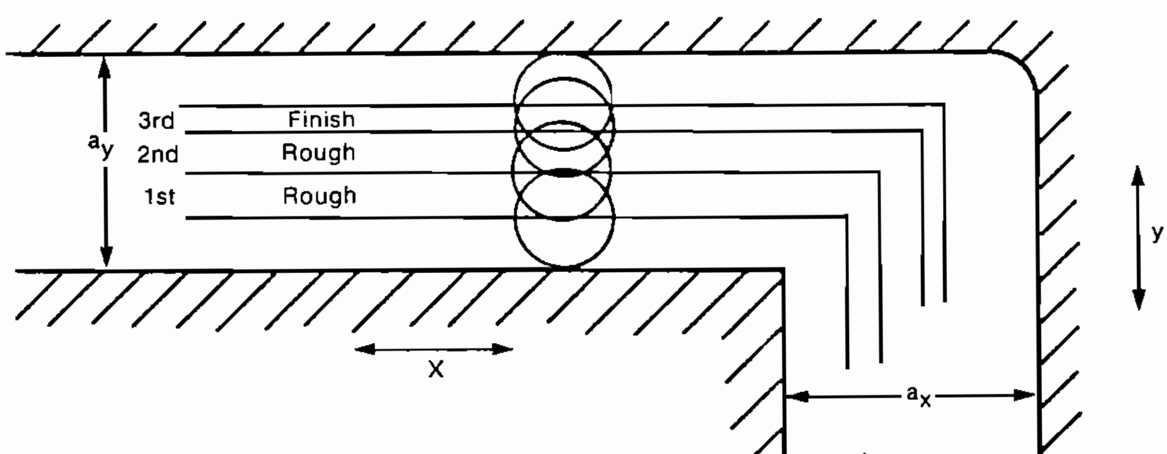
18.4.5 Circular Pockets

To program a circular pocket, omit the X and Y words from the G26 block, that is, do not program X and Y. You must program an R word to define the radius of the pocket, however, and all other words are allowed. A circular pocket will be formed with its center at the XY position just prior to the G26 block.

18.4.6 Enlarging Pockets

You can program the G26 autoroutine to start at a position that is not the center of the pocket. You can use this feature to enlarge existing pockets. Use the D and W words together in the G26 block to use this technique. See figure 18.8.

Figure 18.8 - Enlarging a Pocket with D and W



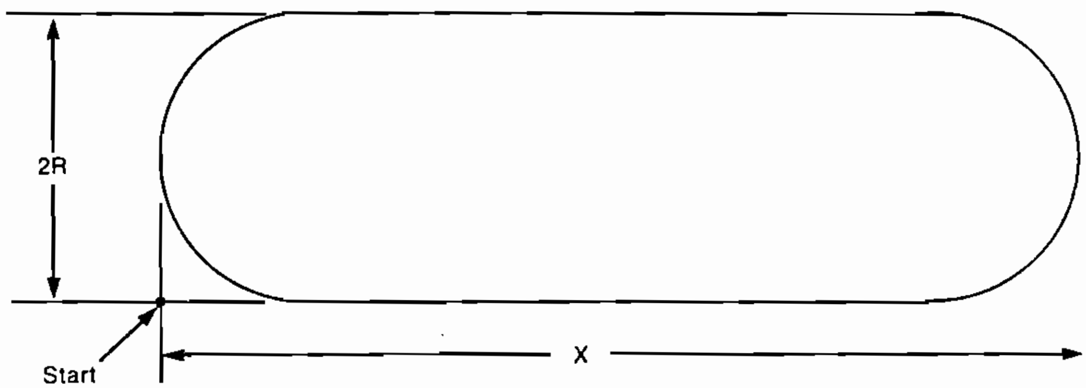
-- use these formulas determine D & W

a_x = [(D - 1) x W] + I + tool diameter
a_y = [(D - 1) x W] + J + tool diameter

18.4.7 Slot Milling

The G26 autoroutine can also be used to mill out a slot. If either the X or the Y word is omitted from G26, then a slot with a width of 2R will be milled. The length of the slot will be equal to the word, X or Y, that is programmed. If R is not programmed, a slot with a width equal to the tool diameter will be formed. See figure 18.9

Figure 18.9 - Milling a Slot with G26



18.4.8 Examples

Figures 18.10 through 18.12 show how the words in the G26 autoroutine are used.

Figure 18.10 - G26 with Corner Radius R

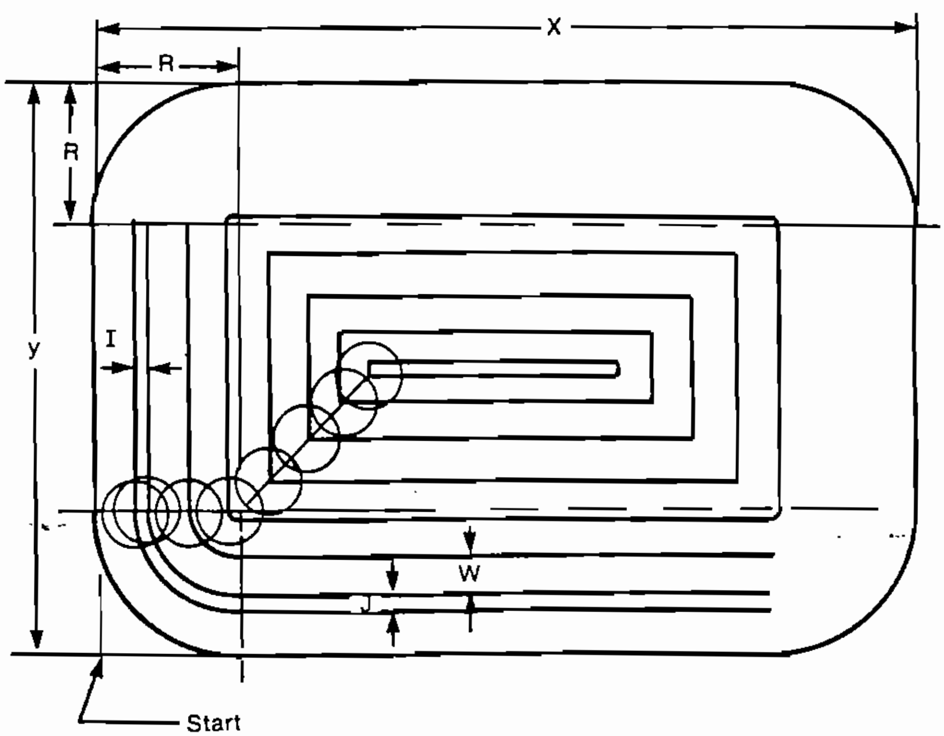


Figure 18.11 - G26 with no Corner Radius R

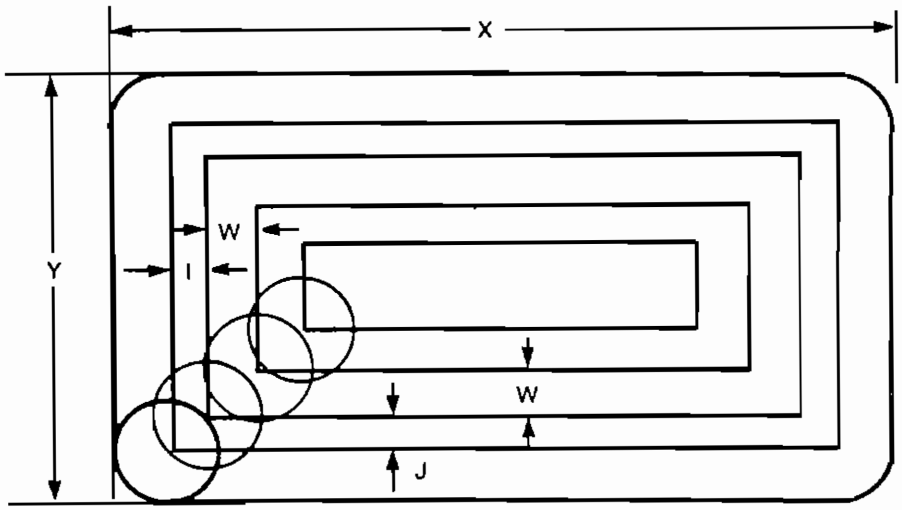
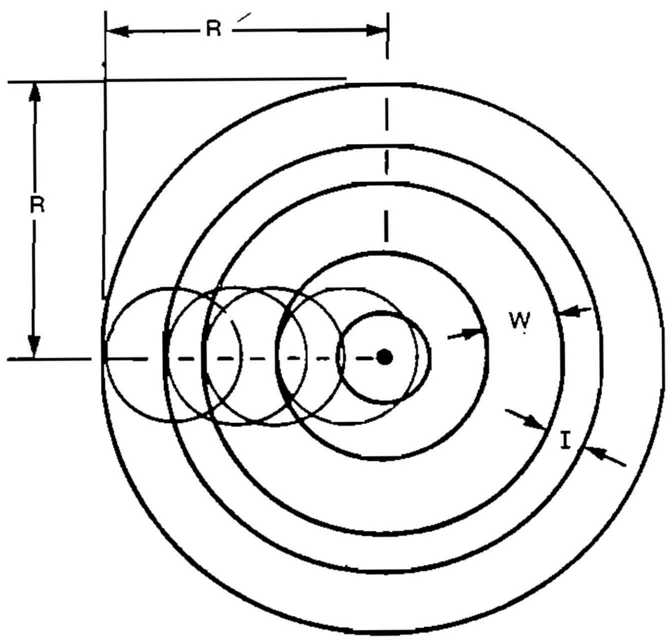


Figure 18.12 - G26 Circular Pocket X = Y = 0



18.5 Post Milling -- G27

A G27 block performs a sequence that is similiar to G26 (described in the previous section), except it removes material outside a specified area (as opposed to inside a specified area for pocket milling).

On completion of the finish pass of a post cycle, the last line is extended past the finish pass starting arc. The length of extension is specified as the cutter path radius of the finish pass arc. If a 'Q' parameter is entered, the pullout move length will not be greater than the 'Q' entry.

All of the prompts, restrictions, and notes applicable to G26 also apply to G27, so refer to the previous section for background information on G27.

The specified area, or "post" is defined by the X and Y parameters.

If you do not program an R word in the G27 autoroutine, the milling pattern will be rectangular, that is, it will have square corners in the cutter path. If R is programmed, then the pattern will produce corners that have a radius equal to R. R0 produces a rectangular post but with 90 degree arcs in the cutter path.

Important: The control must know how far outside the post area to start the milling sequence; therefore, you must program a D word that specifies the number of XY rough cuts.

Figure 18.13 shows the start positions for G27 post milling. The current location of the tool is the position of the axes when the G27 block is defined in the program.

Figure 18.13 - G27 Starting Locations

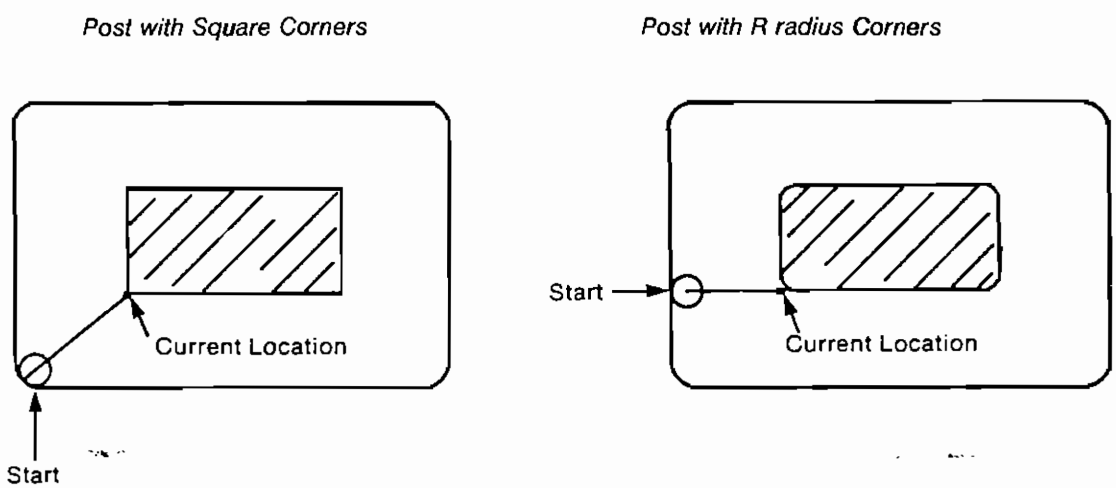
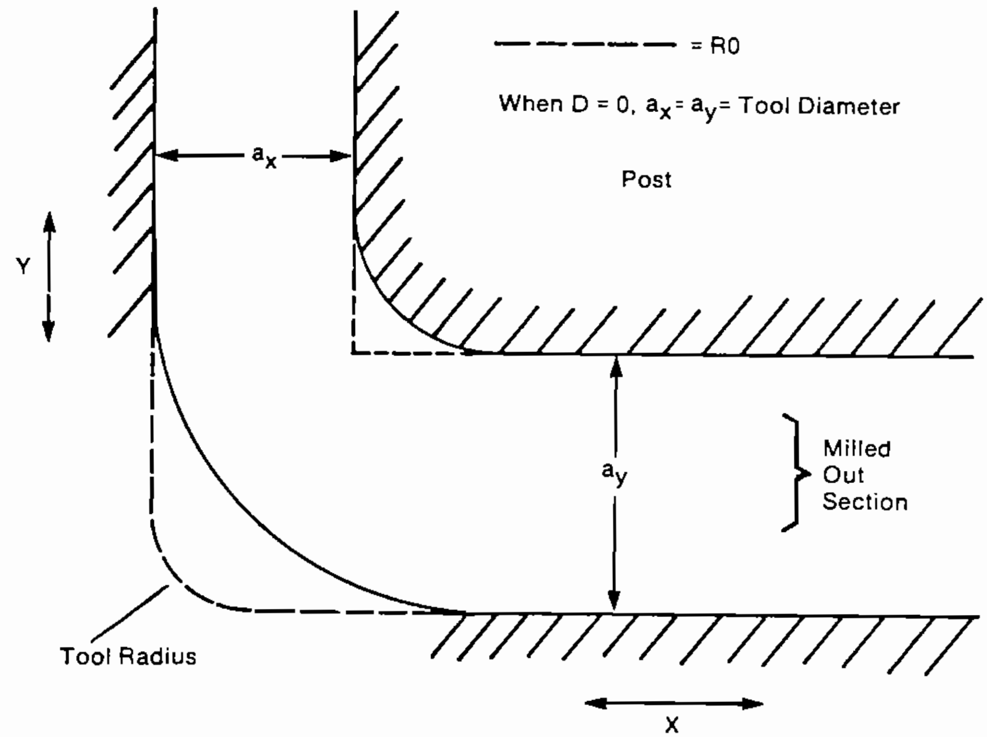


Figure 18.14 shows how the control calculates the area to clear around a post.

Figure 18.14 - G27 Post Clearance Dimensions



-- use these formulas for clearance

a_x = [(D-1) x W] + I + tool diameter
a_y = [(D-1) x W] + J + tool diameter

18.5.1 Examples

Figures 18.15 through 18.17 give examples of how the words in a G27 block are used.

Figure 18.15 - G27 with Corner Radius R

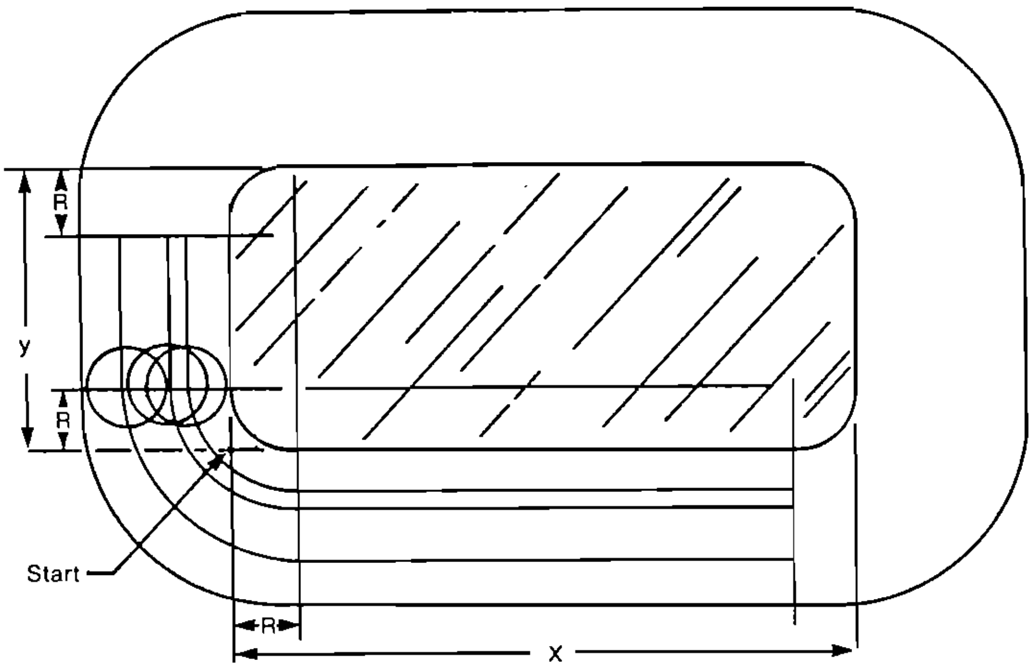


Figure 18.16 - G27 with no Corner Radius R

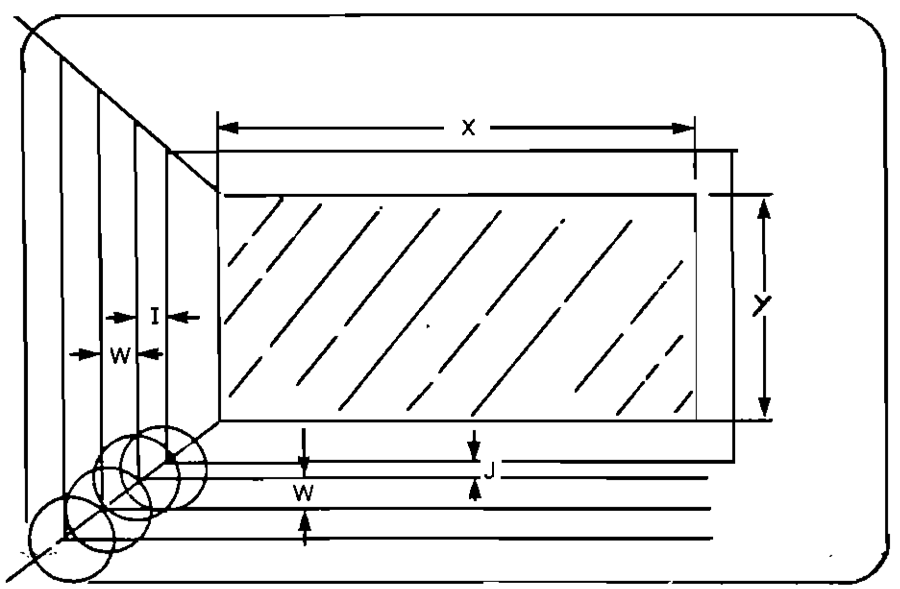
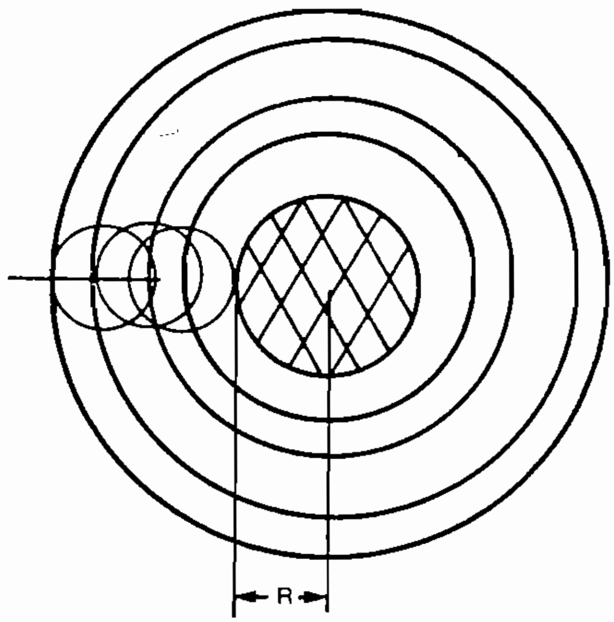


Figure 18.17 - G27 Circular Post X = Y = 0



18.6 Execute Last Autocycle - G29

The G29 word executes the last autocycle that was active in a program. When G29 is programmed, the autocycle is performed even if it has been cancelled by a G80 code. The autocycle is performed only once. The cycle is not restored to active status (if previously cancelled) by the G29 code. The current feedrate need not be rapid (G00) to perform the autocycle.

The G29 block may contain an L word which specifies the number of executions for the entire block up to 255 times.

18.7 Chapter Summary

This chapter covered the G word data block group known as autoroutines. These data blocks perform such things as helix moves, bolt circle patterns and pocket/post milling.

The next chapter covers the CNC control G word group. CNC control governs the graphics display and determines how the control responds to the certain front panel switches and buttons.

Chapter 19 -- CNC Control

19.0 Chapter Overview

This chapter covers the CNC Control group of G word data blocks. CNC Control gives the programmer control of certain operator functions from within the part program. After reading this chapter, you will know:

- how to define soft limits on the six axes of motion
- how to inhibit operator override functions
- how to control features of the graphics display

19.1 Zone Inhibited --G60

The G60 ZONE INHIBITED feature provides the ability to program soft limits on the six axes of motion. The limits may be programmed relative to Machine Zero or relative to the current position.

The G60 Zone Inhibited command can include the following parameters:

Axis	Positive Limit	Negative Limit
X	X parameter	I parameter
Y	Y "	J "
Z	Z "	K "
U	U "	A "
V	V "	B "
W	W "	C "

Each of these parameters can be either absolute or incremental. This mode can be determined by G90 vs G91 or by upper vs lower case, just like axes moves. If the current mode for a letter is absolute, as indicated by an upper case letter on the status page, then an upper case parameter is absolute, a lower case parameter is incremental. If the current mode for a letter is incremental, as indicated by a lower case letter on the status page, then the parameter is incremental, regardless whether it is upper or lower case.

An absolute parameter sets the corresponding limit versus Machine Zero. An incremental parameter sets the corresponding limit versus the current location. For example: G60 X5.3 I-5. Y3. J-4. #

Assuming X, Y, I, and J are all set to Absolute, this defines an inhibited zone on the X axis from '-5.0000' to '+5.3000' and on the Y axis from '-4.0000' to '+3.0000'.

If the negative inhibited zone limit parameter is more positive than the positive inhibited zone limit, an error message of **ZONE INHIB INVERTED** is displayed when the command is executed. For Example: G60 X-2. I2.# will result in this error message.

Note that mirroring (G31) will affect incremental inhibited zone parameters, but not absolute ones. This has the effect of reversing the parameter letters for the positive and negative limits. Thus, if mirroring is ON for an axis, you can not program one of the limits incremental and the other absolute. Attempting to execute such a G60 command will result in an error message of **INC MIRRORS ON ABS**. For example: G60 x5. I-5.# where the X axis is currently mirrored will result in this error message.

Any axis limit not specified in the G60 command is set to the soft limit for that axis, if there is one. Thus a G60# (no parameters) will cancel the inhibited zone, setting all of the axes limits to the soft limits, if they exist. If there is no soft limit for an axis, there will be no limit on that axis when the inhibited zone is cancelled.

If there are soft limits for an axis and only one of its inhibited zone limit parameters is included, the suitable soft limit is used for the other inhibited zone limit. If an axis does not have soft limits, you are not allowed to execute a G60 with only one inhibited zone limit for that axis. If you attempt to, a **NO SOFT LIMIT** error message will be displayed. For example: G60 I-5.# (where the X axis has no current soft limit) will indicate this error.

Rotary limits are set modulus 360 (just like soft limits). The difference between the positive and negative limits for G60 parameters on rotary axes must be less than 360 degrees. (If the difference is greater than 360, the parameters may be ignored or a **ZONE INHIB INVERTED** error message may occur, depending upon the setting of the soft limits.)

If a inhibited zone is active when a subprogram is called, that subprogram cannot change the inhibited zone. If it attempts to, a **ZONE INHIB NESTING** error message is displayed. Returning from a subprogram which has set a inhibited zone, or completing or terminating of a program which set an inhibited zone, will cancel the safe area.

An example of prompt edit G60 inhibited zone prompts appears below.

```
| Program  12                    < operator messages>   GRAPHICS | *
| 00001                          <  error messages  >            |
|  ZONE INHIB---- G60                                       LIST | *
| Z POS LIMIT---Z                                                |
| X POS LIMIT---X                                       INSERT P | *
| X NEG LIMIT---I                                                |
| Y POS LIMIT---Y                                         SEARCH | *
| Y NEG LIMIT---J                                                |
| Z NEG LIMIT---K                                                | -
| U POS LIMIT---U                                                |
| U NEG LIMIT---A                                                | -
| V POS LIMIT---V                                                |
| V NEG LIMIT---B                                        G CODES | *
| W POS LIMIT---W                                                |
| W NEG LIMIT---C                                        M CODES | *
|                                                                |
|               ;PARA_NEST1                            DIRECTORY | *
|                                             #                  |
```

19.2 Override Inhibit -- G62

A G62 block lets the programmer choose whether the operator is allowed to use certain override functions. The default condition is for all overrides enabled.

By programming certain words in the G62 block, the programmer can select which override to affect.

- F -- feedrate override switch
- S -- spindle speed override switch
- Q -- [BLK/BLK] and [CYCLE STOP] button (halting)

If one of these words is programmed in a G62 block, a zero value will inhibit the corresponding override function, that is, not allow the operator to use it. If the word has a non-zero value, the override is enabled and the operator is allowed to use it.

A G62 programmed alone in a block (G62 #) enables all override functions.

19.2.1 Prompts

The table below describes the prompts that the control gives for G62.

Table 19.A
G62 Prompts

Prompt	Function
OVERRIDE INHIBIT--G62	
SINGLE/HOLD--(Q)	"Q" inhibits the operator from halting execution by using the [BLK/BLK] and [CYCLE STOP] buttons when Q is programmed as zero. The operator can only use continuous execution. The operator can still use the feedrate override switch to establish a 0% (feedhold) condition. When Q is programmed as non-zero or a G62# is programmed, the operator can resume using halt functions. If the operator presses the [BLK/BLK] button while Q0 is active, the control will recognize the halt after G62# or non-zero Q. If the operator presses the [CYCLE STOP] button while Q0 is active, and no other key, the control will halt after G62# or non-zero Q. If he presses any other key, the control will not recognize the halt request.
FEED--(F)	"F" inhibits the operator from using the feedrate override switch when F is programmed as zero. The control will assume a 100% setting for the feedrate override switch, and execute moves at programmed feedrates. When F is programmed as non-zero, the operator can resume using the feedrate overide switch to modify the programmed feedrate.
SPINDLE--(S)	"S" inhibits the operator from using the spindle speed override switch when S is programmed as zero. The control will assume a 100% setting for the spindle speed override switch, and execute spindle speeds at programmed rates. When S is programmed as non-zero, the operator can resume using the spindle speed override switch to modifiy the programmed spindle speed.

19.3 Graphics Control - G66

A G66 block controls characteristics of the graphics display from within an executing program, according to the words described below.

If a G66 is programmed by itself in a data block (G66 #), the grapics display is erased automatically.

Refer to chapter 10 for more on the graphics features of the control.

19.3.1 Prompts

The following table describes the prompts that the control gives for G66.

Table 19.B
G66 Prompts

Prompt	Function
GRAPICS--G66	
GRID--(Q)	"Q," when it is programmed toggles the GRID feature. In reponse to Q, the control will turn the GRID on if it is off, or off if it is on. If no Q is programmed in the G66 block, the control takes no action with the GRID.
RAPID--(R)	"R," when it is programmed toggles the RAPID feature. In reponse to R, the control will turn RAPID on if it is off, or off if it is on. If no R is programmed in the G66 block, the control takes no action with RAPID.
W/O ERASE--(W)	"W" inhibits ersasure of the graphics display when programmed. If W is not programmed in the G66 block, the display is automatically erased.

19.4 Chapter Summary

This chapter covered the CNC Control G word group that is used to give the programmer control of some operator functions from withing a part program.

This completes the programming sections of this manual.

Chapter 20 -- Paramacros

20.0 Chapter Overview

The Paramacros feature is actually a collection of very powerful, related features. They are listed below.

- Computable variables
- Computable word address fields in any block type
- Variables to and from PAL
- Access to certain modal system parameters for computations
- Arithmetic operators and expressions for computations
- Arithmetic and Logical functions for computations
- Conditional Branching, Subroutines calls, and Subprograms calls based upon logical function results
- Parametric Programs, Subroutines, and Subprograms
- Parametric Autocycles
- User defineable prompts to aid in program generation and execution Prompting of parameters for main program execution
- Calculator function under prompt edit

20.1 Expressions and Operators

The Paramacros feature allows the computation of the numerical values for most word address fields in any type of block. The exceptions to this are the N and G fields in any block and the P field in G39, G59, and G79 blocks. These blocks must be numeric entries and cannot be computed expressions.

Any other field can be computed by enclosing an arithmetic expression in parantheses following the word address letter. For example: X(2.53 * 7 - 6.2) Note: No spaces in actual entry.

The expression can be any length that will fit on the Prompt Edit screen following the word address letter. As many word address letters as desired in the block may have expressions.

Nesting of parentheses is allowed within the expression. For example: L((2.7 + 14.123) * 17)

The following operators are implemented with this feature:

Unary Operators:	*Plus and Minus + - Unary operators may only come immediately after a left perenthesis '(' You must close the quantity with a right perenthesis ')'.*
Binary Operators:	*+ addition* *- subtraction* ** multiplication* */ division* *! exponentiation (including fractional powers)*

The precedence order of these operators is '!', then '*' and '/', then '+' and '-'. Parentheses take precedence over the operators. Execution not resolved by the precedence order is performed left to right.

Operators may not appear consecutively. In the few cases where this situation could occur, it is easily resolved with parentheses. For example: 3!(-4) is legal while 3!-4 is not.

The Prompt Editor provides significant programming assistance and syntax checking. Syntax is checked prior to acceptance of a block when [EOB] is pressed. Syntax is checked any time the block is redisplayed. This can be forced at any time by pressing the [ENTER] key. Programming assistance includes the automatic maintenance of balanced parentheses. This programming assistance is suspended whenever a SYNTAX ERROR message is being displayed, to allow free editing.

Syntax checking is performed by scanning the block from beginning to end. Characters are displayed on the suitable word address line until a syntax error is detected. Then the balance of the block is displayed on the comment lines with the cursor on the first character of that line, poised for editing. A SYNTAX ERROR message is displayed in the lower left of the screen. The programmer should make the correction and press [ENTER] to again attempt to display the block.

During execution, all fields of the block are computed before the block is executed. If a fundamental block includes an L field (block repeat), the fields are not recomputed with each repeat of the block. Also, G39, G59, and G79, do not assign computed values until the entire block is evaluated.

The precision of arithmetic expressions is 7 digits, from .0000001 to 9999999. When the value is assigned to the word address letter, it is truncated to the precision demanded by that letter for the applicable G code.

20.2 Variables

This feature supports four sets of variables. The variables are referenced by a two letter name with the first letter corresponding to the set type and the second letter corresponding to the specific variable within the set type(i.e. GA,GB,GC,LA,LB,LC). Single letter variables default to an 'L' prefix as a Local Variable (i.e. 'F' is the same as 'LF').

The available variables are:

- Local: LA - LZ or A - Z (except G, E, H, N, P)
- Global: GA - GZ (except G, E, H, N, P)
- PAL: PA - PZ (G, E, H, N, or P are read only)
- System: SA - SZ and Sa - Sz (see Appendix A, Read Only)

Local Variables are variables that are available only to the current program or subroutine. Thus there are six sets of Local Variables: the main program and the five allowable subroutine nesting levels. When a program level is exited, the local variables for that level are lost.

Global Variables are variables that are available throughout a program and its subroutines and subprograms.

PAL Variables are variables that are available to set or use from the program or from the PAL, (note the above restriction).

System Variables are actually system parameters that can be used in computations. They cannot be set from the program. Section 20.12 provides a list of the System Parameters corresponding to specific System Variable names.

Except for the second letter of System Variables, upper versus lower case is ignored.

The precision of all Local and Global Variables is 7 digits, from .0000001 to 9999999. PAL Variables are integer values from 0 to 65535. System Variables precision depends upon the specific parameter referenced(ie. current program number 0 - 250, current X absolute coordinate .0000 - 999.9999).

20.3 Functions

This feature includes a library of arithmetic and logical functions that can be used within expressions. In each case, the function is followed by an expression enclosed in parentheses. The expression is evaluated and the function is applied to the resulting numerical value to yield a final numeric value.

The following functions are included:

Table 20.A
Arithmetic Functions

ABS( ) Absolute value
INT( ) Integer value (Value to the left of the decimal)
FRA( ) Fractional value (Value to the right of the decimal)

LOG( ) Logarithm base 10
LN( ) Logarithm base *e*
EXP( ) e to the power
SQR( ) Square root

SIN( ) Sine of angle in degrees
COS( ) Cosine of angle in degrees
TAN( ) Tangent of angle in degrees

Important: SIN, COS, TAN of arguments greater than or equal to 58761 degrees will cause a 'math overflow' error.

ASN( ) Arcsine with result in degrees
ATN( ) Arctangent with result in degrees
ACS( ) Arccosine with result in degrees

```
Examples:

A(ATN(3.5/4.2))
```

(would assign the local variable A the value 39.80557)

```
W(SQR(3.532 ! 2 + 2.785 ! 2) - .2714)
```

(would assign the local variable W the value 19.95984)

Logical Functions:

EQ( ) Equal to zero
NE( ) Not equal to zero
GT( ) Greater than zero
LT( ) Less than zero
GE( ) Greater than or equal to zero
LE( ) Less than or equal to zero

The logical functions all evaluate to '0' for false or '1' for true. This value can then be used in arithmetic calculations. The logical result can also affect conditional branching, subroutine calls, and subprogram calls (see Section 20.7.).

Note:The logical comparison of two variables is accurate to 10^{-5}, for example: EQ(.000001) will evaluate as true.

```
Examples :

A(EQ(27.01 - 22.2))
```

(would assign the local variable A the value 0 from the false evaluation of the expres-sion. The result does not equal zero)

```
A(EQ(22 - 3 - 19))
```

(would assign the local variable A the value 1 from the true evaluation of the expression. The result is exactlly equal to zero)

20.4 G39 Local Variable Assignment and Parametric Subroutines

G39 provides two functions: Local variable assignment and Parametric Subroutines and Subprograms.

A G39 that contains a subroutine or subprogram call, as referenced by E, H, and/or P fields, is a Parametric Subroutine or Subprogram Call. All word address fields except N, G, E, H, and P pass values to the corresponding Local Variables for the called subroutine or subprogram.

```
Example:

G39 P7 X2.75 Y(LOG(2.65)/2) #
```

calls subprogram number 7. The Local Variable 'LX' for that subprogram is assigned the value 2.75 and the Local Variable 'LY' is assigned the value .2116229 (which is log(2.65)/2). Thus, within the subprogram, the parametric value can be referenced: For example, a command of G01 X(LX) # would position the X axis to 2.75,or a command of G81 Z(2.3 + LY) # would set up a G81 autocycle with a Z of 2.511629.

Similarly, G39 H100 E200 L7 # would jump to a subroutine beginning at N100 and ending at N200 which could then reference the value 7 passed by the 'L' field by use of the Local Variable 'LL'.

A G39 that does not contain a subroutine or subprogram call, serves to set the Local Variables. Thus all word address fields except N, G, E, H, and P provide values to the corresponding Local Variables.

```
Example:

G39 J27.5 U(4.5/2) B(ASN(.5)+25.) #
```

would set Local Variables LJ to 27.5, LU to 2.25, and LB to 55. (which is the arcsin of .5, plus 25.). These can be referenced within the current program, subroutine, or subprogram by using the Local Variables. For example: G01 R(LJ) A(LB/5) # would execute as G01 R27.5 A11. #.

20.5 G59 PAL Variable Assignment

G59 is used to assign PAL Variables. These are variables in the range of 0 to 65535 which can also be accessed by the PAL. The PAL can assign any PAL variable,which can then be referenced in the program by PA through PZ. The program can also assign these variables (except PN, PG, PE, PH, and PP which can only be used by the program, not assigned).

This provides an easy means for a program to accept and use numerical data from the PAL or to generate and pass numerical data to the PAL.

PAL assignments are found on the Diagnostics page. They are the A Flags (variables) from A015 thru A040.

Example:

```
G59 S1456 D(PR ! 2) Q(GT(PV-463.)) #
```

will set PAL Variable PS to 1456, PD to the square of PAL Variable PR, and PQ to 1 if PAL Variable PV is greater than 463 or to 0 if it is not.

G59# (with no parameters) will pause until the control catches up to this block. This would be used if a subsequent command needs to use a variable whose value may be changed by commands that have been interpreted but not yet executed. G59# and G79# behave similarly.

20.6 G79 Global Variable Assignment and Parametric Autocycles

G79 provides two functions: Global variable assignment and Parametric Autocycles.

A G79 that contains a subroutine or subprogram call, as referenced by E, H, and/or P fields, is a Parametric Autocycle. All word address fields except N, G, E, H, and P provide values to the corresponding Local Variables for the called subroutine or subprogram.

Example:

```
G79 P7 X2.75 Y(LOG(2.65)/2) #
```

sets a call to subprogram number 7 as an Autocycle. The Local Variable 'LX' for that subprogram is assigned the value 2.75 and the Local Variable 'LY' is assigned the value .2116229 (which is log(2.65)/2). Thus, within the subprogram, the parametric value can be referenced: For example, a command of G01 X(LX) # would position the X axis to 2.75, or a command of G81 Z(2.3 + LY) # would set up a G81 autocycle with a Z of 2.511629.

Note that the G79 is not recomputed every time the autocycle is executed, only when the G79 itself is executed. The same is true for other autocycles, such as G81 Z(PZ) #.

Similarly, G79 H100 E200 L7 # would set up the Autocycle as a jump to a subroutine beginning at N100 and ending at N200 which could then use the Local Variable 'LL' to reference the value 7 passed by the 'L' field.

A G79 that does not contain a subroutine or subprogram call, serves to set the Global Variables. Thus all word address fields except N, G, E, H, and P provide values to the corresponding Global Variables.

Example:

```
G79 J27.5 U(4.5/2) B(ASN(.5)+25.) #
```

would set Global Variables GJ to 27.5, GU to 2.25, and GB to 55. (which is the arcsin of .5, plus 25.). These can be referenced anywhere in the program or its subprograms by using the Global Variables. For example: G01 R(GJ) A(GB/5) # would execute as G01 R27.5 A11. #.

20.7 Conditional-Branching Subroutines Calls

This set of features includes enhancement to the conditional branching, subroutine calls, and subprogram calls to allow the conditional to be based upon the results of a Logical Function.

If an M word has an expression, with a Logical Function executed last, then the branching or call is based upon the results of that Logical Function. If the function is true, the branch or call will take place; if it is false, the control will proceed to the next block in sequence.

Example:

```
M(GE(GX)) H123 #
```

will branch to N123 if Global Variable GX is greater than or equal to zero.

Note that this expression is not executed as an M function. It is only a conditional expression to determine whether to branch.

If the command were: M(+GE(GX)) H123 # then the Logical Function is not executed last and, because of the "+" before the "GE". Therefore, the block will evaluate and execute as either M1 H123# or M0 H123# depending upon whether GX is greater than or equal to zero.

Similarly: M(NE(LS-456.3)) P25# will call subprogram number 25 if LS is not equal to 456.3.

20.8 Prompting

Under the Prompt Editor, all of the implemented G codes provide prompts for related parameters. This is basically a column of the word address letters with a brief description of the purpose of the word address field to the left of the letter. The area to the right is provided to enter that word address field.

With parametric Subprograms and Autocycles, a means has been implemented to allow the user to define prompts for the related parameters. This is done on the first line of the subprogram, following the subprogram name.

When the programmer enters a G39, or G79 on the Prompt Edit page, prompts will appear for SUBPROGRAM----P, SUB__START__NO--H, and SUB__END__NO----E. When a P number is entered, the name of that subprogram will replace the SUBPROGRAM----P prompt. Further, the defined word address letters and prompts from the first line of that subprogram will be added to the prompt list on the screen. The characteristics of these prompts are equivalent to the prompts for G codes.

Prompt Format:

; **name** , **word address letter 'space'** , **next word address letter . . . . ';' optional comment**

Important: The 'space' field contains a space if the field is optional or a default value. The 'space' field contains an '=' if the field is required and does not have a default value prompt

Example:

```
;SAMPLE,X1.25HORIZ_DIST,Y=VERT_DIST,I_HORIZ_INC,V_VERT_INC;STEP-R
```

Would define a subprogram named SAMPLE with the following prompts:

```
HORIZ_DIST---- X1.25
VERT_DIST----- Y
HORIZ_INC-----I
VERT_INC------V
```

Note that X and Y are required parameters and X has a default value of 1.25. I and V are optional parameters. SAMPLE is the subprogram name. STEP-R is a comment.

These prompts appear when the P field is entered under Prompt Edit entry of a G39, G59, or G79 into the calling program. If the P field is less than three digits, you can press [ENTER] to force the fetching of the prompts. For a three digit P field, the prompts are retrieved automatically after the third digit.

Note that the first block of the subprogram always begins with a semicolon followed by the subprogram name. This can be followed by a comma and one or more prompt fields, each separated by another comma. A comment can be appended to the end by preceeding it with a semicolon.

The prompt fields each begin with the word address letter. If the field is optional, the next character must be a space. If the field is required and you desire a default value, then the default value follows the word address letter.

If the field is required and you do not desire a default value, you may follow the word address letter with an equals sign to separate the word address letter from the prompt for better legibility (an equals after the word address letter is ignored). Next is the prompt characters. Only the first 13 characters will display in the prompt field.

The prompts are strictly designed as a programming aid and have no bearing on the passing of the fields. All fields entered with the G39, G59, or G79 are processed, whether prompted for or not.

Note that G59 does not have a related subprogram. However, it is possible to call a subprogram from a G59. Even through the word address fields in the G59 define PAL Variables and not Local Variables, the subroutine can define prompts which, thereby, identify the PAL Variables to aid the programmer. The subprogram in this case would probably consist of only a M02 to return to the calling program.

Similarly, a G39 can be defined as a parametic subprogram call with prompts. The subprogram could consist of one or more commands to assign the Local Variables passed to it to the equivalent Global Variables. Doing this makes it possible to prompt for Global Variables as well. The prompts are actually for the local variables to the subprogram, but its execution causes these local variable to be assigned to the equivalent global variables.

Example:

Main program: G39 P250 X1.23 K-4.76 L25 #

```
Subprogram P250: ;GLOBALS,X_HORIZ,K_INCREMENT,L_COUNT #
                 G79 X(LX) K(LK) L(LL) #
                 M02
```

20.9 Operator Prompting

If the program number selected for execution on the Auto page or the Checkout Pages is defined with prompts, the control will prompt the operator to enter the values for the required parameters. This is done by pressing the fifth soft key on the Auto Operate page called PARAMETERS. The control will then bring up a screen similar to the Prompt Edit screen, with the prompts that are defined in the first block of the selected program.

The operator can enter the desired word address fields. When [Cycle Start] is pressed, those values become the coresponding local variables for the main program. If the parameters are not defined and the operator exits the PARAMETERS page the control will issue the error message "PARAMS NOT EXECUTED".

This allows for a parametric main program.

20.10 Calculator Function

The calculator function is designed to eliminate the need for a hand held calculator by allowing the programmer to enter an expression under Prompt Edit and then command the control to replace the expression with the computed numeric value of the expression.

To facilitate this, the programmer need only enter the expression as though it were going to be in the program in that form. Then he positions the cursor following the right parenthesis and presses [EQUALS] (=).

The control will respond to the 'equals' by evaluating the expression from the left parenthesis that mates to the right parenthesis immediately in front of the 'equals'. The expression is then replaced by the computed value.

Example:

```
X(2.5 + (27 * (SIN(24.)=)))  will become X(2.5 + (27 *(0.4067366)))

X(2.5 + (27 * (SIN(24.))=))  will become X(2.5 + (10.98189))

X(2.5 + (27 * (SIN(24.)))=)  will become X(13.48189)
```

20.11 System Parameters

The following is a list of the system variables.

Table 20.B
System Variables

VARIABLE	VALUE
SA	NOT USED
SB	NOT USED
SC	NOT USED
SD	PROGRAMMED D PARAMETER
SE	PROGRAMMED ROTARY FEED RATE
SF	PROGRAMMED LINER FEED RATE
SG	NOT USED
SH	NOT USED
SI	CURRENT ABSOLUTE POLAR CENTER X COORDINATE
SJ	CURRENT ABSOLUTE POLAR CENTER Y COORDINATE
SK	NOT USED
SL	NOT USED
SM	NOT USED
SN	NOT USED
SO	TOOL OFFSET NUMBER
SP	CURRENT PROGRAM NUMBER
SQ	NOT USED
SR	NOT USED
SS	CURRENT S CODE
ST	CURRENT TOOL NUMBER
SU	U AXIS POSITION VS MACHINE ZERO
SV	V AXIS POSITION VS MACHINE ZERO
SW	W AXIS POSITION VS MACHINE ZERO
SX	X AXIS POSITION VS MACHINE ZERO
SY	Y AXIS POSITION VS MACHINE ZERO
SZ	Z AXIS POSITION VS MACHINE ZERO
Sa	NOT USED
Sb	NOT USED
Sc	NOT USED
Sd	CURRENT D PARAMETER VALUE
Se	NOT USED
Sf	FEED RATE OVER-RIDE PERCENTAGE (50% = 50.0, 150% = 150.0)
Sg	NOT USED
Sh	NOT USED
Si	CURRENT INCREMENTAL POLAR CENTER X COORDINATE
Sj	CURRENT INCREMENTAL POLAR CENTER Y COORDINATE
Sk	NOT USED
Sl	NOT USED
Sm	NOT USED
Sn	NOT USED
So	FIXTURE OFFSET NUMBER

Table 20.B
System Variables (continued)

Sp	NOT USED
Sq	NOT USED
Sr	TOOL RADIUS (CUTTER COMP) LAST VALUE USED (COMP ON OR OFF)
Ss	SPINDLE RATE OVER-RIDE PERCENTAGE (SSO SWITCH TIMES 10)
St	NOT USED
Su	U AXIS POSITION VS PROGRAM ZERO
Sv	V AXIS POSITION VS PROGRAM ZERO
Sw	W AXIS POSITION VS PROGRAM ZERO
Sx	X AXIS POSITION VS PROGRAM ZERO
Sy	Y AXIS POSITION VS PROGRAM ZERO
Sz	Z AXIS POSITION VS PROGRAM ZERO

20.12 Error Conditions

The following are error messages that can result from this feature:

Table 20.C
Error Messages

ERROR	MESSAGE
Result > permitted word address	X CODE OUT-OF-RANGE
Result < 0 and only positive values allowed	X CODE OUT-OF-RANGE
Missing parameter	X - SYNTAX ERROR
Undefined local variable	LA UNDEFINED
Undefined global variable	GA UNDEFINED
Undefined system variable	Sa UNDEFINED
Function not found	X FUNCTION NOT FOUND
1st letter of 2nd letter variable not G,L,S,P	X - SYNTAX ERROR
Number input > 9999999.	X CODE OUT-OF-RANGE
Number input < .0000001	X CODE OUT-OF-RANGE
Total number of digits entered > 7	X CODE OUT-OF-RANGE

Chapter 21 -- Quick Path and Arc Tangency

21.0 Chapter Overview

This chapter tells you the function of the Quick Path and Arc Tangency Features and shows you how to use them correctly. Applying these features will be easier if you read the chapter completely before implementing this information.

21.1 Quick Path Automatic Blends & Champhers (Q Parameter)

This feature allows the programmer to easily define a blend(radius or fillet) or a champher to be inserted between the current block and the next one.

This feature only blends blocks in the current circular place (G17,18,19).

In addition, this feature provides a means to include a parameter (a Q0) in a G01, G02, or G03 block to cause the control to come into position before proceeding with the next move. A G01 block with this Q0 parameter is like a G73, but is non-modal.

21.1.1 Definitions

Blend: A blend is a radius or a fillet between two moves.

Radius: A radius is considered to be a rounded outside corner.

Fillet: A fillet is considered to be a rounded inside corner. Either of the two programmed moves may be a line or an arc. This feature requires no programmed distinction between a radius and a fillet.

Champher: A champher is a line between two programmed moves which replaces a corner with two less sharp corners. Either of the two moves may be a line or an arc.

FFB: FFB is an abbreviation for a 'Fundamental Feed Block'. That is, a move generated by a G01, G02, G03, G05, or no G. It is not a point-to-point move generated by a G00.

21.1.2 Command Format

The format for this feature simply involves including a Q parameter in any FFB.

This feature applies to current interpolation plane.

If an FFB contains a Q parameter with a value **greater than zero**, it specifies the radius of an arc tangent to the line (or arc) of that block and to the line (or arc) defined by the next FFB. The control will execute the current block to the tangency point with the blend arc, follow the blend arc to its tangency point with the next FFB block, and then execute the balance of the next block.

If an FFB contains a Q parameter with a value **less than zero** (value preceded by minus sign), then the control will use the absolute value as a radius to compute the tangency points as though it was going to insert a blend. However, instead of following a blend arc between the blocks, the control will insert a straight line, thereby forming a champher.

If the G01, 02, or 03 block contains a Q parameter with a value **equal to zero**, the control will wait to come into position before proceeding to the next move. Normally the control will proceed immediately into the next move, which may provide a slight rounding of the corners between moves due to the following error lag. This differs from G73 in two ways: It is not modal; and it works on G02/3 arc.

Consecutive blocks can each contain a Q parameter, thereby giving blends and champhers into and out of a block. The blend is attached to the end of the move defined by the block containing the Q parameter.

21.1.3 Examples

See the G05 Arc Tangency spec for examples of the Blends and Champhers. These two features can be combined to provide very powerful, yet simple to program, capability

21.2 G05 Arc Tangency

The G05 Arc Tangency feature defines linear paths which are tangent to arcs whose radius and center are programmed. The tangency points are automatically determined by the control, alleviating the need for the programmer to calculate these points.

G05 is a powerful companion feature to the Blend & Champher feature. Blend & Champher insert moves between programmed paths and G05 inserts lines to connect specified arcs.

Arc Tangency is an additional modal positioning mode. That is, like G00 and G01, G05 sets the mode of the machine such that subsequent commands which do not contain a G code will default to G05. This mode remains in effect until changed by an applicable G code (G00, G01, G02, G03, or G73).

G05 operates in the current interpolation plane as determined by G17, G18, or G19. This description will assume the XY plane, but the same information applies to the other planes except for the change in axis letters.

Programming a G05 with the center and radius of an arc defined will cause the control to position, at feedrate, along a straight line path from the current location to a point of tangency on the defined arc. The choice between the two possible points of tangency is determined by the sign of the radius specified. The parameters of this arc and this point of tangency can be commanded to be retained (through the D parameter) by the control to allow the programmer to close the path to this arc if he desires.

If the Arc Tangency mode is already active, programming a G05 or a G01 will cause the control to continue along the arc it last approached until it reaches a tangency point from which it can continue (not double back) on a linear path to the new destination. If the command is a G05, the destination can be a tangency to another arc defined in the G05.

If no radius is specified in a G05, or if it is specified as zero, a point is defined instead of an arc.

A G05 with no parameters tells the control to close the Path to the tangency point in an arc previously commanded to be retained(through the D parameter).

21.2.1 Programming Format

The parameters for the G05 command consist primarily of the coordinates for the arc center in the current interpolation plane and a Q parameter to specify the radius of the arc. If no Q is specified, or if Q is zero, the axes coordinates specify a 'point'. If the Q value is **positive**, it specifies the radius of a counter clockwise arc. If the Q value is **negative**, it denotes the radius of a clockwise arc. The direction in either case corresponds to the direction of travel along the arc.

The center of the arc can be programmed using Cartesian coordinated or polar coordinates following the same rules as for the ending point of a G01 block (with no Q). Remember that polar programming is currently available only in the XY plane.

The G05 block is considered to run from the initial tangency point of the current (beginning) arc to the initial tangency point of the arc specified in the G05 block.

Incremental distances are from the center of the current arc.

The G05 parameters are as follows: (Unless otherwise noted, the same details apply as for G01)

A,B,C,I,J,R,X,Y

(Substitute K and Z for other planes) Allowable combinations specify the center of the ending tangency arc.

Q- Specifies the radius of the arc. '+' for counter-clockwise; '-' for clockwise. If **Q** is zero or missing, the destination is a point at the 'center' defined above

N- Is an optional field label

H,E,P- Optionally specify a branch, subroutine, or subprogram call

F- Optionally specifies a feedrate

M,S,T,O- Optionally specify a Miscellaneous function, a Spindle Speed, a Tool Number, or a Tool Offset number, respectively

L- Optionally specifies a block repeat.

D- **Does not** specify block divide in G05 blocks. Instead, if the **D** is present (the value is ignored), it tells the control to save the parameters of the specified arc and its entry tangency point to be used for path closure

A G05 with no arc center specifed is a special command which terminates the G05 mode. The control reverts to the G01 mode.

Important: If a 'D' parameter was specified in a previous G05 block, the G05 mode will continue on to the arc specified in that block. After arriving at the tangency to that arc, the control will follow the arc to the tangency point where the block with the 'D' entered the arc. This provides an easy means to program a closed path.

21.2.2 Ending the Arc Tangency Mode

While the Arc Tangency mode is active:

A G00 will terminate the Arc Tangency mode and position from the current location. (Incremental parameters are still from the current arc center.)

A G01 will follow the current arc to the exit tangency point, just like a G05 block. The control will then complete the G01 block as defined.

A G02 or G03 is only allowable if the control is currently at a 'point' (That is, not on a tangency point of a non-zero arc from the previous G05 command.)

21.2.3 Status Display

To provide the added modal conditions relative to the current positioning, the status display has been changed to provide two modal fields:

```
INTERP MODE:        LINEAR
                    CIRC CW
                    CIRC CCW
                    HELIX CW
                    HELIX CCW

PATH MODE:          CONTINUE
                    PT TO PT
                    RAPID
                    BLEND
                    ARC TAN
```

21.2.4 Errors

The following error messages are new messages which may occur as a result of improper usage of the G05 command or the Blends & Champhers (Q parameter):

Bad blend plane -- A blend was programmed in a block with a non Circle-plane axis.

Q-radius too large -- The Q parameter in the current or previous block specifies a radius which is too large for the space provided. Note: Q-RADIUS TOO LARGE may appear with the block following the block with the faulty Q entry. This block shows the error because Q is only too large after the intersection with the block is known.

Zero move in blend -- A move of zero length was programmed while blending.

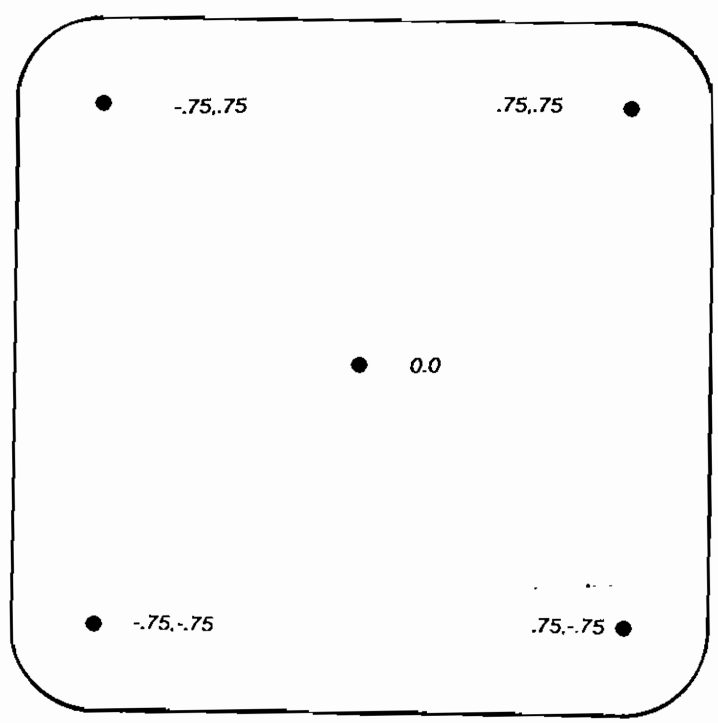
Rapid during blend -- Blending was attempted on a rapid move. Or an arc tangent move was followed by a rapid move.

G02/3 In arc tangent -- A G02 or G03 was programmed following a G05 arc tangent.

21.2.5 Examples

Figure 21-1 - G05 With Q Coordinates

```
G00 X-.5 Y1.1#
G05 X-.75 Y.75 Q.25 D#
Y-.75 Q.25#
X.75 Q.25#
Y.75 Q.25#
G05#
M02#
```

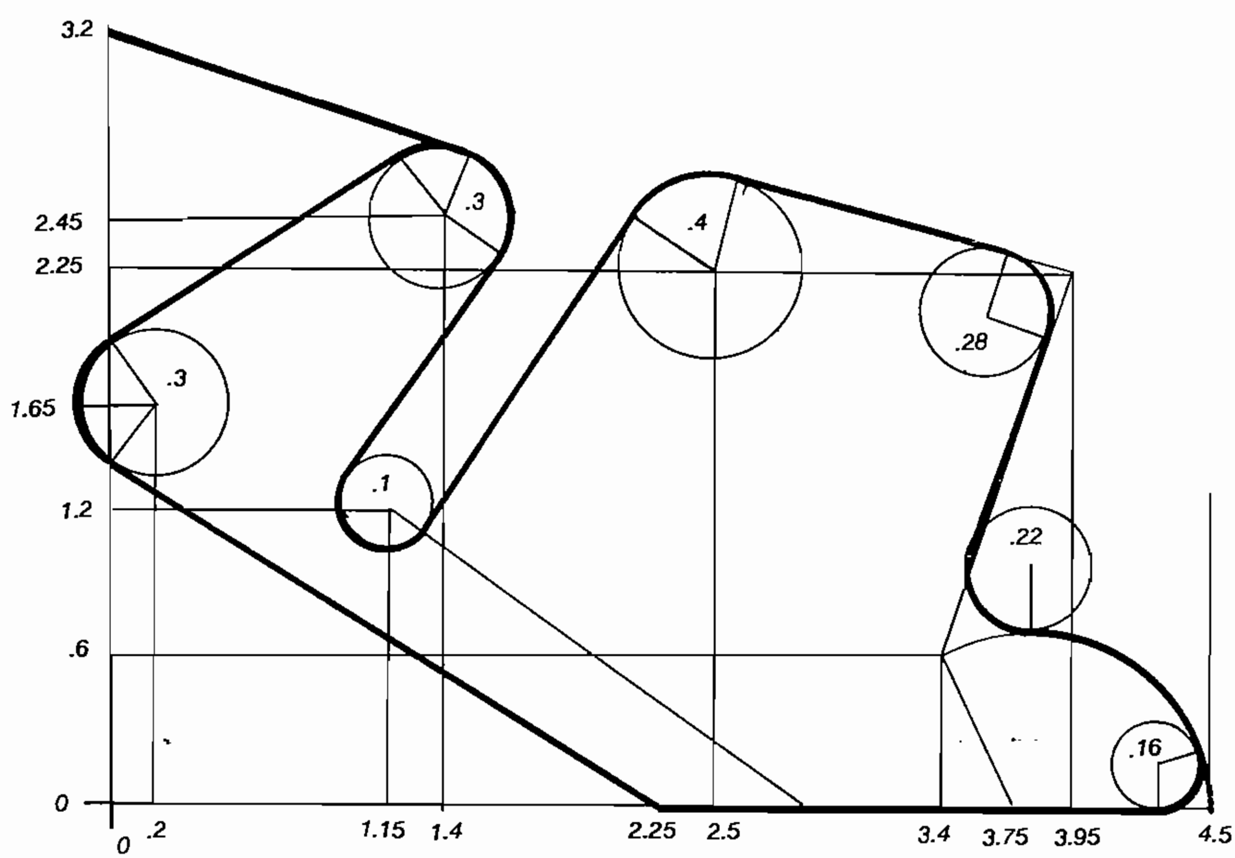


The example of the use of G05, combined with the power offered by the Q parameter of the G01, G02, and G03 blocks.

This example is in absolute Cartesian coordinates. For an appreciation of the power of the G05 Arc Tangency and the Q Blends, you can try to program this example without these features.

Figure 21-2 - Absolute Cartesian Coordinates

```
STARTING AT X = 0,Y = 3.2 IN ABSOLUTE
G05 X1.4 Y2.45 Q-.3D#
X1.15 Y1.2 Q.2#
X2.5 Y2.25 Q-.4#
G01 X3.95 Y2.25 Q+.28#
X3.4 Y.6633 Q.22#
G02 I3.75 J0 X4.5 Y0 Q.16
G01 X2.25#
G05 X.2 Y1.65 Q-.3
G05#
G00 X0 Y3.2#
M02#
```



Appendix Overview

This appendix serves as a guide for the error codes and operator messages that occur while you program and operate the control. The errors and operator messages appear in alphabetical order. We also tell you what the message means, and what you can do to correct the problem.

If you see this message...	It means this...	Notes...
ABORTED BY OPERATOR	You pressed [SHIFT] and [CYCLE START] at the same time.	You have cancelled the operation; control has been reinitialized.
AMP WAS DEFAULTED	AMP was set to a default state and drives may not be enables.	Logon as user 1 and enter Control Parameters page.
ARC ENTRY IN RCOMP	You have programmed an arc (G02, G03) immediately following a G40.	Cutter operation can't be removed with an arc move. Use appropriate linear movement programming to remove compensation before arc entry.
ARITHMETIC OVERFLOW		Contact a qualified Allen-Bradley field service engineer.
AUTOCYCLE REQUIRED	You need a subprogram in G75 or G79.	Correct your program.
AWAITING CYCLE START	Press [CYCLE START] to initiate machine home.	
----AXIS BEING HOMED	The currently displayed axis is being homed.	
AXES NOT IN POSITION	Axes aren't stabilized, possibly due to the auto balance feature.	Press [EXIT], then [CYCLE START].

If you see this message...	It means this...	Notes...
AXIS EXCESS LAG	A particular axis has exceeded the allowed following error; or, an axis is not keeping up with the control's commanded position.	The axis drives will shut down; When you clear the error with EXIT you will get DRIVE NOT on. To set them back on you will have to press and release ESTOP: you must press [EMERGENCY STOP] to remove the condition. You should also examine the machine for any drag producing conditions. You may need to "retune" the drives.
___ AXIS NOT ACTIVE	You have programmed an axis that was not selected in AMP.	Remove the invalid axis word from the program.
--- AXIS HARD TRAVEL	You have hit the hard travel switch.	The axis must be moved off the travel switch with JOG.
AXIS RAN ONTO LIMIT	You hit the soft limits.	You have to jog the axis off the limit.
BAD CRC PLANE AXIS	You've programmed a non-CRC plane axis in the same block as a CRC plane with CRT active.	
BAD FIXTURE NUMBER	You need to enter a fixture number to digitize on fixture offsets page.	
BAD HEX CHARACTER	This may appear when you are loading personality, tool offset, or fixture offset memory, or whenever you are performing a multi-prog. load.	You have to try again or fix the tape.

If you see this message...	It means this...	Notes...
BAD ROUTINE ENTRY	You have programmed a G24 or G25 incorrectly.	Check the G24 or G25 block for faulty coding. You cannot use R and C if both I,
BATTERY LOW	J,X,Y appear. Battery voltage Between 2.04 and 2.12 volts	
BATTERY OPEN	Your battery is dead, removed, or poorly connected.	
BLEND FAILED	The control failed to calculate the correct blend of	Contact a qualified Allen-Bradley field service engineer. two moves.
___CODE NOT KNOWN	You programmed an illegal designator for an axis or an illegal G code	Correct your program as needed
DRIVES NOT ON	The axis drives have not been initialized properly. This is a normal message if you have just exited from the control parameters page.	Press [EMERGENCY STOP] and release,If message remains, refer the problem to qualified service personnel.
____DURING LOOK AHEAD	You have programmed a T,O, multi H,E,L,P, D (block divide), G41, G42, G45, G92, while cutter radius compensation (G40-G42) is active.	Cancel and remove compensation before you execute the program.
EMERGENCY STOP	If the emergency stop button is in, or if you have pressed [EMERGENCY STOP], this message appears.	Make sure operating conditions are safe, rotate the button and pull it out.
END LIMIT <= START	You've tried to insert a program segment that doesn't have the segment start preceding the segment end.	

If you see this message...	It means this...	Notes...
ENTER PATTERN, START	This message is shown on the block search page. You must type in the characters you want the control to search.	1. Enter characters 2. Press [ENTER]. 3. Press [START]. 4. Press [START] to resume the search.
ENTRY OUT OF RANGE	An entry that you programmed on G, M, or Directory Page is out of range.	
EXIT TO CLEAR ERROR	Exit is shown with common error messages.	Press [EXIT] to clear the error condition. Correct the error producing circumstances.
FAULT	The control has detected a programming or system functioning error.	Press [EXIT] or press [SHIFT] and [HALT CYCLE] at the same time to reinitialize the control.
FAULT ABORTED I/O	Punch output was aborted by reception of DLE character from output device.	
FEEDHOLD	The feedhold override has been moved to its 0% setting which holds the axes in position.	
FIRMWARE FAULT		Contact a qualified Allen-Bradley field service engineer.
FUNCTION NOT ALLOWED	You have tried a function that isn't allowed.	This message could appear in a number of places, such as access control restriction or program protection. Check for errors.
FUNCTION NOT ALLOWED	Parametric function such as TAN or ABS not found	correct the paramacro

If you see this message...	It means this...	Notes...
G02/3 IN ARC TANGENT	You programmed a G02 or G03 while G05 arc tangent mode is active.	Switch to G01 before the G02/G03. Yoy may accomplish what you want using the Q (Quick Path) rather than the G05
G ____ILLEGAL ENTRY	You have programmed the shown G code incorrectly.	Check the coding and modify the data block.
G ____IN CUTTER COMP	You have programmed a G17, G18, G19, G30 G31, G90, G91, G92 or G99.	Change the program. Cancel and remove compensation before you execute the codes listed.
G81-G85 REQUIRED	You must program a G81-G85 or G79.	
GA UNDEFINED	Undefined global variable	Define the Global paramacro variable
HALTED	You have stopped execution of a function by pressing [CYCLE STOP] or [BLK/BLK], or an M00-M01 has executed, or axis limits have been violated	Press the green [CYCLE START] or the yellow [BLK/BLK] button reinitialize execution, or use the jog capabilities to move away from the axis limit.
I NEEDED FOR G83	You haven't entered and I value, which defines the maximum initial peck increment.	Define a value to I.
IDLE	The system is error free, out of feedhold, and isn't running a user or system program.	You can perform any operation with the system.
ILLEGAL PAL TRANSFER	The PAL download has received an unknown request for the external download device.	Press [EXIT]

If you see this message...	It means this...	Notes...
IMPROPER CIRC PLANE	Cutter compensation is active and you program simultaneous movement of a non-circ plane with a circ plane axis. The remaining 4 axes may not be combined with the plane axes.	Change the program as required.
IMPROPER CIRCLE AXES	You have programmed an improper axis for linear movement in helical interpolation CW or CCW.	Change the program to specify the total linear movement required for the data block. The remaining axis should be perpendicular to the selected plane of circular interpolation.
IMPROPER G02/G03 ARC	The values you specified in the G02 or G03 circular block result in an impossible move.	Change the program to properly program the center of the arc.
IMPROPER G02/G03 PLANE	Axis entries for G02/G03 circular interpolation don't match the selected plane (XY, YZ, XZ).	Change or correct the plane you selected.
INCH/METRIC CHANGE	You have attempted to change modes from inch to metric or metric to inch.	Normal warning message acknowledging the mode change. Restart the program.
INVALID PROGRAM NO	You have specified an inappropriate program number.	Specify valid program numbers as required.
JI CAN'T BE MINUS	You have entered a jog increment value with a negative number.	Enter a positive increment and use the (+) or (-) axis soft-key to go (+) or (-).

If you see this message...	It means this...	Notes...
J,K missing in G83	You have made a mistake in programming G83, peck drilling autocycle.	Both J and K must appear in programming. If J or K appears in the G83 block, the other must also appear. Correct the program.
KEY HARDWARE FAILURE	The serial link has been lost between the front panel and the processor.	Consult qualified service personnel.
KEY NOT IMPLEMENTED	You have pressed a key which is not used with your particular firmware release.	
LA UNDEFINED	Undefined local variable	define the local paramacro variable
LADDER DISABLED	Downloading ladder	Normal operation during ladder downloading, message will appear until finished.
LADDER EXCEEDS RAM	Downloaded PAL has exceeded its memory.	
LADDER RAM FAILURE	Download ladder doesn't checksum.	
LOAD BREAK ERROR	Load aborted due to "break" condition on input signal.	
LOAD COMPLETE	The loading process is finished.	
LOAD FRAMING ERROR	Serial link data word size does not match what was expected by the control.	
LOAD OVERRUN ERROR	Load aborted due to excess character input.	

If you see this message...	It means this...	Notes...
LOAD PARITY ERROR	Serial link data did not contain the parity expected by the control.	
LOADER ERROR	The control has detected a fault in an external input device.	Examine the device for proper operation.
LOADING AMP	AMP load is in progress.	This message appears during load.
LOADING FIXTURES	Fixtures load is in progress.	This message appears while you are loading the program.
LOADING PROGRAM	A program load is in progress even if a tape reader hasn't been connected to a control.	
LOADING TOOL OFFSETS	A tool offsets load is in progress.	
LOOK AHEAD TOO FAR	The control must process too many non-movement blocks while cutter radius compensation (G40-G42) is active.	Modify the program to reduce the number of blocks between execution of compensation moves.
LOSS OF FEEDBACK	The control has detected a loss of feedback from the motor/encoder packages.	The system will shut down. Consult qualified service personnel.
M-CODE OUT OF RANGE	You have programmed an illegal M code of 100 or greater.	
MACHINE FAULT		Contact a qualified Allen-Bradley field service engineer.
MACHINE HOME REQ'D	You have to perform a machine home before you continue with a particular function.	This message appears after powerup, before MDI, dry run, and auto operate.

If you see this message...	It means this...	Notes...
MATCH FOUND	A program search operation has found a corresponding message.	This message appears during block search.
MEMORY FAIL	The control has detected a fault in its memory while it is running its internal diagnostic program.	The system shuts down and is not functional. Note the address and refer the problem to qualified service personnel.
MISSING POST SIZE	You have programmed a G27 and you're missing a X, Y, or R.	Modify/correct the program block.
MORE THAN 1____CODE	A data block contains more than one word address of the same kind.	Delete the duplicate word address and any numeric value associated with it.
MUST BE IN ESTOP	When you modify personality parameters, the control must be in ESTOP.	Put the control in Emergency Stop.
MUST PRESS ENTER	You must press [ENTER] before you continue with an operation.	
MUST SELECT OPTIONS	You must select options during program insert.	
MUST SET ENTRY MODE	You must initialize mid program start and press [ENTER].	
MUST SET START FIRST	You must press [START] during program insert before you can continue.	

If you see this message...	It means this...	Notes...
NO # IN BLOCK	The control detected a data block that requires an EOB character.	Insert the EOB at the end of the program.
NO # OF ROUGHCUTS	The control detected a G26-G27 that requires you to specify a number for roughcuts.	Check your program for the error.
NO MATCH FOUND	A corresponding code hasn't been found or the search reaches the end of the program.	Press [START] to resume the search.
NO PROGRAMS TO DUMP	You have attempted a save operation, but there are no programs in the control's memory to output.	
NO RETURN FROM JOG	Axis has hit the soft travel limit or ESTOP has occurred and no return from jog was performed.	
NOT IMPLEMENTED	You've tried to use a feature that is not part of the system.	
OUTPUTTING LEADER	The tape is being output.	
P CALL in MDI	You can't call programs or sub-programs for execution from MDI.	Correct your MDI block.
P NESTED TOO DEEP	Successive subprogram calls have gone deeper than 5 levels.	Modify the program to stay within the allowed nesting depth. (More than 5 levels without a return.)

If you see this message...	It means this...	Notes...
PARAMS NOT EXECUTED	You have executed the Parameters screen of the Auto or Check Out pages without defining the variables.	
P--- BAD CHECKSUM (***)	There is checksum error in a sub-program. The error lists the bad program (P) and calling program (***).	Edit the program.
PC INHIBITING START	The integral programmable controller is preventing an operation to begin, when the [CYCLE START] button is pressed.	Make sure you specify initial conditions. Examine the status page for any PC messages.
P____NOT LOADED (____)	You have used a sub-program that isn't in memory. The P number and calling number are indicated.	Enter the required program in memory; change the subprogram call, or renumber an appropriate program.
PROGRAM ___ -NO M02	You haven't ended a subprogram or program with an M02. The control halted execution.	Insert the M02 at the end of the program whose number is displayed.
PROG. CHECKSUM ERROR	You have tried to execute a program that is in memory and has a checksum error.	Programs that have checksum errors are marked in the program directory with a "C" in the "C" flag position. They appear, for example, when a damaged tape is loaded and ACCEPT ERRS option is active.
PROGRAM COMPLETED	A program has been completed.	
PROGRAM #`IN USE	You have tried to renumber a program number that has already been assigned.	Change the program number.

If you see this message...	It means this...	Notes...
PROGRAM LIST FULL	You have stored the limited number of programs (65) in memory.	MDI requires part program memory equal to one part program. It temporarily stores number 252.
PROGRAM # NOT FOUND	The control doesn't recognize the program number you entered.	
PROGRAM NEARLY FULL	You are editing a program and do not have much room to work with.	
PROGRAM SPACE FULL	You have run out of program storage memory.	Delete some programs from memory or transfer them to a storage device.
R ABOVE CURRENT Z	The programmed clearance plane height (R) is above the current Z position.	When the operation is defined and rapid negative Z movement to the R plane is impossible. Change the command or position Z at or above the clearance plane before you define the operation.
RADIUS COMP FAILED		Contact a qualified Allen-Bradley field service engineer.
RADIUS 0 in B DIVIDE	You've programmed a radius of zero length in block divide.	
RADIUS TOO LARGE	The radius is too big for the current tool.	Change the tool.

If you see this message...	It means this...	Notes...
RAPID MOVE IN BLEND	Blending was attempted on a rapid move, or an arc tangent move was followed by a rapid move.	
REC CHECKSUM ERROR	The control has detected a checksum error in the records of loading tool offset data, fixture offset data, or personality parameters.	Load will stop;default values will appear in the personality. Tool offsets and fixture offsets will have zeroes in the positions with faulty checksums. Make a new tape with correct data on it.
REFERENCE LENGTH = 0	You programmed a polar block that has no direction; Or, you have programmed a cavity autocycle with a zero move.	Program a linear move prior to the polar block, or change the polar block. Change the program by programmming a non-zero move.
REQUIRES CHECK RUN	You've tried to make a change in the graphics screen limits while you were in the auto operated mode.	You can only change graphics limits when the control is in check run or test run modes.
REQUIRES IDLE STATE	You've tried to use a facility that requires the IDLE state.	Establish IDLE state before you select the capability.
REQS NON-ZERO PITCH	You must program the lead per revolution as non-zero.	Change the command.

If you see this message...	It means this...	Notes...
RETURN JOG ON START	You are given the option of returning to the point on a program path, OR resuming execution at the point to which you jogged.	Press the green [CYCLE START] to resume execution, OR press [SHIFT] and [CYCLESTART] at the same time to resume execution from the same location.
RTN:CRC PLANE CHANGE	The return from a subroutine changed the CRC plane.	You need to program a G17, G18, or G19.
RUNNING	The control is executing a user or system program.	
SA UNDEFINED	Undefined system variable	Define the system paramacro variable
SAVE COMPLETE	The output has been successfully completed.	
SAVING AMP	Output is in progress.	
SAVING FIXTURES	Output is in progress.	
SAVING PROGRAM	Output is in progress.	
SAVING TOOL OFFSETS	Output is in progress.	
SEQUENCE # NOT FOUND	The control doesn't recognize the sequence # you tried to search.	
SELECT EIA/ASCII	You must select EIA or ASCII format before you initiate a program output.	Select the appropriate format on the save page.
SOFT TRAVEL	An axis has been run to a software defined limit.	The control assumes a halted state; select either jog handwheel or jog continuous to move away from the limit position.
SOFTWIRE LINK ABORT	The PAL download was aborted from the external device.	

If you see this message...	It means this...	Notes...
SOFTWIRE LINK FAILED	The PAL download process was aborted because the external device time expired.	
STORAGE SPACE FULL	You have tried to input data with 65 programs already in memory.	
___ SYNTAX ERROR	There is an illegal, misplaced, or missing value in the block, following the indicated letter. Or the 1st letter of a two letter paramacro is not G, L, S, or P.	Correct the block or the paramacro for these conditions.
SYSTEM IN FEEDHOLD	You have pressed the green [CYCLE START] to initiate an axis move and the feedrate override switch is at 0%.	Select the setting other than 0% to allow the machine to move.
TAPE BLOCK TOO LONG	There is a block that exceeds 64 characters in length.	
TOO MANY CAL POINTS	You have programmed over 1000 CAL points.	Check your program.
TOO MANY ROUGH CUTS	You have programmed too many rough cuts.	Check your program.
U & L IN SAME BLOCK	You have programmed a U & L in a G26.	Delete either U or L.
___UNDEFINED	The indicated letter has no defined value	Probably the result of an incomplete or misplaced paramacro.
VERIFY AMP	Verify in progress.	
VERIFY COMPLETE	The verify function is complete.	

If you see this message...	It means this...	Notes...
VERIFY ERROR	The control has detected an error during verify.	Check your program.
VERIFY FIXTURES	Verify in progress.	
VERIFY PROGRAM	Verify in progress.	
VERIFY TOOL OFFSETS	Verify in progress.	
VERSIONS DON'T MATCH	You have tried to load personality, tool offset or fixture offset memory with data from a control that has an earlier version of firmware.	Firmware versions must be the same level.
WDT TIME OUT	The external system watch dog timer has detected a fault in the system.	The system will shut down. Consult qualified personnel.
X CODE OUT-OF-RANGE	1. Result > permitted word address 2. Result < 0 and only positive values allowed 3. Number input > 9999999. 4. Number input < .0000001 5.Total number of digits entered >7	Correct the paramacro
X, Y, I, J REQ'D IN G25	You have not included X, Y, I, or J when you programmed a G25.	Include either X, Y, I, or J n the block.
XY NEEDED FOR G26	You need to specify X, Y, or R dimensions.	Change the G26 block as required.

If you see this message...	It means this...	Notes...
Z ABOVE R	The final Z axis position in a G26, pocket milling, or G81-G85, is above the established clearance (R) plane.	Change the Z axis position.
Z ABOVE START	Programmed Z in G81-G85 is too small.	Check your program.
ZERO MOVE IN CRC	You have programmed a move having zero length while cutter radius compensation is active.	Modify the program to eliminate the zero length move.
Z NEEDED FOR G81-G85	You need to specify a Z axis movement in a G26, or G81-G85.	Enter an appropriate Z value.

A

B

C

D

E

F

G

H

I

J

L

M

Q

R

S

T

V

W