SIEB & MEYER

SERIES 8 CNC
PRECISION PCB ROUTER
OPERATOR MANUAL

DOCUMENT: 82-50298C

SIEB & MEYER USA, LLC.
4460 Lake Forest Drive, Suite 228
Cincinnati, Ohio 45242
(513) 563-0860 FAX: (513) 563-7576

REVISIONS

Description	Initials	Date
Initial Release	JGE	08/07/06
Add I/O Servolink and HSSB	JGE	08/15/06
Add header programming; restore utility	JGE	02/20/07
Correct mirror code	JGE	03/11/07
Formatting	DJE	12/18/07
Add vision, bar code, G10	JGE	11/19/09

TABLE OF CONTENTS

SERIES 8 CNC Introduction - Precision PCB Router 1
Introduction 2
Control Operations - Basic 1
Starting the System 2
Turning Off the System 2
Display Layout 3
Pull Down Menu 3
Window 1 Display Area (upper left) 3
Operator Panel Area (upper right) 4
Window 2 Display Area (lower left) 9
Window 3 Display Area (lower right) 10
Function Keys (F1- F12) 11
Status Line 12
Control Operations - Detail 1
Main Menu 2
System Menu 2
Show Desktop 2
Exit to Windows 2
Turn off Computer 3
Configure Menu 3
Operator ID 3
Machine ID 3
Change Level 3
Changing to a lower password level 4
Changing to a higher password level 4
Change Password 4
Configure Main Menu 5
Configure Display 8
Offsets Menu 9
Tool offsets 10
Files Menu 11
Delete 12
Edit 12
Backup 13
Copy Files 14
Update APP Files 15
Program Menu 16
De-select Program 16
Document 16
Select Program 16
Draw Program 17
Status Page 20

Program Status 20
MDI 21
Edit Program 22
APP Variables 24
APP Message Page 24
OEM List 25
Search Menu 25
Search Tool 25
Skip Pattern 25
Maintenance Menu 27
Servos Menu 28
PMI Debug Menu 32
PMI Statements Debug Display 32
PMI Variable Debug Display 34
PMI Debug Input/Output Display 36
Servolink Status Menu 39
Load Parameters 41
Store Parameters 41
Recalibrate Touch screen 41
High Speed Serial Bus (HSSB Status) 41
Print Parameters 43
View Log File 43
OEM 44
Parameter Editor 44
PMI Compile 49
Digital Input / Output Definitions 50

Tool Management 1

Tool Tables 2
Automatic Tool Change 3
TOOL CHG Parameters 4
TOOL POD Parameters 5

Part Programming Code Descriptions 1

Letter Code Summary 2
G-Code Summary 2
M-Code Summary 4
Part Programming Examples 5
N-code Sequence Number 6
Program Format Codes (G70-G71, G90-G91) 6
Tool Information Codes (M06, M07) 7
M06 - Tool Feed and Speed Setting with Up and Down 7
M07 - Tool Diameter Setting 8
Tool Number vs. Tool Offset Number: 8
Tool Information Codes using Program Header 9
Position Register Preset (G92) 10
Interpolation Codes (G00-G03) and Plane Selection (G17-G19) 10

G00 Rapid Traverse Positioning 11
G01 Linear Interpolation 11
G02/G03 Circular Interpolation 11
Radius Programming 12
Arc Center Programming: 12
Plane Selection (G17-G19) 13
Part Rotation (G55, G56) 14
Part Scaling (G57, G58) 14
Inserted Arc or Chamfer (G10) 15
M15/M16 Head UP and DOWN Control Codes 16
G40/G41/G42 Cutter Compensation Codes 16
Step & Repeat Part Programming Codes 18
Nested Step and Repeat 20
R-Code: Repeat Step and Repeat Offsets 21
Canned Cycles 21
G81 Drilling Cycle 21
G83 Peck Drilling Cycle 22
G32/G33 Routed Circle 23
G49/50 Rectangular Canned Cycle 23
Fixture Routing Cycles 25
G34/35 Milled Rectangular Cavity 25
G36 Milled Circular Cavity 26
Looping Function to Mill Cavity at Different Depths 26
Program Coordinate Override (G67-G69) 27
Axis Mirror Codes (M931, M941, M932, M942) 28
Feedrate Format Codes (G93, G94) 28
Program Dwell (G04) 29
Fixture Offsets (E-Code) 29
Tool Offsets (D-Code) 29
5.19 Advanced Programming Package (APP) 29
Type I and Type II Data Blocks 30
APP Part Program Rules 30
Use of Variables and Expressions 30
Subscripted Variables 30
Variables 31
System Variables (:) 31
Part Program Variables ($) 33
PMI Variables (@) 34
Global Variables (#) 36
Local Variables (%) 37
Subscripted Global and Local Variables 37
Variable Display Page 38
APP Operators, Functions, and Expressions 39
APP Operators 39
APP Functions 39
APP Expressions 40

APP Type II Commands 40
Assign Value (AV) 41
Zero Global Variables (ZGV) 42
Comment Line (COM) 42
FOR Loop Command (FOR) 42
WHILE Loop Command (WHL) 43
Loop Nesting 43
If-Then-Else (IFT) 43
Combinations of Loops and (IFT) Commands 45
Subroutines and Cycles 45
Storage Location for Subroutines and Cycles 46
Procedure Structure 46
Procedure Naming 47
Subroutine Call 48
Cycle Call 50
Modal G Codes 51
Procedure Nesting 51
Procedure Rules 51
Unconditional Jump (GTO) 51
Part Program Input/Output Codes 52
Display Text (DSP) 52
Wait for Operator Input (KEY) 54
Write Text to File (WRT) 55
Special Procedures: START.APP and END.APP 56
APP Error Messages 57

Part Program Digitizer 1

Setting up the Digitizer 2
Hardware Requirements 2
Parameter Setting 2
Digitizer Operation 3
Starting the Digitizer 3
Entering Tool Information 5
Using the Handheld Pendant 6
Creating a Program Pattern 7
Routing 7
Drilling 8
End of Pattern: Create Step & Repeat 9
Starting a Second Tool 10
Ending the Program 10
Trace Mode: Verifying Program Coordinates 10

Fiducial Recognition 1

Hardware Connection 2
SERIES 8 Ethernet Connection 2
SERIES 8 Parameter Settings 2
Initial Connection Verification 3

Manual Camera Operations 3
Basic Operations 3
Enhanced Camera Functions (M93) 4
Camera Parameter Setting 4
Camera Operation in Part Program 4
Table Fiducial (G86,G87,G88) 4
Zero Shift Based on Fiducials (G63) 6
Zero Shift and Rotation Based on Fiducials (G85) 7
Programming Considerations with Step & Repeat 8
Pattern Skip with Fiducial Check Active 9
Part program review in NO ROUT mode 10
Technical Notes 11
M-codes for Camera Functions 11

Barcode Program Select and Fixture Verification 1

Hardware Connections 2
Parameter Settings 2
Barcode for Part Program Select 3
Mask Type 0 – Numbers Only 3
Mask Type 1 – Actual Barcode 3
Setting the barcode readers 3
PART PROGRAM SELECT BAR CODE READER 4
FIXTURE BAR CODE READER 5
Barcode for Fixture 6
Operation 6
Selecting the program: 6
Scanning the fixture: 6
Special test mode 7
Bar code settings – additional information 8

Software Utilities 1

Restore 2
Basic operation: 2
Application notes: 3
Remote Diagnostics 6
drivemaster2 7
Sprint SmartView 8

SERIES 8 CNC File Locations 1

Chapter:

SERIES 8 CNC Introduction - Precision PCB Router

Introduction

This manual describes the operation and programming of the SERIES 8 CNC control system for the Precision PCB router. This system is based on the machining center package for the SERIES 8 CNC, with some special adaptations for the router requirements.

SERIES 8 Common Components		
Device	SIEB & MEYER Model Name	Description
Motion Control Card	MC84	The main processor for the CNC, this card utilizes the Motorola ColdFire processor, and can be supplied with 2, 4, or 8MB of RAM, as well as 256KB of non-volatile memory, and 1 MB of flash memory. The motion control card communicates with a host PC via an Ethernet interface.
Servo Amplifiers / Frequency Converters	84 Series, SD2	SIEB & MEYER amplifiers incorporate a digital interface. Tuning parameters of the amplifiers are accessed via the CNC or SIEB & MEYER drivemasterII software.
Input and Output Boards		Various I/O configurations are available to interface with the machine. Both sourcing and sinking analog and digital signals are supported.
Industrial PC		Allows the operator to easily back up and select a program. Easy file transfer using Microsoft Windows.
Keyboard / Mouse		Standard with some systems, optional on others, a keyboard and mouse provide access to programming the system.
Handheld Pendant		A handheld pendant is used on systems requiring a part program digitizer.
Touchscreen Monitor		A touchscreen monitor that can stand up to the industries challenging environment. The operator can monitor the status of the machine and get real time diagnostic feedback.

Chapter:

Control Operations - Basic

Starting the System

To start the system, press the front panel pushbutton labeled "CONTROL ON". Pressing this starts power to the control system, and initiates the following sequence:

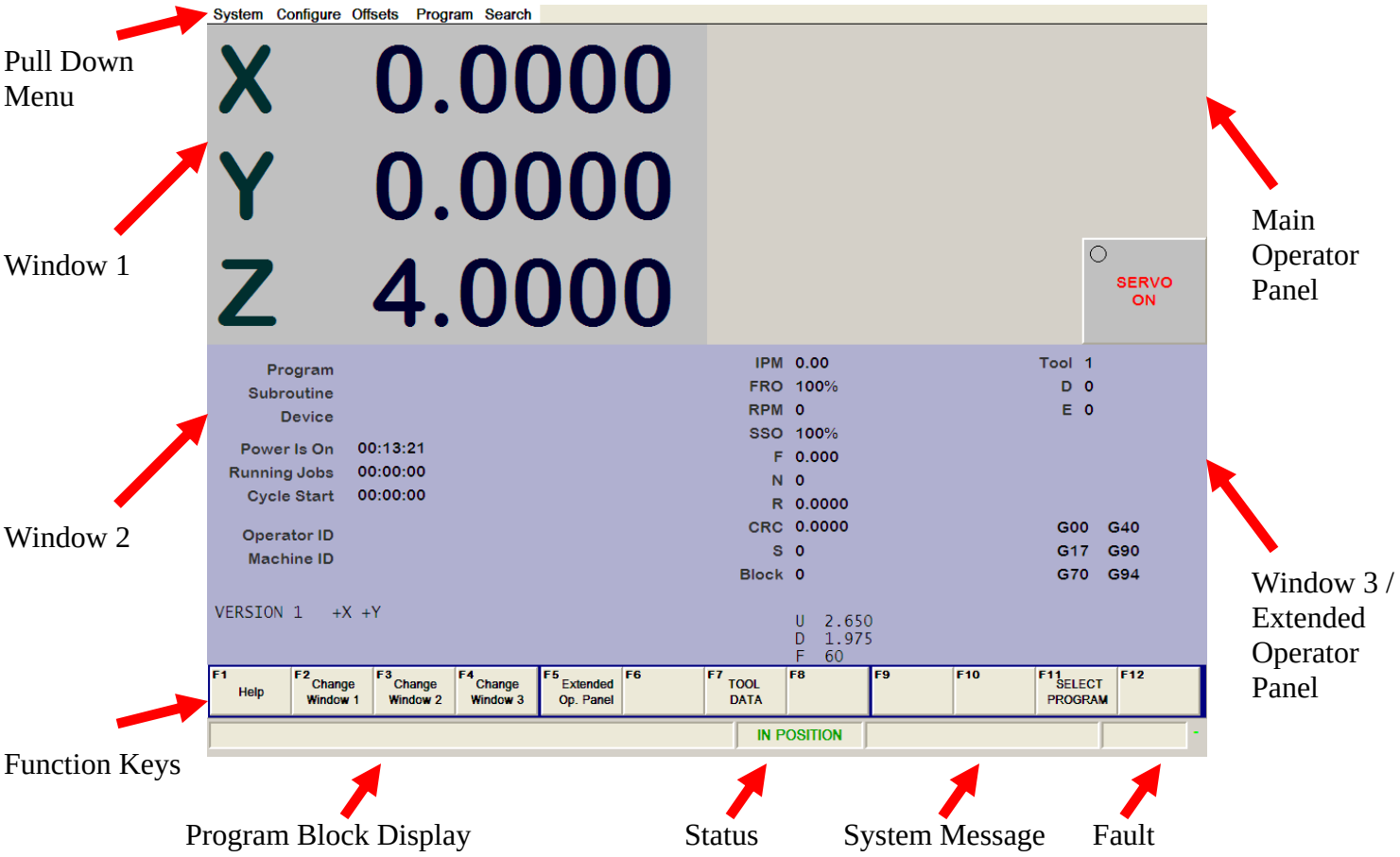
1. The PC loads the Windows software
2. Immediately after windows is loaded, the SERIES8.EXE file is executed, which start the PC part of the SERIES 8 software
3. The primary CNC software is then downloaded from the PC to the MC84 board
4. Parameter files for the machine are loaded
5. The machine interface program (PMI) is compiled and copied to the MC84 board
6. The total CNC software begins execution

Turning Off the System

When turning off the CNC system, it is important that the software be properly shut down before removing power. Pressing the pushbutton labeled "CONTROL OFF" initiates the power down sequence. There is also a power off command in the SYSTEM pull-down menu.

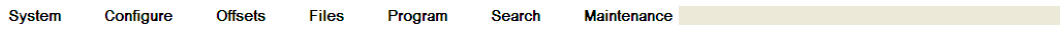
In order to shut the system down, the servos must be off. When the CONTROL OFF request is made, there is an 'Are you sure?' prompt which must be acknowledged prior to the system beginning its shut down sequence. Once the control power is off, then it is OK to disconnect main machine power.

Display Layout



Pull Down Menu

Most control commands are initiated from the pull-down menu at the top of the screen. These can be accessed via touch (on systems with touch screen), or using the cursor with a mouse or trackball. Menu commands can vary by specific machine, and by the active password level. The typical main menu is shown below:



Touching the screen in the area of any of these words displays menu options for specific commands.

Window 1 Display Area (upper left)

The Window 1 area typically shows the current axis positions.

Other typical displays in this window include:

- spindle status information
- servo status information
- auxiliary axis information

Pressing the F2 function key cycles through available displays in the Window 1 display area. The display can also be changed by 'double-clicking' this window.

Operator Panel Area (upper right)

The operator panel area contains touch keys for machine operation. This window contains the primary operator function keys; additional keys are available in Window 4 (lower right), which can be accessed by pressing the F5 (Extended Operator Panel) softkey.

When software is first loaded, the following keys are displayed:

DOOR OPEN			SERVO ON

Operator Panel Touchkey Definitions	
SERVO ON	Turns the servo drives on, enabling power to the servo motors. Servos must be on for the axes to move. To turn off the servos, press the red SERVO OFF pushbutton on the front panel of the machine
DOOR OPEN	Raises / lowers the machine front door. The red LED is on if the door is open. The door automatically closes when a program starts, and opens when a program completes.

After pressing SERVO ON, the touchkey display is:

X	Y	Z	CONTROL RESET
JOG -	JOG +	JOG MODE	AUTO MODE
JOG LOW	JOG .1 IN	HOME ONE AXIS	CYCLE START

The operator may move axes around if needed, but generally the first step after turning servos on is to home all axes by pressing AUTO HOME. Axes must be homed in order for part programs to be executed.

SERVOS ON Touchkey Definitions	
X, Y, Z	Axis selection keys. An axis must be selected prior to jogging or homing for manual moves. If the LED is on, then the axis is selected. If the color of this key is red, then the axis has not yet been homed.
CONTROL RESET	Pressing this does the following: clear any fault messages, assuming the cause of the fault has been corrected; reset the part program to the beginning and clear any program codes to their default state. The Z-axis also retracts to its zero position.
JOG - / +	Pressing this key jogs the selected axis in the requested direction. The type of jog depends on the status of the "JOG HIGH/LOW" key (continuous jogging), or the "JOG .xxx" key (incremental jogging).
JOG MODE	This key toggles between key displays of jogging functions, and cycle control functions. JOG MODE is active upon initial startup of the software, until all axes are homed.
AUTO HOME	Home all axes in sequence to locate their starting position. All axes must be homed after starting the CNC before a program cycle can be executed. Once all axes are homed, this key is available only on the extended operator panel (F5).
JOG LOW (HIGH)	Pressing this key switches between HIGH and LOW jog feedrates for continuous jog moves. Continuous jogs move the selected axis for as long as the JOG + or JOG - key is pressed.

SERVOS ON Touchkey Definitions	
JOG .1 IN (.xxx IN)	Pressing this key selects among various jog increments for incremental jogging mode. In incremental jogging, the selected axis moves the selected amount for each pressing of the JOG + or JOG - key.
HOME ONE AXIS	Pressing this key runs the homing cycle for the selected axis only.
CYCLE START	Pressing this key runs the active part program. Prior to running a part program, all axes must have been homed.

After all axes have been homed, JOG MODE is turned off, and other display keys appear. It is possible to toggle JOG MODE off and on at any time when a program is not executing to access the jog keys or cycle keys as needed. With JOG MODE off, the keys are displayed as follows:

OPTIONAL STOP	BLOCK DELETE	RAISE PRS FOOT	CONTROL RESET
TABLE FRONT	TABLE REAR	JOG MODE	Z RETRACT
OPEN DOOR	SINGLE	FEED HOLD	CYCLE START

JOG MODE Touchkey Definitions	
OPTIONAL STOP	Pressing this key toggles the OPTIONAL STOP feature ON/OFF. If the LED is on, then OPTIONAL STOP is active. If ON, then part program block execution stops on an M01 block, otherwise the M01 block is ignored.
BLOCK DELETE	Pressing this key toggles the BLOCK DELETE feature ON/OFF. If the LED is ON, then BLOCK DELETE mode is active. If ON, then blocks that contain the "/" character at the front of the block is skipped during program execution.
RAISE PRS FOOT	Pressing this key overrides the automatic pressure foot control and forces the pressure foot to its up position. This feature is active with the LED is ON.
CONTROL RESET	Pressing this does the following: clear any fault messages, assuming the cause of the fault has been corrected; reset the part program to the beginning and clear any program codes to their default state. The Z-axis is also retracted to its zero position.

JOG MODE Touchkey Definitions	
TABLE FRONT	Pressing this key starts a sequence to park the machine table: raise the Z-axis to its full UP position, and then move XY so the table is at the front of the machine.
TABLE REAR	Pressing this key starts a sequence to park the machine table: raise the Z-axis to its full UP position, and then move XY so the table is at the rear of the machine.
JOG MODE	This key toggles between key displays of jogging functions, and cycle control functions.
Z RETRACT	Pressing this key retracts the Z-axis to its full UP position. The LED is ON when Z is fully retracted.
DOOR OPEN	Raises / lowers the machine front door. The red LED is on if the door is open. The door automatically closes when a program starts, and opens when a program completes.
SINGLE	Pressing this key toggles SINGLE STEP mode off and on. If ON, then executing a program with the CYCLE START key executes one block at a time. When OFF, blocks execute continuously
FEED HOLD	Pressing this key during an active axis move or part program execution stops machine motion immediately. If a part program move is stopped, then pressing CYCLE START resumes execution (pressing CONTROL RESET cancels move).
CYCLE START	Pressing this key runs the active part program. Prior to running a part program, all axes must have been homed.
CYCLE STOP	Pressing this key with a part program executing halts program execution at the end of the current move (program block). This key appears only if a program is currently executing.

The extended operator panel area is located in Window 3, directly below the main operator touch keys, and contains additional touch keys for machine operation.

Pressing the F5 function key displays the extended operator panel. Pressing F5 again returns to the normal Window 3 display.

The typical extended operator keys are shown below:

OPTIONAL STOP	BLOCK DELETE	RAISE PRS FOOT	ZERO SET
	SPINDLE ON	VACUUM ON	VACUUM OFF
COLLECT OPEN	PRS FOOT DOWN	PRS FOOT CLAMP	SERVO ON

Extended Operator Touchkey Definitions	
OPTIONAL STOP	Pressing this key toggles the OPTIONAL STOP feature ON/OFF. If the LED is on, then OPTIONAL STOP is active. If ON, then part program block execution stops on an M01 block, otherwise the M01 block is ignored.
BLOCK DELETE	Pressing this key toggles the BLOCK DELETE feature ON/OFF. If the LED is ON, then BLOCK DELETE mode is active. If ON, then blocks that contain the "/" character at the front of the block are skipped during program execution.
RAISE PRS FOOT	Pressing this key overrides the automatic pressure foot control and forces the pressure foot to its up position. This feature is active with the LED is ON.
ZERO SET	Pressing this key resets the position offset of the selected axis so that its current position is Program Zero. Manually set ZERO SET positions are overridden if the part program contains a G92 code.
SPINDLE ON	This key provides a manual method to turn the spindle on. The spindle is normally automatically turned on from the part program.
VACUUM ON	This key provides a manual means to force the vacuum door open. Normally the vacuum is controlled automatically via part program execution.
VACUUM OFF	This key provides a manual means to force the vacuum door closed. Normally the vacuum is controlled automatically via part program execution.
COLLET OPEN	This key is only active if the spindle is stopped. When pressed and the LED is on, the spindle collet is open. Pressing the key again closes the collet.
PRS FOOT DOWN	Pressing this key lowers the pressure foot; pressing it again raises it.
PRS FOOT CLAMP	Pressing this key lowers the pressure foot with high air pressure.

Extended Operator Touchkey Definitions	
SERVO ON	Turns the servo drives on /off, enabling/disabling power to the servo motors. Servos must be on for the axes to move. The LED is on if servos are enabled.

Other keys may appear in the main touch key area, or the extended touch key area depending on machine options and current status. These keys are defined below:

Other Operator Touchkey Definitions	
SKIP W/ FID CHCK	This key appears if the camera fiducial check is turned on (G86), and is used when it is desired to skip patterns in a program but still read the fiducials. Press this key first, then CYCLE START to run program up to the fiducial check blocks. Then a SKIP PATTERN command may be executed.
Z POSN CAMERA	On systems with cameras, this key will automatically position the Z axis to the camera height set in status management. This height applies for digitizing (teach mode), or part fiducial checking.
CAMERA CONNECT	This key is available when servos are off on systems with a camera installed. It defaults to being on, but when turned off, it will have the software not attempt to connect to the camera.
SEL PROG BAR CODE	On systems using the bar code option to select part programs, this key toggles between using the bar code scanner, or manually selecting the part program from the program menu. If bar code verification of the fixture is enabled in the Parameter Editor, then it is also required that **SEL PROG BAR CODE** is on to have the program do the fixture check.

Window 2 Display Area (lower left)

The specific display in the Window 2 area can be configured for different machine applications, but typically contains part program and cycle information. The initial display for this window is called the "production page":

Production Page	
Program	The name of the currently selected part program.
Subroutine	If a subroutine is executing, this displays the name of the active subroutine.
Device	Device name from where the program was selected (e.g. “C:”)
Power is On	Timer that indicates how long the CNC has been powered on
Running Jobs	Timer that indicates how much time the CNC has been running a program
Cycle Start	Timer that indicates how long the current program has been running
Operator ID	Text area for operator ID information
Machine ID	Text area for machine ID information

In addition, there are three lines for text messages that can be displayed from the PMI program.

Other typical displays in this window include:

- part program blocks

- fault messages
- digital inputs

Pressing the F3 function key cycles through available displays in the Window 2 display area. The display can also be changed by 'double-clicking' this window.

Window 3 Display Area (lower right)

The specific display in the Window 3 area can be configured for different machine applications, but typically contains information regarding the modal status of certain part program codes.

Program Page	
Text	Description
IPM (or MMPM)	Current vector feedrate of any axes in motion
FRO	Feedrate override
RPM	Current spindle speed
SSO	Spindle speed override
F	Current programmed F-code (feedrate)
N	Current programmed N-code (block label)
R	Current programmed R-code
CRC	Current tool radius used for cutter compensation
S	Current programmed S-code (spindle speed)
Tool	Current programmed T-code (tool number)
D (upper right area)	Current programmed D-code (tool offset)
E	Current programmed E-code (fixture offset)
U (lower left area)	Current programmed clearance position for Z-axis
D	Current programmed rout depth for Z-axis
F	Current programmed infeed rate for Z plunge moves
G00 G40 G17 G90 G70 G94	Current status of G-codes in modal groups

Other typical displays in this window include:

- digital outputs

Pressing the F4 function key cycles through available displays in the Window 3 display area. The display can also be changed by 'double-clicking' this window.

Function Keys (F1- F12)

The 12 function keys across the bottom of the display are defined as follows:

Function Keys		
Text		Description
F1	HELP	Displays the current release version of SERIES 8 CNC software
F2	Change Window 1	Changes the display of window 1 area to the next display option.
F3	Change Window 2	Changes the display of window 2 area to the next display option.
F4	Change Window 3	Changes the display of window 3 area to the next display option.
F5	Extended Op. Panel	Changes the window 3 area to show the extended operator panel softkeys. Pressing it a second time will revert back to the normal window 3 display.
F6		User Defined
F7	Tool Data	Selects the Program Status window, enabling access to tool information.
F8	MDI	Activates the Manual Data Input function to allow entering in a single program block to be executed by the control.
F9	Digitize	This key activates the program digitizer function, if this option is available on the machine.
F10	Tool Offsets	This key calls up the TOOL OFFSET page, primarily used to set the tool diameter.
F11	Select Program	This key is used to select a part program for execution.
F12		User Defined

Status Line

The status line at the bottom of the display contains the following elements.

Status Line	
Text	Description
Program Block Display	Shows the currently executing part program block. Touching the screen in this area displays the part program block display page (in Window 2)
Status	Shows current status of axes 'IN POSITION"
System Message	Contains one of the following messages: • EMERGENCY STOP (servos are off) • FEEDHOLD(axis motion halted) • FRO = 0(feedrate override is set to zero)
Faults	Contains the text "FAULT" is there is a new message shown on the faults page display (Window 2 area). Touching the screen in this are displays the faults page

Chapter:

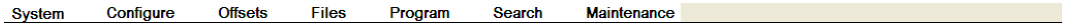
Control Operations - Detail

The SERIES 8 is a touch-screen based system, allowing most operations to be done directly on the screen. In cases where data entry is required, a touch keyboard is displayed on the screen. If a keyboard and mouse are available, it is possible to use touch keys or keyboard keys interchangeably.

In the menu descriptions shown below, all of the possible commands are described even though some may not be present on certain systems due to password level settings or machine functionality.

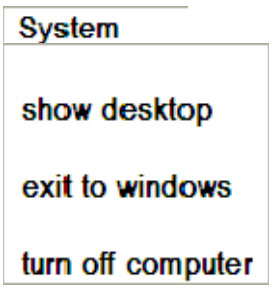
Main Menu

Many operator commands are performed by using a pull-down menu. The main menu appears at the top of the screen. The menu choices vary depending on the operator's password level and the menu configuration of the specific machine. Here is a typical main menu:



To select a menu, either click on the text for that menu header, or hold down the Alt key on the keyboard and press the key corresponding to the first leader of the menu header (i.e., Alt-C for the Configure menu, Alt-O for the Offsets menu, etc.). Alt-S activates the SEARCH menu, Alt-Y activates the SYSTEM menu.

System Menu



The system menus interact with Windows functions. Typically, these functions require that servos be off in order to function.

Show Desktop

This displays the Windows desktop, yet maintain the SERIES 8 CNC software executing, allowing the operator to perform other Windows operations. Normally, this requires servos to be off, but this requirement can be over-ridden in the SERIES 8 INI file.

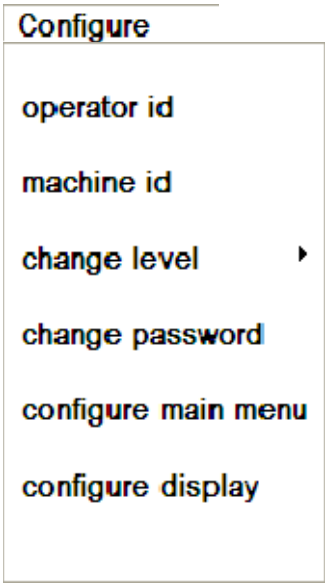
Exit to Windows

This halts execution of the SERIES 8 software and shows the Windows desktop. To restart the SERIES 8 software, it is necessary to double-click the SERIES 8 icon on the desktop.

Turn off Computer

This initiates the Windows shut down sequence to turn off the computer in a proper sequence. This same function is generally available on most systems with a CONTROL OFF pushbutton on the operator panel.

Configure Menu



This menu allows changes to the Operator and Machine Identifiers, current password level, passwords, timers, and the format of the main menu and displays.

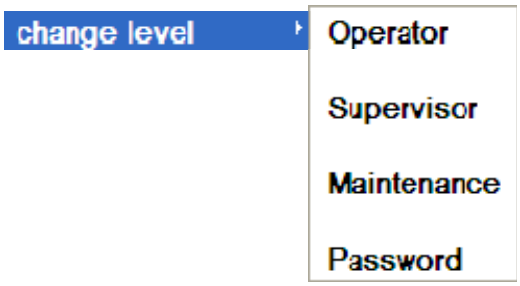
Operator ID

This command allows changing of the operator identifier. The operator identifier is displayed on the production panel and stored in certain log files. To change the operator identifier, use the keyboard to enter a new identifier.

Machine ID

The machine identifier is displayed on the production panel and stored in certain log files. To change the machine identifier, use the keyboard to enter a new identifier.

Change Level



The SERIES 8 supports four levels of operator control. At each increasing password level, more commands are available to the user. The password levels from lowest to highest are as follows:

- Operator (default on power up)
- Supervisor
- Maintenance
- OEM*

** OEM level is reserved for the machine tool builder, and this level should not be used on a regular basis as it allows changing of basic machine parameters which could cause damage to the system.*

The password level determines which commands are available. At power up, the system is always at the operator level. Changing the password level is handled differently depending on whether the new level is higher or lower than the current level:

Changing to a lower password level

When the change level command is pressed, a command window is displayed showing all password levels below the current level, as well as the command "password". To move to a lower level, simply press the name of the level to be made active.

Changing to a higher password level

When the change level command is pressed, a command window is displayed showing the command "password". Pressing the password command displays a touch keyboard, and the password for the desired level can be entered.

To change the password level, use the keyboard to enter the password for the new level.

Note: If the Enter key is press without entering a password, the password level is set to Operator (the lowest password level, with the least number of commands).

Inherent in the password is the level that is associated with it:

- Supervisor passwords must start with an "S"
- Maintenance passwords must start with an "M"
- OEM passwords must start with an "O"
- (There is no password for Operator level)

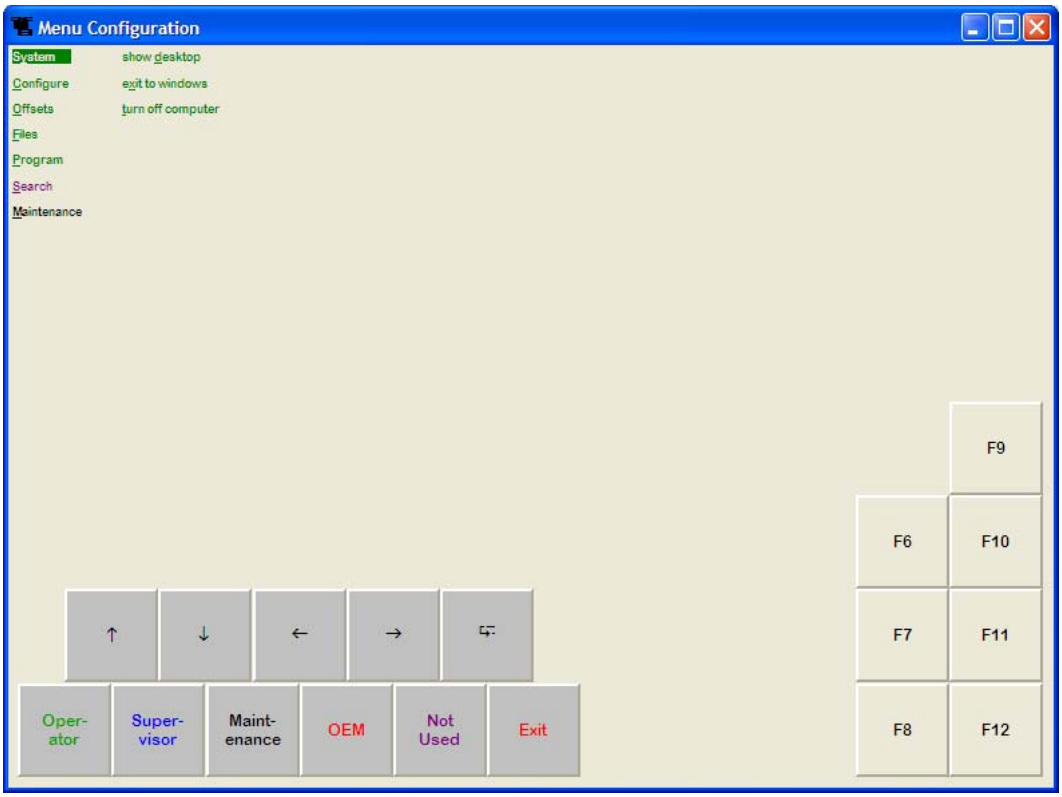
Change Password

To change the password for a given password level, a higher password level must already be selected (i.e., at Maintenance level, the passwords for the Supervisor or Maintenance levels). The first letter of the password must match the password level: S for Supervisor, M for maintenance, and O for OEM.

Password must be from 6 to 8 characters long, and the first character must match the password level associated with that password. From any given level, the password for that level and for any lower levels may be set.

Pressing the change password key causes the touch keyboard to display. The password must be entered twice exactly the same for the password to be changed.

Configure Main Menu



This command allows configuration of system commands: The following capabilities are provided:

- Password level access can be assigned to all commands; commands may also be defined as 'not available' at any password level.
- Location of commands within the pull-down menu can be changed.
- Commands may be assigned to the F6 through F12 function keys.

The configuration page lists the top-level pull-down commands on the left. All commands are color-coded by the password level currently assigned to them:

Green	Operator
Blue	Supervisor
Black	Maintenance
Red	OEM

Magenta not used

With the cursor on a specific top-level menu command, the next column shows the complete list of commands possible in that pull-down. These commands are color coded also with the password level associated with the command. Moving the cursor to the right (into the column with the displayed commands), allows access to changing the password level for these commands.

For some commands, there is a third tier available which displays in a third column when the cursor is placed on a command that supports additional commands.

Positioning the cursor to a specific command, and then pressing the key on the bottom of the screen that corresponds to a password level sets the command to that password level.

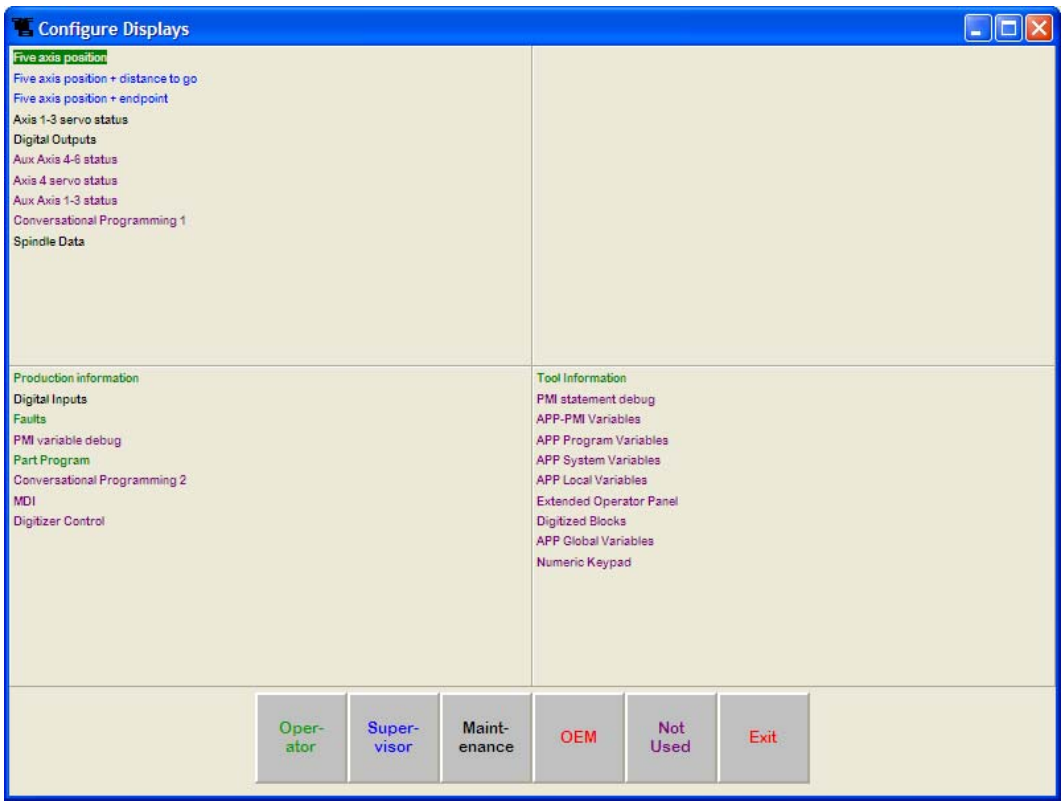
With the cursor positioned on a command, it is also possible to assign this command to a function key (F6-F12) by pressing the appropriate key on the right-hand side of the display. The key shows in green when it is assigned to a command:

Note: With this function, assign a function key to a specific command. The text displayed on the function key is not automatics however, and must be defined in the "NEXTEXT.ENG" file.

Configure Main Menu	
Button	Description
↑	Moves the cursor up one position
↓	Moves the cursor down one position
←	Moves the cursor left one position
→	Moves the cursor right one position

Configure Main Menu	
Button	Description
↳	Moves the command at the current cursor position down one row in the menu. Keyboard shortcut “Alt + S”.
Oper-ator	The menu or menu entry at the cursor is accessible to all password levels. Keyboard shortcut “Alt + 0”
Super-visor	The menu or menu entry at the cursor is accessible to Supervisor or higher password levels. Keyboard shortcut “Alt + 1”
Maint-enance	The menu or menu entry at the cursor is accessible to Maintenance or higher password levels. Keyboard shortcut “Alt + 2”
OEM	The menu or menu entry at the cursor is accessible to OEM or higher password levels.
Not Used	The menu or menu entry at the cursor is inaccessible to all password levels. Keyboard shortcut “Alt + 4”
Exit	Exit and save the changes made. The changes are automatically saved to the hard disk in a setup file and are restored every time software is loaded. Keyboard shortcut “Esc”
F6-F12	User-definable keys. Any of the pull-down menu commands may be configured to execute in 'short-cut' fashion by being assigned to one of these keys.

Configure Display



This page allows the user to change the display system as follows:

- Specify the password level required to access any of the display panels. To do this, touch the line which describes the display (or click on it with the mouse). Then press the corresponding key on the bottom of the screen to set the desired level.
- Change the order of displays within a window. To do this, touch and hold the line which describes the display and move it up or down within the window (may also be dragged with a mouse).
- Move a display to any of the three display windows. To do this, touch line to be moved, and drag it to the new window.

Note: Certain display lines default to "not used" and only appear when a specific command is executed, such as "PMI variable debug". Changing the password level of these has no effect, nor changing their order in a display. It is possible however, to move these displays to a different window.

Configure Display	
Button	Description

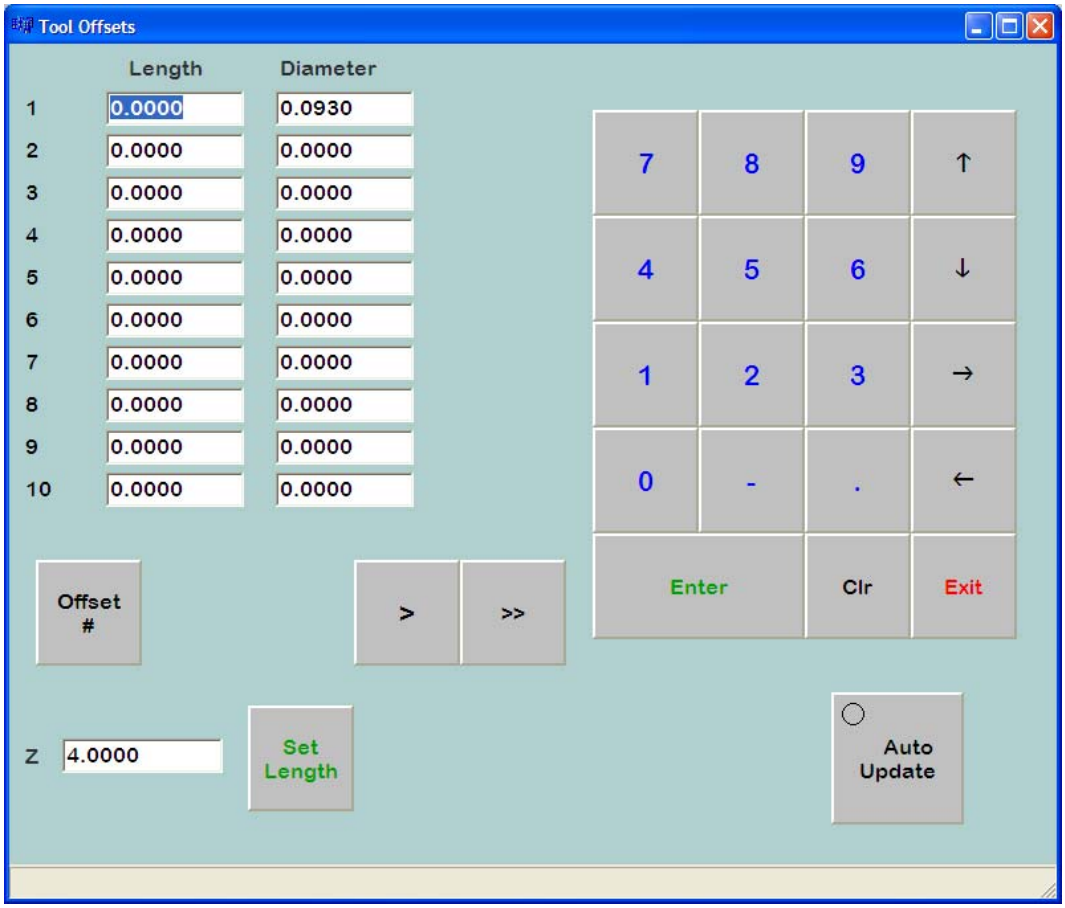
Configure Display	
Button	Description
Oper-ator	The display entry at the cursor is accessible to all password levels. Keyboard shortcut “Alt + 0”
Super-visor	The display entry at the cursor is accessible to Supervisor or higher password levels. Keyboard shortcut “Alt + 1”
Maint-enance	The display entry at the cursor is accessible to Maintenance or higher password levels. Keyboard shortcut “Alt + 2”
OEM	The display entry at the cursor is accessible to OEM or higher password levels.
Not Used	The display entry at the cursor is inaccessible to all password levels. Keyboard shortcut “Alt + 4”
Exit	Exit and save the changes made. The changes are automatically saved to the hard disk in a setup file and are restored every time software is loaded. Keyboard shortcut “Esc”

Offsets Menu

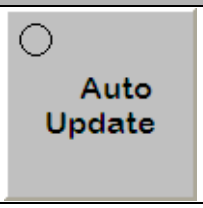
This menu allows access to the Tool Offset.

Tool offsets

On this page, the operator can modify the tool length and diameter offsets.



Tool / Cycle Offsets	
Button	Description
Offset #	Enter the tool offset you want to see; it will be displayed and selected. Keyboard shortcut “#”
Set Length	Sets the length offset for the selected tool to the current Z axis position. Keyboard shortcut “Z”
Clr	Clears the selected cell value. Keyboard shortcut “Delete”

Tool / Cycle Offsets	
Button	Description
	On systems with this feature implemented, this allows a remote device to capture position of the axes and assign it to the current fixture.
	Selects the cell below the current cell. If the cell is the in the bottom row of a column the top cell in the next column is selected. Keyboard shortcut “Enter”

Up to 200 tool offsets may be defined. Tool offsets are activated by programming a D-code.

Example:

```
D5    ; activate tool offset number 5
```

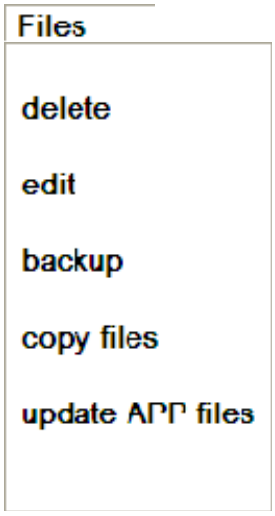
In many applications, if a D-code is not specified with a new tool code, the system defaults to assigning the D-code the same as the T-code.

Example:

```
T4    ; selects tool 4 and activates tool offset number 4
```

Files Menu

This menu allows for the manipulation of files.

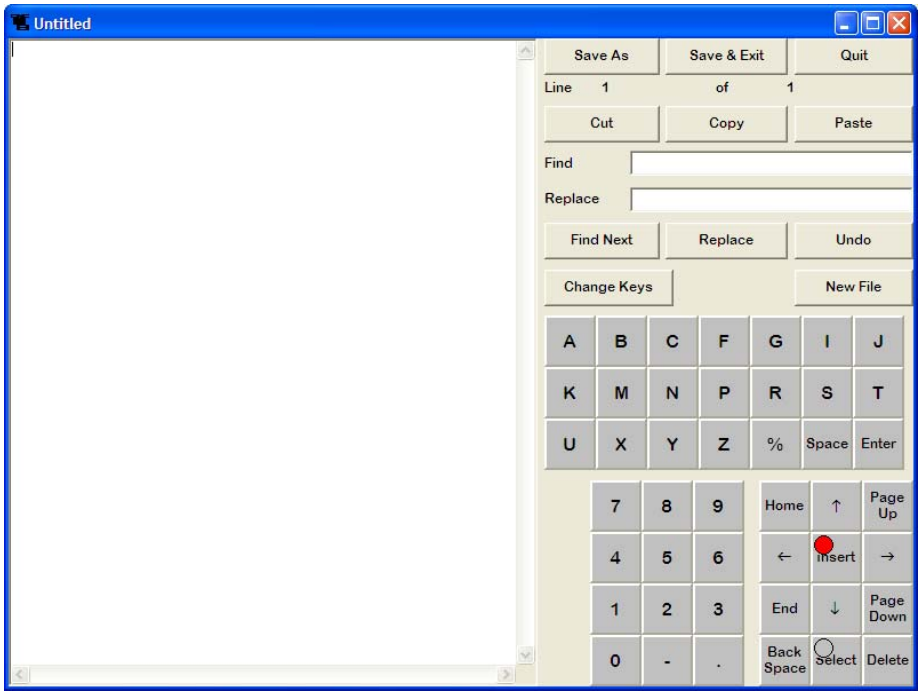


Delete

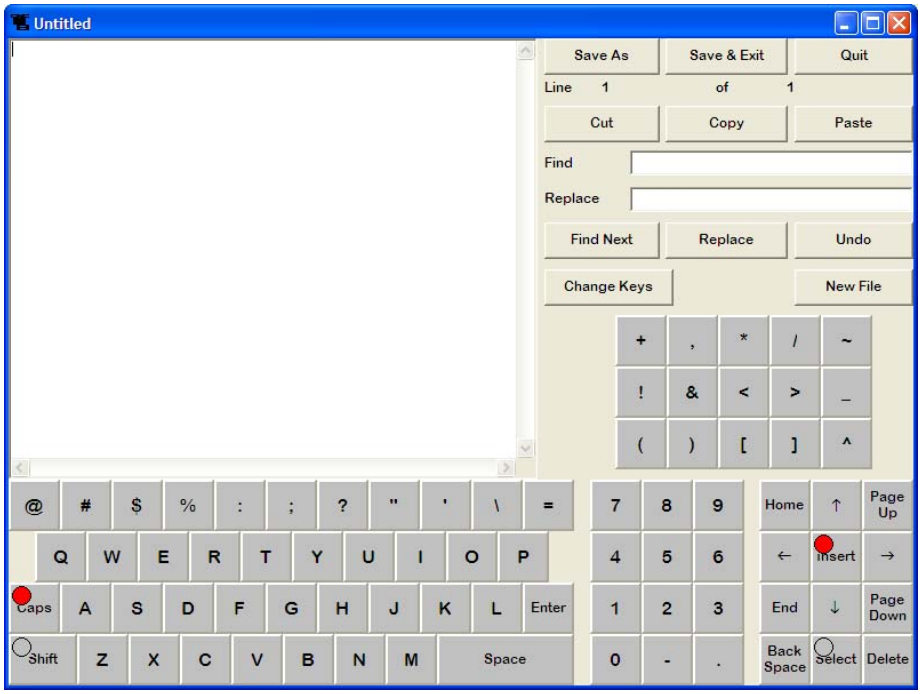
A Windows dialog box is shown, after selecting a file, the selected filename is shown along with a prompt "Are you sure?". If "OK" is selected, the file is deleted. If "Cancel" is selected, the file is not deleted.

Edit

Use to edit any text file on the CNC.



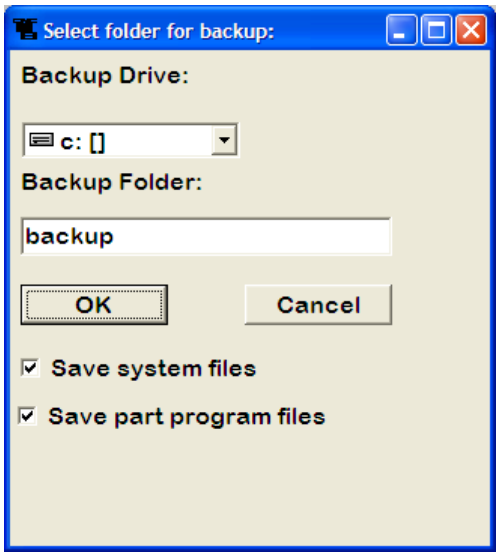
Pressing the “Change Keys” softkey to display a QWERTY keyboard and common symbols used in SERIES 8 CNC programming.



Pressing the “New File” softkey opens a Windows dialog box to open an existing text file.

Opening a new text file will close the current text file without saving any changes.

Backup



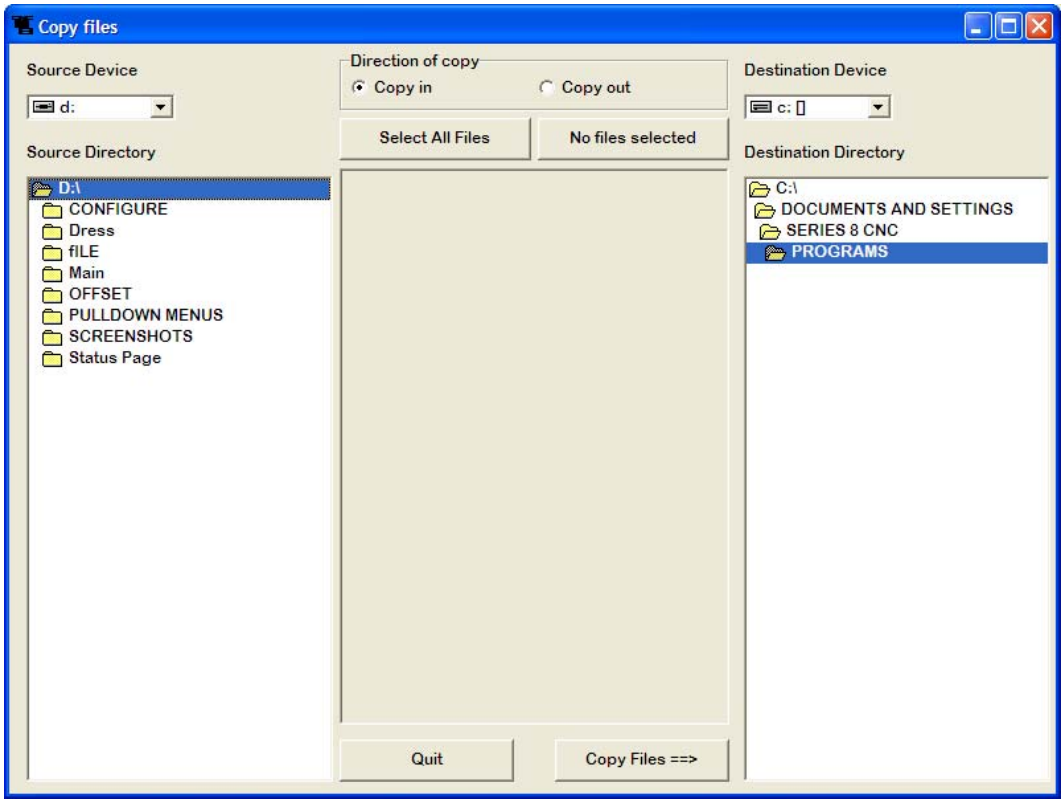
The SERIES 8 software provides an easy way to create a backup of all the CNC files. The default storage location is the C: drive, in the \BACKUP folder. The destination drive can be changed by clicking on the down arrow in the drive window, and selecting another device. The folder may be changed by clicking on the "Backup Folder" window and entering in a new name (a keyboard will appear when the window is selected.)

By default, the backup utility will save both the system files and part program files, as indicated by the checkmarks on these settings. Either item can be unchecked to reduce storage requirements if the entire backup is not needed.

In the backup folder, other folders will be created to store files by their use within the SERIES 8:

Backup Folders	
Folder Name	Description
APP	user APP files
OEMAPP	OEM supplied APP files
DEFAULT	default files
PMI	PMI files
SERIES8	Executive files, library files, and the INI file

Copy Files



The copy file utility is an easy way for the operator to select files to copy either from one location to another. The left column “Source Device” is the location of the original files. The center column is the list of files to be copied. The right column is the “Destination Device”, this specifies the folder where the copied files will be placed.

The default selection is “Copy in” which places the d: drive (typically USB flash drive) as the source device and c: (CNC’s hard disk) as the destination device. Copying in files from a flash drive to the CNC. “Copy out” sets the source device as c: (CNC hard disk) and the destination device to the d: (USB flash drive). Copying out file from the CNC to a flash drive.

Update APP Files

APP files are subroutines that can be called from the part program. There are two main types of APP file subroutines:

APP File Types	
File Type	Description
Canned Cycles	Certain M and G codes may be defined as canned cycles, which perform various functions based on parameters passed to the cycle from the program. Most common of these cycles are the drilling cycles (G81-G89) for machining centers.
Subroutines	These are part program routines that are saved in separate files and may be called up from the part program using the "CLS" block. Subroutines may also be included with conversational programming files to create a 'fill-in-the-blank' for of data entry for setting the subroutine parameters.

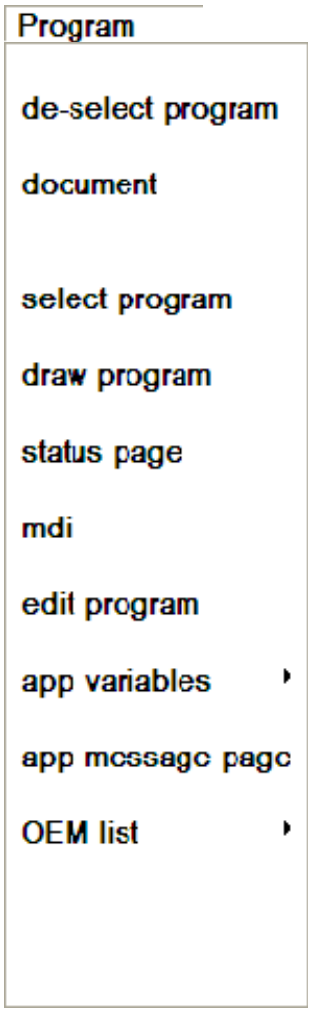
APP files are stored on the PC hard disk in directories identified by the SERIES8.INI file. These files are automatically copied from the hard disk to the motion control card during initial load of the SERIES 8 software.

The Update APP Files command provides a way to copy the APP files to the motion control card after the SERIES 8 software is already running. This is helpful when writing and testing new APP files to avoid having to restart the entire system.

In order to update APP files, there must not be a part program selected. Use deselect program from the Program menu to deselect any current program. Then execute Update APP Files to copy the new files to the motion control card. A progress bar for the copy is shown. This may take several seconds depending on the amount and size of the APP files for the machine.

Program Menu

This menu allows for the creation, selection, and drawing of a part program. The parameters controlling how a part program is run can also be modified.



De-select Program

After making this menu selection, no part program is active. A new part program must be selected to be able to run any program

Document

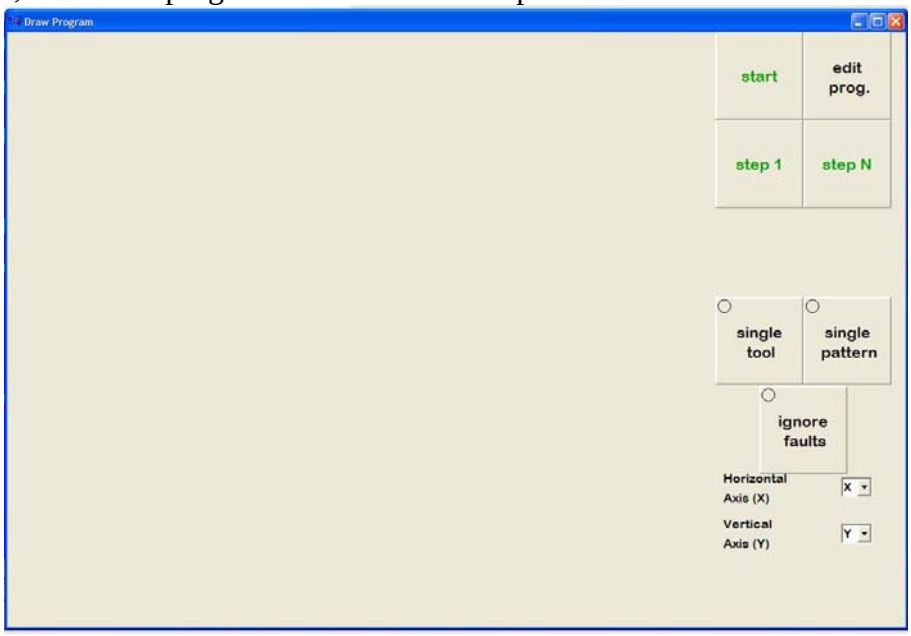
This menu selection opens the SERIES 8 CNC Precision PCB Operator's Manual in Abode Acrobat.

Select Program

Select a file, which becomes the active part program. When CYCLE START is pressed, part blocks are read from this file and executed.

Draw Program

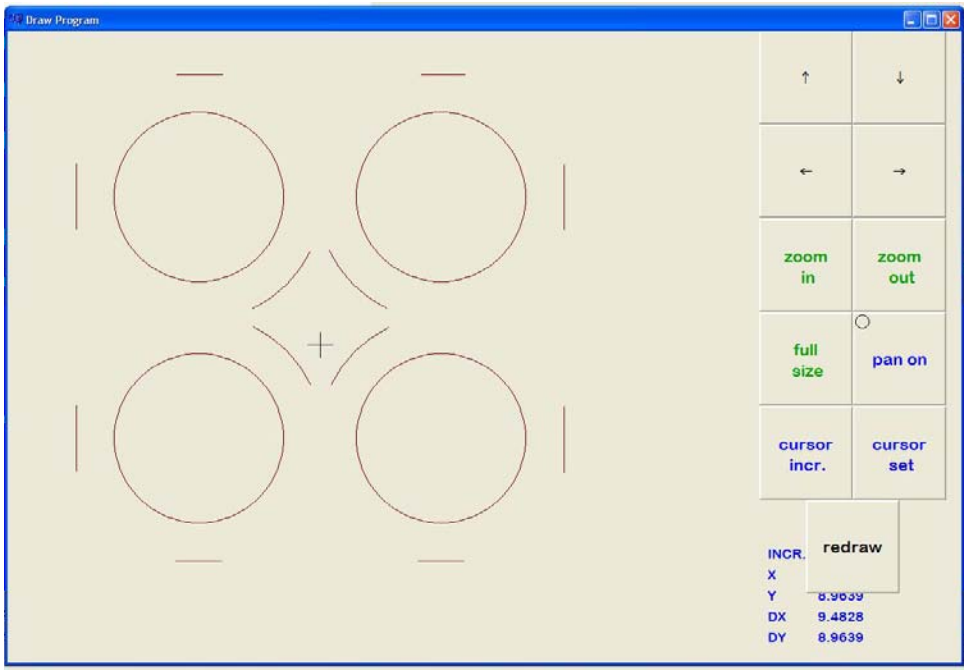
This window displays a two-dimensional projection of the part program. Specify which two axes to draw. In drawing mode, the blocks are graphically displayed sequentially on the screen. When the program is complete or by pressing the "done" touchkey, viewing mode starts. In viewing mode, the drawn program can be zoomed or panned to show more detail.



Draw Program Touchkey Definitions	
start	Blocks are drawn continuously until the end of the program, or until "stop" touchkey is pressed.
done	Stops drawing blocks.
edit prog.	Opens the Notepad text editor, with the current part program loaded.
step 1	Draw the next block in the part program.

step N	Draw a specified number of blocks.
single tool	Only draw the moves for a specified tool number.
single pattern	On programs with repeated patterns, draw only the first occurrence of a pattern
ignore faults	Ignore program faults, such as travel limits

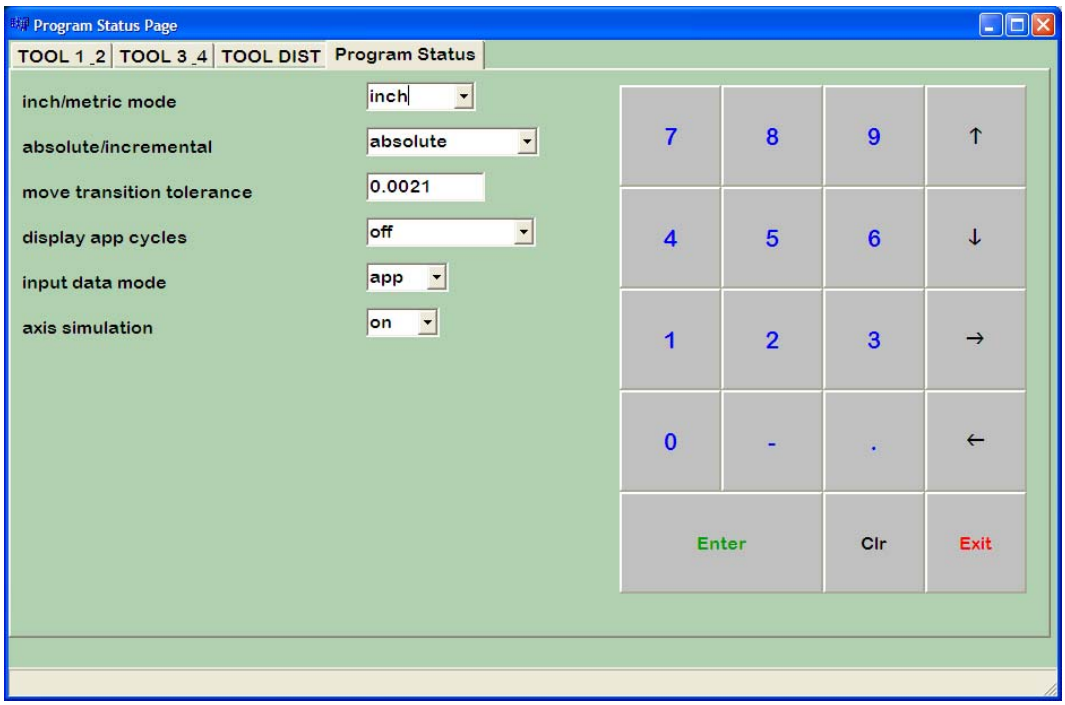
In viewing mode, the cursor position is shown in X and Y coordinates from program zero, and distance from a designated reference point. INCR is the distance the cursor moves when an arrow key is pressed.



Viewing Mode Touchkey Definitions

↑ ↓ ← →	Moves the cursor in the direction indicated by the arrow, by the distance shown as the current cursor increment.
zoom in	Zooms in on the drawing at 125% of its current size.
zoom out	Zooms out on the drawing, to 80% of its current size.
full size	Resets to zoom status to 'full size' – this sets the drawing limits equal to the travel limits of the machine.
pan on	Turns on panning mode, the cursor is no longer shown, and pressing a directional arrow key shifts the entire drawing in the direction of the arrow pressed.
cursor incr.	Changes the incremental move distance of the cursor, cycling through a set of pre-set increments.
cursor set	Sets the cursor position as "zero" on the display. Subsequent move of the cursor relative to this set position can then be read on the display.
redraw	Pressing this key reverts back to the initial menu of the DRAW PROG screen, allowing the program to be re-drawn from the beginning. The current zoom level is maintained.

Status Page



These pages let the operator examine and modify parameters. The Program Status page is standard and available on all systems. Other pages within this section are controlled by the PMI program and vary depending on the application.

Program Status

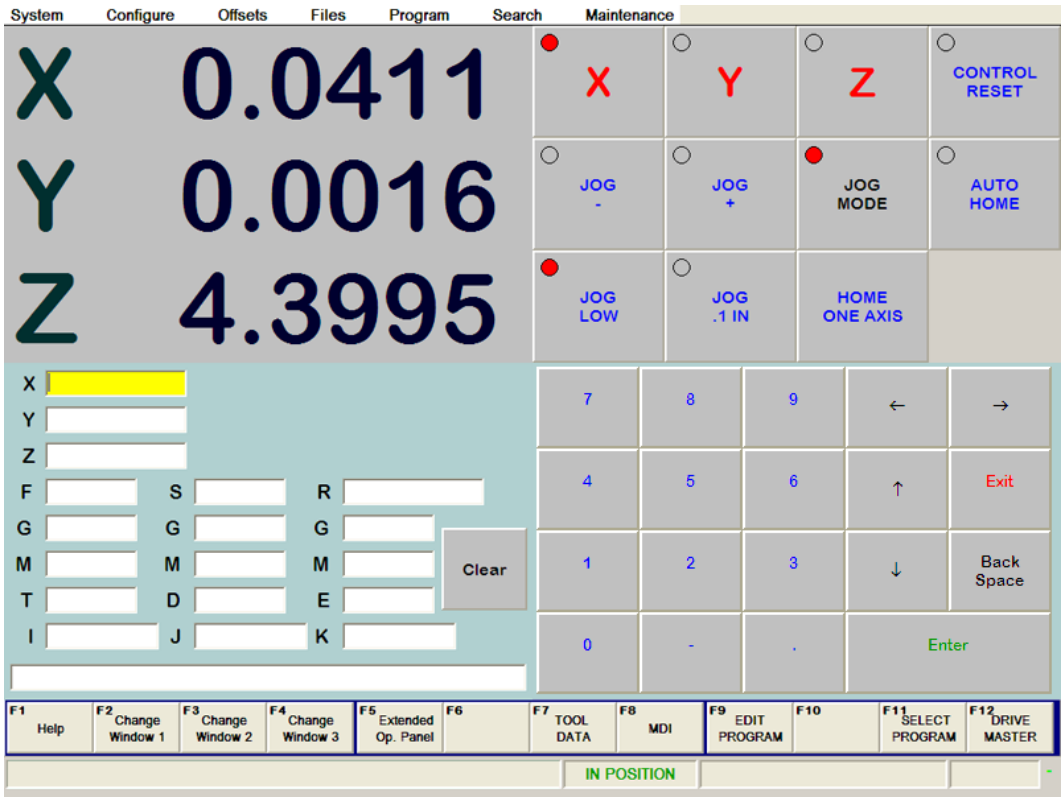
This page allows the operator to change the parameters that control how part programs are executed and interpreted.

Program Status Page	
Parameter Names	Description
Inch / Metric Mode	Determines how part program coordinates are interpreted and displayed (inches or millimeters)
Absolute / Incremental	Sets the default mode for program coordinates as absolute position values, or incremental values based on current position.
Move Transition Tolerance	Sets the allowable amount of corner rounding when 'round corner' mode is active. Round corner mode attempts to maintain velocity while making move transitions that are not tangent. If this parameter is kept very small, it creates more accuracy at move transitions, with a possible sacrifice in velocity. If this parameter is set to zero, it allows the maximum round corner amount allowed for the programmed velocity of the axes.

Program Status Page	
Parameter Names	Description
Display APP Cycles	Normally set to "OFF". If set to "ON" it allows display of each block of APP cycles to be shown on the part program display page to assist in debugging cycles. If executing in SINGLE BLOCK mode with APP CYCLES ON, program execution halts with each block in the cycle; otherwise with APP CYCLES OFF, the entire cycle executes as one block.
Input Data Mode	Determines the numeric format of data in the program. The options are: flpt - floating point tzs - trailing zero suppression lzs - leading zero suppression app - special version of floating point that allows APP codes in blocks Normally, the input data mode should be set to app to allow the greatest flexibility in programming.
Axis Simulation	If ON, it allows running the control in a simulated mode without actual servo motors running.

MDI

Manual Data Input (MDI) allows entering of a single block of information. When this command is executed, the Window 2 display shows the available part program codes, and Window 3 provides data entry touch keys. Data may also be entered directly from a keyboard.



As a block is being 'built' by entering data, it is displayed on the lower line of Window 2.

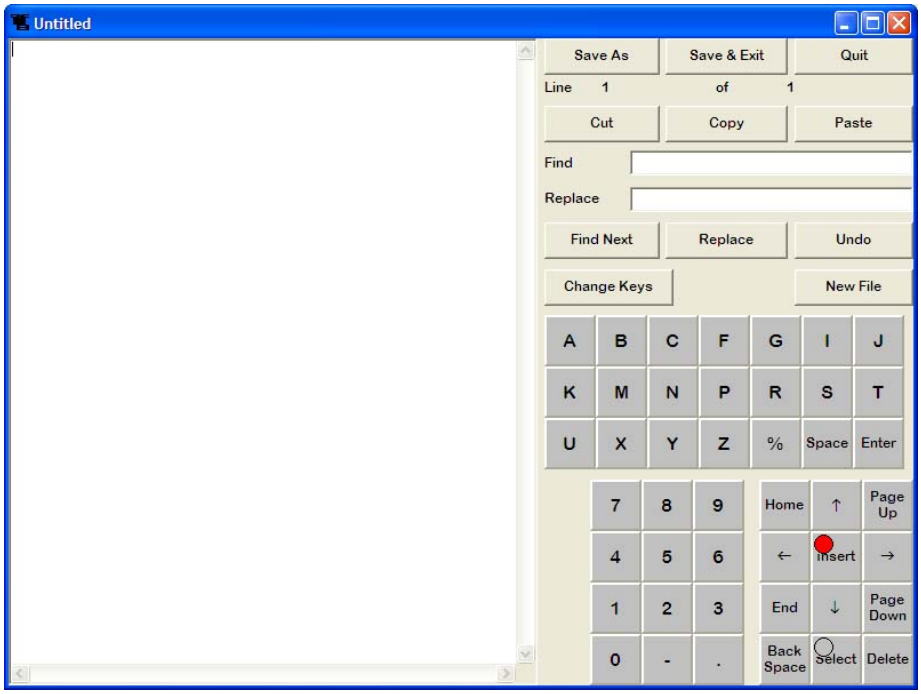
The block may be created in one of these ways:

1) Touch the letter code in Window 2, and then type in the value for that code.
2) Type the letter code desired (the cursor in Window 2 moves to the correct display), and then enter the value for that code.
3) Touch the lower line in the Window 2 area and the cursor blinks on this line; then type in the block directly from the keyboard.

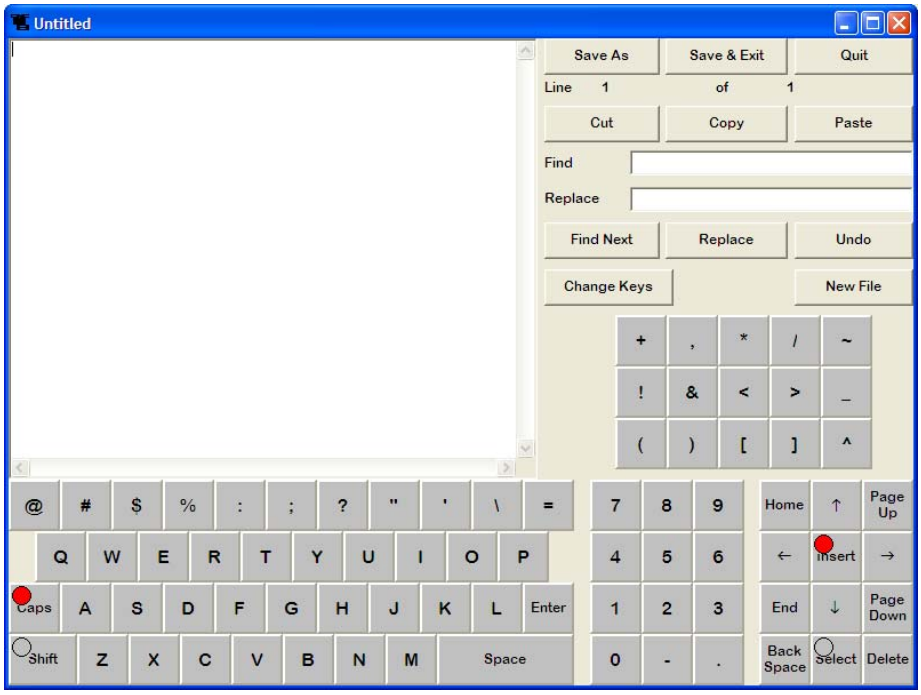
After the block information has been entered, press the Execute softkey to run the programmed block.

Edit Program

Selecting Edit program opens a text editor with the current program opened for editing.



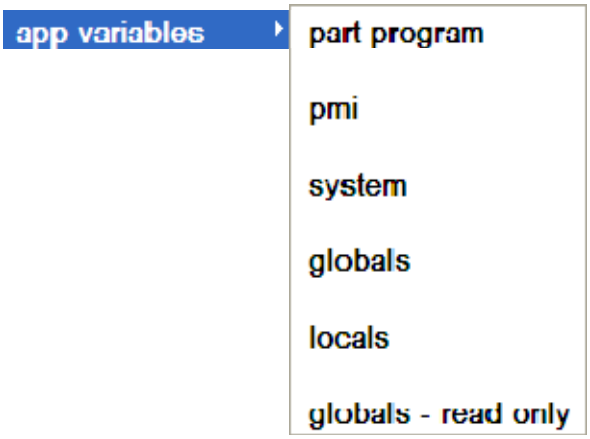
Pressing the “Change Keys” softkey to display a QWERTY keyboard and common symbols used in SERIES 8 CNC part programming.



Pressing the “New File” softkey opens a Windows dialog box to open an existing text file.

Opening a new text file will close the current text file without saving any changes.

APP Variables



This menu allows access to the Advanced Programming Package (APP) variables. User access to variables is generally not required, so only a brief description of their meaning is supplied here. Refer to the "APP" section of this manual for a description of the APP programming language.

In general, these variables can only be read by the operator, and provide a means to debug APP cycles. Global Variables may also be changed by the operator from this menu to define certain parameters to APP cycles. Global Variables are user-specific to APP cycles.

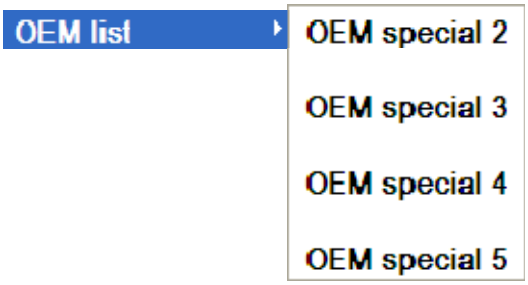
APP Variables	
Variable Name	Description
APP Part Program Variables	" $ " Variables with status for part program data.
APP PMI Variables	" @ " Variables that are passed to/from the PMI program.
APP System Variables	" : " Variables relating to real-time status of the control.
APP Global Variables	" # " Variables created by APP programs that can be accessed at all APP subroutine levels.
APP Local Variables	" % " Variables used within an APP subroutine.

APP Message Page

This page allows part programs to use the Advanced Programming Package (APP) to display data and prompt the operator for a value. It can be used to create custom cycles with parameters that can be entered at the time the cycle is run.

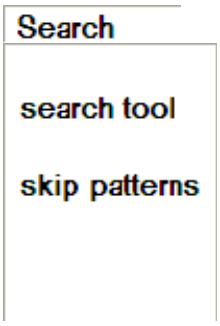
Note: The numeric keypad appears on this page only when an APP (KEY) command is active; otherwise, only the Esc key appears.

OEM List



Selecting OEM list and then OEM special # starts a program specified in the PMI.

Search Menu



Search Tool

The part program skips blocks until a tool change to the specified tool number is encountered. None of the skipped blocks are executed. The program can run from the new part program position.

Skip Pattern

If the fiducial check mode is active (G86), then it is necessary to inhibit a "skip pattern" request until the fiducials have been checked, otherwise the part location may be incorrect. If an attempt is made to skip patterns at the beginning of the part program with the fiducial check active, the operator will get the message:

"Command not allowed"

The correct procedure to skip in this case is as follows:

1. Press the operator key labeled SKIP W/ FID CHECK. This sets a mode where the part program will halt execution after running the G63 or G85 fiducial cycle.

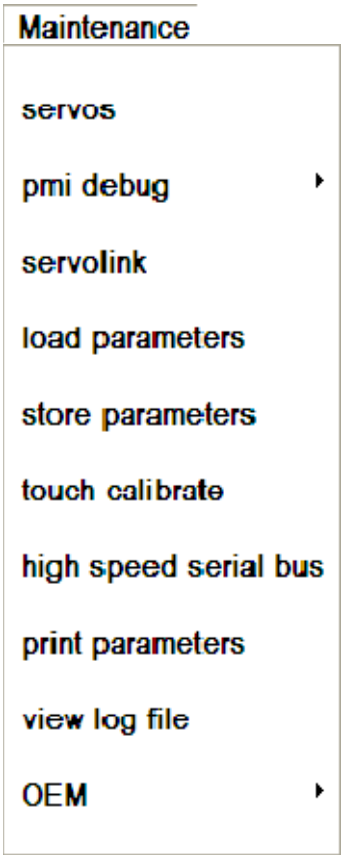
2. Press CYCLE START to start the part program. The program will execute the table fiducial check, followed by either a G63 or G85, whichever is in the program, and then stop.

3. The operator may then use the SKIP PATTERN command in the SEARCH menu to skip the desired number of patterns.

4. Pressing CYCLE START will resume program execution from the point that was skipped to.

Maintenance Menu

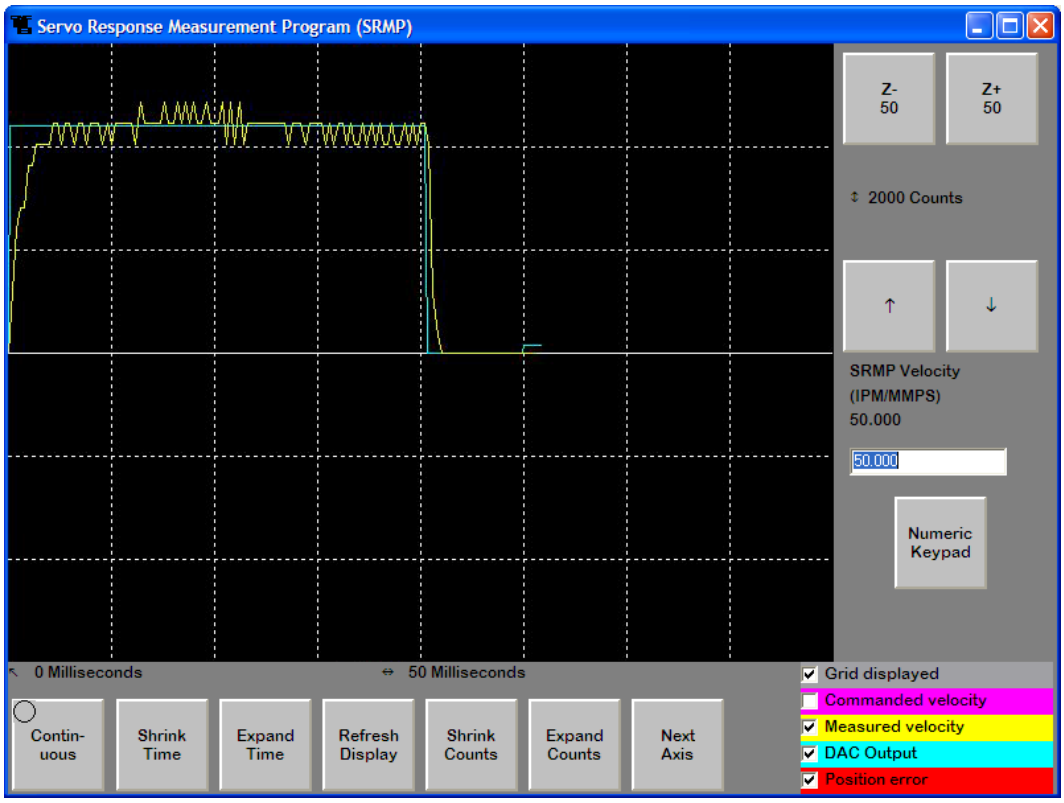
This menu allows data to be collected about the machine, to tune the servo drives, to load or save configuration files that set up the software for a particular machine, and to set up the touch screen.



Servos Menu

Note: The SERVOS command is used for tuning the machine servo parameters and should be used by a qualified service technician only.

This page permits adjustment of servo system parameters to optimize machine performance. The graph is of



Servo Response Measurement Program (SRMP) Softkeys	
Softkey Label	Description
Contin-uous	Makes continuous moves in reversing directions for the selected axis.
Shrink Time	Decreases the total time displayed on the graph.

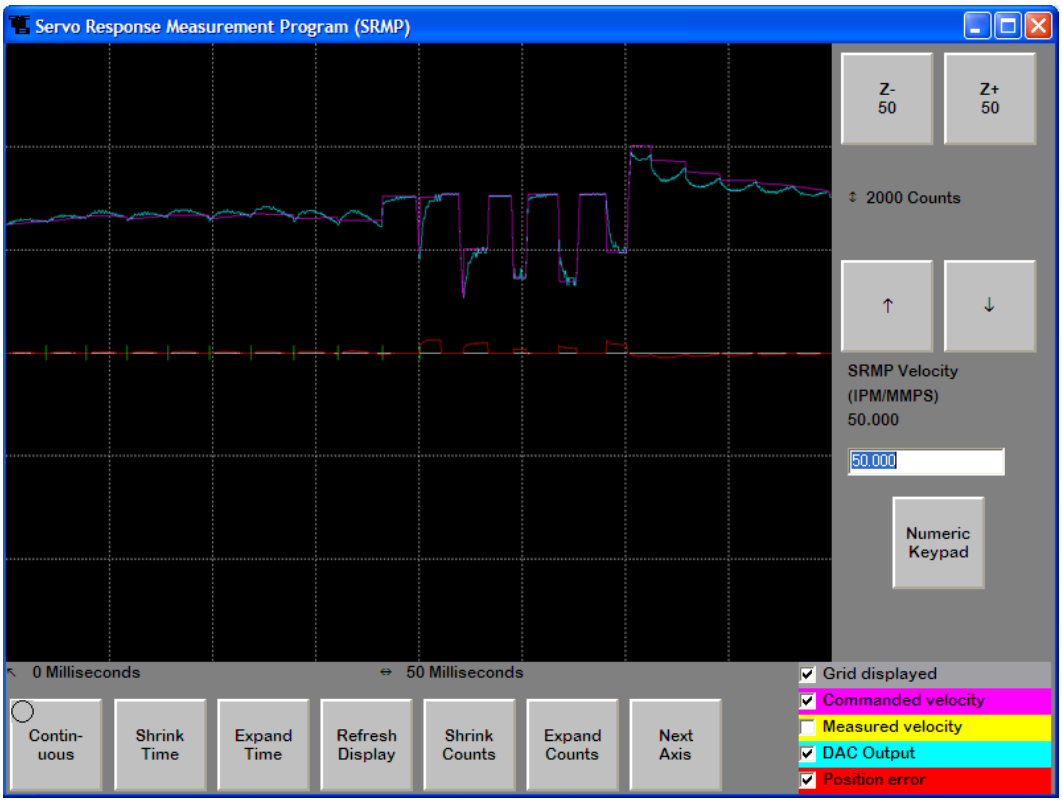
Servo Response Measurement Program (SRMP) Softkeys	
Softkey Label	Description
Expand Time	Increases the total time displayed on the graph.
Refresh Display	
Shrink Counts	Decreases the total counts displayed on the graph
Expand Counts	Increases the total counts displayed on the graph
Next Axis	Cycles through the available axes on the machine. Double check any axes selected is capable of moving without hitting/interring with its surroundings.
Numeric Keypad	Toggles a numeric keypad on the screen for modifying SRMP parameters. To close the numeric keypad press the numeric keypad softkey, to exit SRMP press the Exit softkey located in the numeric keypad.
↑	Cycles through the SRMP parameters.
↓	Cycles through the SRMP parameters in the opposite direction.

Servo Response Measurement Program (SRMP) Softkeys	
Softkey Label	Description
Z- 50	Moves the selected axis (in this example Z) in the negative direction at the specified speed for the specified time.
Z+ 50	Moves the selected axis (in this example Z) in the positive direction at the specified speed for the specified time.

Servo Response Measurement Program (SRMP) Parameters	
Parameter Name	Description
SRMP VELOCITY	The velocity of the move is made for the selected axis, when the + or - softkeys is pressed.
SRMP PULSE WIDTH	The time allowed for the axis move, in milliseconds
REFERENCE GAIN +	The nominal value for this is 1000, which is the standard gain for the parameters specified in the Parameter Editor for this motor/amplifier combination. Small adjustments may be made to correct for variations in commanded vs. actual velocity for moves made in the plus direction by changing this parameter.
REFERENCE GAIN -	The nominal value for this is 1000, which is the standard gain for the parameters specified in the Parameter Editor for this motor/amplifier combination. Small adjustments may be made to correct for variations in commanded vs. actual velocity for moves made in the minus direction by changing this parameter.
P GAIN	Proportional gain. This gain represents the sensitivity of how fast the servo amplifier reacts while maintaining a velocity control loop. A low P-gain results in a 'mushy' servo, while a P Gain which is too high results in oscillation (high pitched squeal). It is generally desirable to set the P Gain as high as it can be without creating an oscillation.
I GAIN	Integrator gain. This gain sets the sensitivity of the amplifier's response to changes in velocity command (acceleration/deceleration). A higher I Gain results in better servo performance. Excessive I Gain causes an overshoot at the end of acceleration.
OFFSET	Velocity command offset. A value of zero represents no offset, and a value of +128 to -128 may be entered to correct for offset variations.

Servo Response Measurement Program (SRMP) Parameters	
Parameter Name	Description
CURRENT LIMIT	Entered as a percentage of max, the current limit sets the maximum allowed current to be sent to the servo motor as a part of the velocity control loop. Normally, the current limit should be set to 100% (actual current limit in amperes as set in the servo amplifier), but may be lowered for certain tests.
NOTCH FREQUENCY #1	Notch filter setting to suppress certain machine resonances. A value of 2500 causes this filter to have no effect.
NOTCH FREQUENCY #2	Notch filter setting to suppress certain machine resonances. A value of 2500 causes this filter to have no effect.
Program Block	Specify a program block number to start a 2 second recording of the axis' measured velocity, commanded velocity, position error and DAC output. The data will be recorded for the axis selected before entering the SERVO tuning page. After running the part program return to the SERVO page and press the Refresh Display softkey to display the graph.

An example of a program block SRMP looks like the image below. The green vertical lines on the graph represent the start of a new block.



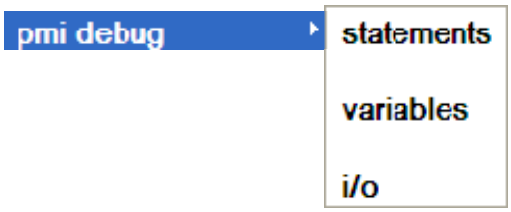
Note 1: Before setting any other servo parameters for an axis, the OFFSET value should be set. To do this, select the desired axis (NEXT AXIS key), and look at the value for "DAC" in the lower right hand corner of the display. With servos on and no move commanded, the DAC value should be near zero. Change the offset value + or - until a reading near zero is achieved.

Note 2: If an axis is making a loud squeal at certain speeds, lower the P Gain until the noise goes away. If the P Gain end up below a value of 5, changes to the notch filter settings may be necessary to achieve a better P Gain.

Note 3: If the commanded and actual velocity curves differ by more than 5%, there is likely a problem with the basic motor/amplifier settings in the Parameter Editor. Check with the machine manufacturer for the correct settings.

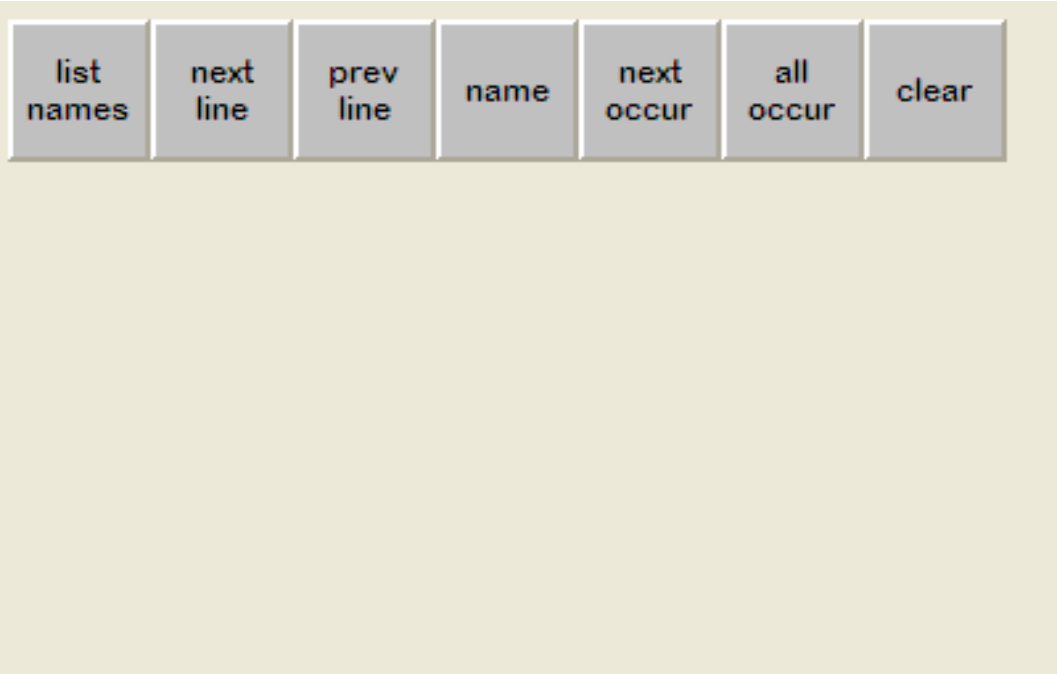
PMI Debug Menu

PMI is the Programmable Machine Interface that interfaces the executive software to a specific machine. It controls all of the digital inputs and outputs, as well as the operator touch screen softkeys, spindle speed control, and auxiliary axis moves. PMI DEBUG is a utility that provides real-time monitoring of PMI statements, PMI variables, and the digital I/O of the machine.



PMI Statements Debug Display

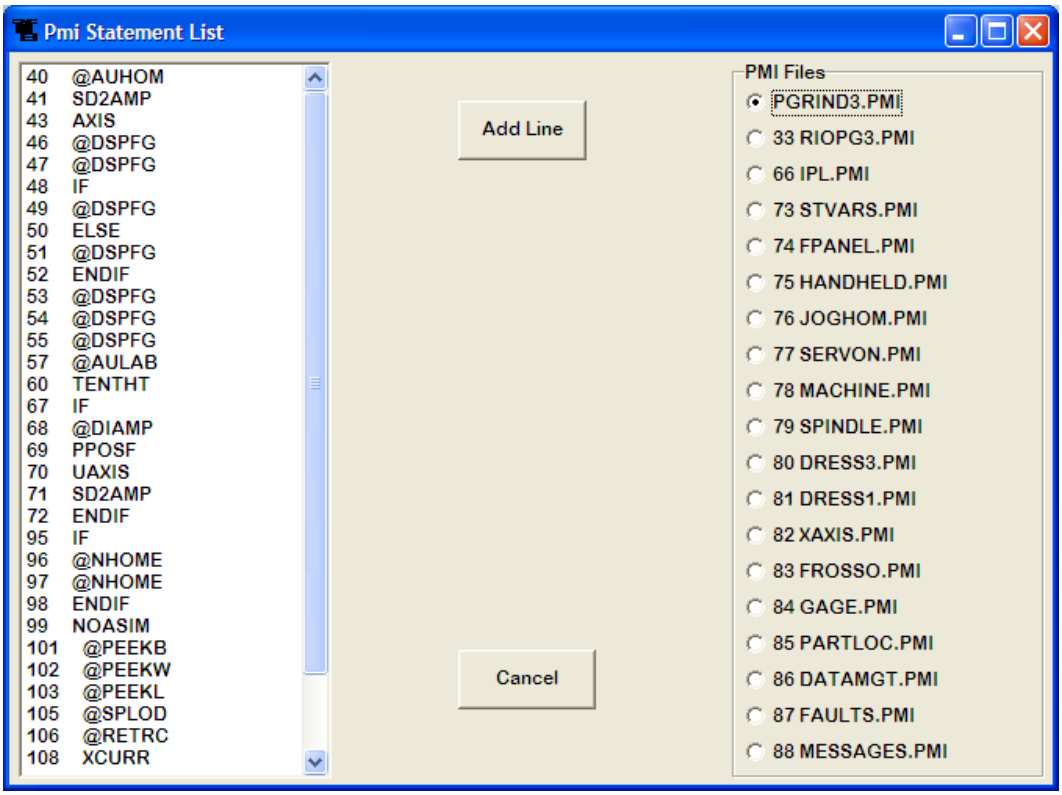
This window displays PMI statements, with the variable values shown in real time. The operator must define which statements are displayed.



When displaying PMI statements, the variable name to the left of the equal sign (=) determines the name of the line. Since more than one line may contain the same variable name left of the equal sign, the first occurrence is initially displayed.

PMI Statements Debug Softkeys	
Softkey Label	Description
list names	In response to the "list names" touchkey, a new window is opened (see image below table.) The scroll box on the left lists the first symbol in each line of the selected file. The "PMI Files" box on the right lists all files used in the PMI program: first the main program, then any included files. To display a line, click on it in the left-hand scroll box to highlight it, then click the "Add Line" softkey. Click "Cancel" to close this window. Keyboard shortcut "L"
next line	Displays the next line of PMI code following the line last displayed. Keyboard shortcut "N"
prev line	Displays the previous line of PMI code relative to the line last displayed. Keyboard shortcut "P"
name	The operator is prompted to enter a PMI variable name. The line containing the first occurrence of this name in the PMI program is displayed. Keyboard shortcut "M"

PMI Statements Debug Softkeys	
Softkey Label	Description
next occur	Based on the PMI variable name last entered with the 'name' command, this causes the next occurrence of this variable in the PMI statements to be displayed. Keyboard shortcut “X”
all occur	The operator is prompted to enter a PMI variable name. All lines containing this variable name (left of the equals sign) are displayed. If more lines contain this variable than can be displayed in a single page, only the first several occurrences that can be displayed, are shown. Keyboard shortcut “A”
clear	Clears the display. Keyboard shortcut “C”



PMI Variable Debug Display

This window shows the current state of PMI variables. The operator can select which variables are displayed.

PMI Variable Debug Softkeys	
Softkey Label	Description
display	Opens a keypad to input a variable to display. For array variables, specify an array index of “*” to display the entire array. Keyboard shortcut “D”
change	Change the value of constants and timers. Keyboard shortcut “H”
erase	Clears the window. Keyboard shortcut “E”

For array variables, there are several options:

VAR[2] - displays array element "2" for the variable VAR
VAR[*] - displays all array elements for variable VAR
VAR[3-6] - displays array elements VAR[3], VAR[4], VAR[5], and VAR[6]

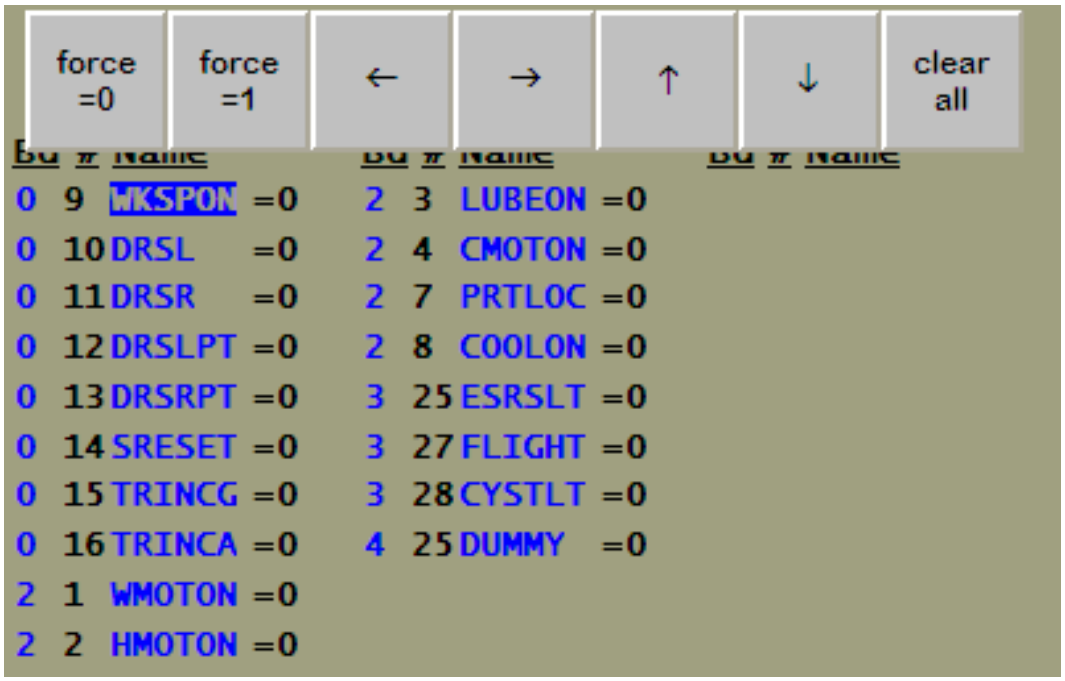
If a timer variable is displayed, four values are displayed.

CTIMRT = 0
CTIMRD = 1
Count = 500
Timer = -1

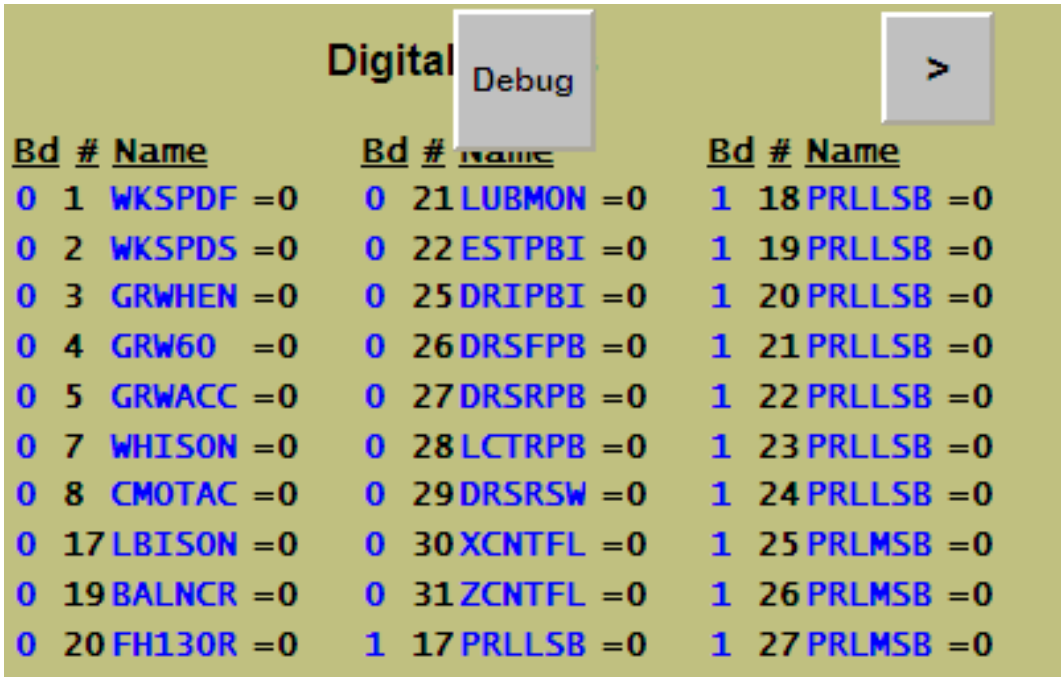
The first two entries are the input-timer variable and the output-timer variable. The "Count" is the programmed value of the timer (each count is equal to 20 ms). The "Timer" is the current value of the timer. If the current value is 0 or -1, then the timer has completed counting.

PMI Debug Input/Output Display

The PMI Input/Output debug allows viewing the status of all inputs and outputs of the system, as well as allowing the operator to override the current value of an input or output for testing purposes.



PMI Debug Inputs



PMI Debug Outputs

Note: Use caution when overriding I/O, because machine damage may result. When a value is overridden, PMI can no longer control its state.

When the PMI Debug I/O display is selected, both the INPUT and OUTPUT page listings appear. The OUTPUT page is selected by default for overriding values. The INPUT page may be selected by pressing the DEBUG key at the top of the INPUT page.

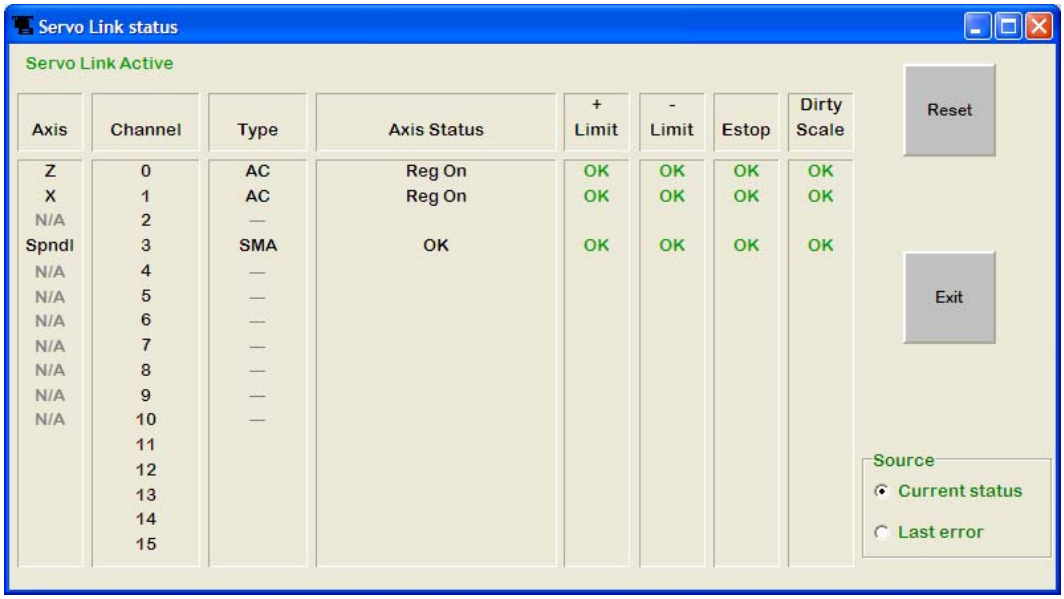
For the active I/O page, the following menu allows overriding of values:

PMI Variable Debug Softkeys	
Softkey Label	Description
force =0	Forces the selected input or output to be 0.
force =1	Forces the selected input or output to be 1.
cancel	Cancels any forced status on the selected input or output, returning it to normal operation.

PMI Variable Debug Softkeys	
Softkey Label	Description
←	Moves the cursor left.
→	Moves the cursor right.
↑	Moves the cursor up one entry in the display.
↓	Moves the cursor down one entry in the display.
clear all	Cancels forced status on all inputs, returning them to normal operation.
<	Displays the next column to the left.
>	Displays the next column to the right
Debug	Selects either the digital inputs or outputs for override control.

Servolink Status Menu

SERVOLINK is the high speed communications bus for the servo system (X, Y, Z, and Spindle/SFU). If a "SERVOLINK" fault is shown, looking at this page helps isolate the cause of the problem. SERVOLINK faults can generally be reset from this page, when servos are off, by pressing the RESET key.



Axis	Channel	Type	Axis Status	+ Limit	- Limit	Estop	Dirty Scale
Z	0	AC	Reg On	OK	OK	OK	OK
X	1	AC	Reg On	OK	OK	OK	OK
N/A	2	—					
Spndl	3	SMA	OK	OK	OK	OK	OK
N/A	4	—					
N/A	5	—					
N/A	6	—					
N/A	7	—					
N/A	8	—					
N/A	9	—					
N/A	10	—					
	11						
	12						
	13						
	14						
	15						

The SERVOLINK status page shows each axis of the system, plus the spindle (SFU) interface. If there is a fault associated with any of the amplifiers, the fault is shown in the AXIS STATUS column.

Axis over travel limits, and dirty scale signals (on systems equipped with scale feedback) are wired directly to the servo amplifiers, and the status is communicated to the CNC over the SERVOLINK bus.

Master Status (upper left corner)	
Term	Description
Servo Link Active	Indicates that SERVOLINK communication is working and active.
Servo Link Not Defined	
Servo Link Interface Missing	
DPR Self-test failed	
Timer Vector Test Failed	
Tx Complete Vector Test Failed	
Servo Link WDT Test Failed	
Internal Loop Back Test Failed	

Master Status (upper left corner)	
Term	Description
Communications Failed During Initialization	
External Loop Back Test Passed	
External Loop Back Test Failed	
Unknown Servo Link Failure	

Type (Motor)	
Term	Description
AC	AC brushless rotary motor
DC	DC rotary motor
Linear	Linear motor
xx AMP	Rated current of amplifier (or frequency converter)
Probe	This channel is used for a probe input (probe interface board)
Spndl	This channel uses a spindle interface board (external analog speed command)
???	Motor typed not defined

Axis Status	
Term	Description
Tx Error	Data transmission error detected
Rx Error	Data transmission error detected
EE Read Error	
EE Checksum Error	
Servo Power Fault	
Regulator On Failed	Unable to enable the servo
Feedback Rx Error	Data transmission error detected.
I squared T Fault	Amplifier drew excess current for too long of a time. Caused by mechanical binding in an axis or and undersized motor for the application.
Motor Over Temp	Indicator from the thermal sensor in the motor of an over temperature.
Heat Sink Over Temp	
Short Circuit	Short circuit detected in motor leads
Over Voltage	
Motor Feedback Fault	Missing signals from resolver
Amplifier Detected	
Slip Fault	Used on spindle drives with spindle feedback, indicating excessive slip during spindle rotation
Sp Temp	Indicator of over temperature from spindle sensor
Ramp Up	Spindle speed is ramping up
Ramp Down	Spindle speed is ramping down

Axis Status	
Term	Description
0 Speed	Spindle is at 0 speed
At Speed	Spindle is at programmed speed

Load Parameters

This allows recall of previously stored parameters for the Parameter Editor. Under normal conditions, the parameter settings are saved upon exiting the Parameter Editor to a file named "DEFAULT.PRM", and stored in the DEFAULT directory specified in the SERIES8.INI file. This file s then loaded automatically whenever the SERIES 8 CNC software is started.

This LOAD PARAMETERS command allows recall of a backup copy of the parameter settings previously saved with the STORE PARAMETERS command.

Note: When a new parameter file is loaded, it is NOT automatically saved as the DEFAULT.PRM file. If the SERIES 8 is shut down and then restarted, the previous data is restored. In order to save parameters loaded with the LOAD PARAMETERS command as the default, it is necessary to enter the Parameter Editor and then exit. Exiting the Parameter Editor creates a new DEFAULT.PRM file with the latest parameter settings.

Store Parameters

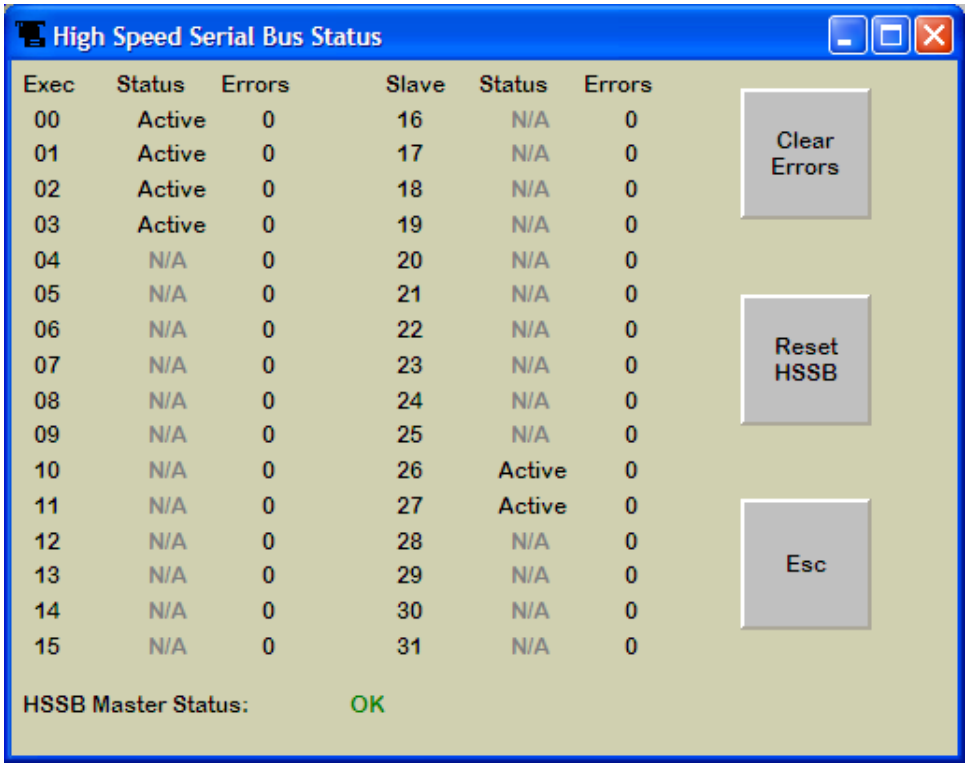
The store parameters command creates a backup file storing the current value of parameters in the Parameter Editor. Parameters are automatically saved to the file DEFAULT.PRM (located in the default directory of the SERIES8.INI file) any time the Parameter Editor is exited. This store parameters command allows saving a separate copy of the parameter file under a different name, for possible later recall with the load parameters command.

Recalibrate Touch screen

This function creates an alignment of the physical touch surface of the screen to the image being displayed. Follow the on-screen instructions to re-calibrate.

High Speed Serial Bus (HSSB Status)

The high speed serial bus (HSSB) is the hardware function of the control for handling digital inputs and outputs. Up to 32 HSSB addresses are available in the system. Each typically relates to a single I/O board, but in some cases multiple addresses may be assigned to a single board.



The status window shows the HSSB address, its status, and its error count. The error count indicates the number of times that a communications re-try had to occur due to a checksum failure of the data. It is not uncommon for some errors to be present after running the machine; if five errors occur consecutively, the system turns servos off. To restart the system, press "Reset HSSB" softkey, verify that all addresses are “Active”, press the “Esc” softkey to exit the High Speed Serial Bus Status window and then press SERVO ON to restart the servos.

In the image above the following addresses are present:

Address 00 I/O board with 8 inputs and 8 outputs
Address 01 I/O board with 8 inputs and 8 outputs
Address 02 I/O board with 8 inputs and 8 outputs
Address 03 I/O board with 8 inputs and 8 outputs
Address 26 4x4 board
Address 27 4x4 board

HSSB Address Status Definition	
Status	Description
Active	This address is active and communicating with the host.
N/A	This address is not present on the bus.
N/U	This address is present on the bus, but not being used by the system.
IDLE	This address is present and able to communicate with the host, but is not presently in use.
MISSING	An address requested by the system is not present.

HSSB Address Status Definition	
Status	Description
COMM	A communications error has occurred between this address and the host controller.
CFG	The board address matches one needed by the system, but the board is not able to perform the needed function.
EXEC	The address of this board was requested by both the executive software and the PMI software.

HSSB Troubleshooting	
Machine Status	Possible Solutions
A newly installed I/O board	• Check that the HSSB address is the same as the board being replaced. The address is typically set using DIP switches located on the board. • Make sure the proper power is supplied and all connections made.
A system that has been running OK:	• If the number of errors listed for a specific address is exactly 5, this is a likely sign there was a reset of the controlled chip on the I/O board. A reset can occur if there is a voltage drop on the incoming power, or if there is a loose connection on the HSSB signals. Check the power to the board, and verify all connections are secure. • If the number of errors for the failed address is more than 5, it is likely there is a noise or grounding problem. Noise on the HSSB communication lines causes corrupted data, resulting in error retries. • If more than one address has failed simultaneously, this is an even stronger indication of a loose connection on the HSSB line. HSSB board connections are typically 'daisy-chained' by running the power connections and the HSSB communication lines into one board, then out to the next. Boards need not be wired in the order of their addresses. If multiple boards fail at the same time, determine which of these addresses is physically first in the wiring chain. This likely is the failed board or has a loose connection.

Print Parameters

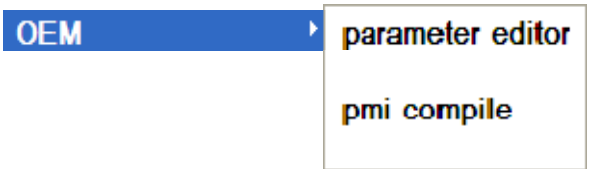
The print parameters command creates a text file of the controls parameters and current values. After selecting the print parameters command a Windows dialog box opens prompting for the file location. The default location is: C:\Document and Settings\SERIES 8 CNC\Default

View Log File

Allows the log file to be viewed without exiting the SERIES 8 CNC software. Selecting view log file will open the text editor with the log file selected. The log file adds a time stamp to major events and errors in a text file. The file is located in the root directory of the C: drive.

OEM

The OEM menu allows for the modification of the machine parameters and PMI.

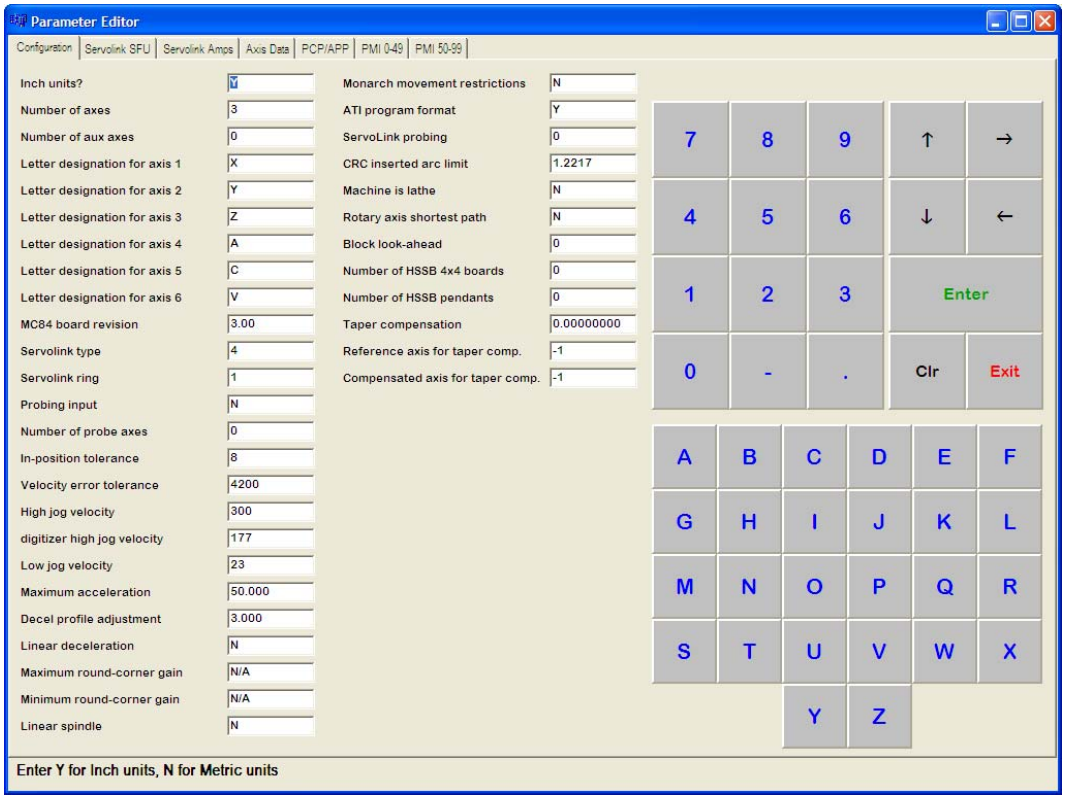


Parameter Editor

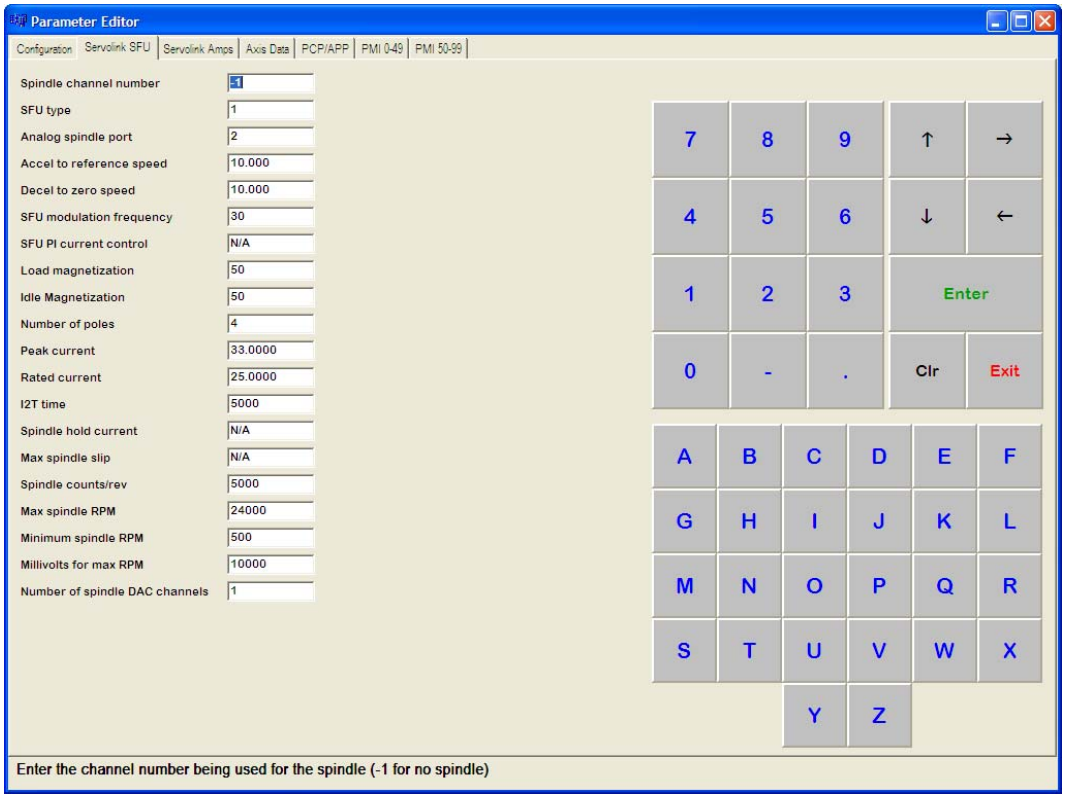
Each SERIES 8 CNC configuration has a different set of parameter tabs. The parameters and information on the tabs are the same for each machine. The machine parameters include numbers of axes, type of axes, motor counts per inch of travel, etc. When a parameter cell is selected, a line of text appears in the lower left corner of the screen to assist in setting the parameter.

This is a very important part of the SERIES 8 CNC, any modifications to these parameters need to be an educated adjustment. Wrong parameters could make the machine inoperable or cause axes to crash.

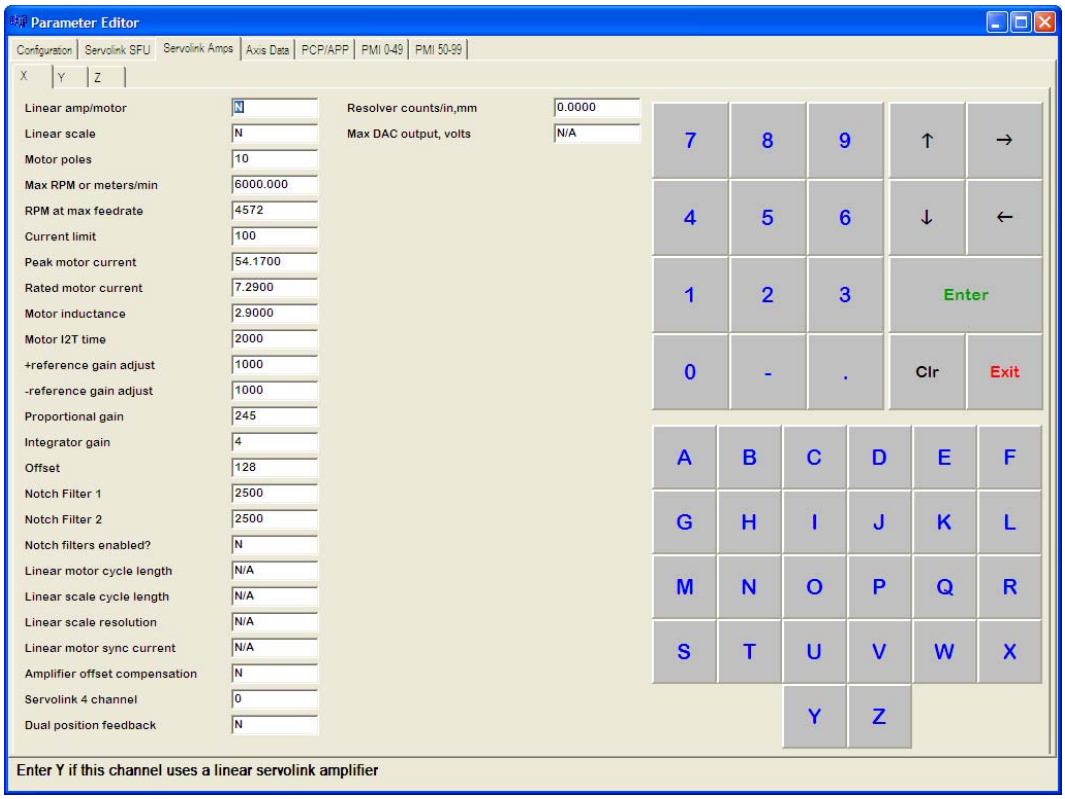
The first tab in the Parameter Editor is the Configuration tab. It sets the number of axes on the machine and their letter designations. It also sets some of the hardware configurations (SERVOLINK type, HSSB boards, etc.)



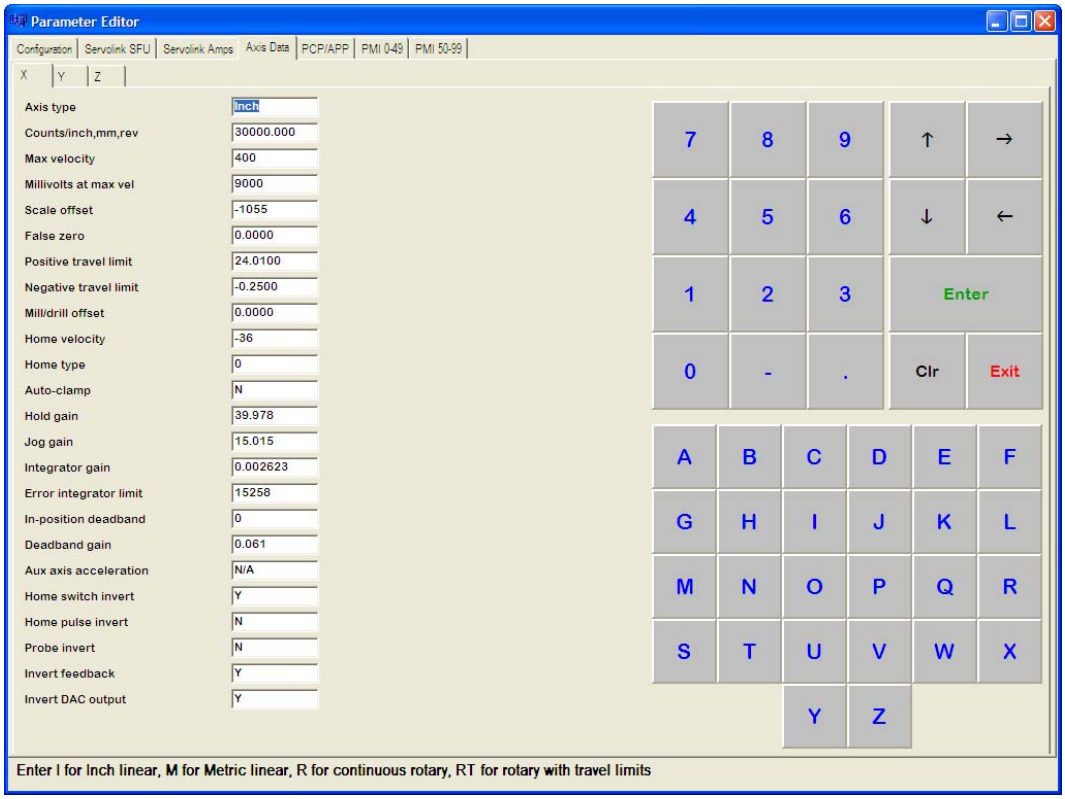
The second tab is the Servolink SFU tab. This tab sets the drive parameter for a frequency converter.



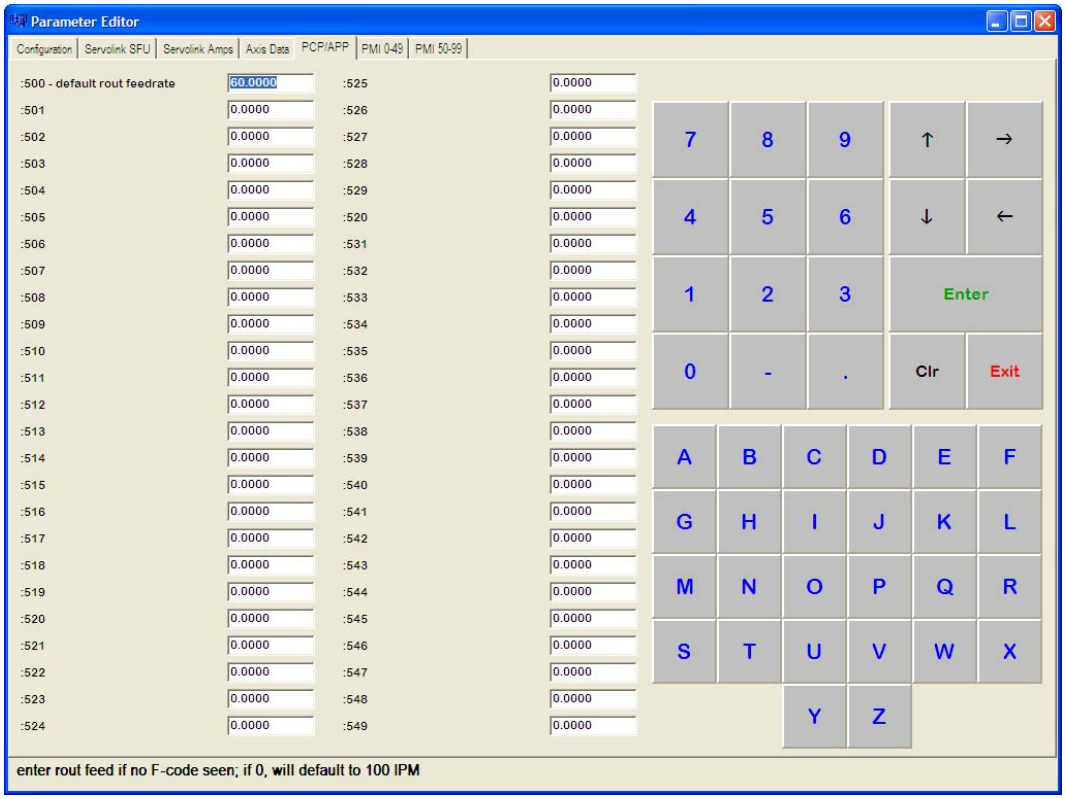
The third tab is the Servolink Amps tab. This tab has multiple nested tabs inside of it for each machine axis. This example has 3 axes: X, Y and Z. Each axis has its own tab for their drive and motor parameters.



The fourth tab is labeled Axis Data and has more axis parameters.



The fifth tab is PCP/APP.



The sixth and seventh tabs are PMI 0-49 & PMI 50-99. The tabs contains machine options that as different on every style of machine. The example below is of a router. These parameters are unique to this machine and will be different from other machine. The PMI 50-99 is used on machines that require more than 50 options. In this example it is blank because there are less than 50 machine options.

Parameter Editor

Configuration | Servolink SFU | Servolink Amps | Axis Data | PCP/APP | PMI 0-49 | PMI 50-99

0 - stripper plate option	0	25 - Auto tool change	0
1 - Coolant check enabled	0	26 - X tool pod 1 location, part 1	0
2 - Main air check enabled	0	27 - X tool pod 1 location, part 2	0
3 - Door overide touchkey	0	28 - Y tool pod 1 location, part 1	0
4 - Wait for door up/down	0	29 - Y tool pod 1 location, part 2	0
5 - light curtain option	0	30 - X tool pod spacing, part 1	0
6 - X front park position	100	31 - X tool pod spacing, part 2	0
7 - Y front park position	235	32 - Z tool dropoff position, part 1	0
8 - X rear park position	100	33 - Z tool dropoff position, part 2	0
9 - Y rear park position	10	34 - Z tool pickup position, part 1	0
10 - Z park position	40	35 - Z tool pickup position, part 2	0
11 - digitizer option	0	36	0
12 - Door not controlled	1	37	0
13 - Spindle on time at M30 (secs)	0	38	0
14 - vacuum check enable	0	39	0
15 - pw lvl for zero set	3	40 - camera IP address 1	0
16 - door light curtain check	0	41 - camera IP address 2	0
17 - operator verify at start of progr	0	42 - camera IP address 3	0
18 - broken tool drop option	0	43 - camera IP address 4	0
19 - X tool drop position	0	44 - Z position for manual tool chg	40
20 - Y tool drop position	0	45 - number of spindles (1 or 2)	2
21 - pw lvl for ext op pnl	0	46 - Rockwell cycle enable ovride	2
22 - camera option for fiducial	0	47 - invert home limit switches	0
23 - bar code option	0	48 - invert hood switch	0
24 - barcode prog de-selct @M30	0	49 - Digitizer simulate mode	0

7 8 9 ↑ →
4 5 6 ↓ ←
1 2 3 Enter
0 - . Clr Exit
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z

1= machine has stripper plate, 0= no stripper plate

Parameter Editor

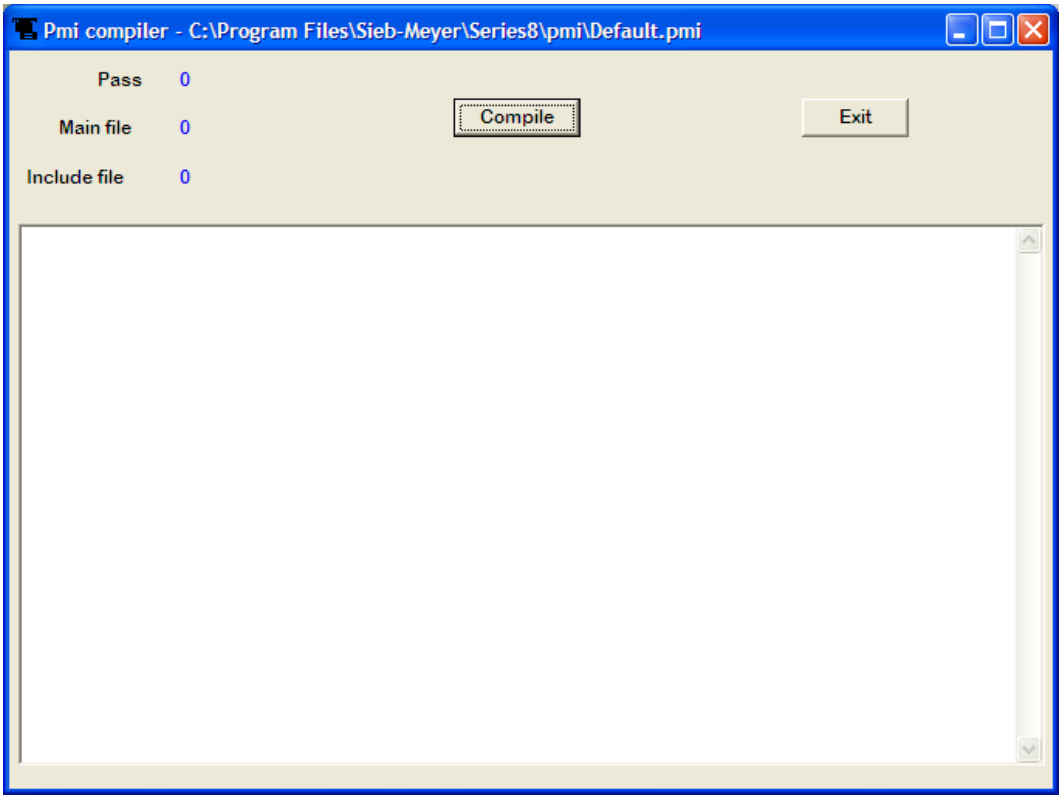
Configuration | Servolink SFU | Servolink Amps | Axis Data | PCP/APP | PMI 0-49 | PMI 50-99

50	0	75	0
51	0	76	0
52	0	77	0
53	0	78	0
54	0	79	0
55	0	80	0
56	0	81	0
57	0	82	0
58	0	83	0
59	0	84	0
60	0	85	0
61	0	86	0
62	0	87	0
63	0	88	0
64	0	89	0
65	0	90	0
66	0	91	0
67	0	92	0
68	0	93	0
69	0	94	0
70	0	95	0
71	0	96	0
72	0	97	0
73	0	98	0
74	0	99	0

7 8 9 ↑ →
4 5 6 ↓ ←
1 2 3 Enter
0 - . Clr Exit
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z

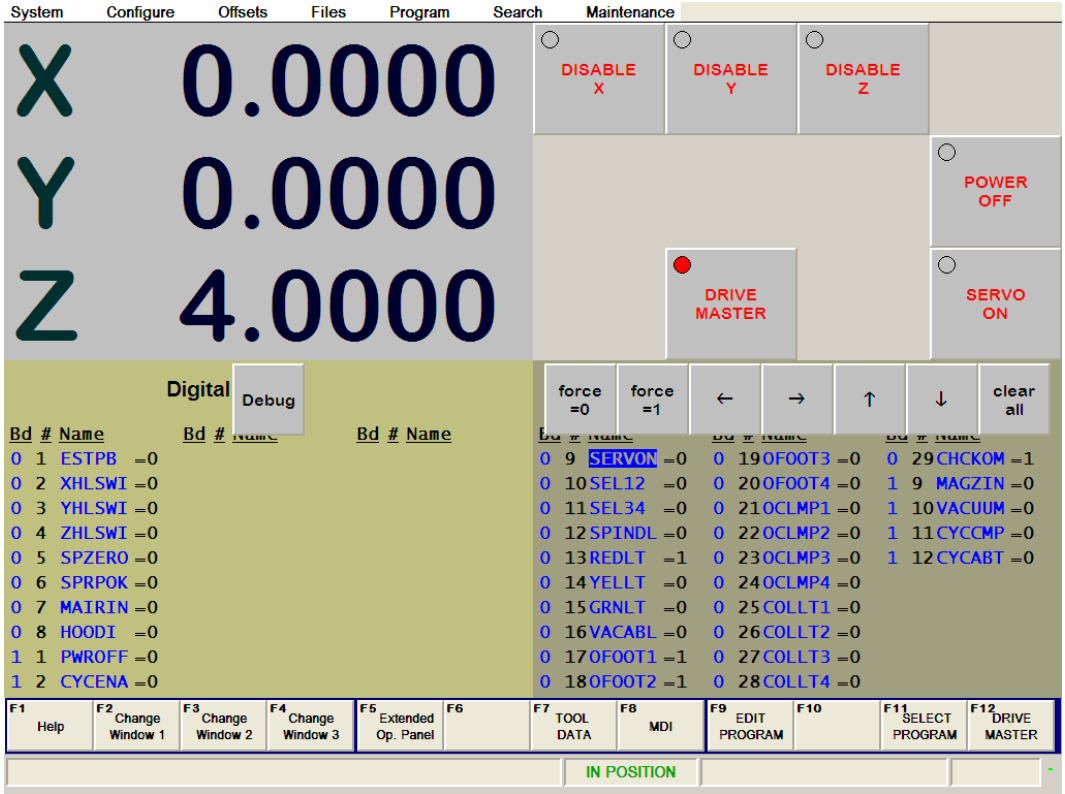
PMI Compile

Launches a Windows dialog window to open an existing .PMI file. After selecting the .PMI file a SERIES 8 windows is displayed similar to the one below. To compile the new file select the "Compile" softkey. After the checks are completed errors are shown in the window. If no errors are found, press the "Exit" softkey to load the compiled .PMI file return to the SERIES 8



Digital Input / Output Definitions

Digital inputs may be viewed in the Window 2 area, either by pressing the Window 2 display key, or by using the PMI DEBUG I/O command.



Digital Inputs	
PWROFI	The POWER OFF pushbutton from the front panel.
ESTPB	The ESTOP pushbutton from the front panel.
MAIRIN	Air pressure OK input to machine, indicating sufficient air pressure to run machine.
DOOR1	Switch on front door of machine, indicating it is open.
DORKYI	Door override key switch.
WATRPI	Water pressure switch.
PROGS1	(optional) Table mounted switch indicating program 1 is to be run. Used on systems where part program execution is determined by boards present on the machine table.
MANCLI	Manual collet open switch.
DOORSW	Door open request switch.
PROGS2	(optional) Table mounted switch indicating program 2 is to be run. Used on systems where part program execution is determined by boards present on the machine table.
LITCUR	Light curtain (optional). If light curtain is broken during machine operation, the axis motion is stopped immediately.

Digital Inputs	
CYSTSW	Remote cycle start switch (optional).
FHLDSW	Remote feedhold switch (optional).

Digital Outputs	
SERVON	Comes on when SERVO ON is requested to enable power to the servos.
VACUUM	Opens the vacuum door during routing operation.
COLLT1	Open spindle collet to release tool.
OFOOT1	Raise the pressure foot.
OCLMP1	Increasing pressure foot pressure to clamp (drilling) level.
DRCLOS	Close the machine door.
STRPUP	Raise the stripper plate. (Optional)
REDLT	Red light on totem pole.
YELLT	Yellow light on totem pole.
GRNLT	Green light on totem pole.
VACABL	Vacuum enable. Provides main power to vacuum.
ZBRKRL	Z axis brake release.

Chapter:

Tool Management

The SERIES 8 router control system provides special capabilities to manage router bits. Many machines allow only manual tool changes, but some systems contain an automatic tool changer.

Up to 4 different tool types (T codes 1-4) may be defined in the system (manual or automatic tool change).

Tool number "T0" is used to designate no tool is in the spindle.

Up to 10 tool pod locations may be defined for an automatic tool change system. The 10 pods may be user-assigned to any of the 4 possible tool numbers.

Each tool type may have the following parameters defined:

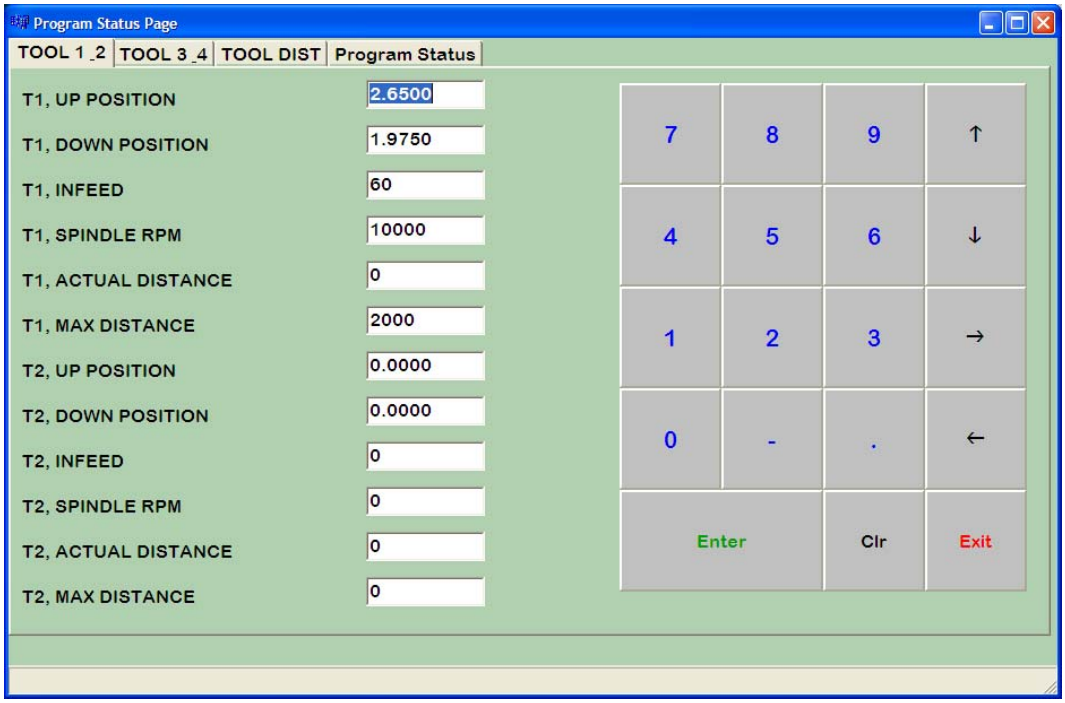
- Routing depth
- Routing clearance height
- Infeed rate for plunge moves
- Spindle speed
- Maximum routing distance allowed for new (sharp) tool

Tool data may be entered manually by the operator, or set via part program codes.

Tool Tables

Tool tables are available for display to define and view the current status of the tools. These are accessible under the Program Menu – status page, or by pressing the F7 function key.

Tool numbers 1-4 are split on two different parameter pages, with the first page showing tools 1 & 2, and the second page showing tools 3 & 4. The information is the same for all 4 tools:



Tool Table	
T1, UP POSITION	Clearance height of tool when making moves at rapid rate and not routing (M16 height).
T1, DOWN POSITION	Routing depth for tool for routing moves (M15 height)
T1, INFEED	Feedrate for Z-axis move when moving from UP position to DOWN position. *Note: When going from the DOWN position to the UP position, this is done at the rapid traverse rate.*
T1, SPINDLE RPM	Spindle speed for this tool.
T1, ACTUAL DISTANCE	Distance moved (in inches) for this tool during actual routing moves (M15). This distance is only tracked by the control if a MAX DISTANCE value has been set. This distance is cleared with the REPLENISH TOOL touch key during a tool change when max distance for this tool has been reached.
T1, MAX DISTANCE	Operator specified maximum rout distance to allow for this tool. When the maximum rout distance has been reached, the next cycling of the part program requires the operator change this tool.

Automatic Tool Change

On machines equipped with an automatic tool change system, two additional tabs appear on the program status page:

Automatic Tool Change Tabs	
TOOL PODS	Status information about the tool change pods
TOOL CHG	Tool pod configuration codes; auto tool change enable/disable

Note: System parameters relating to the automatic tool change sequence are defined in the Parameter Editor, and accessible only by the OEM. They are explained briefly here for general information purposes:

X-axis pod location for tool 1
Y-axis pod location for tool 1
Spacing of pods (always one row, along X-axis)
Z height for tool drop-off
Z height for tool pickup

On systems with automatic tool changers, the operator may configure the tool pod locations with any combination of tool types. If there are multiple tools of the same type defined (e.g. 5 pods defines as being tool 1), then the system automatically selects the next tool of the same type when the current tool is 'expired' (i.e., the maximum allowed rout distance has been reached for that tool), or 'broken' (on systems with broken bit detection capability). When all tools of the same type are expired or broken, the operator must replace the tools with new ones, and then indicate to the control that the tools have been replenished.

TOOL CHG Parameters

The tool pod layout may only be defined or changed when there is no tool active. To set no tool as active, program a "T0" via MDI.

Up to 10 tool pods may be available on the machine. Tool numbers may be assigned to tool pods with the following rules:

- Pods must be defined from left to right, lower tool numbers to higher tool numbers
- Pods left undefined are considered 'empty', and must be to the right of pods that are defined
- Total number of pods defined may not exceed 10
- Enter the desired number of tool pods for each tool code, and then set "CHANGE POD LAYOUT" to ON to have the system reset the tool information

Tool CHG Parameters	
AUTOMATIC TOOL CHG	ON/OFF - if ON, tool changes are made to the tool pod locations; if OFF, tool changes are manual.
CHANGE POD LAYOUT	Setting this ON signals the system to re-assign tool pods as defined in the following four parameters. The system automatically sets this back to OFF when the tool pods have been established. When the pod layout is changed, all pods that are defined are set to a 'sharp' tool status (no rout distance accrued).
T1, NUMBER OF PODS	Number of pods to be assigned for tool 1
T2, NUMBER OF PODS	Number of pods to be assigned for tool 2

Tool CHG Parameters	
T3, NUMBER OF PODS	Number of pods to be assigned for tool 3
T4, NUMBER OF PODS	Number of pods to be assigned for tool 4

TOOL POD Parameters

The tool pod parameters show the current status of the tool in each tool pod

TOOL-POOD-Parameters	
POD 1, TOOL 1	empty / broken / expired / sharp / active / in use
POD 2, TOOL 1	empty / broken / expired / sharp / active / in use
POD 3, TOOL 1	empty / broken / expired / sharp / active / in use
POD 4, TOOL 1	empty / broken / expired / sharp / active / in use
POD 5, TOOL 1	empty / broken / expired / sharp / active / in use
POD 6, TOOL 2	empty / broken / expired / sharp / active / in use
POD 7, TOOL 2	empty / broken / expired / sharp / active / in use
POD 8, TOOL 3	empty / broken / expired / sharp / active / in use
POD 9, TOOL 3	empty / broken / expired / sharp / active / in use
POD 10, TOOL 4	empty / broken / expired / sharp / active / in use
REPLENISH TOOL(S)	1 / 2 / 3 / 4 / all

Tool Status	
Empty	Setting a tool pod to empty designates to the control not to use this pod.
Broken	On machines with broken bit detection capability, when a broken bit is detected it is placed back in its pod and marked in this table.
Expired	A tool is marked expired when its routed distance exceeds the max distance setting for this tool type.
Sharp	Sharp signifies that this is a new tool, with no distance routed.
Active	Active denotes that this tool has some routing use (distance routed), but is not yet at its maximum distance, i.e. this tool is neither sharp nor expired.
In use	This tool is currently in the spindle.

The status of the tools is updated automatically by the CNC during the course of normal operation. The operator may override the status of tool pods by changing them here.

Chapter:

Part Programming Code Descriptions

The SERIES 8 CNC part program follow the standard EIA RS-274D, with several adaptations for circuit board routing.

Letter Code Summary

Letter Code Summary	
Letter	Description
G	Up to 3 G-codes can be set in one block
M	Up to 3 M-codes can be set in one block
T	Tool Number
X	X Coordinate
Y	Y Coordinate
Z	Z Coordinate
I	Offset in the X direction to the center of circle
J	Offset in the Y direction to the center of circle
K	Offset in the Z direction to the center of circle
E	Fixture offset
D	Tool offset
R	Circular interpolation radius
A	Circular interpolation radius
F	Feedrate command
S	Spindle speed command

G-Code Summary

NM = non-modal
* denotes default state for modal group after startup or reset

G-Code Summary		
G Code	Modal Group	Description
* G00	1	Rapid traverse - axis moves made at maximum feedrate
G01	1	Linear interpolation - axis moves made at programmed feedrate
G02	1	Circular interpolation, clockwise
G03	1	Circular interpolation, counter-clockwise
G04	NM	Program dwell
* G17	2	XY plane select for circular interpolation and cutter radius compensation
G18	2	ZX plane select for circular interpolation and cutter radius compensation
G19	2	YZ plane select for circular interpolation and cutter radius compensation
G21	1	G01 mode with sharp corner activation: deceleration between moves

G-Code Summary		
G Code	Modal Group	Description
G22	1	G02 mode with sharp corner activation: deceleration between moves
G23	1	G03 mode with sharp corner activation: deceleration between moves
G32	1	Rout full circle, clockwise
G33	1	Rout full circle, counter-clockwise
G34	1	CCW milled rectangular cavity
G35	1	CW milled rectangular cavity
G36	1	Milled circular cavity
* G40	3	Cancel cutter radius compensation
G41	3	Activate cutter radius compensation, right of path
G42	3	Activate cutter radius compensation, left of path
G49	1	CCW rectangular pattern
G50	1	CW rectangular pattern
G63	NM	Part fiducial check with XY offset
G67	NM	Absolute coordinate position programming for current block
G68	NM	Incremental coordinate position programming for current block
G69	NM	Machine absolute coordinate position programming for current block
G70	5	Inch programming
G71	5	Metric programming
G80	4	Cancel canned cycle
G81	4	Standard drilling cycle
G83	4	Peck drilling cycle
G85	NM	Part fiducial check with XY offset and rotation
G86	NM	Set table fiducial and camera offsets; activate fiducial checking
G87	NM	Turn off fiducial checking
G88	NM	Verify table fiducial position
G90	6	Absolute dimension programming
G91	6	Incremental dimension programming
G92	NM	Program zero set = defines new program zero offset from machine home
G93	7	Inverse time feedrate programming (1/T)
G94	7	Feed per minute feedrate programming (IPM or MMPM)
G98	NM	Program zero set - defines new program zero offset from machine home

M-Code Summary

M-Code Summary	
M Code	Descriptions
M00	Program stop
M01	Optional program stop
M03	Spindle on
M05	Spindle off
M06	Tool parameter definition
M07	Tool offset definition
M13	Spindle on, hold program until spindle at speed
M14	Spindle off, hold program until spindle stopped
M15	Head down - move Z axis to routing depth
M16	Head up - move Z axis to routing clearance height
M17	Head up without pre-vacuum shutoff
M24	Step and Repeat: end of pattern
M25	Step and Repeat: start of pattern
M26	Step and Repeat: offset definition for pattern repeat
M27	Step and Repeat: end of offset definitions
M30	End of program
M36	Cancel M37
M37	Pressure foot clamp override
M42	Vacuum on
M43	Vacuum off
M44	Ignore M43 block
M45	Cancel M45
M47	Program stop with operator message
M51	Pressure foot up
M52	Pressure foot down
M53	Cancel M51 and M52
M60	Laser etch function
M80	Mirror X - step & repeat only
M90	Mirror Y - step & repeat only
M98	Check fixture bar code
M99	Enable manual camera commands
M931	X-axis mirror
M941	Cancel X-axis mirror
M932	Y-axis mirror
M942	Cancel Y-axis mirror

Part Programming Examples

For most router programs, the part program follows a typical format. This section describes a typical part program to help illustrate the use of the program codes.

Note 1: It is typical to define all tool parameters at the top of the program, although this is not required. It is necessary to define the tool parameters (M06, M07) prior to activating the tool (T01)

Note 2: It is typical to rout the outside of a part in the counter clockwise direction, with tool radius compensation to the right of the path

Typically, a part program follows this structure:

- Tool information (M06, M07)
- Define XY coordinate system (G92)
- Activate first tool (T)
- Start of pattern code (M25)
- Rapid to initial plunge point on the part (G00)
- Enable cutter compensation (G41)
- Plunge into part (M15)
- Make routing moves (G01, G02, G03)
- Raise Z to clearance position (M16)
- Repeat sequence of plunge-rout-retract to complete all needed routing moves
- End of pattern (M24)
- Repeat pattern with offsets (M26)
- End of pattern repeats (M27)
- Activate second tool, if needed, and repeat steps as above
- End of program (M30)

Example:
(The block numbers "Nxxx" are not required, but are used here for reference.)

```
N001    M06 T1 F50 R3.0 Z1.0 S40000; Defines parameters for tool #1
                                   ; - Infeed of 50 IPM
                                   ; - Clearance plane of 3.0"
                                   ; - Routing depth of 1.0"
                                   ; - Spindle speed of 40000 RPM
N002    M07 D1 X.093     ; Tool diameter for tool offset #1, et to .093"
N003    G92 X1.5 Y2.5    ; Defines program zero relative to machine absolute
                         ; zero
N004    T01              ; Activates parameters for tool #1
                         ; Also makes tool change to tool #1 if not already
                         ; active
N005    M25              ; Start of routing pattern (for later step and
                         ; repeat)
N006    G00 X.8 Y1.      ; Position at rapid rate to edge of part
N007    G41              ; Define cutter compensation to be right of part
N008    M15              ; Move Z down to routing depth
```

```
N009    G01X3.F100.      ; Feed at 100 IPM to X3
N010    Y2.5             ; Feed to Y2.5
N011    M16              ; Raise Z axis to clearance position
N012    G00Y3.           ; Move at rapid rate to Y3;
N013    M15              ; Move Z down to routing depth
N014    G01Y5.           ; Feed to Y5.
N015    G03X2.5Y5.5R.5   ; Rout 90 degree counter-clockwise arc
N016    G01X.5           ; Feed in linear move to X.5
N017    G03X0Y5.R.5      ; Rout 90 degree counter-clockwise arc
N018    G01Y3.           ; Feed in linear move to Y.3
N019    M16              ; Raise Z axis to clearance position
N020    G00Y2.5          ; Move at rapid rate to Y2.5
N021    M15              ; Move Z down to routing depth
N022    G01Y.8           ; Feed in linear move to Y.8
N023    M16              ; Raise Z axis to clearance position
N024    G40              ; Cancel cutter compensation
N025    M24              ; End of pattern
N026    M26X5.Y0         ; Repeat pattern with 5" shift in X axis
N027    M26X0Y6          ; Repeat pattern with 6" shift in Y axis
N028    M26X-5.Y0        ; Repeat pattern to -5" shift in X axis
N029    M26              ; 'Dummy' M26 is ignored
N030    M27              ; End of pattern repeats
N031    M30              ; End of program
```

N-code Sequence Number

The N-code lets the programmer give a sequence number to a program block to assist in identifying it. The part programmer may sequence every block with an N-code of up to 4 digits.

The N-code is essentially ignored by the control for specific block execution, but may be used in conjunction with the APP "GTO" command to identify a block to be skipped to.

```
N102X.5Z.5
N3Z1.2
N2000M30
```

Program Format Codes (G70-G71, G90-G91)

The G90 and G91 codes determine how X and Z coordinate information in the part program are interpreted.

In G90 (absolute) mode, the program coordinates specify the desired axis position as an absolute position from program zero:

```
G90
X1.5Z1.5

The machine will move to the position X1.5 Z1.5, and these values will be
the displayed axis position on the Production Page.
```

In G91 (incremental) mode, the program coordinates specify the desired axis position as an incremental distance from the last programmed axis position:

```
G90
X1.5Z1.5
G91
X.5
X-.75
```

The first two blocks move the machine to the position X1.5 Z1.5, as described above. The G91 block then sets up incremental mode. The block "X.5" then moves X a distance of .5 to 2.0. The X-axis displayed position will be 2.0. The last block, "X-.75", moves X in the minus direction .75 to 1.25.

Both G90 and G91 are modal codes. Upon system startup or a Control Reset, G90 mode is active.

Tool Information Codes (M06, M07)

Tool information can be predefined so that at any time in the part program a tool code (like "T1") can be entered and all the related parameters for that tool are automatically set. Tool parameters include:

- Spindle speed
- Z clearance height
- Z rout depth
- Z feedrate for plunge move
- Tool diameter for cutter compensation

Tool parameters may be set via the Program Status page by the operator, but are more typically written into the part program.

M06 - Tool Feed and Speed Setting with Up and Down

The M06 block is used to pre-set certain parameters to a specific tool number. No machine motion or tool change occurs when this block executes. This block enters data into the tool table.

M06: set up tool information

```
M06 T- R- Z- F- S-
```

Data is copied into the tool table as follows:

M06 Tool Parameters	
Letter	Description

M06 Tool Parameters	
Letter	Description
T	Specifies the tool number which these parameters apply to. Valid range is 1-4.
R	Retract plane, also known as the "UP" position for an M16, or top of drill stroke. (inches/millimeters)
Z	Z depth plane, also known as the "DOWN" position for an M15, or bottom of drill stroke. (inches/millimeters)
F	Infeed rate of the Z-axis when moving to it's down position for an M15 or drill stroke. (IPM / MMPM)
S	Spindle Speed (RPM)

This same information may be entered directly on the program status page, also accessible from the hotkey labeled “TOOL DATA”.

M07 - Tool Diameter Setting

The M07 is similar to the M06 command, except it applies to the tool diameter offset.

M07: enter tool diameter

```
M07 D- R- (X-)
```

Data is copied into the tool offset table as follows:

M06 Tool Parameters	
Letter	Description
D	Tool offset number
R	Tool diameter
X	Tool diameter

The tool diameter specified in the tool offset number is used for tool radius compensation when cutter compensation is active.

The X and R code have the same meaning. The R code is the newer format; X is maintained for backward compatibility. If both are programmed, the X code is used.

Tool Number vs. Tool Offset Number:

"Tool number" (which is used with the M06 block), and "Tool offset number" (which is used with the M07 block) are not the same. Data associated with the tool number becomes effective when that tool is programmed, and relates to its spindle speed, UP/DOWN, and infeed. The tool offset applies to the tool diameter used for cutter compensation.

Programming the tool offset can be done with the D-code (shown below), but more typically is programmed directly with the T-code:

```
T0104

The first two digits after the "T" specify the tool number; the second two
specify the tool offset number.  In this example, it is T1, offset 4.
```

If the second two digits are not programmed, then it is assumed that the offset number is the same as the tool number:

```
T02

This is interpreted by the control as T2, offset 2.
```

Alternately, the D-code can be used to set or change the current tool offset number:

```
D05

Activates tool offset 5 for the current tool.
```

Tool Information Codes using Program Header

As an alternative to programming M06 and M07 tool setup codes and for the purpose of backward compatibility to previous part programming formats, a 'Program Header' may be used. This header must appear at the beginning of the part program, and start with a "%", and end with a "%".

Within the header, the parameters for a tool are defined as follows:

Tool Parameter Definitions	
Letter	Description
T	Tool number for which parameters are being set
S	Spindle speed, in hundreds of RPM
F	Infeed rate of Z-axis for M15 moves, in inches per minute
U	UP position of the Z-axis for M16 moves
D	DOWN position of the Z-axis for M15 moves

```
T01S300F30U2.5D1.2

• T01 - data for tool #1
• S300 - spindle speed of 30,000 RPM
• F30  - infeed rate of 30 IPM
• U2.5 - M16 UP position
```

Within the header, the tool diameter is specified as follows:

```
CP01 .066

Where "CP01" refers to tool offset #1, and ".066" sets defines the diameter
for offset #1.
```

Position Register Preset (G92)

This block defines a new XY coordinate system zero reference point. The values for XY entered on this block represent the distance from the machine home position to the program zero position.

```
G92X1.5Y2.2

The program zero position will be 1.5" from the X absolute home position of
the machine, and 2.2" from the Y absolute home position of the machine.
```

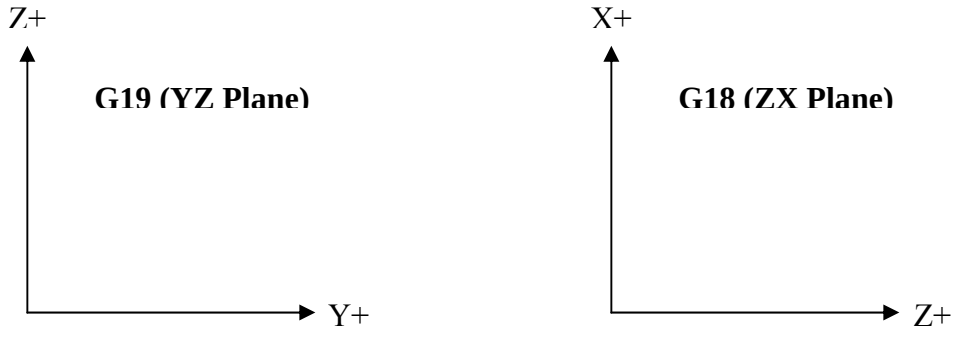
Interpolation Codes (G00-G03) and Plane Selection (G17-G19)

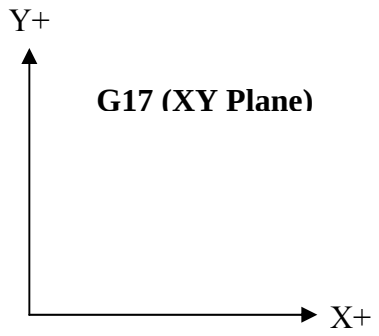
The SERIES 8 CNC provides linear, circular, and helical contouring of the machine axes. Circular contouring may be programmed in of the three orthogonal planes:

- XY (G17)
- ZX (G18)
- YZ (G19)

In addition, any other programmed axes not in the defined plane may be programmed also, and a linear move is made in that axis along with the programmed circular move.

For circular contouring modes, as well as cutter compensation offsets, directions are determined by the 'right hand rule'. This means that determination of clockwise (CW) and counter-clockwise (CCW) as well as right/left must be made when viewing the axes in the correct orientation, as defined below:





G00 Rapid Traverse Positioning

The G00 code sets rapid traverse positioning mode. Subsequent moves are at the maximum feedrate available for the machine. G00 also cancels a canned cycle, if it was active (G81).

```
G00X1.5Z2.          Move at maximum speed to position X1.5, Z2.
X5.6                Move at maximum speed to position X5.6
```

G01 Linear Interpolation

The G01 code sets linear interpolation mode. Subsequent moves are in a straight line at the current program feedrate. The feedrate is determined by the last programmed F-code. When a G01 move is executed, the machine moves from its current position to the programmed position in a straight line.

```
G00XZ         Position to X0 Z0 at max speed
F50           Set feedrate to 50 IPM
G01X2.5       Feed to X2.5, Z make no move at stays at Z0
X1.Z4.6       Feed to X1., Z4.6
```

G02/G03 Circular Interpolation

The G02 and G03 commands are used to create circular motion. The G02 specifies a clockwise (CW) direction, and the G03 specifies a counterclockwise (CCW) direction. Motion is at the current feedrate.

The starting point is the current axis position when this block begins execution. The coordinates in the G02 or G03 block specify the arc endpoint. There are two different methods for defining the center of the arc:

1) The radius of the move may be programmed with an R-code, and the control automatically calculates the arc center; or
2) The arc center may be programmed and the control calculates the arc radius.

Method 1 is generally the simpler to program, but either method is valid.

Radius Programming

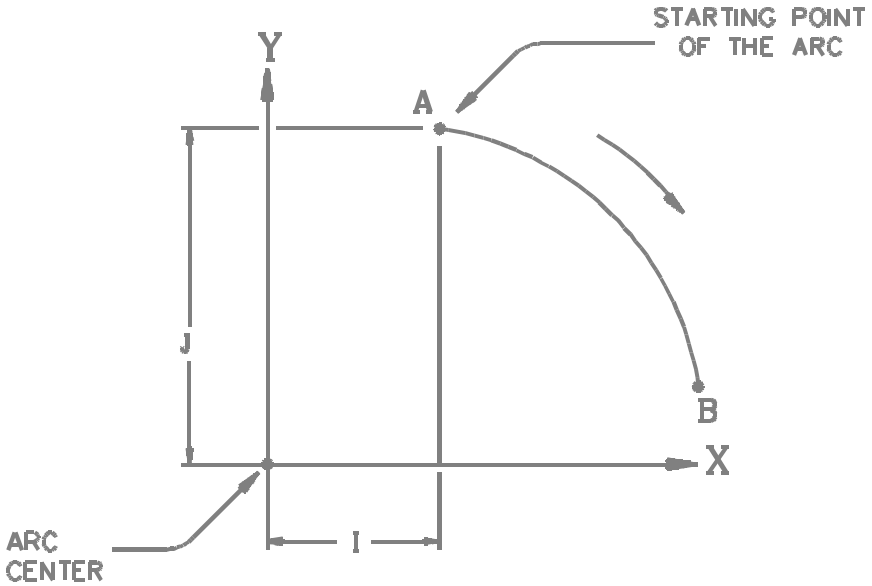
Radius programming of a circular move is accomplished with an R-code. The circular move is between 0° and 180° if the specified R value is positive or between 180° and 360° if the specified R value is negative. If the absolute value of the R value specified is less than one-half the distance between the move start and end points (i.e., the chord), the R value is adjusted so a 180° move is made (the R value equals half the chord). If no R is specified, the last value of R is used.

```
G00X0Y0       Position to location X0 Y0
F10           Set feedrate of 10 IPM
G02X2.Y0R1.   Move in CW direction to X2. Y0 at a radius of 1.; this would
              result in a 180° move
G03X1.Y1.     Move in CCW direction to X1. Y1. at a radius of 1. (last
              programmed R-code); this would result in a 90° move.
X0Y0          Move in CCW direction to X0 Y0 at a radius of 1.; the G03 and
              the last programmed R-code are modal.  The move made would be
              90°.
X1.Y1.R-1.    Move in CCW direction to X1 Y1 at a radius of 1.; this would
              result in a 270° move.
G01X1.Y1.     Move in linear mode to X1. Y1.
```

Special care must be taken when programming with the R-code, as the control always makes some type of circular move to the desired endpoint, even if a programming mistake is made and the wrong coordinates or the wrong radius is specified.

Arc Center Programming:

The I and J codes are used in circular interpolation as arc center offsets (in the XY plane). The offsets are unsigned, incremental axis values defining the coordinates of the center of the arc with respect to the starting point of the arc. The programmed arc must not cross a quadrant boundary or the error message INVALID ARC is displayed (the arc center is used as the origin for defining quadrants). If an arc center offset value is zero, it need not be programmed. The calculated arc endpoint based on the programmed I and K codes must be located on the arc within an allowable tolerance or the error message INVALID ARC is displayed.



The diagram illustrates the usage of the I and J codes in arc center programming of a G02 command. The I-code specifies the arc offset in the X-axis, and the J-code specifies the arc offset in the Y-axis. Both are unsigned values. The coordinates of point A are the endpoint of the preceding block, and the coordinates of point B are the coordinates in the G02 block.

```
Assume a starting point of X0 Y0, and the desired move is a CW 90° arc of
1." radius to the location X1. Y1. The center of the arc is at X1. Y0.  The
distance from the starting point to the center is 1." along the X-axis, so
the I-code will be 1.  The distance from the start point to the center
along the Y-axis is 0, so the J-code will be 0 (or it can be omitted since
it is equal to zero). The programmed block would then be:

  G02X1.Y1.I1.
```

Plane Selection (G17-G19)

The plane selection codes define the plane in which circular interpolation codes (G02, G03) is interpreted. The XY plane (G17) is the default if not otherwise specified.

- G17 - XY plane
- G18 - YZ plane
- G19 - XZ plane

If a G02 or G03 is programmed with all three axes (X, Y, Z), the two axes in the active plane move in a circular motion, while the third axis moves linearly. This creates a helical type move.

```
G17                 Sets the XY plane selection
G01X0Y0Z0F10.       Move XYZ to their zero position at 10 IPM
G02X2.Y2.R2.        Makes a 90 degree arc with 2" radius in XY plane; Z does
                    not move
```

```
G17                 Sets the XY plane selection
G01X0Y0Z0F10.       Move XYZ to their zero position at 10 IPM
G02X2.Y2.Z2R2.      Makes a 90 degree arc with 2" radius in XY plane; along
                    with arc move, the Z-axis moves from 0 to 2" in a linear
                    fashion.
```

```
G19                 Sets XZ plane selection
G01X0Y0Z0F10.       Move XYZ to their zero position at 10 IPM
G02X2.Y2.Z2R2.      Makes a 90 degree arc with 2" radius in XZ plane; along
                    with arc move, the Y-axis moves from 0 to 2" in a linear
                    fashion.
```

Part Rotation (G55, G56)

Part Rotation	
G55	Turns on part rotation
G56	Turns off part rotation

The G55 code is programmed with an R code (in degrees) to activate and define the angle for part rotation. While this is active, all endpoints in the active plane (G17-G19) are rotated from the Program Zero by the angle defined by the R code.

```
G00X5Y5       ; MOVE 1
G55R45        ; TURN ON ROTATION WITH AN ANGLE OF 45 DEGRESS
G01F50X10     ; MOVE 2
Y10           ; MOVE 3
G56           ; TURN OFF ROTATION
M30

Move 1 to X5Y5
Move 2 to X7.0711 Y7.0711 (X10Y10 rotated by 45 degrees)
Move 3 to X5 Y14.1421
```

Part Scaling (G57, G58)

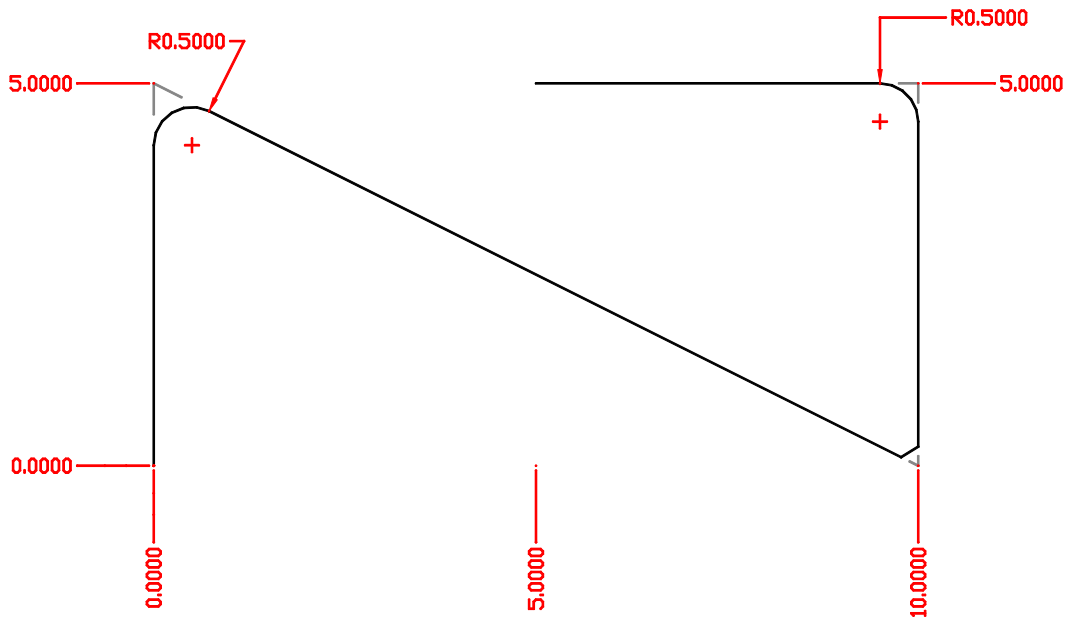
Part scaling	
G57	Turns on part scaling
G58	Turns off part scaling

The G57 code is programmed with an R code to scale all program dimensions in the active plane by the value of the R code.

Inserted Arc or Chamfer (G10)

The G10 code provides a simplifier means to add an arc or chamfer move between two linear moves, without requiring the programmer to calculate the arc/chamfer endpoints. For arc insertions, the arc is inserted such that the start and end positions are tangent to the programmed lines.

Programmed along with an L or R code, G10 inserts a chamfer (L) or radius (R) between two linear moves. Both moves must be linear. The second move may also have a G10 if another insertion is desired.




```
G00X0Y0          ;Rapid move to 0,0
G01F50G10Y5R.5   ;Move 1 - inserted .5" arc
G10X10Y0L.25     ;Move 2 - inserted .25" chamfer
G10Y5R.5         ;Move 3 - inserted .5" arc
X5               ;Move 4
M30
```

Move 1 does a linear move from X0Y0 to X0Y4.1909 and then does a clockwise arc inserted move to blend with move 2. This arc has a radius of .5" and ends at X.7236Y4.6382.

Move 2 starts at this point and does a linear move to X9.7764X.1118 and then does a linear .25" chamfer move to X10Y.25.

Move 3 does a linear move to X10Y4.5 and then does a counter-clockwise .5" arc to X9.5Y5.

Move 4 is a linear move to X5Y5.

M15/M16 Head UP and DOWN Control Codes

The M15 and M16 codes are used in routing mode to control the head up and down moves. The UP and DOWN position associated with these codes are set with the M06 block.

Programming an M15 causes the Z-axis to move to the down position at the infeed associated with the active tool, and turns on the machine vacuum.

Programming an M16 causes the Z-axis to move to the up position at the programmed retract feedrate, or at the maximum speed for the Z-axis if no retract feedrate is specified. Both of these codes turn off cutter compensation, if it is active, while the head is up, and turn off the machine vacuum.

If X or Y coordinates are programmed in the same block with any of these codes, the Z-axis motion occurs first, followed by the simultaneous motion of the X and Y axes.

```
M06T1F50S40000R.1Z.5     Set UP position to .1, DOWN position to .5
T1                       Activate T1 parameters set in previous block
G00X1.5Y1.5              Position to X1.5 Y1.5 in rapid traverse
M15                      Lower Z-axis to .5
G01X3.F75                Move to X3. Y1.5 at 75 IPM
M16                      Raise Z-axis to .1
M30                      End program
```

One-time change of Z position for M15 or M16:

Optionally, the M15 or M16 block may contain a "Z" coordinate to specify the Z-axis position for the move:

```
M15Z2.2            Moves the Z-axis to Z2.200
```

```
M16Z3.8            Moves the Z-axis to Z3.800
```

When the Z-code is used on the M15 or M16 block, it overrides the Z up or down position previously defined in the M06 block, but this override is effective for only the current M15 or M16 block. A subsequent M15 or M16 without a Z-code move to the previously defined up or down position.

G40/G41/G42 Cutter Compensation Codes

The cutter compensation codes are used to activate and deactivate a feature of the control which permits programming of the actual finished part dimensions, without performing special calculations to compensate for cutter radius. When cutter compensation is active, the control

automatically compensates for the cutter radius by shifting the tool position relative to the programmed position.

The diameter of the tool is specified in the tool offset table (set with the M07 block), and activated when a T-code is programmed. The tool offset number normally matches the tool number, although it may be specified separately.

Cutter Compensation Code	
Code	Description
G40	Cancels cutter compensation
G41	Cutter compensation to the right of the tool path
G42	Cutter compensation to the left of the tool path

The G41 or G42 code must be programmed prior to the M15 head down command for routing. This allows the cutter compensation offset to be applied prior to lowering the tool into the work piece.

The G40 code to cancel cutter compensation should be programmed prior to the end of pattern (M24) if a step and repeat construct is used.

Cutter compensation is inserted by the control in a direction perpendicular to the current direction the tool is moving. Thus, the actual direction of compensation cannot be determined until a move is programmed. The direction of compensation is best understood by thinking in terms of the tool position and looking forward from the tool in the direction it is moving. Compensation can then be applied to the left or to the right of this motion.

The G41 code selects cutter compensation right, and G42 selects cutter compensation left. The G40 code cancels cutter compensation. Both linear and circular moves are compensated when G41 or G42 modes are active.

In order to operate correctly, cutter compensation requires that the control "look ahead" in the part program to calculate block intersection points and modify move endpoints accordingly to avoid gouging the part. Whenever a block intersection occurs where the cutter is on the outside of two moves which create an angle of less than 70°, an arc move is inserted by the control to minimize the cutter path.

If a G41 or G42 is programmed while the Z-axis is in the UP position, a subsequent M15 code is held up until the block look-ahead is performed to determine the direction of the initial cutter compensation. The compensation move occurs prior to the actual head down motion.

```
T0101   ; Select tool number 1, offset number 1
G00XY   ; Position to X0 Y0
G41     ; Activate cutter compensation right
M15     ; Request head down (actual motion will not occur until the next
        ; block is processed )
```

```
G01X1.  ; rst a move will be made to compensate for the cutter radius, then
        ; the head will lower, and finally a move will be made along the X-
        ; axis.  The final endpoint is determined by the cutter radius.
Y1.     ; Continue routing a square pattern
X0
Y0
M16     ; Raise Z-axis to UP position
G40     ; Cancel cutter compensation
M30     ; End program
```

Step & Repeat Part Programming Codes

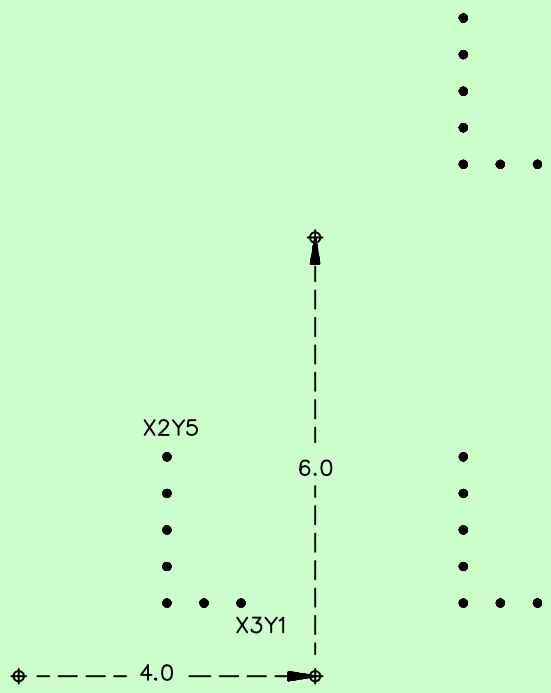
Step and repeat is a programming feature which permits programming a sequence of moves, and then repeating that pattern in a new XY location, with optional mirroring of the pattern.

The original pattern is programmed with the drilling or routing codes described in previous sections, with an M25 block indicating where the pattern starts. Then a series of blocks at the end of the pattern can be programmed which cause the pattern to repeat, and specify the move offsets or axis mirroring desired. A summary of the codes used in step and repeat follows:

Step & Repeat Code	
Code	Description
M25	Start of pattern
M24	End of pattern
M26	Offset block
M27	End of offset blocks
M80	Mirror X-axis coordinates
M90	Mirror Y-axis coordinates

```
  T01        ; Select tool 1
  M25        ; Start of pattern
  X3.Y1.     ; Drill hole at location X3. Y1.
  X2.        ; Continue drilling
  X1.        ; Continue drilling
  Y2.        ; Continue drilling
  Y3.        ; Continue drilling
  Y4.        ; Continue drilling
  Y5.        ; Continue drilling
  M24        ; End of pattern
  M26X4.0    ; 1st repeat pattern offset
  M26Y6.0    ; 2nd repeat pattern offset
  M26        ; Dummy offset block
  M27        ; End of offsets
  M30        ; End of program

Visual representation of above program
```



The first block performs a tool change and activates the tool parameters for tool #1.

The M25 defines the start of pattern.

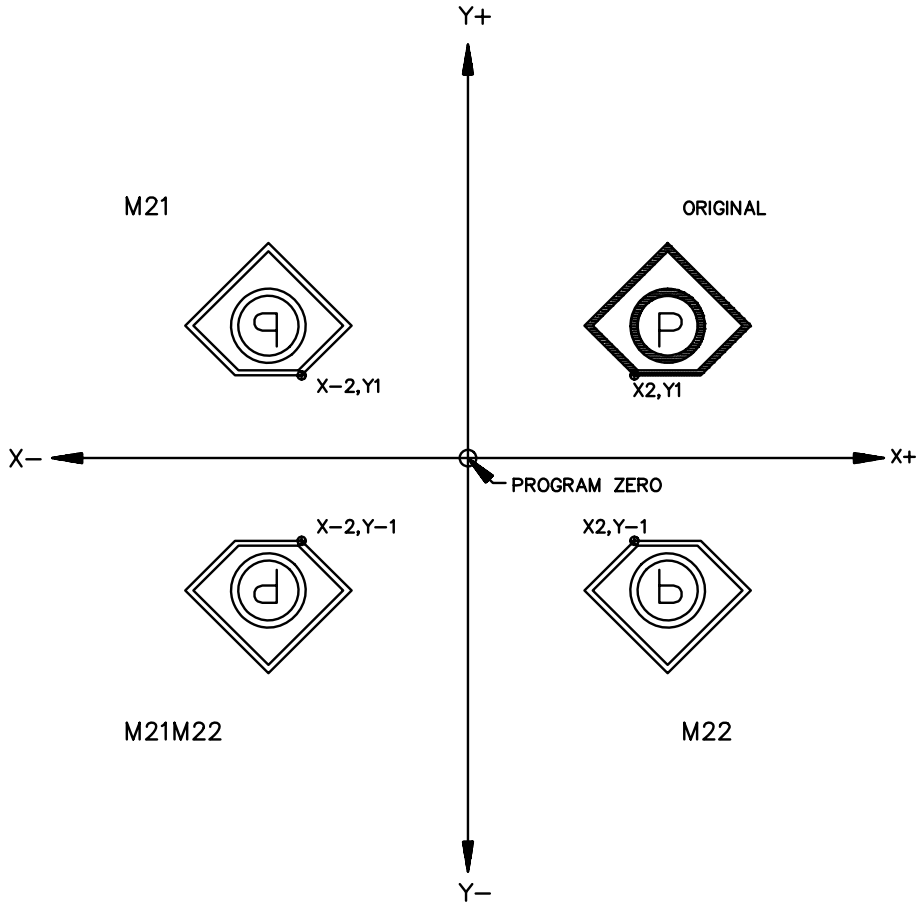
The next several blocks define XY locations for holes to be drilled.

The M24 denotes the end of the pattern.

The first M26 block specifies that the pattern is to be repeated offset from the first pattern by 4.5" in the X positive direction. The repeated pattern starts with the block "X1.Y1.". The second M26 block specifies that the pattern is again to be repeated offset from the second pattern by 4.5" in the Y positive direction. Successive repeats of the same pattern accumulate the step and repeat offsets. The final M26 block is a "dummy"; the last M26 block in a step and repeat offset section is always ignored, even if it contains XY coordinates.

The M27 indicates that the step and repeat section is complete.
This program causes a pattern of four holes to be drilled three times: once in the original programmed location, once with all coordinates offset by X4.5", and once with all coordinates offset by X4.5" and Y4.5".

In addition to XY offsets in the M26 block, mirror and axis swap codes may be programmed. A mirror code (M80 or M90) inverts the sign of all programmed coordinate for the mirrored axis (M80=X, M90=Y). That is, all negative coordinates become positive and all positive coordinates become negative.



The above drawing illustrates the effect of executing a step and repeat with axis mirror codes. The patterns shown assume the M26 offset block contained no XY offset.

Note: After executing a pattern repeat with mirror or swap codes, these codes are automatically canceled when the pattern is completed. A subsequent offset block must repeat the mirror or swap codes if they are still desired.

Nested Step and Repeat

Step and repeat patterns may be nested by defining one pattern to be made up of a previous pattern with the step and repeat offsets. This is accomplished by placing an M01 at the end of the first set of step and repeat offsets, and then defining additional step and repeat offsets. Up to five levels of step and repeat nesting are permitted.

```
T01          ; Select tool 1
M25          ; Start of pattern
X1.Y1.       ; Drill hole at location X1. Y1.
X2.          ; Continue drilling
Y1.5
X3.Y4.
M24          ; End of basic pattern
M26X4.5      ; 1st repeat pattern offset
M26Y4.5      ; 2nd repeat pattern offset
```

```
M26           ; Dummy offset block
M24           ; End of pattern; create 2nd level of nesting
M26X10.M80    ; Repeat above pattern offset in X positive direction by 10.",
              ; with all X coordinates mirrored.  The pattern repeated
              ; includes all three sets of holes drilled by the first step
              ; and repeat nest level.
M26           ; Dummy offset block for 2nd level nest
M27           ; End of offsets
M30           ; End of program
```

1) The first 9 blocks are identical to example 1 and will drill three patterns of four holes each as defined above.
2) The second M24 programmed indicates that nesting is to occur. It marks the "end of pattern" for the entire construct of the first set of step and repeat blocks.
3) The block "M26X10.M80" then causes a repeat of the entire 1st pattern, which includes the first set of offsets. This program will drill six sets of the four hole pattern defined in blocks 3-6.

R-Code: Repeat Step and Repeat Offsets

The R-code specifies a repeat of a step and repeat offset pattern. The format is as follows:

R#M26X#Y#

Repeat Step & Repeat Offsets	
Letter	Description
R	Number of repeats
X	Offset in the X axis, between pattern calls
Y	Offset in the Y axis, between pattern calls

If no M27 follows this block, the last repeat of the block does not cause the step and repeat pattern to be run (in conjunction with the standard step and repeat programming format of requiring a dummy M27 offset block).

Canned Cycles

Certain canned cycle programs are provided with the control system. These are APP routines supplied by the machine tool builder to facilitate certain types of common routing or drilling requirements.

G81 Drilling Cycle

The G81 code is a canned cycle that creates a special move sequence. The format is:

G81 X# Y#

- Position XY at the rapid traverse rate
- Move Z to the UP position at the rapid traverse rate

- Feed Z to the DOWN position at the infeed rate
- Raise Z to the UP position at rapid traverse rate

The Z drill UP and DOWN positions are the same as the routing depth and clearance positions used for the M15 and M16 commands.

The G81 code is 'modal', meaning that once a G81 is programmed, all subsequent blocks with and X or Y move runs the drilling cycle. G81 mode is canceled with a G80, G83, or G00 code.

G83 Peck Drilling Cycle

The G83 code is similar to the G81 drill cycle, with the added capability to divide the drill stroke into multiple pecks. It is generally used for deep holes, or difficult material. Block format:

G83 X# Y# I# J#

The drill sequence is as follows:

- Position XY at the rapid traverse rate
- Move Z to the UP position at the rapid traverse rate
- Feed Z to the position specified by the I-code at the infeed rate
- Raise Z .050" at the rapid traverse rate
- Feed Z down the amount specified by the J-code at the infeed rate
- Continue raising Z by .050: and feeding down the J-code amount until the DOWN position is reached
- Raise Z to the UP position at rapid traverse rate

The final Z drill depth (DOWN position) and UP position are the same as the routing depth and clearance positions used for the M15 and M16 commands.

The G83 code is 'modal', meaning that once a G83 is programmed, all subsequent blocks with and X or Y move runs the drilling cycle. G83 mode is canceled with a G80, G81, or G00 code.

In subsequent blocks after a G83, it is not necessary to reprogram the I and J codes if the values are to be the same.

```
  G83 X5. Y2. I.7 J.2


Assume that the UP position is set to 1.0, and the DOWN position is set to
0.2.  The following move sequence would occur:

  - move in rapid traverse to X 5.", Y 2."
  - move in rapid traverse to Z 1." (if not there already)
  - feed in Z to .7"
  - rapid in Z to .75"
  - feed in Z to .5"
```

```
  - rapid in Z to .55"
  - feed in Z to .3"
  - rapid in Z to .35"
  - feed in Z to .2"
  - rapid in Z to 1."

Note that the last feed move did not move the full .2" increment; instead
the control will stop at the programmed final depth.
```

G32/G33 Routed Circle

This canned cycle routs a complete circle.

Routed Circle	
Code	Description
G32	Clockwise Circle
G33	Counter-Clockwise Circle

Block format:

G32 X# Y# A# Z#
G33 X# Y# A# Z#

Where:

G32/G33, Routed Circle	
Letter	Description
X	X coordinate of the center of the circle
Y	Y coordinate of the center of the circle
A	Circle radius
Z	Routing depth, if omitted the M15 DOWN position is used

The final radius routed by the control will be offset by the currently active tool diameter.

The routed circle is cut by having the Z-axis plunge .010" in from one edge and make a 90 degree arc to blend into the circle edge. The full circle is then routed, followed by a 90 degree arc to blend away from the circle edge, ending .010" off the edge of the circle, where the Z-axis is retracted to its UP position.

G49/50 Rectangular Canned Cycle

This canned cycle routs an inside rectangular pattern with automatic cutter diameter compensation toward the inside of the rectangle. The sides of the rectangle are parallel to the machine axes. The Z-axis plunges to the inside of the rectangle, off the final surface edge, and

make a lead-in move. The direction of routing may be either clockwise (G50) or counter-clockwise (G49). There are two possible command formats for this code:

G49X#Y#I#J#

G49, Routed Rectangle	
Letter	Description
X	X coordinate of the center of the rectangle
Y	Y coordinate of the center of the rectangle
I	Total length of the rectangle in the X direction
J	Total width of the rectangle in the Y direction

Where X#Y# is the center of the rectangle; I is the total length of the rectangle in the X-direction, and J is the total width in the Y direction.

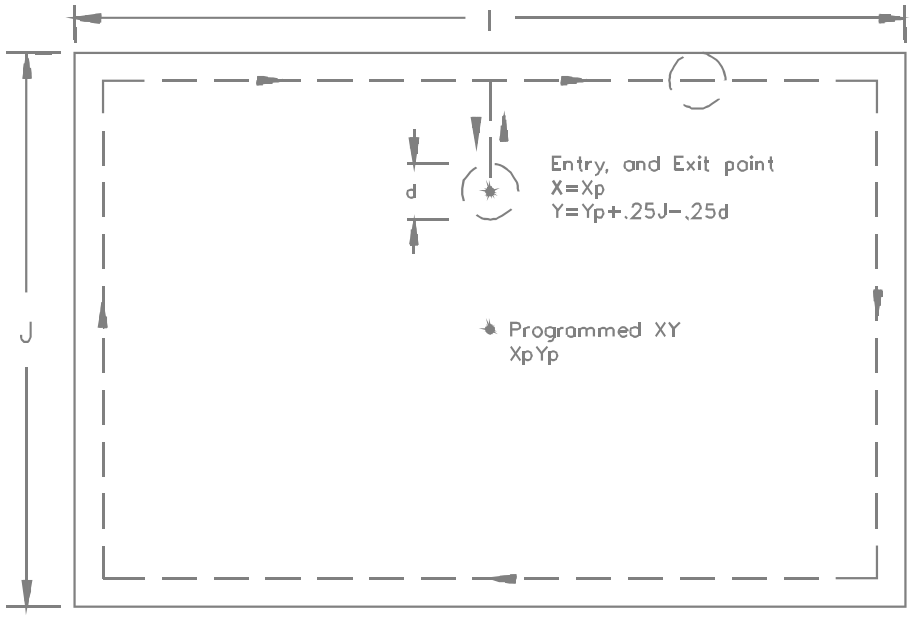
G49X#Y#R#

G49, Routed Rectangle	
Letter	Description
X	X coordinate of the center of the square
Y	Y coordinate of the center of the square
R	Length of one side of the square

Where X#Y# is the center of the square; R is the length of one side of the square.

The complete cycle consists of the following steps:

1) Raise Z-axis to the UP position.
2) Position XY to the entry point shown in figure below.
3) Plunge Z-axis to the DOWN position.
4) Rout the rectangle as shown below, ending at the exit point.
5) Raise Z-axis to the UP position.



G50

G49 is the same but moves CCW.

Fixture Routing Cycles

In making fixtures for holding the circuit boards to be routed, it is often necessary to mill out rectangular or circular cavities to provide clearance for components on the board. The following sections describe cycles and programming techniques to facilitate this process.

G34/35 Milled Rectangular Cavity

These canned cycles are programmed the same as the G49/G50 cycles. The difference is that all material within the rectangular area is milled out. A rectangle is first routed around the center of the area and successively larger overlapping rectangles are routed until the programmed dimensions are reached. G34 routs in a clockwise direction and G35 in a counter-clockwise direction.

G34X#Y#I#J#

G34/G35, Milled Rectangular Cavity	
Letter	Description
X	X coordinate of the center of the rectangle
Y	Y coordinate of the center of the rectangle
I	Total length of the rectangle in the X direction
J	Total width of the rectangle in the Y direction

G34X#Y#R#

G34/G35, Milled Rectangular Cavity	
Letter	Description
X	X coordinate of the center of the square
Y	Y coordinate of the center of the square
R	Length of one side of the square

The complete cycle consists of the following steps:

1) Position XY to the center of the rectangle.
2) Plunge Z-axis to the DOWN position.
3) Rout the rectangle in successively larger moves, removing material from the center out.
4) Raise Z-axis to the UP position.

G36 Milled Circular Cavity

This cycle routs a circular pattern starting in the center, with successively larger circles until the programmed radius is reach. Each step of a larger circle is based on the programmed tool diameter.

G36X#Y#A#Z#

G36, Milled Circular Cavity	
Letter	Description
X	X coordinate of the center of the circle
Y	Y coordinate of the center of the circle
A	Radius of the circle
Z	Depth of the rout

The complete cycle consists of the following steps:

1) Position XY to the center of the circle.
2) Plunge Z-axis to the DOWN position.
3) Rout the circle in successively larger moves, removing material from the center out.
4) Raise Z-axis to the UP position.
5) Position XY to the center of the circle.

NOTE: if the Z code is omitted, the preset DOWN value for the M15 command will be used for the rout depth.

Looping Function to Mill Cavity at Different Depths

The G34, G35, and G36 cycles can be combined with APP codes to provide a looping construct where the cavity can be milled in successively deep Z-levels. To accomplish this, the M06 tool code block that defines the routing depth must be inside the loop, changing the Z depth each pass.

Refer to the APP programming section later in this manual for a description of the APP codes used in this example.

```
M07D01X.093              ; Sets tool offset #1 diameter to .093"
(FOR,%1=3.2,3.0,-.1)     ; Start of loop: variable "%1" will be set to 3.2"
                         ; for the first pass, then will decrease by .1"
                         ; each pass until it reaches the value of 3.0".
                         ; This variable will be used to set the Z depth
M06T01S20F50Z%1R3.5      ; Define the parameters for tool #1; note the Z-
                         ; code for the M15 depth is assigned to the
                         ; variable "%1"
G34X3.Y2.I1.5J1.0        ; Rectangular pocket cycle
(ELP)                    ; End of loop - goes back to the "FOR" statement an
                         ; runs the G34 again with a new value for %1
M30
```

The entire loop structure above may be combined with step and repeat codes if necessary to rout multiple pockets of the same size.

```
M07D01X.093
M25                      ; Start of outer loop, step & repeat
(FOR,%1=3.2,3.0,-.1)
M06T01S20F50Z%1R3.5
G34X3.Y2.I1.5J1.0
(ELP)
M24                      ; End of step & repeat pattern
M26X5.                   ; Repeat rectangular pocket at new location
M26Y4.                   ; Repeat rectangular pocket at new location
M26X-5.                  ; Repeat rectangular pocket at new location
M26
M27                      ; End of step & repeat offsets
M30
```

Program Coordinate Override (G67-G69)

Although valid at any point in a part program, the G67-G69 non-modal codes were specifically developed to assist in writing APP cycles without destroying any modal codes prior to programming of a cycle. They permit a "one-block" deviation from the current programming mode without having to restore the original mode.

Program Coordinate Override	
G67 ABSOLUTE BLOCK	This code functions similar to a G90 block in setting of absolute programming mode format. Any valid axis letter code programmed in this block is interpreted as absolute data, regardless of the modal status of G90/G91 prior to this block. Subsequent blocks return to the G90/G91 status that was in effect prior to this block's execution.

Program Coordinate Override	
G68 INCREMENTAL BLOCK	This code functions similar to a G91 block in setting of incremental programming mode format. Any valid axis letter code programmed in this block is interpreted as incremental data to be added to the last programmed position, regardless of the modal status of G90/G91 prior to this block. Subsequent blocks return to the G90/G91 status that was in effect prior to this block's execution.
G69 MACHINE ABSOLUTE BLOCK	This code has no equivalent modal code. This block permits programming axis coordinates to a specific location in the machine coordinate system, ignoring any coordinate system offsets that may be in effect. Any valid axis letter code programmed in this block are interpreted as absolute data from machine zero, regardless of the modal status of G90/G91, or any zero offsets prior to this block. Subsequent blocks return to the G90/G91 status, and program coordinate system that was in effect prior to this block's execution.

Axis Mirror Codes (M931, M941, M932, M942)

The M931/M941, M932/M942 codes affect the interpretation of the letter codes X and Y in the part program. An axis which has its coordinates "mirrored" reverses the sign of the programmed value. That is, a positive value becomes negative and a negative becomes positive.

An M931 code mirrors all subsequent X-axis commands, and M932 mirrors the Y-axis commands. M941 and M942 respectively cancel the mirror commands.

Note: When axes are mirrored, the interpretation of clockwise/counter-clockwise for circular moves and right/left for cutter compensation may become reversed. If only one axis is mirrored, then the circle sense and CRC direction is reversed. If neither or both axes are mirrored, then these orientations do not change.

Feedrate Format Codes (G93, G94)

Standard programming feedrates are in inches per minute, or inches per millimeter depending on the status of inch/metric programming. This is G94 mode, and is active as the default method.

Alternately, program moves can be made in what is called 'inverse time mode' where the time for the program move is specified, not the feedrate. This is programmed with a G93 code. G93 is a modal code, and all subsequent feedrate commands are interpreted as inverse time until a G94 is programmed, or there is a CONTROL RESET.

In G93 mode, the feedrate is specified with an F-code in seconds:

```
G00 X1. Y1.   ; Position X and Y to 1.0"
G93 F1.5      ; Set inverse time feedrate of 1.5 seconds
G01 X2.5      ; Move X to a position of 2.5" (starting from 1.0"), and make
              ; this move in 1.5 seconds
```

Program Dwell (G04)

The G04 code causes a pause in program execution for a time specified in seconds (or spindle revolutions) by an X or F coordinate in the same block. The X and F codes are interchangeable. No operator interaction is required to continue program execution.

The format of the G04 command is:

```
G04X5.

This would cause a 5 second delay in program execution.
```

Fixture Offsets (E-Code)

Fixture offsets are provided as a way to create a program coordinate system offset from the base machine coordinate system. 50 fixture offsets are provided which are defined by the operator and activated in the part program by the E-code.

Fixture offsets apply to any of the interpolated axes.

The E-code is programmed as a value from 0-50, where E0 cancels any previously active fixture offset.

If the E-code is programmed on a block along with an X, Y, Z, or A move, the axes move to their programmed position with the new fixture offsets. If one or more axes are not programmed in the block with the E-code, the axis does not move, but its position register is updated to the current position in the new coordinate system.

Tool Offsets (D-Code)

The D-code specifies the current tool offset values to use. Tool offsets include a length offset and a radius offset. The length offset takes place immediately upon execution of the D-code, to update the position register of the Z-axis accordingly. The radius offset applies only when cutter radius compensation (CRC) is active.

The D-code may be any value from D0 - D200, where D0 is a zero offset for length and radius.

5.19 Advanced Programming Package (APP)

The Advanced Programming Package (APP) is a powerful and flexible extension to the standard programming language. This power and flexibility is derived from the following features:

- The capability for the OEM or end user to write custom cycles which are called by conventional G, M or T codes or by a special subroutine call command.
- Powerful looping and conditional testing capability.
- Access and control of global and local variables.
- Embedded arithmetic expressions.

In short, APP provides the user with the programming tools that are associated with a high level language.

Type I and Type II Data Blocks

APP allows use of Type I and Type II data blocks in a part program as defined in EIA Standard RS-447.

Data outside parentheses are Type I which is conventional data defined in EIA Standard RS-274.

Data inside parentheses are Type II which consists of a mnemonic (which defines an operation) usually followed by arguments. Many Type II commands are defined for use by the APP processor and these are described later in this section.

APP Part Program Rules

The following rules are in force when running part programs in APP mode:

- Some blocks can get very long and the programmer is reminded that blocks can be up to 3 lines long by using a semicolon (;) as a continuation character in the first (and second) line. An MDI block cannot contain a semicolon.
- All numeric data are read in floating point format.
- Only one APP command is allowed per block.
- Blocks containing APP commands may not contain any letter codes except for N.

Use of Variables and Expressions

The numeric value for any Type I letter code or Type II argument can be defined by a constant, a single variable, a subscripted variable or an expression. Constants, variables and expressions are processed in floating point format with seven significant digits. Expressions can consist of variables, functions and constants all related by operators. Data is rounded only when required by a part program letter code, a PMI variable, a function or a (FOR) loop construct.

```
Type I (Digits beginning with # are variables)

Constants    :    G00 X1.2 Z-7.689 F.015
Variables    :    G00 X#3 Z#2
Expressions  :    G01 X#3+2.7 Z#10*1.3 F.010*#1
Mixed        :    G00 X1.2 Z#1*(#3-2)
```

Subscripted Variables

A subscripted variable is written as a variable identifier followed by an expression inside square brackets:

```
#3[%5+2];   %1[0];   %1[2]
```

The expression inside the brackets is first evaluated to an integer (whole) number. The result is used as an index which is added to the location of the programmed variable giving the location of the required value. The index value for the first element of a subscripted variable is 0 not 1. The index value can be negative.

To demonstrate, the expression #14[-4] accesses global variable #10 and the expression #4[3] accesses the global variable #7. An attempt to access a variable beyond the legal range results in an APP error.

Variables

System Variables (:)

System Variables (:101 - :159)	
:101	probe trigger position, from machine zero, axis 0
:102	probe trigger position, from machine zero, axis 1
:103	probe trigger position, from machine zero, axis 2
:104	probe trigger position, from machine zero, axis 3
:105	probe trigger position, from machine zero, axis 4
:106	probe trigger position, from machine zero, axis 5
:111	probe trigger position, from program zero, axis 0
:112	probe trigger position, from program zero, axis 1
:113	probe trigger position, from program zero, axis 2
:114	probe trigger position, from program zero, axis 3
:115	probe trigger position, from program zero, axis 4
:116	probe trigger position, from program zero, axis 5
:120	if 1, then probe changed state during last move; otherwise 0
:121	for APP command (PRB,2), this flag =1 when probe signal changes state
:123	counts per inch, axis 1
:124	counts per inch, axis 2
:125	counts per inch, axis 3
:126	counts per inch, axis 4
:127	counts per inch, axis 5
:128	counts per inch, axis 6
:130	Y offset in pixels, camera fiducial check
:131	X offset in pixels, camera fiducial check
:150	table fiducial check is active
:151	X axis position for table fiducial check
:152	Y axis position for table fiducial check
:153	Z axis position for table fiducial check
:154	table fiducial check tolerance
:155	table fiducial offset, X axis
:156	table fiducial offset, Y axis

System Variables (:101 - :159)	
:157	calculated pixels per inch for camera offsets

:1000 - :1299 = fixture offset table values

These variables give the programmer direct read/write access to the data in the fixture offset table as follows:

Fixture Offset [offsnum] [axis] offsnum = id / 6 axis = id % 6

System Variables (:1000 - :1299)	
:1000	fixture offset 1, axis 1
:1001	fixture offset 1, axis 2
:1002	fixture offset 1, axis 3
:1003	fixture offset 1, axis 4
:1004	fixture offset 1, axis 5
:1005	fixture offset 1, axis 6
:1006	fixture offset 2, axis 1
:1007	fixture offset 2, axis 2
:1008	fixture offset 2, axis 3
:1009	fixture offset 2, axis 4
:1010	fixture offset 2, axis 5
:1011	fixture offset 2, axis 6
↓	↓
:1298	fixture offset 50, axis 5
:1299	fixture offset 50, axis 6

These variables give the programmer direct read/write access to the data in the tool offset table as follows:

Tool Offset [offsnum] [index] offsnum = id / 2 index = id % 2
0 = length
1 = diameter

System Variables (:2000 - :2399)	
:2000	tool offset 1, length
:2001	tool offset 1, diameter
:2002	tool offset 2, length
:2003	tool offset 2, diameter
↓	↓
:2398	tool offset 200, length
:2399	tool offset 200, diameter

Part Program Variables ($)

These variables are identified by a dollar sign ($) followed by a number referencing the variable. The following background variables are defined along with their Read-Only (R) or Read-Write (R/W) status:

Part Program Variables ($0 - $103)	
$0	undefined variable; (R)
$1[ ]	current programmed position for each axis (6 max); (R)
$9	active cutter radius compensation value; (R)
$10	X axis current position offset; (R)
$11	Y axis current position offset; (R)
$12	current modal G code: 0=G0; 1=G1; 2=G2; 3=G3; (R)
$21[ ]	current position offset for each axis (6 max); (R)
$30[ ]	*commanded RPM per axis to rotate in open loop mode {M320-M325} (R/W)
$40[ ]	*endpoint per axis when exiting open loop mode {M330-M335} (R/W)
$100	absolute/incremental mode: 0 = ABS; 1 = INCR; (R)
$101	plane selected: 0=XY; 1=YZ; 2=ZX; (R)
$102	English or Metric input mode: 0 = English; 1 = Metric; (R)
$103	current feedrate; (R)

** $30[ ] and $40[ ] are used with the M320-M325, and M330-M335 codes to request an axis to run in open loop mode. For example, to run what is normally a positioning rotary axis at some constant RPM.*

Variables $200-$220 are written to by various canned cycles to set certain values as defaults for later cycle calls:

Part Program Variables ($200-$220)	
$200	R-plane
$201	Z-plane
$202	(not used)
$203	G83 initial depth (canned cycles)
$204	G83 incremental depth (canned cycles)
$205	(not used)
$206	M06 seen – used for SKIP routines
$207	M07 seen – used for SKIP routines
$208-220	spare

Part Program Variables ($221)	
$221	used in router systems that have M15/M16 head down/up control to indicate when to activate or de-activate CRC

$0 is a system variable whose value is always undefined. Its use is illustrated by examples in a later section of this document.

$1[ ] is an array of variables containing the current programmed position of each axis in the system. Since there are a maximum of six programmed axes in the SERIES 8, the array contains six elements which would be accessed by subscripts 0, 1, 2, 3, 4, and 5. Of course, the number of axes in a specific system varies.

$101 applies only to systems with the plane select feature.

PMI Variables (@)

These variables are identified with a @ followed by a number between 1 and 99. Data may be passed between the part program and PMI by using special PMI general purpose interface variables that have been set aside for that purpose. There are 99 read-write variables numbered @1 to @99 and 99 read-only variables numbered @101 to @199.

The PMI program uses inputs @PIN[1] to @PIN[99] (which are the same as @1 to @99) and outputs @POUT[1] to @POUT[99] (which are the same as @101 to @199).

PMI variables are created by the machine builder and built into the machine interface program for special functions unique to a specific machine.

Assume the machine has an interchangeable fixture plate. A set of three switches on the machine table provides a binary code to identify the fixture (0-7). Within the part program, a check needs to be made to see which fixture is mounted on the machine, and by this determine which part of the program to execute.

In PMI, the following statements would be needed:

```
$INPUT
FIXTUR       = 0/0-2/B   ? HSSB address '0', bits 0-3 define the fixture
inputs
@POUT[1]     = FIXTUR   ? assign PMI-APP output variable 1 to the value of
FIXTUR
```

In the part program, the program blocks check the status of the PMI-APP variable to determine which blocks to execute:

```
(IFT, @101 = 1)              ; fixture #1 is present ("@101" is the PMI
variable @POUT[1])
 .
 .                           ; blocks for program #1 (skipped if @101 does
not equal 1)
 .
(EIF)                        ; end of IFT section
(IFT, @101 = 2)              ; fixture #2 is present
 .
 .                           ; blocks for program #2 (skipped if @101 does
                             ; not equal 2)
 .
```

```
(EIF)                          ; end of IFT section
```

PMI INPUT VARIABLES (@PIN)	
@1	programmed tool number (1-4) (M06)
@2	programmed UP position (M06)
@3	programmed DOWN position (M06)
@4	programmed INFEED (M06)
@5	programmed spindle speed (M06)
@6	set in START.APP to indicate program has started
@7	X position error from fiducial check
@8	Y position error from fiducial check
@9	calculated rotation angle from fiducials
@10	table fiducial location, X axis
@11	table fiducial location, Y axis
@12	fiducial code
@13-99	(spare)

PMI OUTPUT VARIABLES (@POUT)	
@101	UP position for current tool
@102	DOWN position for current tool
@103	INFEED for current tool
@104	spindle speed for current tool
@105	programmed D-code error
@106	CRC correction, tool 1
@107	CRC correction, tool 2
@108	CRC correction, tool 3
@109	CRC correction, tool 4
@110	X axis front park position
@111	Y axis front park position
@112	programmed S-code
@113	MDI block in progress
@114	part program select switch #1
@115	part program select switch #2
@116	current tool offset number
@117	skip check of part program verification
@118	bar code fixture check required
@119	(not used)
@120	bar code read shift position (Y offset from park)
@121	Z height for camera focus
@122	X axis camera offset

PMI OUTPUT VARIABLES (@POUT)	
@123	Y axis camera offset
@124	X axis simulated camera reading
@125	Y axis simulated camera reading
@126	use simulated camera readings
@127	current coordinate version active
@128	draw prog active, skip fiducial checks
@129	current Z absolute position
@130	fixture bar code error
@131-199	(spare)

Global Variables (#)

Global variables are identified by a pound sign (#) followed by a number between 1 and 200. They are Read-Write variables which are accessible to any program, subroutine or cycle. A global variable can be accessed as a single variable or as an array. The advantage of an array is that its elements can be accessed by subscript from within a loop. The total number of global variables used by a program may not exceed 200.

Global variables also retain their value after the end of a part program, and therefore can be used from one program to the next. Global variables are set to zero whenever software is first loaded in the control.

For the Precision PCB router, certain global variables are reserved for system APP routines, and are listed in the table below:

Reserved Global Variables	
#1-#25	Used by camera fiducial routines
#95	CRC saved status for head up/down moves
#96	Indicator that inch/metric conversion variables are set
#97	Z axis counts per millimeter
#98	XY axis counts per millimeter
#99	Position conversion for inch/metric
#100	Feedrate conversion for inch/metric
#240	Saved M07 D-code for search
#241	Saved M07 R-code for search
#242	Saved M07 X-code for search
#243	Saved M06 T-code for search
#244	Saved M06 R-code for search
#245	Saved M06 Z-code for search
#246	Saved M06 F-code for search
#247	Saved M06 S-code for search

Local Variables (%)

Local variables are identified by a percent sign (%) followed by a number. They are Read-Write variables that are accessible from the main program and subroutines or cycles. Local variables are temporary memory locations that are automatically allocated on an "as needed" basis when a subroutine or cycle is called, and are de-allocated when part program execution returns to the calling program. Local variables used by a main program are maintained until an End of Program command or another program is selected. Data from local variables that have been de-allocated (by an (EDF) command) are lost and can no longer be accessed. The allocation and de-allocation process is done for each level of subroutine nesting.

```
Assume a part program calls subroutine SUB1 and that SUB1 in turn calls
SUB2; also assume that SUB1 requires 3 local variables and SUB2 requires 5.

The 3 local variables for SUB1 are referenced as %1, %2, %3.

The 5 local variables for SUB2 are referenced as %1, %2, %3, %4, %5.

Note: If %1 for SUB1 is not the same variable as %1 for SUB2.

When SUB1 calls SUB2 a total of 8 local variables are in use.  If this
total exceeds 50 the message "local variable table overflow" is displayed
and part program processing stops.
```

On entering a subroutine or cycle, all local variables for that level are undefined. Local variables are first allocated as dummy variables for subroutine and cycle arguments. Other local variables become defined as they are assigned a value in the procedure.

Subscripted Global and Local Variables

Global and local variables may also be accessed with a subscript (index). This is what happens when a subscripted variable is accessed. The index, which may be an expression, is evaluated to a whole number. This number is then added to the specified variable number. The result is the new variable number to be accessed.

```
The following expressions all access global variable #10 :

  #9[1]
  #10[0]
  #12[-2]
  #1[9]
  #6[4]
  #10
```

```
If six global variables, #9 through #14, are to be increased by some amount
stored in local variable %2 then instead of writing these six expressions:

  (AV,#9 = #9 + %2)
```

```
   (AV,#10 = #10 + %2)
   (AV,#11 = #11 + %2)
   (AV,#12 = #12 + %2)
   (AV,#13 = #13 + %2)
   (AV,#14 = #14 + %2)

the following (FOR) loop could be used instead:

   (FOR,%1=0,5)
   (AV,#9[%1] = #9[%1] + %2)
   (ELP)
```

APP does not require subscripted variables to be declared before use. It only checks that the effective variable number is within legal range. It is the programmer's responsibility to prevent a subscripted variable from overlapping one that is already being used. This could happen, for example, if the index were miscalculated.

Variable Display Page

The variable display page is accessed from the PROGRAM MENU, then selecting APP Variables, then selecting the variable type to view.

On the variable display pages the variable identifier is displayed first, followed by its value. The value is displayed as seven significant digits plus a decimal point and an optional minus sign. There is always at least one digit displayed before the decimal point.

Global variables are displayed as follows:

```
                              GLOBAL VARIABLES (#)

1  23.78934  2 -1.745723  3  0.016278  4  0.000000  5  1.000000 6  0.000000
7  1.000000  8  440.0000  etc....
```

Local variables are displayed in groups according to the program currently executing. The level number, device and program name are displayed first, followed by the variable number and value in use at that level.

Values that are too large, too small or undefined are displayed as follows:

Value Range	Display
x > 9,999,999	+>>>>>>>>
0 < x < 0.0000001	+<<<<<<<<
x < -9,999,999	->>>>>>>>
-0.0000001 < x < 0	-<<<<<<<<
undefined	********

Global variables have an option of being displayed as READ-ONLY, or as READ-WRITE. In READ-WRITE mode, the operator has the ability to enter new values for the global variables.

APP Operators, Functions, and Expressions

Variables and letter codes may be assigned values which are computed from an expression. Expressions may consist of operators, functions and constants.

APP Operators

Operators can be arithmetic, relational or logical. The following operators may be used in an expression:

Arithmetic Operation	Symbol
Exponentiation	**
Addition	+
Subtraction	-
Multiplication	*
Division	/
Negation (unary minus)	-

Relational Operation	Symbol
Equal	=
Not Equal	/=
Greater than	>
Greater than or equal	>=
Less than	<
Less than or equal	<=

Logical Operation	Symbol
And	&
Or	!

APP Functions

The following functions may be used in any expression:

APP Function		
Function	Mnemonic	Range of x
Absolute Value	ABS	
Integer (truncate)	INT	
Round (nearest integer)	RND	
Sine	SIN	
Cosine	COS	

APP Function		
Function	Mnemonic	Range of x
Tangent	TAN	
Arc Sine	ASN	-1 <= x <= 1
Arc Tangent	ATN	
Log (natural)	LOG	x > 0
ex	EXP	
Square Root	SQR	x >= 0

In order for the control to distinguish between a letter code and the start of a function, the function must always be preceded by an open parenthesis.

Note: Trigonometrical functions SIN, COS and TAN take arguments in degrees. ASN and ATN return values in degrees. If the value of the argument passed to a function is not within the required range an error is reported. See section 4.20.11 for a list of APP error messages.

APP Expressions

Expressions are evaluated from left to right. Parentheses may be used to change the order of evaluation, with the lowest level being evaluated first.

```
1 - 3 - 5 + 2 = -5

but

(1 - 3) - (5 + 2) = -9
```

In the absence of parentheses the following hierarchy applies to expression evaluation:

Functions, Negation	Highest, performed first
**	↓
*,/	
+,-	
Relational Operations	
Logical Operations	Lower performed last

APP Type II Commands

The following table lists all the Type II commands available in APP mode. Refer to the notes for further operational details.

APP Type II Commands		
Command	Syntax	Notes

APP Type II Commands		
Command	Syntax	Notes
Assign Value	(AV,var=expr,var=expr,...)	A,C
Zero Globals	(ZGV)	A,C
Comment	(COM,text)	A
For Loop	(FOR,var=expr1,expr2,expr3)	A
While Loop	(WHL,relational expression)	A
End of Loop	(ELP)	A
If-Then	(IFT,relational expression)	A
Else	(ELS)	A
End if	(EIF)	A
Define Cycle	(DFC,name,arg,arg,...)	A,B
Define Subroutine	(DFS,name,number of args)	A
Call Subroutine	(CLS,name,arg,arg,...)	C
End of Definition	(EDF)	A
Go To	(GTO,Nxxxx)	A
Display Text	(DSP,row,column,"format string"[,arg,arg,...][,attribute])	
Wait for Input	(KEY,var)	
Write Data to File	(WRT,"format string"[,arg,arg,...])	

A) The block is not processed like a regular program block. It does not stop in single block mode, nor does CRC treat it as a non-motion block. It cannot be executed from MDI unless otherwise specified.
B) Arguments are letter codes.
C) This command can be executed from MDI.

Assign Value (AV)

The (AV) command sets a variable to a particular value. Multiple variables may be assigned in one command. The form of the (AV) command is:

(AV,var=expr,var=expr...)

Assign Value Parameters	
var	any Read-Write variable
expr	expression to be evaluated for var

```
(AV,#5 = 10.24, #7 = 5*#8)
(AV,%4 = %1*(%2 + %3))
```

The 'assign value' command is also an implied command if no APP command is given. That is, the format of "AV," can be omitted, simply setting a variable equal to a value:

```
(#5 = 10.24, #7 = 5*#8)
```

```
(%4 = %1*(%2 + %3))
```

An (AV) command may not assign an undefined value to a system variable.

Zero Global Variables (ZGV)

The (ZGV) command sets all global variables to zero. There are no parameters.

Comment Line (COM)

The (COM) command is used to insert comments into a part program. The text may consist of any alphanumeric characters.

(COM,START OF FINISHING PASS)

FOR Loop Command (FOR)

The (FOR) command has the form:

(FOR,var=expr1,expr2,expr3)

For Loop Parameters	
var	controlled variable
expr1	initial value of var
expr2	final value of var
expr3	incremental value which is added to var each time the loop is completed. If omitted, it defaults to +1

The expressions are evaluated as signed floating point numbers. If expr3 is programmed it must be non-zero. When the (FOR) command is executed, var is set to the value of expr1. Part program blocks are executed until the end loop command (ELP) is found. The (FOR) command is executed again and var is incremented by expr3. If expr3 is positive and the new value of var is less than or equal to expr2, or expr3 is negative and the new value of var is greater than or equal to expr2, execution continues with the first block after the (FOR) command. Otherwise, execution continues with the block after (ELP). If var is greater than expr2 (expr3 positive) or less than expr2 (expr3 negative) on the first pass through the loop, the loop is skipped and execution continues with the block after (ELP).

```
G00G95X-1.Z-1.              ; rapid to position X-1.0, Z-1.0
(FOR,#2=-.005,-.02,-.005)   ; set up loop parameters:
                            ;  starts at -.005, increments by -.005
                            ;  each pass until it reaches -.020.
G33Z-2.K.055                ; make threading pass to Z-2."
G00X-.5                     ; retract X-axis
Z-1.                        ; bring Z back to start
X-1.+#2                     ; move X in to new depth
(ELP)                       ; loop until complete
```

```
This program would make four threading passes along the Z-axis, each time
moving in .005" on the X-axis dimension (depth of thread).
```

WHILE Loop Command (WHL)

The (WHL) command has the form:

(WHL,expression)

While Loop Parameters	
expression	an expression which usually contains at least one relational operator

The loop is terminated by an (ELP) command. The loop repeats itself as long as expression is true. When expression becomes false, program execution continues with the block after (ELP).

```
Roughing pass that removes .1" each pass until X is <= 5"

  G90 G00 X7.05 Z5
  (WHL,$1[0]>5.05)      ; $1[0] is the current X coordinate
  G91 G00 X-.15
  G01 Z-3 F50
  G00 X.05
  Z3
  (ELP)
```

Loop Nesting

Loop nesting is allowed to a level of five; that is, there may be up to four loops within a loop. These loops may be any mix of (FOR) or (WHL) loops. Each loop must have its own (ELP). Loop nesting is independent of subroutine and cycle nesting.

If-Then-Else (IFT)

The (IFT) command allows conditional execution of a section of part program blocks. When used in conjunction with the (ELS) command, one of two alternate sections of part program blocks are executed. This is a standard if-then-else structure.

The (IFT) command has the form:

(IFT,expression)

If-Then-Else Loop Parameters	
expression	an expression which usually contains at least one relational operator

If the value of the expression is non-zero the result is true, else the result is false. The section of the part program controlled by the (IFT) command is always terminated with an (EIF) command and may or may not include an (ELS) command.

```
Without (ELS)

   (IFT,#5 = #8)
  N0100........
  N0110........
   (EIF)
  N0200........

When #5 is equal to #8, the sequence is:

  N0100, N0110, N0200.

When #5 is not equal to #8, the sequence is:

  N0200
```

```
With (ELS)

  N0050.......
   (IFT,#5 = #8)
  N0100.......
  N0105.......
   (ELS)
  N0120.......
   (EIF)
  N0200.......

When #5 is equal to #8, the sequence is:

  N0050, N0100, N0105, N0200

When #5 is not equal to #8, the sequence is:

  N0050, N0120, N0200
```

There is no limit to (IFT) nesting but any nested (IFT) structure must be within the same section (either all before an (ELS) or all after an (ELS)) of the outer (IFT) structure.

Combinations of Loops and (IFT) Commands

If an (IFT) command is within a loop, it's associated (ELS) and (EIF) commands must be within that loop. If a (FOR) or (WHL) command is within an (IFT) section of blocks, its associated (ELP) must be within that same section.

Permitted combinations:	Illegal combinations:
(IFT,.....) ..... (WHL,.....) ..... (ELP) ..... (ELS) ..... (FOR,.....) ..... (ELP) ..... (EIF) (IFT,.....) ..... (WHL,.....) ..... (ELS) ..... (ELP) ..... (EIF)	 (FOR,.....) ..... (IFT,.....) ..... (ELS) ..... (EIF) ..... (ELP) (FOR,.....) ..... (IFT,.....) ..... (ELP) ..... (EIF)

Subroutines and Cycles

Subroutines and cycles are collectively referred to as procedures. A subroutine is an independent part program module that is called by a command from the executing part program. The call may contain arguments that are input to or output from the subroutine. A cycle is an independent part program module that is called by an M or G code from the executing part program. Any M or G code not used by the executive may call a cycle, though there is nothing to prevent an APP cycle replacing an executive M or G code. APP G cycles are modal; that is, every succeeding block is treated as the G code cycle until the mode is canceled by programming a non-APP G code. M code cycles are not modal.

Arguments provide communication between a calling program and a subroutine or cycle. In addition, subroutines can pass data back to the calling program in arguments. By using arguments, a procedure can be written in a completely general fashion that does not specify the

use of any particular constants or variables except its own local variables. Inputs to and outputs from the procedure are defined only when it is called and can be different for each call.

NOTE: A cycle may not call another cycle. That is, all G, M, and T-codes within a canned cycle must be executive type codes, and cannot be another APP cycle. Subroutines, however, may call other subroutines or they may call an APP cycle.

Storage Location for Subroutines and Cycles

Subroutines and cycles are stored in folders on the PC hard disk, and then automatically copied to the MC840 board when the SERIES 8 software is first started. There are two different folders where these APP routines are stored: one for user-defined cycles, and one for OEM-defined cycles. OEM cycles are supplied with the control by the machine builder, and should not be changed; user cycles are created by the end user as needed.

When the SERIES 8 copies the APP cycles to the MC840 disk, it copies the OEM cycles first, then the user cycles. In the event that a user cycle is defined with the same name as an OEM cycle, the user cycle overwrites the OEM cycle.

The pathnames for the folders for user and OEM cycles are defined in the SERIES8.INI file. The typical locations for these are as follows:

USER APP FILES: C:\Documents and Settings\Series 8 CNC\APP\

OEM APP FILES: C:\Program Files\SIEB-MEYER\Series8\APP\

As mentioned above, files in these folders are copied to the MC840 board when software is first loaded. If files are added or edited after the SERIES 8 software is loaded, the command "UPDATE APP FILES" in the PROGRAM menu is used to copy the current contents of the APP folders to the MC840 board.

In addition, it is possible to create subroutines (but not M or G code cycles) by appending them to the end of the part program. After the M30 block indicating the end of the part program, the 'define subroutine' code (DFS) code can be programmed followed by the subroutine blocks and ending with (EDF). Multiple subroutines can be appended to the end of the part program file. These subroutines get copied to the MC840 board at the time the part program is selected.

Procedure Structure

The first block of a procedure must be a Type II definition command.

For subroutines:

(DFS,name,n)

Subroutine Parameters	
name	name of the subroutine, may be 8 characters or less

Subroutine Parameters	
n	number of arguments for that subroutine even if the number is zero

For cycles:

(DFC,name[,arg1,arg2,...])

Cycle Parameters	
name	name of the cycle which must be Gxx, Mxx (xx is the G or M code which calls the cycle), or T. G and M code cycle names must be three characters long
arg1, arg2, etc.	are optional letter code addresses that are passed as arguments for that cycle

The body of a procedure follows the definition command. The procedure must never contain an end of program code (M30,M02). An End of Definition (EDF) command is the last block of every procedure. If the file contains more than one procedure, the definition command for the next procedure follows the (EDF) block of the previous one.

```
Main program with one subroutine and one cycle.


  (DFS,SUB1,5)

  body of SUB1

  (EDF)
  (DFC,G85,X,Z,R)

  body of G85

  (EDF)

  main program

  M30
```

Procedure Naming

Each procedure is stored in a single file, and must be copied to the MC840 board or attached to the end of the main program before the main part program is selected. All APP procedure files must have a file name that is identical to that used in the (DFS) or (DFC) command for that procedure and in addition must have the extension ".APP".

For subroutines, any legal Windows filename may be used as the subroutine name, and the file name is that same name with an ".APP" extension:

Subroutine "drill" filename: DRILL.APP

For M and G code cycles, the name of the procedure is the name of the M or G code, for example:

G83 cycle filename: G83.APP
M06 cycle filename: M06.APP

To avoid possible conflict with other G or M codes used by the CNC software or the PMI program, it is required to pre-define in PMI those M or G codes used as APP cycles. The exception to this is the G80-G89 drilling cycles, which are automatically assumed by the CNC software to be APP cycles.

To define an M or G code to be an APP cycle, it is specified in PMI in the following manner:

```
$MCODE
@M06/A

$GCODE
@G52/A
@G53/A
```

The "A" parameter designates this code as an APP cycle.

It is also possible to create an APP cycle code that is already an existing code in the CNC software.

Creating a canned cycle for the "G01" code. By defining in PMI "@G01/A" and creating an APP procedure called G01.APP, when a G01 code is encountered in the part program, the control executes the cycle G01.APP instead of activating linear interpolation (the standard definition of G01). In order to achieve the original linear interpolation function, it is necessary to program a "G101".

This means, all standard CNC software G-codes can also be commanded as the original value, +100 (G102 is the same as G02, G141 is the same as G41, etc.). The root code (G02, G41, etc) can be redirected to run as a cycle instead of its standard function.

Subroutine Call

The subroutine call has the form:

(CLS,name[,arg,arg,...])

Subroutine Call Parameters	
name	name of the subroutine
arg's	the subroutine arguments, if any

The call must contain the same number of arguments as defined in the (DFS) command of the subroutine, otherwise one of the error messages "Too few arguments" or "Too many arguments" is displayed. The arguments must appear in the order required by the subroutine. The order should be documented in the subroutine using the (COM) command.
If a dummy argument is passed in the calling list, it is assigned the value of undefined within the subroutine.

Within the subroutine, the data passed as arguments are referenced as local variables. There is a one-to-one correspondence between the passed arguments and the local variables.

```
(DFS,SUB1,3)          ; define subroutine name, 3 arguments
(COM,X COORD)         ; comment: 1st argument is X coordinate
(COM,Z COORD)         ; comment: 2nd argument is Z coordinate
(COM,FEEDRATE)        ; comment: 3rd argument is feedrate
G00 G95 X%1 F%3       ; set rapid mode, IPR mode, rapid to X
                      ; coordinate passed in 1st argument
                      ; set feedrate to 3rd argument passed
G01Z%2                ; rapid to Z position passed in 2nd
                      ; argument
(AV,%4 = %3 - 0.2)    ; calculate a value for local variable %4
G00Z%4                ; move Z to calculated position
(EDF)                 ; end subroutine definition
.....
.....
(CLS,SUB1,2.5,-7.6,.010)
.....

Within the subroutine:

1st argument (%1) = 2.5
2nd argument (%2) = -7.6
3rd argument (%3) = .010
```

Note that %4 is a local variable but not an argument. The subroutine has only 3 arguments as defined by the (DFS) command. %4 is undefined until the (AV) command is executed. The subroutine would execute as follows:

```
G00 X2.5 F.010
G01Z-7.6
G00Z-7.8
```

In this example the arguments were simply constants, but they could also be system, global or local variables or expressions. For example:

```
(CLS, SUB1, $200, #5, %4)
```

Here, the third argument, %4, is a local variable at the calling level, not a local variable of the subroutine. The arguments may be input or output; however, an output may not be an expression.

Cycle Call

A cycle is called by simply programming the cycle G, M or T code with its required letter code argument values. Only M and G codes not used by the executive should be used as cycles although there is nothing to prevent an APP cycle from replacing an executive M or G code. M and G codes used by PMI should not be used as cycles.

The following G code cycle is equivalent to SUB1 in the previous section except that arguments are passed with letter codes.

```
(DFC,G50,X,Z,F)
(COM,X COORD)
(COM,Z COORD)
(COM,FEEDRATE)
G00 G95 X%1 F%3
G01 Z%2
(AV,%4 = %3 - 0.2)
G00 Z%4
(EDF)
.....
.....
G50 F.010 X2.5 Z-7.6
.....
```

For G code cycles, the arguments in the call are not required to be in any particular order as they are for subroutines. Argument and local variable relationship is defined by the order of the arguments in the (DFC) command.

```
1st argument (%1)  X = 2.5
2nd argument (%2)  Z = -7.6
3rd argument (%3)  F = .010
```

The execution sequence is the same as for SUB1. As with subroutines, the arguments may be constants, variables or expressions; however they may only be inputs, not outputs.

```
G50 X$1[0] Y#5 Z%4 F2.5*#6
```

Arguments for G code cycles may be omitted, resulting in its associated local variable being undefined. If that local variable is used to define a letter code, the letter code is ignored in that block.

```
G50 X2.5 F.15

%1  =  2.5 (X)
%2  =  undefined since Z is not used in the call
%3  =  .15 (F)
```

Since %2 is undefined, there is no Z motion. However, X moves to its programmed position.

It is important to note that cycle arguments are read in the standard format for the particular letter code. Thus, if X has a 2.4 format, X as an argument also has a 2.4 format.

Modal G Codes

APP G code cycles are modal. Once a G cycle has been executed by programming a G code, each successive block continues to execute the same cycle until a non-APP G code is programmed.

If a G85 cycle exists that performs a certain machining operation starting at the XZ coordinates passed in the block, the following blocks would repeat that sequence three times:

```
N010 G85 X1 Z4
N020 X2 Z5
N030 X3 Z6.5
N040 G1
N050 XZ
```

Blocks N020 and N030 are modal and repeat the G85 cycle. Because block N040 contains a non-APP G code the modal G cycle is canceled. Block N050 therefore moves the axes to (0,0).

Procedure Nesting

Nesting of procedures is allowed to a level of 5 - a main program may call PROC1 (procedure 1) which calls PROC2 which calls PROC3 which calls PROC4 which calls PROC5. The procedures may be any mix of subroutines and cycles. There is no limit to the number of procedure calls at the same level. Procedure nesting is independent of loop and IFT nesting.

Procedure Rules

The following rules must be observed when writing and calling procedures:

1. All subroutines and cycles must be written in APP format; numbers are read as floating point.
2. A cycle may call a subroutine. Cycles may, but should not, be called from procedures.
3. A G code cycle cannot call another G code cycle.
4. An APP cycle block cannot contain another G or M code that is also a cycle.

Unconditional Jump (GTO)

The (GTO) command allows the program to skip forward to a block that begins with the N-code specified in the command.

(GTO,Nexpression)

Unconditional Jump Parameters	
expression	an integer (whole) number between 0 and 9999

The block with sequence number Nexpression must come after the (GTO) command in the program. Skipping backwards is not allowed. Nexpression must be in the same program level as the (GTO). The target N-code must be at the beginning of the block and contain a positive

number between 0 and 9999. If the target sequence number cannot be found an END OF FILE error is reported. It is the programmer's responsibility to ensure that expression evaluates to a number that can be found in the program.

```
....
(GTO,N0250)
....
N0250
....
```

Part Program Input/Output Codes

The (DSP), (WRT) and (KEY) commands provide a means to create display messages to the screen, write data to a follow, and prompt the operator for an input.

Display Text (DSP)

The (DSP) command is used to display information on the APP display page. The APP display page may be called up by the operator from the PROGRAM pull-down menu. The form of the (DSP) command is:

(DSP,row,column,"format string"[,arg,arg,...][,attribute])

Unconditional Jump Parameters	
row	The row of the start of the text
column	The column of the start of the text
format string	A combination of text and format control
arg	An optional expression(s) evaluated and displayed
attribute	A combination of foreground and background colors. The attribute is optional, programmed as the last field of the command and has the following format: [foreground]/[background]

The foreground and background colors are modal - that is, they remain in effect for all subsequent DSP commands until changed.

Attribute Colors			
Identifier	Color	Identifier	Color
BL	Black	GR	Green
RE	Red	YE	Yellow
BU	Blue	CY	Cyan (pale blue)
MA	Magenta (purple)	WH	White

If either row or column is omitted the text is output starting at the current value.

```
If global variable #4 has a value of -3.
```

```
  (DSP,,,"Here it is %2", #4,WH/BU)

Prints in white on blue:

  Here it is -3
```

The format string must be enclosed in double quotes. It consists of ordinary characters, control codes and, if arguments follow format string, format specifications. There must be one format specification for each argument. Ordinary characters (which may be upper or lower case) and control codes are output to the APP page in the order of their appearance. A control code is used for cursor control and consists of a forward slash (\) followed by a character. The following control codes are available with the (DSP) command:

Display Control Codes	
\R	position cursor to beginning of this line
\N	position cursor to beginning of next line
\T	move cursor to next tab location
\C	blank APP display and position cursor to beginning of first line
\L	clear to end of line
\P	force the APP message display for the operator to see

```
(DSP,1,1,"\C\TLINE ONE\NLINE TWO\NLINE THREE",YE/BL)


Prints in yellow on black:

    LINE   ONE
LINE TWO
LINE THREE
```

Tab markers are set every 5 characters at columns 6, 11, 16, 21 etc.

If arguments follow the format string then the string must contain format specifications that describe how the arguments are to be displayed. A format specification has the following form:

%ndp_before[.ndp_after]

The first character must be the percent sign (%) - this has no connection with the symbol for local variable. ndp_before is the number of digits to be displayed before the decimal point including a minus sign if one is expected. This is followed by an optional decimal point and the number of decimal digits, ndp_after, to be displayed. If the number is to be displayed as an integer both the decimal point and ndp_after would be omitted.

```
Local variable #23 had a value of 25.78 and #7 was set to 1.672456.
```

```
  (DSP,1,1,"Angle = %2.5   Radius = %2.4", #23/2, #7)

Displays the following text at the beginning of line 1

  Angle =  12.890   Radius =  1.6724    (where      represents   the   space
character)

If the command was,

  (DSP,,,"Angle = %3   Radius = %1.6", #23/2)

The display would be

  Angle = __31   Radius = _1.672456  at the current cursor position
```

The first character of a displayed number always is a blank or a minus sign depending on whether the value is positive or negative. If the integer part of the number cannot fit the specified ndp_before, asterisks (*) are displayed for the entire field width. Because a number can be displayed with a maximum of 7 significant digits any format that requires less than 7 causes the number to be truncated. Digits to the right are lost.

Here is a sample of the format and display for the number -3.141592:

Format	Display
%1.6	-3.141592
%2.5	-_3.14159
%1.4	-3.1415
%1	-3
%2.	-_3.
%3.0	-__3.
%1.1	-3.1
%0.4	-3.141592 (value too large for format)

Although a specified format may not be big enough to contain the number (%0.4 in the TABLE above), APP always tries to display the value to seven significant digits to help the programmer spot mistakes.

Wait for Operator Input (KEY)

The (KEY) command is used to read operator input from the keyboard into specified variables.

The form of the (KEY) command is:

(KEY,var)

Unconditional Jump Parameters	
var	A single or subscripted variable that must be Read-Write

The subscript can be an expression. If only the return key is entered, the value of the variable is unchanged. The part programmer is responsible for checking the validity of the entered number. Using (KEY) in conjunction with (DSP) allows a prompt to be displayed.

```
  (DSP,10,1,"ENTER THE NUMBER OF HOLES: \P")
  (KEY, #3)

Displays the following at the beginning of line 10:

  ENTER THE NUMBER OF HOLES: _

The cursor is positioned one space after the semicolon and program
execution waits for a response to be entered from the keyboard.  The APP
Display page must be active prior to entering the response otherwise the
keys are ignored.  The "\P" in the DSP line forces the APP display page in
the system to alert the operator that a response is needed.  The value is
then entered by the operator followed by the return key and the value of
the digits is stored in global variable 3.  If no digits are entered the
variable #3 remains unchanged.  The programmer should check the entered
value for legal range before continuing the program.
```

Write Text to File (WRT)

The (WRT) command allows writing data from a part program to a file on the CNC hard disk. The filename and path for writing the data is set in the SERIES8.INI file with the "APPWRITEFILE" directive. The typical pathname used is:

C:\DOCUMENTS AND SETTINGS\SERIES 8 CNC\PROGRAMS\APPOUT.TXT

The form of the (WRT) command is:

(WRT,"format string"[,arg,arg,...])

Write Text to File Parameters	
format string	a combination of text and format control
arg	an optional expression(s) evaluated and displayed

If arguments follow the format string then the string must contain format specifications that describe how the arguments are to be displayed. A format specification has the following form:

%ndp_before[.ndp_after]

The first character must be the percent sign (%) - this has no connection whatever with the symbol for local variable. ndp_before is the number of digits to be displayed before the decimal point including a minus sign if one is expected. This is followed by an optional decimal point and the number of decimal digits, ndp_after, to be displayed. If the number is to be displayed as an integer both the decimal point and ndp_after would be omitted.

```
 Local variable #23 had a value of 25.78 and #7 was set to 1.672456.

  (WRT,"X axis position = %2.5   Y axis position = %2.4", $1, $2)

Outputs one line of information to the write file:

  X axis position = 4.890   Y axis position =  0.6724
```

The following special characters can be inserted into the text string area to better format the data written to file:

Display Control Codes	
\N	position cursor to beginning of next line
\T	move cursor to next tab location

```
M43.APP cycle, which writes the values of global variables #1-#50 to the
APP write file:

   (DFS,M43)
   (FOR, %1=0,49)                    ; FOR loop, from 0 to 49
   (AV,%2= %1 + 1)                   ; %2 is always +1 of loop
                        ; count
   (WRT,"%2 = %2.4\N",%2,#1[%1])     ; write %2 and global variable
   (ELP)                             ; end of loop
   (EDF)                             ; end of subroutine

The output file would look like this:

   1 = 1.2345
   2 = 4.8764
   3 = 0.8876
   .
   .
   .
   50 = 10.4653

The "\N" causes a new line to be generated for each new output of a global
variable.
```

Special Procedures: START.APP and END.APP

There are two special APP procedure calls built in to the SERIES 8 CNC. These procedures run at the start and end of every execution of a part program. They provide a unique capability to perform certain standard functions that are needed for every program, and save the need for the programmer to build these functions in every program.

For example, at the end of every program, it may be desired to turn off the spindle, turn off all coolants, and move the axes to a known position for part unloading.

When a part program is started with CYCLE START, the routine "START.APP" is automatically called; subsequently when the M30 end of program code is executed, the routine "END.APP" is automatically called.

The START.APP and END.APP procedures must exist in one of the APP directories (user or OEM), or else a fault occurs when the part program executes. The routines themselves may have a 'dummy' function.

The format of these procedures is slightly different than standard APP subroutines:

1. The START.APP and END.APP routines do not have an initial (DFS) block, and they must have the ending (EDF) block.
2. These routines cannot have parameters passed to them.
3. The END.APP must contain an M30 code for official designation of end of program. (Note that the programmer's code of M30 actually causes the routine END.APP to be executed, and then within the END.APP routine, another M30 must be executed for the true end of program).

The minimum requirement for these cycles to run as 'dummy' cycles is as follows:

START.APP

(EDF)

END.APP

M30
(EDF)

```
Typical usages of these routines may be as follows:

START.APP

  G00G69Z0         ; retract Z axis
  (EDF)

END.APP

  G00G69Z0         ; retract Z axis
  M05              ; stop spindle
  M09              ; coolant off
M30
  (EDF)
```

APP Error Messages

APP Error Messages	
Message	Possible Cause

APP Error Messages	
Message	Possible Cause
APP command error	Invalid APP command or command syntax.
	No comma where required after command name
Procedure loading error	Improper Procedure Name
	Procedure file or syntax error
Program file error	Procedure has no (EDF)
Improper structure	(IFT), (WHL), (FOR), (GTO) structure improper
Loop error	Loops nested too deeply
	(ELP) without a (FOR) or (WHL)
Illegal sequence number	Too many procedure nests
	More than five levels of procedure nesting attempted
Invalid procedure name	Too many characters or invalid character
Too many/few arguments	Different number of arguments than specified in (DFS)
Invalid procedure argument	Referenced argument not defined
Illegal cycle call	G cycle cannot call another G cycle
	(DFC) M, G or T does not match that of cycle call
Procedure definition error	(DFS) or (DFC) encountered without corresponding call
Invalid Variable	Specified variable does not exist
Variable is Read-Only	Attempt to change Read-Only variable
Too many local variables	
Illegal system variable value	
Expression Error	Invalid expression
Command not allowed in MDI	
Unbalanced parentheses	
Divide by zero error	
Invalid argument for LOG function	
Invalid argument for SQR function	
Row or column out of range	
DSP or PRN format error	

Chapter:

Part Program Digitizer

Setting up the Digitizer

Hardware Requirements

In order to use the digitizer feature, it is required to have the handheld pendant device. The handheld pendant connects to the high speed serial I/O bus of the SERIES 8, using I/O address X'24'.

The handheld pendant contains an ESTOP switch, handwheel (manual pulse generator), and six illuminated pushbuttons. This pendant comes with a screw-type military connector; when not using the digitizer feature, the handheld pendant may be removed from the system.

Parameter Setting

Note: In the parameter editor, under PCP-PMI, the digitizer option must be enabled (variable #11) in order for it to function.

In the program status menu, under DIGITIZER, the following parameters must be set:

Parameter Settings	
Z position for camera	This parameter should be set to the Z-axis position where the camera is properly focused on the work piece. When digitizer mode is active, a touchkey is available that permits jogging the Z directly to this position. The value for this can be entered directly as a number, or recorded using the parameter described next.
Enter 'ON' to set Z posn	This allows setting the current position to be the "Z position for camera". With the Z-axis located at the correct position for camera focus, set this parameter to ON, and the current position is be captured and saved as the "Z position for camera". The software automatically sets this parameter back to OFF.
Camera offset to spindle, X axis	The value entered here is the distance along the X-axis from the camera to the spindle. It is used in determining the shift in coordinate systems from the record position of the camera to the spindle position.
Camera offset to spindle, Y axis	The value entered here is the distance along the Y-axis from the camera to the spindle. It is used in determining the shift in coordinate systems from the record position of the camera to the spindle position.

If the machine is configured for fiducial recognition capability, then the XY camera offset to the spindle should be set by using the G86 command described in the FIDUCIAL section of this manual. The G86 command uses the table fiducial to accurately determine the XY camera offset.

Digitizer Operation

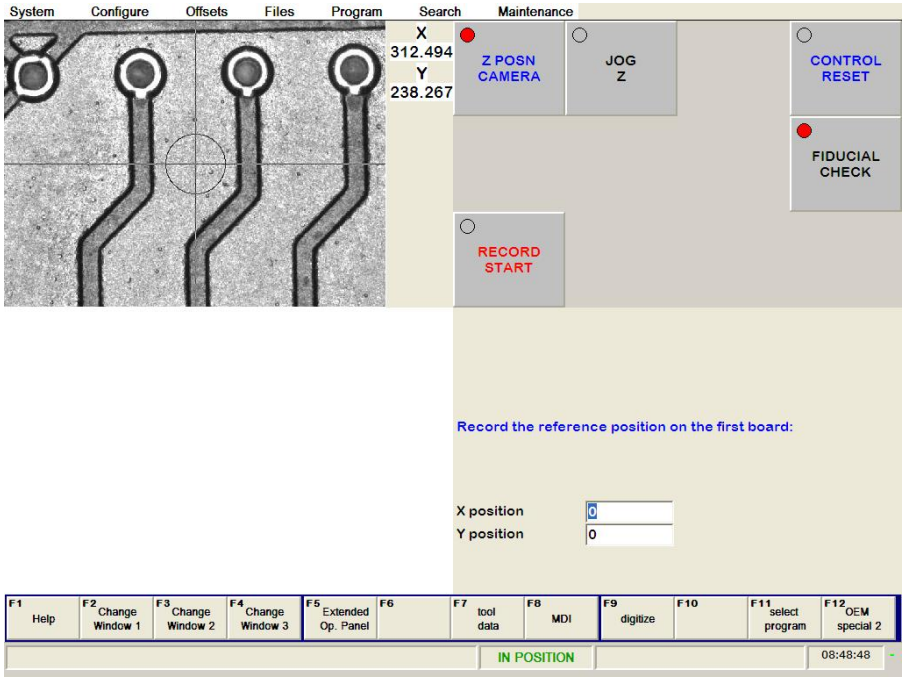
Starting the Digitizer

The digitizer program may be started with the digitizer command in the Program menu, or by pressing the F9 function key. Make sure that the handheld pendant is connected to the machine.

After executing the digitizer command, a program select window appears. The operator may select an existing filename, or enter a new file name.

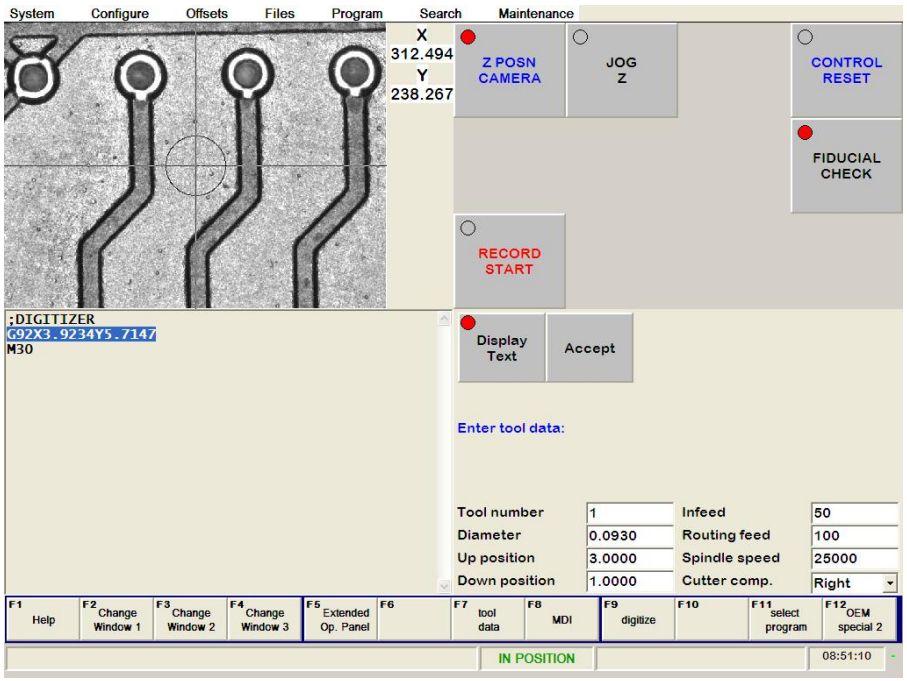
Note: Selecting an existing file name, the file must have been previously created with the SERIES 8 digitizer. It is not possible to edit a file with the digitizer that did not originate from the digitizer.

Starting a new file, the operator is prompted to set the program zero position as the first step.



Position the camera so that it is located at the reference location of the part to be digitized. Then enter the XY program coordinates for this location. In many cases, the XY coordinates for the reference position would be zero. Press the RECORD touchkey on the handheld pendant to set this reference position.

When the digitizer is active, the F3 window (lower right) provides a status indication and data entry boxes for the program. The F2 window (lower left) shows a graphic outline of the tool path that has been digitized, or it shows a text file of the program blocks that have been generated. The DISPLAY TEXT key allows changing the F2 window between a graphic or text display. The LED for this key is show in red if text display mode is active.

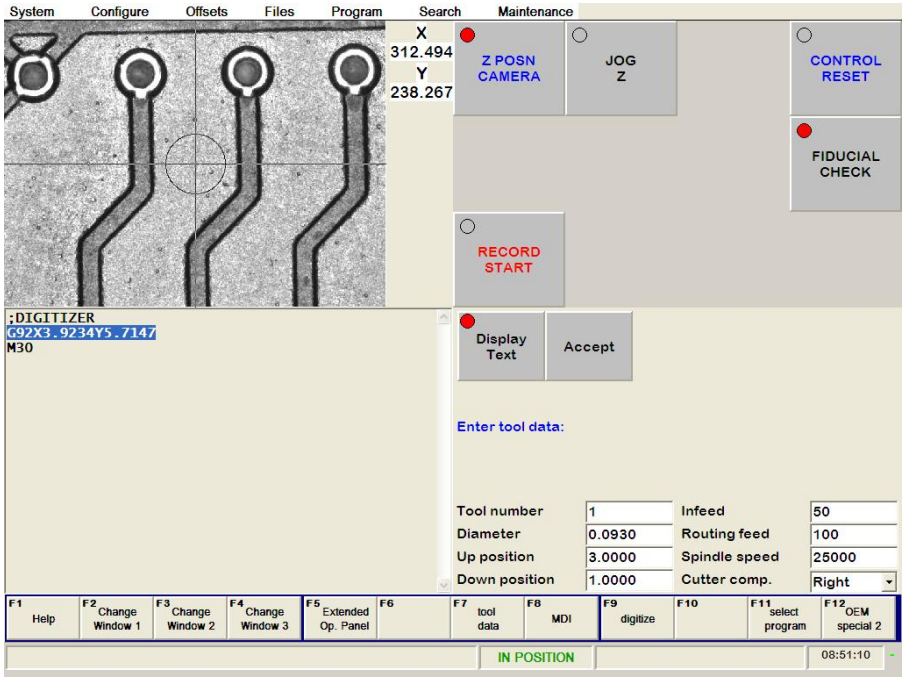


In the main operator touchkey window, there is a key labeled Z POSN CAMERA. Pressing this moves the Z-axis to the position defined in the program status DIGITIZE page, and is intended to be the correct location for the Z to have to camera focus properly on the part to be digitized.

If the camera image is not focused when is lowered to the Z camera position, press the JOG Z key, then jog the Z axis up/down until the image is in focus. Press the F7 'tool data' key, then select the 'digitizer' tab and set the parameter "Enter ON to set Z posn" to "ON" – this will teach the new Z position for digitizing.

Entering Tool Information

After the initial zero position has been set, or after completing a step and repeat pattern, the operator is prompted to enter in tool information:



Tool Information	
Tool Number	This is used to select the tool, and the tool offset number is the same as the tool number.
Diameter	This is the diameter of the tool, and is used if cutter compensation is selected.
UP Position	This is the Z position when a head up command is made (raised position for making moves without routing).
DOWN Position	This is the Z position when a head down command is made (the routing position).
Infeed	The feedrate at which the Z axis is lowered into the work piece.
Routing feed	Feedrate used when routing with the head down.
Spindle speed	Spindle speed in RPM.
Cutter comp	Enables cutter radius compensation, in the selected direction. Cutter comp may be set to OFF to disable it.

When finished setting the tool data, press ACCEPT to save this information, and to begin digitizing.

Using the Handheld Pendant

When the digitizer is active, the handheld pendant is used to position the axes to the positions that are to be recorded.

Handheld Pendant Button Definitions	
Switch 1 OFF/ON /TRACE	Activates the entire handheld pendant. When switched ON, the axes may be jogged with the handheld pendant. TRACE option is where the machine is positioned to the XY coordinates on the current block. In trace mode, the JOG +/- act moves the cursor on the program blocks. The handwheel is not active in TRACE mode.
Switch-2 LINE, ARC, DRILL	Selects the current mode of capturing data points. LINE and ARC are only valid to record if the head is down; DRILL is only valid to record if the head is up.
REC	Pressing this button records the current XY position to the program file. This may be for a line, arc, drill position, or step & repeat offset.
HEAD DOWN	Pressing this button toggles between head up and head down. The light illuminates if the head is currently in the down state. When the button is pressed to create the head down move, the current position is automatically recorded as the positioning move prior to the head down command.
Y	Axis select for jogging. When the button is not illuminated, the X-axis moves when a jogging move is requested. Pressing the button once selects the Y-axis for jogging, and the button illuminates; pressing it again goes back to the X-axis. Pressing and holding the button selects both X and Y at the same time (the button light blinks), and any jogging moves are made on both axes simultaneously.
HIGH	HIGH/LOW select - affects speed of jog keys or handwheel moves. The light is illuminated when in HIGH mode.
JOG -	JOG MINUS (continuous jog) - the selected axis moves at a constant speed in the minus direction as long as the button is pressed. The speed is high or low, depending on the rate selected. In TRACE mode, this key moves the program pointer back one line and move to the XY coordinates on that line (only if that program line has XY coordinates).
JOG +	JOG PLUS (continuous jog) - the selected axis moves at a constant speed in the plus direction as long as the button is pressed. The speed is high or low, depending on the rate selected. In TRACE mode, this key moves the program pointer forward one line and move to the XY coordinates on that line (only if that program line has XY coordinates).
HANDWHEEL	Anytime the JOG + or JOG - keys are not pressed and TRACE mode is not active, then the handwheel is active. Turning the handwheel clockwise moves the selected axis in the plus direction; counter-clockwise moves in the minus direction. The rate of move relative to the handwheel distance moved is based on the HIGH/LOW speed switch.

Creating a Program Pattern

After acceptance of the tool parameters, the control is ready to accept data points to build the part program. There are two basic modes to the type of moves to be made:

Program Patterns	
Routing	The machine positions XY at the rapid traverse feedrate to a location, lowers the Z-axis at the infeed rate to the rout depth, moves XY at the routing feedrate to make the cuts, then raises the Z axis to its clearance position. While routing, there are two types of moves possible: linear or circular (arc).
Drilling	Starting with the Z-axis in the UP position, the XY axis are moved at the rapid traverse feedrate to the programmed location, then the Z axis is moved down to the programmed depth at the infeed rate , then raised back to the clearance position at the rapid traverse rate.

The digitizer automatically creates the program codes necessary to start a 'pattern', which can later be set up in a 'step & repeat' format to repeat the programmed pattern multiple times.

Routing

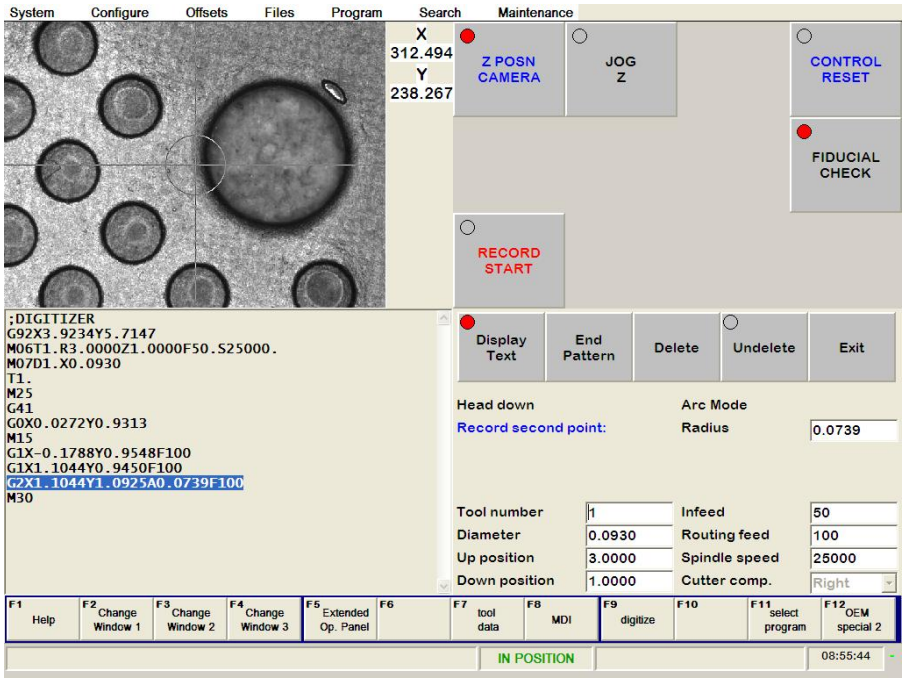
1) Set the move type selector switch to "LINE".
2) To define the start point of a routing pattern, position the camera to the location where the head is moved down into the part, and press the HEAD DOWN key. The RECORD key will not function until a head down is active.
3) Now the system is ready to record points for the routing moves:

   LINE: Move the X or Y axis to the next endpoint, and press RECORD

   ARC: An arc move requires recording of two points. The first point must be some intermediate point on the arc, and the second point is the endpoint of the arc move.
4) When the routing moves are complete, press the HEAD DOWN touchkey again to cancel the head down.
5) If another routing segment is needed in this pattern, repeat steps 2-4.

Special notes recording arc setting:

- recording an arc move requires two presses of the record key (one for an intermediate point on the arc, and one for the endpoint. The start point of the arc is the endpoint of the previous move
- if the intermediate point is recorded, and the system is waiting for the endpoint, switching the selector switch to "LINE" will cancel the arc setting sequence
- once the arc move is created by the control, the calculated arc radius will appear on the screen. The operator may enter a different value if the exact radius size is known – the start and end points of the arc move will not be changed.



Drilling

1) Set the move type selector switch to "DRILL".
2) Position XY to the location to be drilled, and press RECORD.
3) Continue positioning XY and pressing RECORD until all drill moves are recorded.

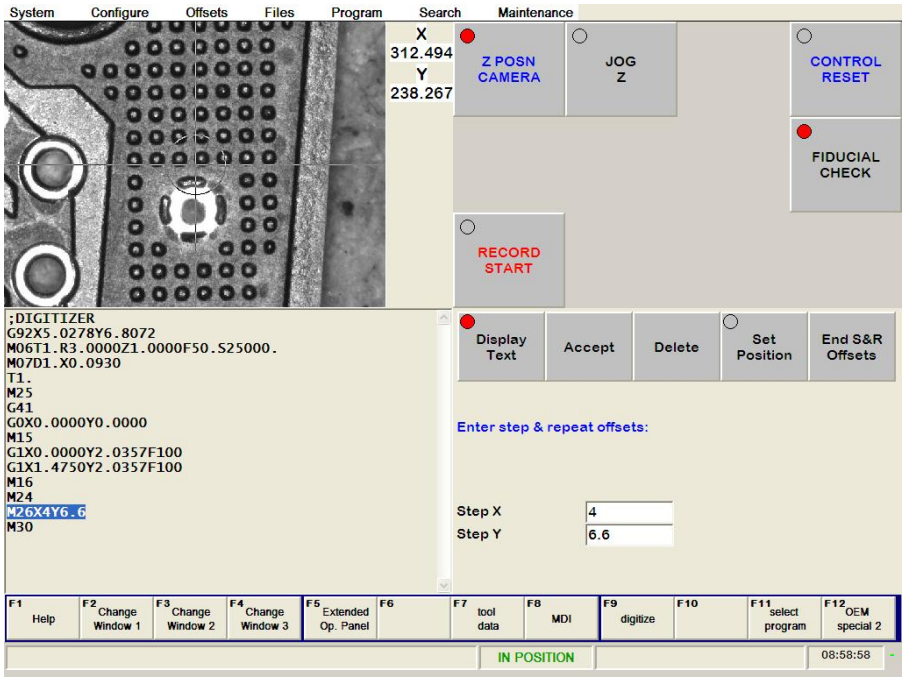
Note 1: When recording points, if neither the X nor Y axis has been moved since the last RECORD, then pressing RECORD has no effect.

Note 2: If a point is recorded in error, it is possible to move the program pointer back to that point (using the JOG+ and JOG- keys in TRACE mode), and pressing the DELETE key to remove that point. If a point has been deleted in error, it is possible to UNDELETE the last point that was deleted. After moving the program pointer or deleting a point, be sure and return the program pointer to the correct location to record the next point. (The JOG + and JOG - pendant pushbuttons move the program pointer, when TRACE mode is active) If in DISPLAY TEXT mode, points are recorded after the highlighted line in the program; if in graphic display mode, points are recorded after the move shown in red.

Note 3: It is not possible to delete the camera zero set position, tool change data, or step & repeat codes. If these are recorded in error, it is necessary to restart the entire program.

End of Pattern: Create Step & Repeat

When complete with recording points for a pattern, press the END PATTERN key. There are three keys shown that allow defining step & repeat of this pattern, and data entry windows to type in XY offsets:



If a step & repeat pattern is not necessary, simply press END S&R OFFSETS.

To create a stepped copy of the current pattern, the distance to move in X or Y from the start of the first pattern to the start of the second pattern must be entered in one of the two following ways:

1) Type in the step offsets directly in the data entry boxes
2) Move the XY axes to the same reference point on the second part as was recorded on the first part, and press the SET POSITION key. The coordinates of the step are automatically inserted in the XY step data entry boxes.

After defining the step offset, press the ACCEPT key to create the program codes needed to step and repeat the pattern.

Additional steps can be added using the same technique as described above to define the step amount. The step amount is always relative to the position of the last stepped pattern.

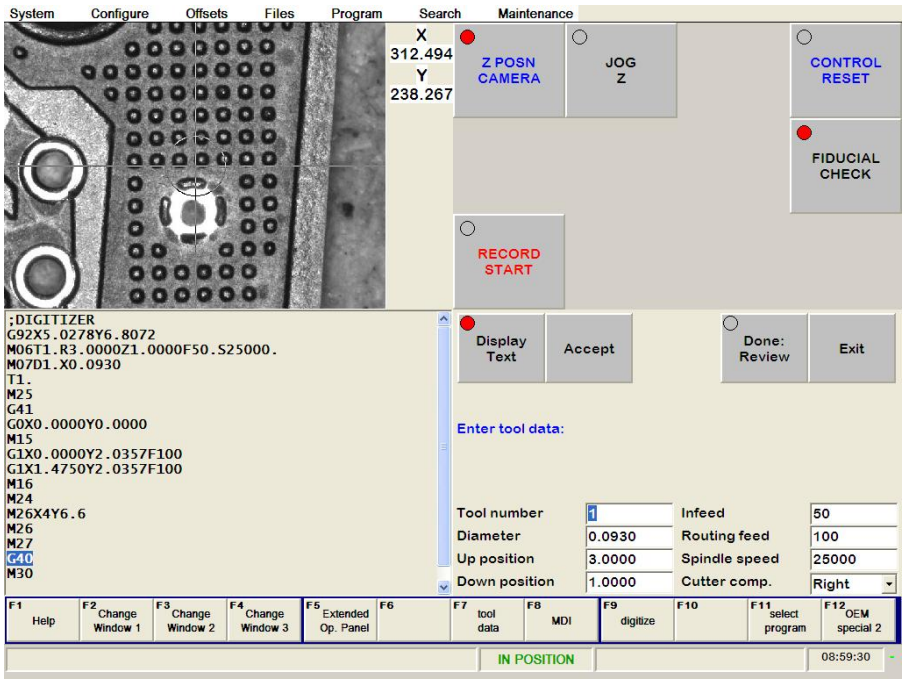
When finished with all the step and repeats, press END S&R OFFSETS to complete this operation.

Starting a Second Tool

After completing a pattern and defining the step & repeat offsets, the digitizer software prompts again for tool data. At this point a complete new pattern with a new tool can begin.

Ending the Program

Pressing the EXIT key exits the digitizer and store the current program. Alternately, the operator can press the "DONE: REVIEW" key which will allow TRACE MODE to be entered, allowing the operator to retrace the program moves for verification.



Trace Mode: Verifying Program Coordinates

Turning the left selector switch on the handheld pendant to TRACE activates the trace mode.

In this mode the JOG + and JOG - softkeys on the handheld pendant moves up and down through the part program blocks that have already been recorded and move the XY axes to the coordinates on that block. This function provides a way to verify if the captured positions are accurate.

Chapter:

Fiducial Recognition

The fiducial recognition feature gives the ability to shift the program coordinate system based on real time measurements of the panel to be routed.

Hardware Connection

The fiducial recognition feature utilizes a camera and display system. Connect the ethernet port of the display to the USB-Ethernet connection on the SERIES 8 PC.

SERIES 8 Ethernet Connection

The SERIES 8 PC must have its ethernet connection to the camera defined.

1) Exit the SERIES 8 CNC software.
2) Select the Windows START menu, then select CONTROL PANEL.
3) Double-click "network connections".
4) Right-click on "Local Area Connection"
5) In the center window, click to highlight "Internet Protocol (TCP/IP)"
6) Click the PROPERTIES button.
7) Select the option "Use the following IP address"
8) Enter in the IP address: 169.254.4.1
9) Enter in the Subnet mask: 255.255.255.0
10) Leave the 'default gateway' blank.
11) Click OK to exit, then click OK again, and then close the CONTROL PANEL window.

SERIES 8 Parameter Settings

The camera/fiducial check option is enabled in the Parameter Editor, in the PCP-PMI section.

PCP-PMI #22 – camera option for fiducial	set this =1 to enable camera
PCP-PMI #40 – camera IP address 1	enter 1 to 3 numeric digits
PCP-PMI #41 – camera IP address 2	enter 1 to 3 numeric digits
PCP-PMI #42 – camera IP address 3	enter 1 to 3 numeric digits
PCP-PMI #43 – camera IP address 4	enter 1 to 3 numeric digits
PCP-PMI #45 – number of part fiducials	enter value from 1-9

The camera IP address to be entered must match the IP address of the camera system. Code-in-motion should supply the value for this address; typically it is 169.254.4.246.

```
If the IP address of the camera system is:    169.254.4.246
Then, enter the following values into PCP-PMI:

  #40 = 169
  #41 = 254
  #42 = 4
  #43 = 246
```

The location and allowable tolerance of the table fiducial must be set in the PCP-APP section:

PCP APP #502	Swap XY data from camera
PCP APP #503	X axis table fiducial defines the X axis coordinate of the table fiducial, in machine coordinates.
PCP APP #504	Y axis table fiducial defines the Y axis coordinate of the table fiducial, in machine coordinates.
PCP APP #505	Table fiducial tolerance defines the tolerance allowed when verifying the table fiducial (typical value is .010").
PCP APP #506	½ X axis camera resolution in pixels (nominally 320)
PCP APP #507	½ Y axis camera resolution in pixels (nominally 240)
PCP APP #508	Invert the X axis camera reading
PCP APP #509	Invert the Y axis camera reading

Initial Connection Verification

With the camera enabled in the Parameter Editor, a new softkey will appear whenever the control is in ESTOP, labeled "CAMERA CONNECT".

When the SERIES 8 starts running and the camera connect is enabled, the software will attempt to connect to the camera automatically, and login to it. If the process fails, the following message will appear:

Asynchronous socket error xxxx

To avoid this fault (and disable the camera option without changing the Parameter Editor option), press the CAMERA CONNECT key to temporarily disable the camera connection. The fault described above will occur if the ethernet cable is unplugged, or if the IP addresses are not set correctly (check items 2 & 3 above).

Manual Camera Operations

Basic Operations

With the camera connected, three softkeys are provided for sending manual commands to the camera. They are accessed by pressing the F5 softkey (extended operator panel).

Manual Camera Operation Softkeys	
CAM LIVE IMAGE	Press this to put the camera into 'live image' mode. In this mode, the camera display updates as the machine table is moved.
CAMERA TEACH	Press this key to teach the camera the fiducial image
POSITION CAPTURE	Press this key to have the camera take an image and determine the XY offset of this image relative to the taught fiducial image. The XY coordinates of this capture are stored in APP variables :130 (X) and :131 (Y).

Enhanced Camera Functions (M93)

The actual commands between the CNC and the camera are broken down into single messages. For testing purposes, it is possible to send each message explicitly to the camera. To make these keys available, simply execute an M93 code via MDI. These keys can be turned off by pressing CONTROL RESET.

Camera Parameter Setting

Settings must be made to record the XY offset of the camera position to the spindle, and the Z height at which the camera should be for proper camera focus and position scaling. These settings are found with the STATUS PAGE command under the PROGRAM menu, under the TL DIST/DIGITIZER tab.

Camera Parameter Settings	
Z position for camera	Position for camera to be used when capturing fiducials; can be entered directly as a Z position, or may be set using the next parameter shown below
Enter 'on' to set Z posn	With the Z position at its correct height for proper camera focus (by manual jogging), selecting this option to 'on' will capture that position and store it as the 'Z position for camera'
Camera offset to spindle, X axis	Camera offsets to the spindle in X and Y must be properly measured and entered as a part of setup. They are generally set with the G86 command, and will be modified when a G88 cycle is run. To manually set the XY camera offset, do the following: - drill or locate a whole on the machine fixture that is in a known location - press the extended operator key "CAM LIVE IMAGE" - jog the Z-axis to the correct height for focusing - jog XY to that the camera cross-hairs are centered on the hole - subtract the current position of the XY axes from the hole position and enter these values as the camera offsets
Camera offset to spindle, Y axis	

Camera Operation in Part Program

In order to use the fiducial checking system, it is necessary to have a table fiducial that is set upon initial setup of the camera. Setting the table fiducial determines the camera offset from the spindle as well as scaling of the camera pixels to machine distance. Once the table fiducial is set, then part fiducials may be checked.

Table Fiducial (G86,G87,G88)

The table fiducial feature provides a way to set, and later verify accuracy of the camera position. The table fiducial is set initially (G86) to establish its location, and then is checked every cycle (G88) before performing a fiducial check on the part(s) to be routed. G87 may be used to cancel checking of the table fiducial.

To set the table fiducial position, the operator must enter the following block with MDI:

G86

Once the G86 code is executed, the camera display will be put into 'live' mode, and the jog keys enabled so that the operator can position the camera over the table fiducial. XY are used to center the fiducial, Z is used to focus the camera. If a handheld pendant is available on the machine, this device will also be active for axis jogging.

When the camera is properly set to the table fiducial, the operator must press the CYCLE START key to record the XYZ positions of the table fiducial. Additional moves will be made after pressing cycle start in the X axis which are used by the CNC to calculate and save the camera resolution relative to machine resolution.

The G86 command should be executed when the camera system is first installed to establish the table fiducial. It will also need to be used any time the camera position has moved relative to the spindle.

Once the G86 command is executed, all subsequent starts of a part program will check the table fiducial automatically prior to executing the program. If the check falls outside the tolerance allowable (default is .010"), then program execution will be halted.

Fiducial checking may be enabled/disabled by the operator by pressing the "FIDUCIAL CHECK" softkey.

A G87 command may be executed to completely cancel the G86 function, turning of fiducial checking altogether. To allow fiducial checking again, the operator will have to run the G86 sequence.

The G88 code performs the table fiducial check. This code is automatically run inside the START.APP routine for normal operation. However, the G88 code may be executed in MDI, or within the part program itself if a separate check is required. If the G88 cycle determines the table fiducial is within an allowable tolerance (default .010"), then it will pass the fiducial check, and it will also make adjustments to the XY camera offsets to re-calibrate the camera position. This corrects for slight errors in camera positioning due to machine vibrations. If the G88 routine determines the table fiducial is out of the allowable tolerance, then the operator will be required to re-run the G86 cycle to reset the camera offsets.

Zero Shift Based on Fiducials (G63)

In the part program, the G63 code is used to measure fiducials and create a program zero shift. Command format:

G63 X# Y# I# J# Z#

G63, Zero Shift Parameters	
X	X coordinate of first fiducial
Y	Y coordinate of first fiducial
I	X coordinate of second fiducial (optional)
J	Y coordinate of second fiducial (optional)
Z	camera height

If only one fiducial point is programmed (i.e. XY values, no IJ values), then the control will measure the one fiducial, and adjust the program zero based on the location of the fiducial.

If two fiducial locations are programmed, the control will measure both fiducials, and calculate a program zero shift based on an average of the two measurements.

At the end of the G63 cycle, the zero shift will be active based on the values seen by measuring the fiducial(s).

NOTE: The programmer should not place a G92 code after the G63 block, as this will cancel any zero correction made by the G63 cycle.

Prior to running the G63 block, the operator may program a G92 to establish the program zero position for X and Y. The G63 XY locations will be made relative to this G92 setting, and the corrections made by the G63 fiducial readings will adjust the values relative to the original G92.

Zero Shift and Rotation Based on Fiducials (G85)

In the part program, the G85 code is used to measure two fiducials and create both a program zero shift and part rotation. The two fiducials measured must be at the same X axis position on the part.

Command format:

G85 X# Y# I# Z#

G85, Zero Shift and Rotation	
X	X coordinate of first and second fiducial
Y	Y coordinate of first fiducial
I	X coordinate of second fiducial (optional)
Z	camera height

If only one fiducial point is programmed (i.e. XY values, no I value), then the control will simply measure the first fiducial point, and calculate an XY zero shift based on that reading (no part rotation). If the I code is programmed, a second camera reading will be made and a rotation angle calculated based on the variation in Y position seen between the two readings.

Programming Considerations with Step & Repeat

The fiducial measurement can be set up to be made once for every program cycle, or it can be set up to measure a fiducial for every part. How this occurs depends on the placement of the G63 code in relation to the M25 step and repeat code.

```
Example: Measuring one fiducial for the entire program

%
T01F200S20.D3.0U3.5
CP01 .0787
%
G92X2.24Y3.827
G63X1.205Y.2          ; G63 located before the M25
M25
T01
G42
X3Y0.5
M15
G01X0.8F70
M16
X3.076Y0.72
M15
Y2.
X2.986
Y0.72
M16
M24
M26X6.540Y-.017
M26X6.583Y-.01
M26
M27
M30
```

```
Example:  Measuring a fiducial for each pattern

%
T01F200S20.D3.0U3.5
CP01 .0787
%
G92X2.24Y3.827
M25
G63X1.205Y.2          ; G63 located after the M25
T01
G42
X3Y0.5
M15
G01X0.8F70
M16
X3.076Y0.72
M15
Y2.
```

```
X2.986
Y0.72
M16
M24
M26X6.540Y-.017
M26X6.583Y-.01
M26
M27
M30
```

Pattern Skip with Fiducial Check Active

If the fiducial check mode is active (G86), then it is necessary to inhibit a "skip pattern" request until the fiducials have been checked, otherwise the part location may be incorrect. If an attempt is made to skip patterns at the beginning of the part program with the fiducial check active, the operator will get the message:

"Command not allowed"

The correct procedure to skip in this case is as follows:

1. Press the operator key labeled SKIP W/ FID CHECK. This sets a mode where the part program will halt execution after running the G63 or G85 fiducial cycle.
2. Press CYCLE START to start the part program. The program will execute the table fiducial check, followed by either a G63 or G85, whichever is in the program, and then stop.
3. The operator may then use the SKIP PATTERN command in the SEARCH menu to skip the desired number of patterns.
4. Pressing CYCLE START will resume program execution from the point that was skipped to.

NOTE: If it is desired to measure a fiducial on every part (pattern), then the above search procedure is not required. It is allowable to skip patterns directly, because the fiducial will be checked on every pattern.

Part program review in NO ROUT mode

The camera used with the fiducial checking system can be used in a 'NO ROUT' mode to verify a part program without actually routing. This option is selected by pressing the JOG MODE key to turn on JOG MODE, then a key will appear labeled NO ROUT. Turn on NO ROUT to activate this feature.

When in NO ROUT mode, the fiducials will be checked as normal (assuming fiducial checking is turned on), but the head will not be moved for an M15 or M16 (the Z axis will instead be positioned at the camera height set in status management), and cutter radius compensation will not be enabled. The XY positioning of the spindle will be shifted for the camera offset, and the camera will be turned on to 'LIVE MODE'. The part program will execute at reduced feedrate.

The result of this will be that the crosshair of the camera will follow the finish path of the routing part program at a slow rate, allowing the operator to verify that the program is correct prior to running an actual cycle.

Technical Notes

M-codes for Camera Functions

The following M-codes are used by the camera canned cycles to perform various camera functions:

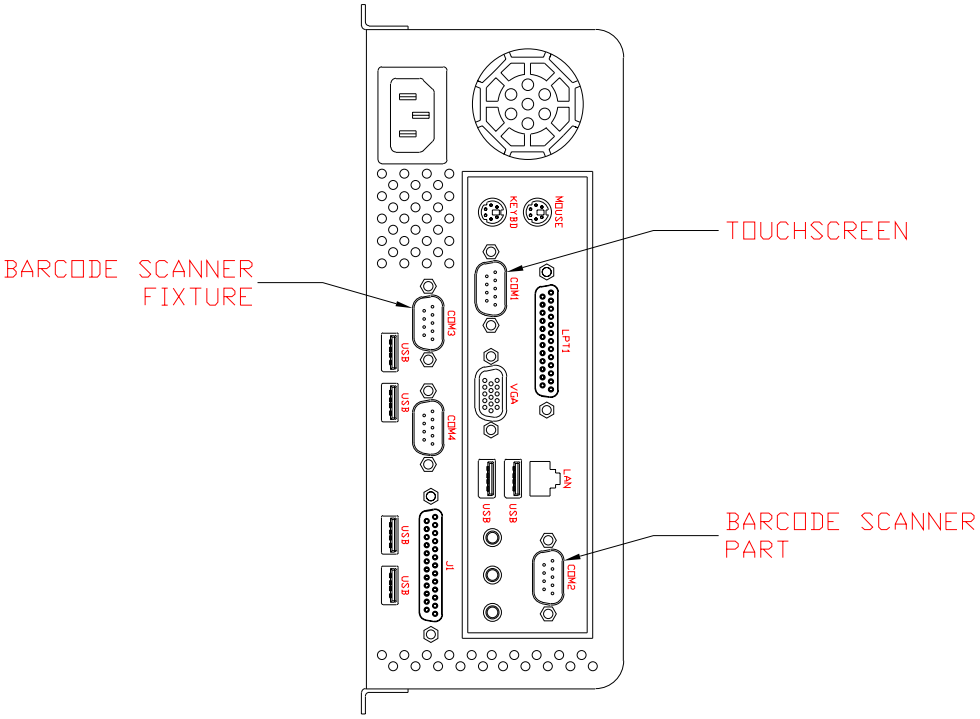
M - Codes for Camera Functions	
M150	take camera reading for XY offset, active fiducial (:130, :131)
M151	display camera offset readings
M152	cancel M151
M153	teach camera active fiducial
M154	read fiducial code number
M155	stop with camera fault, regular fiducial
M160	take camera reading for XY offset, table fiducial (:130, :131)
M161	put table fiducial on live image
M163	teach camera table fiducial
M164	tell PMI to save new camera offsets
M165	stop with table fiducial error
M166	stop with message to operator for table fiducial check
M167	go to MANUAL mode if fiducial skip is on
M168	set fiducial check mode in PMI (:150=1)
M169	reset fiducial check mode in PMI (:150-0)

Chapter:

Barcode Program Select and Fixture Verification

Hardware Connections

There are two barcode readers used – one for scanning the part, and the other for scanning the fixture. The reader for scanning the part must be connected to COM2 on the PC, and the reader for scanning the fixture must be connected to COM3. See drawing below for COM port locations.



Parameter Settings

The barcode reading is enabled in the Parameter Editor, in the PCP-PMI section.

Barcode Parameters	
PCP-PMI #23: barcode option	set this =1 to enable the barcodes
PCP-PMI #24: barcode prog de-select @M30	set this =1 to have the part program automatically de-selected after every run, forcing the operator to scan the part for every cycle
PCP-PMI #44 barcode port timeout	set this to a value between 10-5000 for the maximum number of milliseconds for the software to wait in reading the next character when scanning a barcode before cancelling the read – nominal value is 1000

Barcode Parameters	
PCP-PMI #46 barcode mask type	set this to a 0 or 1 depending on the desired handling of the barcode read – see the next Section for more details
PCP-PMI #47 disable fixture check	set this to a 1 to disable the fixture checking on systems where the barcode option is enabled

Barcode for Part Program Select

Two different types of reading the barcode are available, and set in the Parameter Editor, PMI 0-49 section with variable #46 – "barcode mask type". The resulting barcode value will be used as the name of the part program to select.

Mask Type 0 – Numbers Only

If the barcode mask type is set to "0", then the barcode that is scanned will ignore any leading characters that are non-numeric, and stop reading characters when a non-numeric is seen.

```
Barcode =  ABC12345XYZ                    Program name  = 12345

Barcode =  12345XYZ                       Program name  = 12345

Barcode =  ABC12345                       Program name  = 12345

Barcode =  ABC12345X72                    Program name  = 12345

Barcode =  4-ABC12345XYZ                  Program name  = 4
```

Mask Type 1 – Actual Barcode

If the barcode mask type is set to "1", then the barcode that is scanned will be used directly as the requested part program name.

NOTE1: barcode characters scanned must be valid for Windows file names

NOTE2: lower case letters are not accepted, and will result in a failed match for a program name.

```
Barcode =  ABC12345XYZ              Program name  = ABC12345XYZ

Barcode =  12345XYZ                 Program name  = 12345XYZ

Barcode =  12345abc                 Program name = NOT VALID
```

Setting the barcode readers

The part program select barcode reader is typically a handheld unit activated by a trigger, and the fixture barcode reader is a fixed mount unit that scans continuously. If a unit is not able to scan

correctly, the communication parameters may not be set correctly. Setting the parameters is generally done by scanning certain control codes. The control codes included here are the settings required to make the bar code readers communicate to the SERIES 8.

PART PROGRAM SELECT BAR CODE READER

Scan the following codes in sequence to set up the part program reader. The first code resets all parameters to their default; the following codes set the ones that need to be changed from their default.

Set All Defaults

***1 Stop Bit**

(01h)

Host: RTS High

(01h)

***Baud Rate 9600**
(06h)

***None**

(04h)

FIXTURE BAR CODE READER

The fixture reader is set up similar to the program select reader, however, when the code is scanned to set the defaults, the reader will no longer scan. Since this reader does not have a trigger, the only way to program the necessary settings (and to turn on continuous scan mode) is via software connection. The fixture scanner is also capable of the Symbol Technologies "SSI" interface, which permits the reader settings to be made via software.

If the scanner is on, then start by scanning the 'set all defaults' code shown below.

Set All Defaults

If the scanner is not on to start with, enter the code M96 via MDI from the CNC to enable continuous scanning mode, then scan the 'restore defaults' code shown above.

After setting the defaults, the scanner will turn off. Next, enter the code M95 via MDI from the CNC to set all the correct communications settings. The M95 code will take 10-20 seconds to complete. The last step of the sequence is to turn on the continuous scan, signaling that the parameter setting is complete.

Barcode for Fixture

The fixture barcode is read at the start of every program cycle, and must match the correct value for the selected part program, or the program will not be allowed to continue. The feature to check for the fixture barcode can be disabled with PCP-PMI variable #47.

Fixture barcode masking is handled exactly the same as the part program name described in the previous section.

By default, the fixture code is equal to the part program name, however, it can be overridden and set to a specific value in the program header:

```
%
FIX, 123456
T01 F300. S30. U3.2 D2.7
CP01 .066
%
G92 XY
...
```

Checking the fixture barcode may be enabled/disabled by the operator by pressing the "FIXTURE CHECK" key in the F5 extended operator key window.

Operation

Selecting the program:

With the barcode enabled, the system will default to having the program selected via the barcode scanner. On the touchkey display, after homing, there will be a key labeled "SEL PROG BARCODE". When this is active, the part program may only be selected via the barcode reader. In order to use the "select program" menu command, the SEL PROG BARCODE needs to be turned off.

Scanning the fixture:

The fixture is scanned automatically at the start of every program cycle. The fixture barcode is scanned when the machine table is at the "TABLE FRONT" position. The program will start by moving 3" away from the TABLE FRONT position, then will move to the TABLE FRONT to read the code. If this code matches what is required for the part program, then the program may continue; otherwise a fault message will display and the operator will need to reset the cycle.

Special test mode

The fixture barcode reader parameters may also be set using touchkeys from the SERIES 8. Executing an M94 code from MDI will activate the test mode. The F5 extended operator keys will then show keys to pass parameters to the reader. Pressing the EXIT key on this page will turn off the test mode.

BAR CODE SEND – sends a custom message to the bar code. The message must be a valid SSI code for SYMBOL readers. Prior to hitting this key, the following parameters must be set via PMI DEBUG:

BCOPC - op code
BCPAR - parameter code
BCVAL - parameter value

SET DEFAULTS – sets the bar code reader back to all its default settings

CONTIN SCAN – puts the reader into continuous scan mode

SET CODES – sets all the communication parameters (takes 10-15 seconds)

* one stop bit
* RTS high
* 9600 baud
* no parity
* send DATA <SUFFIX1> <SUFFIX2>

Bar code settings – additional information

The bar code parameters should be set according to the method described earlier in this chapter. This section provides the complete list of bar code settings that are used for the fixture scanner for reference only.

Continuous

***1 Stop Bit**

(01h)

Host: RTS High

(01h)

***Baud Rate 9600**
(06h)

***None**

(04h)

<DATA> <SUFFIX 1> <SUFFIX 2>
(03h)

Scan Suffix 2
(08h)

1

0

1

0

Scan Suffix 1
(06h)

1

0

1

3

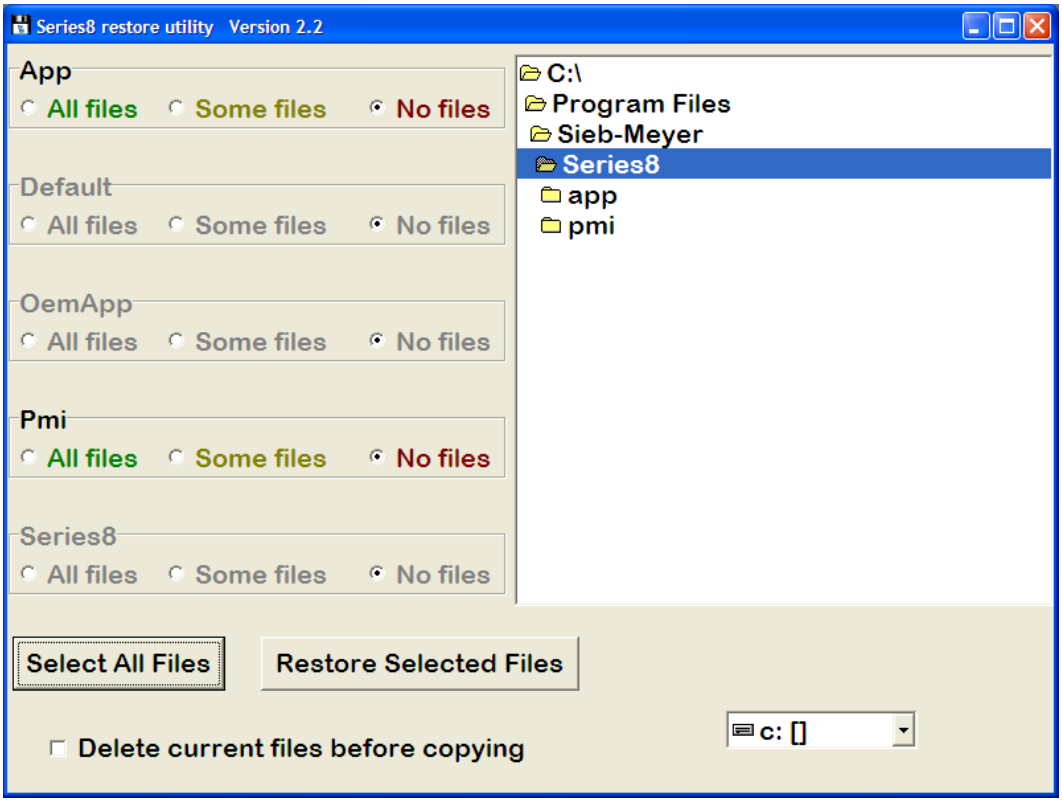
Chapter:

Software Utilities

Restore

The SERIES 8 Restore utility runs from the desktop of the SERIES 8 PC. Its function is to allow restoring of a previous system backup, or to install new software files. The restore software looks at the current SERIES8.INI file to determine where to copy the files to.

Note: This restore utility does NOT restore part program files.



Basic operation:

If installing files that were e-mailed in a ZIP file, this file must be unzipped prior to starting the restore program.

1. Exit the SERIES 8 software before running this program. From the SERIES 8 SYSTEM menu, select the option exit to windows.
2. If the files are to be loaded from an external source, such as a USB memory stick, insert this stick prior to starting the SERIES 8 RESTORE program. A primary folder must contain the sub-folders and files to be restored. The primary folder name is unique to the files being restored; the sub-folder names must follow the SERIES 8 folder names:

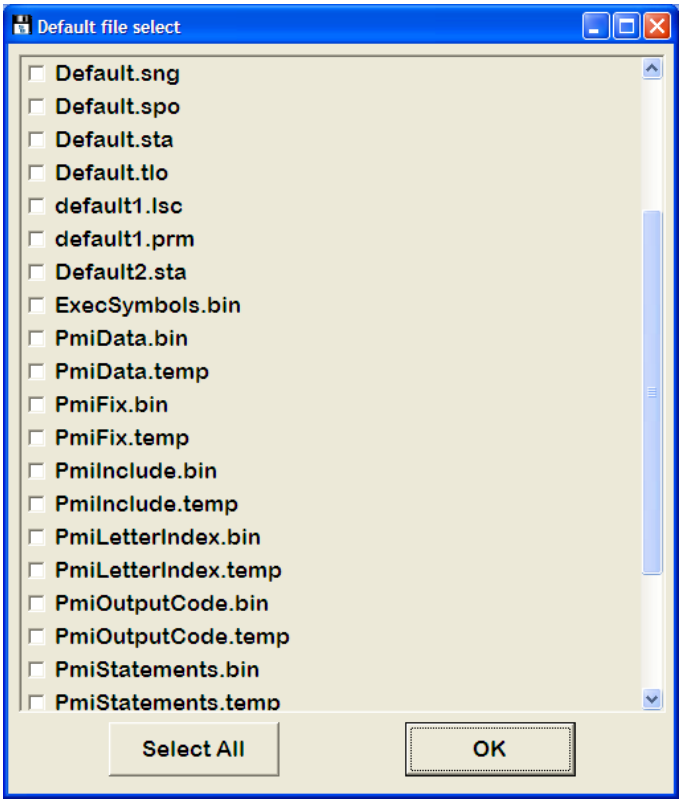
SERIES 8 Folder Names	
File Name	Description
APP	User APP files
OEMAPP	OEM supplied APP files
DEFAULT	Default files
PMI	PMI files
SERIES8	Executive files, library files, and the INI file

Not all sub-directories need to be present, but all files to be restored must be located in one or more of these directories.

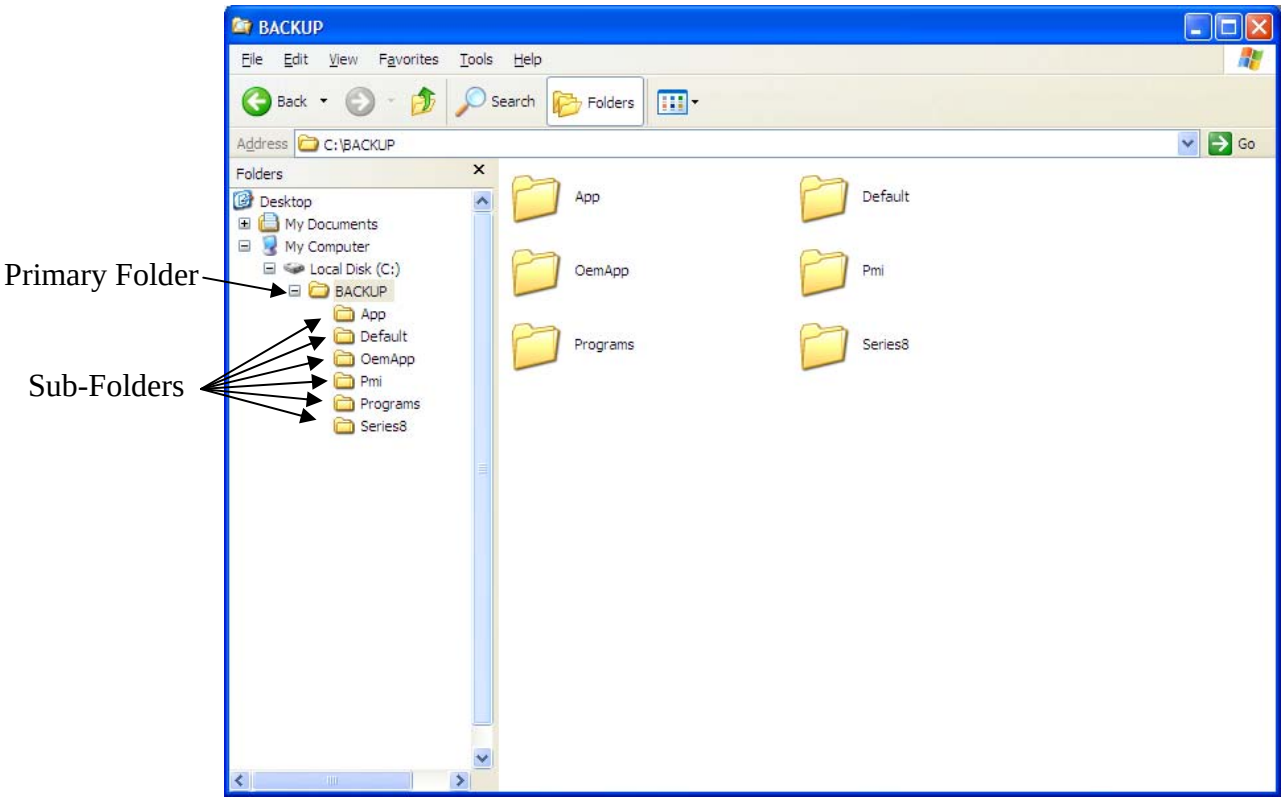
3. Double-click the desktop icon: SERIES 8 RESTORE.
4. Use the disk drive selection box in the lower right hand corner to select the disk drive where the restore files are located. Then select the correct directory path for the files. On the left side of the display, any sub-directories containing files that can be restored are shown in color.
5. If restoring a complete backup, press the key "Select All Files". Then press "Restore Selected Files" and the files are copied. Skip step 6 and continue with step 7.
6. If installing just some software files that were sent in a software update, press the "All Files" touchkey for each directory that is shown in color on the left side of the display. Then press the "Restore Selected Files" touchkey.
7. Exit the restore utility program and restart the complete CNC by powering down the PC and starting it again.

Application notes:

- This restore utility is designed to work in conjunction with the backup utility in the SERIES 8. Under the FILES menu of the SERIES 8, the backup command creates a directory on the C: drive called "backup" and contains all the subdirectories and files needed for restoring the system. To restore a complete system that was previously backed up, simply select the C:\backup directory, press "Select All Files", then "Restore Selected Files".
- This restore utility permits selective backup of individual files, without copying the entire directory. To do this, first select the device (e.g. C:), and the primary directory as described above. Then select the "Some Files" option for the directory (e.g. PMI directory). A listing of the files in the restore directory is shown. The files to be restored can be selected by checking the box next to the file name. Exit this window by pressing the OK softkey.



- When setting up a folder/sub-folders to be restored, be certain to structure the folders properly to insure the files are seen by the restore utility program. The listing below shows the typical locations of files in a SERIES 8 system, although actual directory locations are determined from the Series8.INI file:



Sub-Folder SERIES 8 Folder Locations	
File Name	Description
APP	C:\documents and settings\series 8 cnc\APP\
OEMAPP	C:\program files\sieb-meyer\series8\APP\
DEFAULT	C:\documents and settings\series 8 cnc\default\
PMI	C:\program files\sieb-meyer\series8\PMI\
SERIES8	C:\program files\sieb-meyer\series8\

Remote Diagnostics

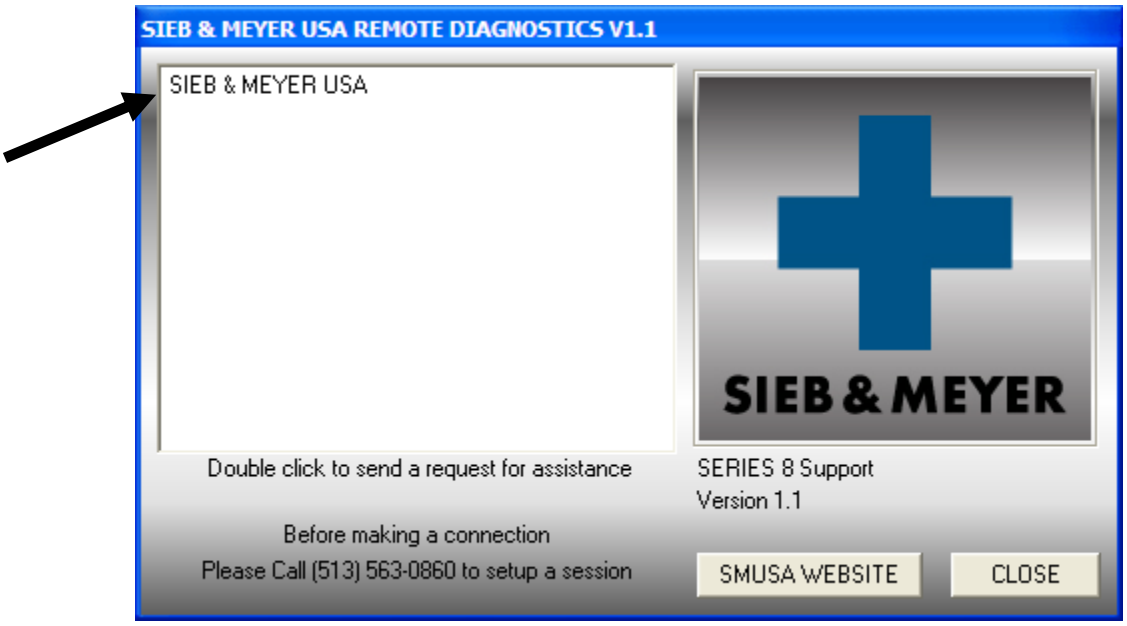
The Remote Diagnostics program is launched from the desktop of the SERIES 8 CNC. The software allows the engineers at SIEB & MEYER USA to look at and control the SERIES 8 CNC remotely.

If there is no Remote Diagnostic icon on the desktop, the software can be downloaded from the SIEB & MEYER USA website. The file is located on the bottom of the product literature page (link).

A working internet connection must be supplied to the SERIES 8 PC. For PC's with two ethernet ports, plug in the network cable to the empty port to give the PC internet access. If the PC only has one ethernet port on the motherboard, a second port must be added by using a USB to Ethernet device or (if the building has an existing wireless network) a wireless USB can be added to the system.

When adding a new device to the PC make sure the proper drivers are installed to insure the device will work properly.

Once the PC has internet access double click the desktop icon (seen above). After the window has loaded double click SIEB & MEYER USA, to start the connection.



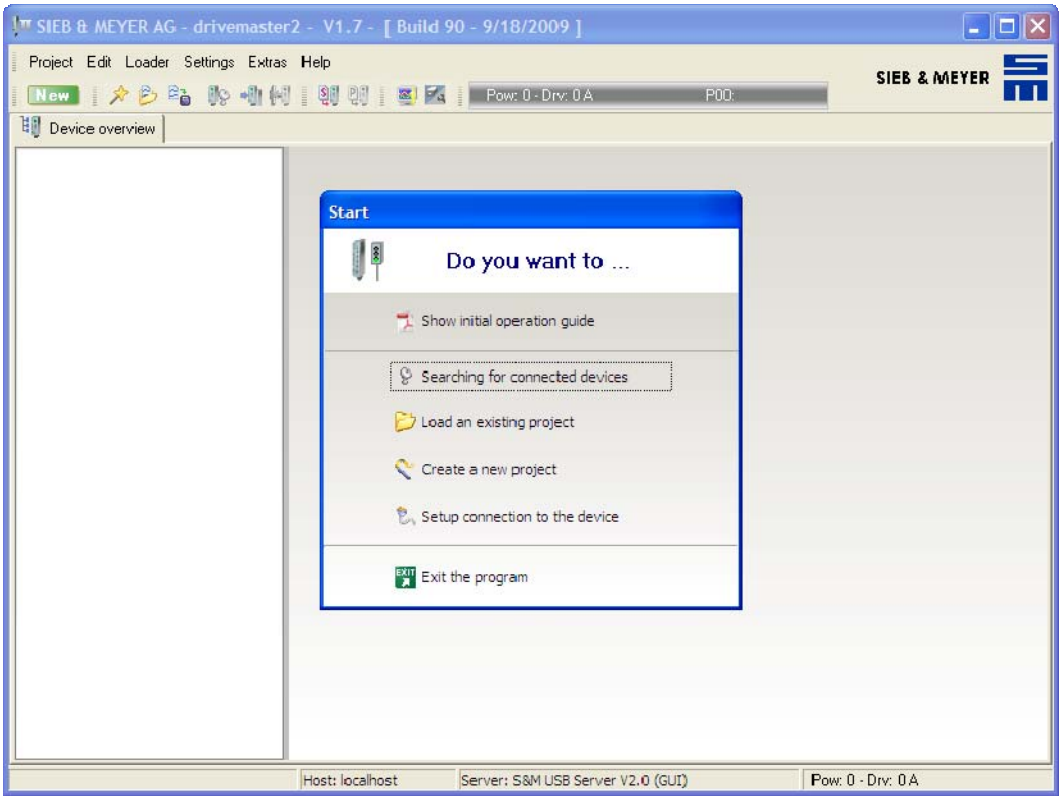
After the connection is established, the display will look like it normally does. The only difference is the machine can be controlled locally and remotely.

To close the remote connection right-click the SM logo in the PC's utility are and select Exit.

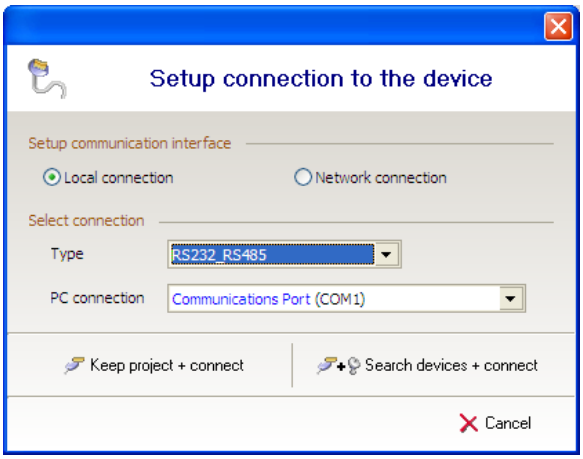
drivemaster2

Drivemaster2 is launched from the desktop of the SERIES 8 CNC. Drivemaster2 controls the drive and motor parameters for SIEB & MEYER SD2/SD2S servo amplifiers and frequency converters.

When the program is started there is a splash screen to initialize the drivers and software. After loading, a start window is displayed asking what to do next.



To check for the correct connection type, select the "Setup connection to the device".



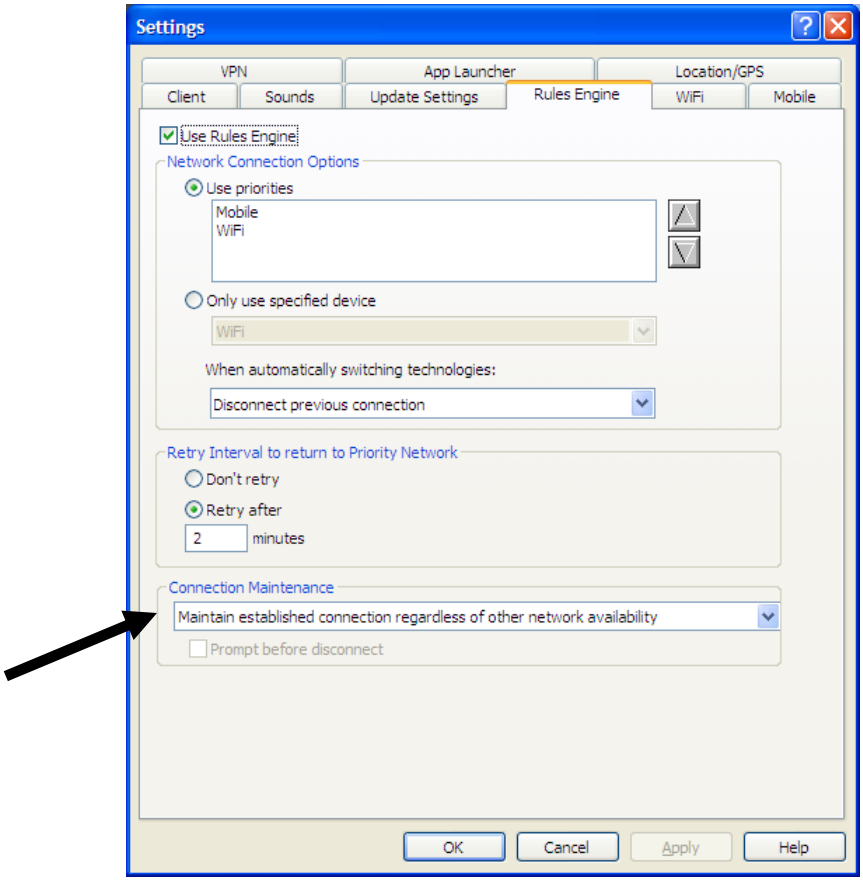
Then select the Type from the pull-down menu either RS232_RS485 or USB. If the RS232_RS485 type is selected then the COM port must also be selected from the pull down menu PC Connection. After selecting the correct connection type select "Search devices + connect".

For more information on the drivemaster2 see the SIEB & MEYER software manual for drivemaster2.

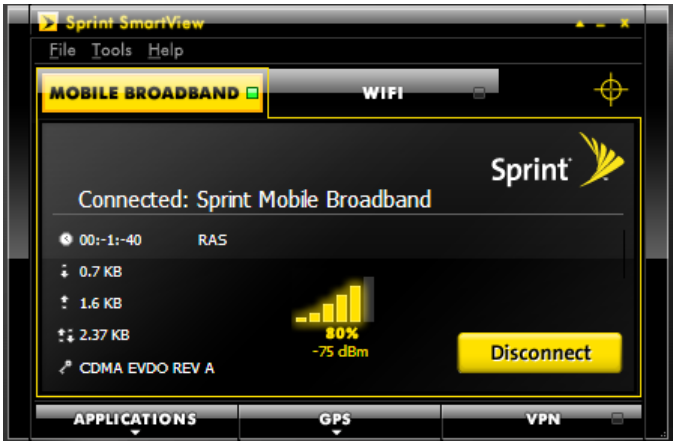
Sprint SmartView

The Sprint SmartView software is launched from the desktop of the SERIES 8 CNC. The Sprint SmartView software when used with the USB modem allows the computer to access the Internet without having to connect to the local network. When the software is started for the first time a setting needs configured correctly.

Click Tools and select Settings. Select the “Rules Engine” tab and change the Connection Maintenance selection to “Maintain established connection regardless of other network availability.” Click Apply and then OK. This allows the SERIES 8 CNC software and the wireless modem to work simultaneously.



Now the connection can be started. Click the yellow Connect button to begin the connection. Once the connection has been made the connection time, uploaded and download information, and signal strength are displayed. To close the connection click the yellow “Disconnect” button



For more information on the Sprint SmartView software see the Sprint user guide for the SmartView software.

Chapter:

SERIES 8 CNC File Locations

The primary files required to run the SERIES 8 CNC are stored in the directory:

C:\Program Files\Sieb-Meyer\Series8\

File Name	Description
series8.exe	Main program that runs on the PC
mc.cnc	CNC executive software that runs on the MC84 motion control card
series8.ini	File location definitions and option setting
series8.eng	Text file for English text (standard text)
newtext.eng	Text file updates for English text

PMI files provide specific interface of the CNC software to the machine, they are stored in:

C:\Program Files\Sieb-Meyer\Series8\pmi\

PMI files have a ".PMI" file extension.

APP files are used for canned cycles that may be supplied by SIEB & MEYER America, or written by the machine builder or end user. APP files may be stored in two different directories. For cycles that use the same name, the end user files overwrites the SMA files

C:\Program Files\Sieb-Meyer\Series8\app\ Cycles supplied by SMA ('OEMAPP')

C:\Documents and Settings\series 8 cnc\app\ Cycles written by end user

Part programs are stored in the following directory:

C:\Documents and Settings\series 8 cnc\programs\

System parameters are stored in files on the PC hard disk, and are updated automatically as data is changed. These files are loaded every time SERIES 8 software is started. These are called 'default' files, and are stored in the following directory:

C:\Documents and Settings\series 8 cnc\default\

File Name	Description
default.prm	Parameter Editor settings
default.fky	Function key definitions (mapping of function key to a command)
default.fxo	Fixture offsets
default.tlo	Tool offsets
default.smd	Page display layout and order
default.smm	Command order (in pull-down menus) and password level settings
default.sta	Status management variables (@STVAR[0] - @STVAR[59])
default.lsc	Leadscrew and backlash compensation values