HARDWARE WARRANTY

Mektronix Technology, Inc. ("Mektronix"), Mundelein, IL, warrants to the purchaser that all products manufactured by Mektronix will be free from defects in material and workmanship for one year from the date of shipment from the factory or from any authorized distributor of Mektronix. Mektronix's obligation under this warranty shall be strictly and exclusively limited to the repair or replacement, at the factory or a service center of Mektronix, of any such equipment or parts thereof which an authorized representative of Mektronix finds to be defective in material or workmanship under normal use. This warranty does not apply to any equipment that has been tampered with or altered in any way, which has been improperly installed or which has been subject to misuse, neglect or accident. There is no expressed or implied warranty of merchantability or fitness for a particular purpose. Mektronix shall have no liability whatsoever in any event for payment of any incidental or consequential damages, including, without limitation, damages for injury to any person or property.

NOTICE

The information furnished in this manual by Mektronix Technology, Inc. is believed to be accurate and reliable. However, no responsibility is assumed by Mektronix Technology, Inc. for its use; nor any infringements of patents or other rights of third parties which may result from its use. Mektronix Technology, Inc. reserves the right to make engineering refinements to all products and to make changes in this manual without prior written notification. No part of this manual may be reproduced in any form or by any means without prior written permission from Mektronix Technology, Inc.

*P-CNC and Personal-CNC are registered trademarks of Mektronix Technology, Inc.

LIMITED USE SOFTWARE LICENSE AGREEMENT

The terms and conditions of this Agreement shall apply to the P-CNC software supplied on the distribution disk and derivatives obtained therefrom, including any copy. If you have executed a separate Software Agreement covering the P-CNC software supplied herewith, such Software Agreement will govern.

1. TITLE AND LICENSE GRANT

   P-CNC is copyrighted and contains proprietary information protected by law. All supplied software, and all copies thereof, are and will remain the sole property of Mektronix Technology, Inc. ("Mektronix"). Mektronix hereby grants you non-transferable and non-exclusive rights to use all supplied software, in whatever form recorded, which is furnished to you under this Agreement. This grant is limited to use with one Mektronix MCP-04 board that has been purchased as part of the P-CNC System Development Package. The MCP-04 serial number and the name of the licensee are to be entered using the supplied OEM setup program. Any other use of this Software shall automatically terminate this license.

   You agree to use your best efforts to see that any user of the P-CNC software licensed hereunder complies with the terms and conditions of this License Agreement and refrains from taking any steps, such as reverse assembly or reverse compilation, to derive a source code equivalent of the software.

2. SOFTWARE USE

   A. You are permitted to make a single archive copy, provided the Software shall not be otherwise reproduced, copied, or disclosed to others in whole or in part.
   B. The Software together with any archived copy thereof, shall be either returned to Mektronix or destroyed when no longer used in accordance with this license Agreement.

3. LIMITED WARRANTY

   A. Mektronix warrants that the supplied Software will be in good working order and will replace, without charge, any Software which is not in good working order if returned to the location where you obtained it within (90) days of delivery to you.
   B. Mektronix does *not* warrant that the functions of the P-CNC system will meet your requirements or that Software operation will be error-free or uninterrupted.

C. Mektronix has used reasonable efforts to minimize defects or errors in the Software. HOWEVER, YOU ASSUME THE RISK OF ANY AND ALL DAMAGE OR LOSS FROM ITS USE, OR INABILITY TO USE THE SOFTWARE.

D. YOU UNDERSTAND THAT, EXCEPT FOR THE 90 DAY LIMITED WARRANTY RECITED ABOVE, MEKTRONIX AND ITS AGENTS MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, AND SPECIFICALLY DISCLAIM ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

4. <u>EXCLUSIVE REMEDIES AND LIMITATIONS OF LIABILITIES</u>

A. YOU AGREE THAT YOUR SOLE REMEDY AGAINST MEKTRONIX AND ITS AGENTS FOR LOSS OR DAMAGE CAUSED BY ANY DEFECT OR FAILURE IN THE SOFTWARE REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT, INCLUDING NEGLIGENCE, STRICT LIABILITY OR OTHERWISE, SHALL BE THE REPLACEMENT OF MEKTRONIX FURNISHED SOFTWARE, PROVIDED SUCH SOFTWARE IS RETURNED TO MEKTRONIX WITH A COPY OF YOUR PURCHASE INVOICE.

B. Regardless of any other provisions of this Agreement, neither Mektronix nor its agents shall be liable for any indirect, incidental, or consequential damages (including lost profits) sustained or incurred in connection with the use, operation, or inability to use the Software, or for damages due to causes beyond the reasonable control of Mektronix or its agents attributable to any service, products or action of any other person.

You acknowledge that you have read this Agreement and understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that, except for separate written agreements between Mektronix and you, this Agreement is the complete and exclusive statement of the rights and liabilities of the parties.

TABLE OF CONTENTS

Table of Contents

Section 1

P-CNC OVERVIEW

This manual describes the Mektronix Technology, Inc. P-CNC (*PERSONAL*-Computer Numerical Control®) system development package, which converts a personal computer into a full featured CNC controller. P-CNC uses a PC compatible computer to run industry standard NC programs written in familiar G and M codes (EIA RS-274D). The System Development Package (SDP) consists of the P-CNC application program with utility software, the MCP-04 four-axis motion controller board, the MCP-R16 I/O module and a set of three interconnect cables. P-CNC controls up to four servo motors plus a speed reference for the spindle drive. The fourth axis may be set up as a linear or rotary axis. Each axis is independently controlled by a dedicated microprocessor with onboard I/O for interfacing to external switches or devices. The MCP-04 connects to the MCP-R16 I/O module that provides sixteen 5Amp relay outputs and 28 contact closure inputs. The I/O module mounts to a standard DIN rail and provides screw terminal connections to external devices. Standard PLC operations are built-in but may be augmented with an external PLC connected to the PC's parallel port. The P-CNC control panel features include: feed rate override, feed hold, emergency stop, cycle start, single cycle, axis select, handwheel mode and handwheel positioning. Other operations are handled using a keypad or standard keyboard.

P-CNC interprets NC programs written as text files, which may be stored on a hard disk or transferred to the computer using a floppy disk or over a communications network. Your program may also be manually entered and/or edited by means of a resident text editor. P-CNC supports DOS shells so that other programs can be accessed within P-CNC or during execution of a user's NC program using specialized M-code functions.

Mektronix, in its commitment to providing only the finest motion control products, has used the highest quality components and construction techniques. Thus ensuring long-term, trouble-free, operation of the MCP-04 motion controller in the demanding conditions typical of an industrial environment. The MCP-04 uses surface mount components to provide minimal board size and fits into a half size computer chassis.

1.1. MANUAL ORGANIZATION

The P-CNC Operations Manual is divided into the following four sections:

- **P-CNC Overview**
- **Configuring P-CNC**
- **P-CNC Operation**
- **Programming with P-CNC**

P-CNC Overview gives a general description of the manual organization and provides a brief overview to acquaint you with the main P-CNC operational features.

Configuring P-CNC describes how to install the distributed software, giving details on how to interface with external hardware and how to setup the P-CNC software parameters for your specific machine.

P-CNC Operation describes in detail how the P-CNC program operates including descriptions of all operational menus. Operating P-CNC is intuitive and straight forward due to all the menu options being described on the display monitor.

Programming with P-CNC describes the function of each command and its effect on program execution. This section serves as the operator's reference for all G and M codes available in P-CNC. If you are familiar with G-code programming, you will recognize many of the codes and their implementation.

1.2. FEATURES OF THE P-CNC SYSTEM

- Operates on 80486 and Pentium personal computers
- Uses G and M-codes, conforming to EIA standard RS-274D
- Functional panel displays with GUI interface
- Three and four axis simultaneous profiling
- Linear, circular, helical, and spline profiles
- Look-ahead feature for continuous profiling and feed rate adjust
- Digital compensation for smooth, accurate motion profiles
- Virtual memory manager can handle huge NC files

- Tool length and cutter radius compensation
- Custom tool changer sequences easily developed
- Input and output function control (even during profiles)
- Simulation capabilities with 2-D graphics (3D surfacing optional)
- Digital Read-Out (DRO) mode for manual operation
- Full browsing capability with execution break-point setting
- MDI - Manual Data Input within Run menu
- Feed hold & feed rate override switches
- Handwheel jog control with override and mode selector
- Dimensional scaling and mirror image
- Canned cycles for drilling operations
- DOS shells from within a NC program or P-CNC Utility menu
- Six user defined work coordinates with adjustable local coordinates
- Exerciser program for motion control debugging
- LAN - Local Area Networking option available

P-CNC allows the user to program in familiar G and M-codes maintaining compatibility with EIA standard RS-274D and other CNC controllers. Mektronix's implementation of the standard closely follows that of GE/Fanuc controllers. The easy-to-use menu driven software provides the capabilities to simulate and run NC programs while displaying current status and axis positions. The screen display and functional menus ensure that the user will learn to operate P-CNC quickly. Your favorite text editor may be used to write new programs or modify existing ones. P-CNC may also be configured to run your CAD/CAM software directly from the Utilities panel.

P-CNC features include three (optionally four) axis simultaneous profiling, tool function control, input and output control and full jogging control. The program may be executed either in Block mode or Look-ahead mode, allowing the program to interpret a single block or read through and assemble motion sequences into continuous profiles. P-CNC has many built-in features that should be used to get the maximum benefit from the software.

Other development software is provided to assist machine designers and system integrators in writing software utilities to expand P-CNC’s capabilities. An Exerciser program is included for motion control debugging by issuing commands interactively to the MCP-04 board using the keyboard. It also allows system programmers to become familiar with commands used in the Programming Interface Library. This library is provided for high level language programmers to develop their own application programs using a C compiler. Custom tool changer and spindle orient sequences can be developed using these libraries. The advantage of this approach over using a separate PLC is the ability to move axes during the tool change sequences. The source code for these library routines are provided on the MCP-04 distribution disk. User application programs can be interfaced with P-CNC to obtain more functionality or to expand the number of I/O points. This is typically done for complicated tool changers.

1.3. OPERATIONAL OVERVIEW

This section provides a quick overview of the operation of the P-CNC program. The main operations are to execute NC programs and to perform manual functions. Other support features are used to assist in these two basic components of P-CNC. The demonstration program, **demo.cnc**, is included on the distribution disk to give an example of what a NC program may look like and to illustrate basic programming features. Once the system has been installed per the instructions given in Section 2, you may wish to run this program to become familiar with P-CNC. The CNC demonstration program is programmed to make 3 identical parts as shown next in Figure 1-1.

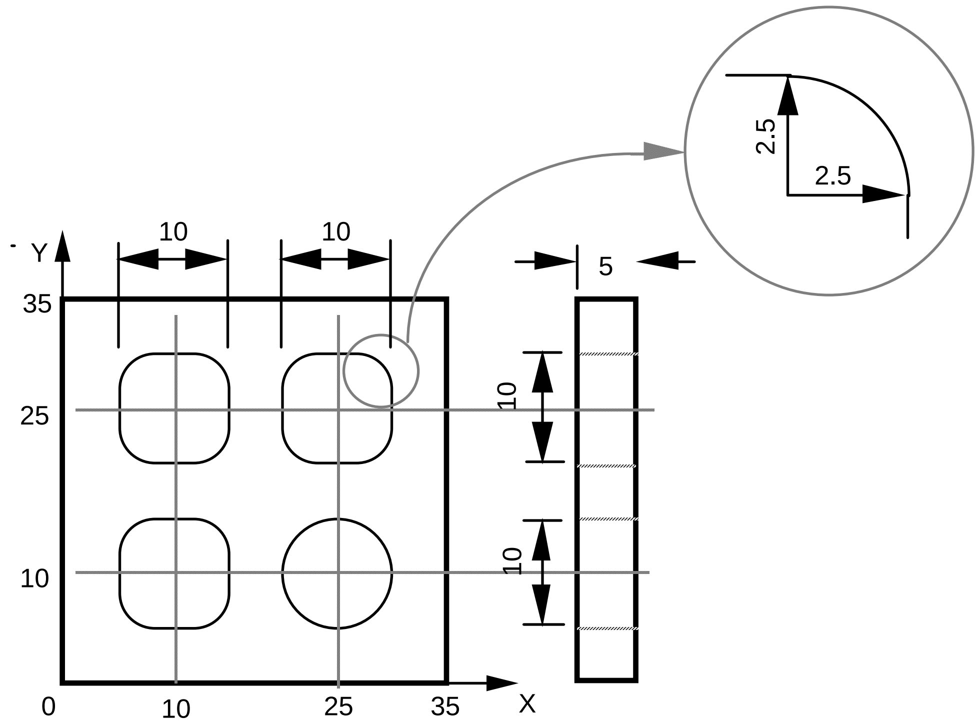


Figure 1-1. Part Machined by DEMO.CNC Program

Operational functions are given on the screen allowing you to run the P-CNC program without additional assistance after a bit of practice. An on-line help facility is available at the computer by typing the '?' key.

1.3.1. The Run Screen

The Run screen is the start-up screen once P-CNC licensing information has been displayed as well as an initial keystroke. The bottom control panel display ten function keys that allow the operator to switch to other screens and modes of operation. Initially, the machine needs to establish a Home position. This operation is executed by hitting the F7 key or through the Jog screen. After homing is complete, the operator may load and execute a NC program file.

1.3.2. Loading a NC File

From the Run screen, select F1, and the Load screen will appear. The top of the panel gives the name of the directory from which files are listed and the search extension. If you wish to load a NC program file contained in another directory, you can move to that directory by changing the default directory. Each time you re-open the Load screen, the last directory you moved to will be given. All ∗.CNC files in the directory are listed on the screen, with the selected file highlighted. To select a different file, use the cursor keys (↑↓←→) to move the selection highlight.

In order for a file to be listed, the filename must have the extension .CNC appended, however, filenames are listed without the .CNC extension. For example, the demonstration file is named “demo.cnc” but is listed as “demo”. If you create new NC files, be sure to use the .CNC extension in the filename.

You may also select the file by typing [F], the “Enter Filename” option. Enter the file name and hit Enter to load the file. If the file is not listed in the default directory, you may include a full directory path to load a file in another directory (refer to the DOS manual for information about directory paths and path names). You may also create a new file by typing the new CNC file name. When you type the filename, you may use the .CNC extension or you may omit it. For example, you may type **demo.cnc** or **demo**; the program will accept either entry.

If you wish to always load CNC files from a different directory, you must change the “Default Directory” by typing [D]. Do *not* include the filename in the path. When you have typed an existing path name, the program will print the path name in the default directory box and all CNC files in the directory will be listed. If there are no CNC files in the directory, the message “No CNC file found in this directory” will appear. From now on, every time you enter the Load screen, the same default directory will appear.

1.3.3. Text Editing

The default editor is supplied with DR-DOS 7.2, but can easily be changed to load the editor of your choice. Note that the editor used should be a text editor or one that can store a file in text only format. See the instructions described in Section 2.4 to change the default editor in the OEM Setup Utility.

P-CNC converts all text to uppercase when displayed in the RUN panel. P-CNC is not case sensitive and treats each new line as a control block. NC code must not conflict within any given block, so it is often beneficial to keep blocks from becoming too long or unclear. There is no advantage to packing command codes without spaces since P-CNC reconstructs the NC program into its own internal format. For example, the following

```
G01 X3.45 Y2 F150  (Linear move to IP at 150 ipm)
```

is much easier to read than

```
G1X3.45Y2F150
```

1.3.4. Running NC Programs

The Simulation and Run options allow for execution of NC programs. Simulation operates identically to Run except for re-routing position commands into a graphics buffer. Thus, Simulation is an effective way to test and verify a NC program without having to run it on the machine tool. A VGA graphics adapter is required for displaying the programmed tool path.

The NC program is listed on the left side of the panel and the results of program operation are displayed on the right side. As the program runs, you will see the control functions and axis positions change as the program is executed. Figure 1-2 shows the display during a simulated execution.

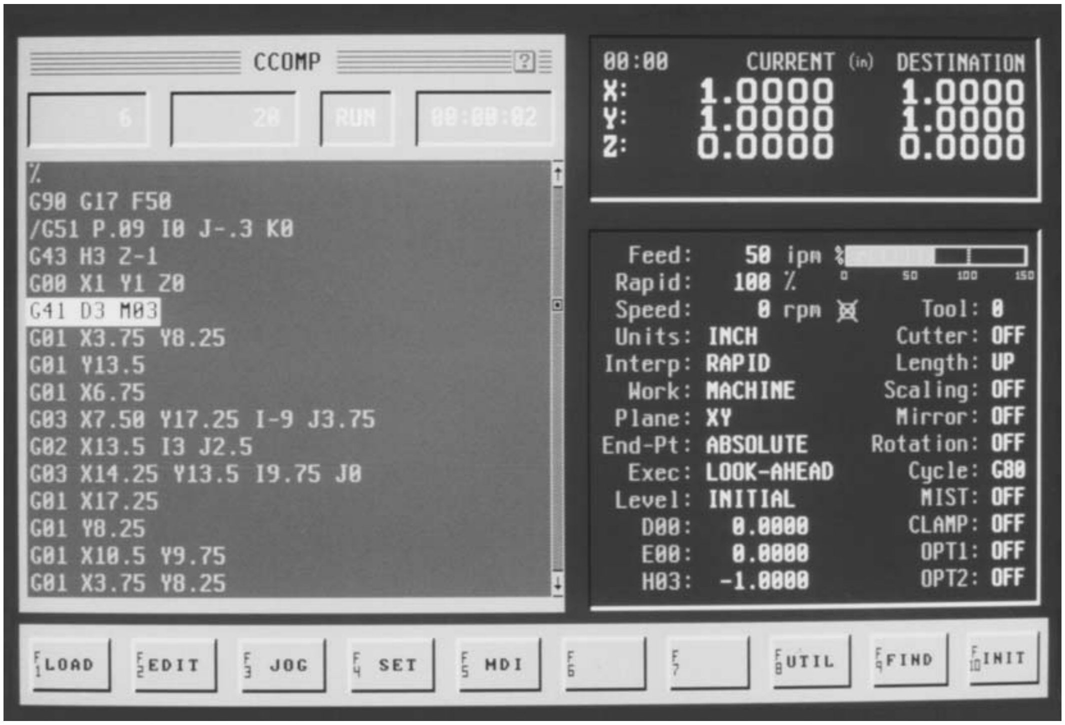


Figure 1-2. Run Screen

Note the function buttons at the bottom of the screen which select other run options. The operator may also enter the Status area of the panel by hitting the CTL key. Once in the Status area, the modes are highlighted for easy access. You may also cursor to the option and edit the mode by typing a new value or use the space bar for toggling to an alternate setting. Once complete, hitting the CTL key again to leave the Status panel area.

To browse through the program without running the NC program, use the up/down cursor keys (↑ ↓) or Home and End keys to quickly return to the last executed block or to the break-point, respectively. The End key on the numeric keypad is used to set an execution break-point at the current cursor location. Hitting the End key again will remove the break-point. Execution of a program begins with the block that is highlighted. Type [I] or [F10] to initialize all the execution registers to their defaults and start execution from the beginning of the currently loaded NC program. To run the program continuously, type [R] or hit the Cycle Start button on the control panel. Initially, you may want to run a new program in step mode or in the simulation mode. Step mode is set by a control panel switch and simulation mode is set in the Plot panel of the Setup screen. Hit [F4] to enter the Setup screen and then Plot and toggle the simulation mode using the space bar.

To manual input NC code from the keyboard, type [M] or [F5] to enter MDI mode. Multiple blocks can be entered in the MDI editor by terminating each block with a semicolon. Type [Esc] to quit the MDI mode.

1.3.5. Jogging/Homing Screen

Jogging operations are performed by selecting the JOG option from the Run screen. Operational keys are shown in the lower panel. Additional instructions will be given in this same panel for operations which require more choices.

1.3.5.1. Jogging Axes

The cursor keys perform jogging according to the Velocity and Increment parameter setting in the "Axis Control" panel. Once these values are set, the operator can either jog in velocity or increment modes. Only the axes highlighted are selected for jog operations. The Axis Select switch is used to select the desired axis. Use the up/down cursor keys (↑↓) to jog the axes at the selected position increment defined previously. Hold down the left/right cursor keys (←→) to jog continuously in velocity mode and then release the key to decelerate to a stop. The feed rate override switch will further adjust the jog velocity.

Jogging is also possible through a handwheel after typing **[F4]** from the Jog screen. The jogging speed is related to the speed at which the handwheel is turned. A Handwheel mode switch provides two positioning and two velocity settings for the handwheel. The feed rate override switch further adjusts the handwheel multiplier ratio from low to high. The handwheel allows the operator to quickly move to an area of the work space and accurately position the tool by changing rotation speed.

1.3.5.2. Switch Toggling and Tool Selection

The Shift plus F1-F5 function keys are used to toggle outputs highlighted in the "Relay" control panel at the left side of the screen. This option allows the spindle, coolant, clamp, and two other options to be turned on and off. The display shows the current status of the switch settings. The [F10] function key is used to toggle the position display between actual, command and position error.

1.3.5.3. Homing to Machine Coordinate Reference Frame

Type [F7] from the Jog screen to home the selected axes. Use the [F1-F4] keys to deselect and select an axis. Type [F7] again to initiate an automatic homing sequence. If the home switch is not detected, a time-out will occur with warning issued. The machine coordinate frame may be manually entered when automatic homing is disabled. This option is only available when homing switches are not installed. All “out of range” errors are referenced from the machine coordinate frame determined by this homing procedure. Type [Esc] to return to the Jog menu.

Section 2

CONFIGURING P-CNC

This section gives details on installing and configuring the P-CNC development package. The hardware installation should be checked along the way using the Exerciser program described in section 2.4.2. The system developer can make use of its powerful diagnostic tools to simplify trouble shooting external components. It is always beneficial to perform operational verification checks on interface components as the project progresses in a step-by-step fashion.

A PC-compatible computer serves as the host for the system. The recommended computer configuration required for the P-CNC system is listed below.

8 MB of RAM
0ne 1.44 MB 3.5" floppy and 500 MB hard disk
1.5 MB Flash Memory bootable disk
14" VGA color monitor
66 MHz 80486DX or faster microprocessor

There are many computer related options that may be used in developing the P-CNC controller. The hard disk should be shock mounted and have high impact resistance typical of drives used in portable computers. A solid state disk may be considered for use in extremely harsh environments. The enclosure should be designed for the operating conditions and temperature of the factory. It is possible to replace or augment the standard keyboard with a membrane keypad and discrete operational buttons using a keyboard encoder board. The keyboard encoder board allows configuration of interface switches to simulate standard PC/AT keyboard keystrokes. P-CNC can be configured to use keyboard macros that assign one key to perform multiple keystrokes. This makes frequently used features easier for the operator to perform.

2.1. SOFTWARE INSTALLATION

2.1.1. Installing P-CNC Software

P-CNC runs most efficiently from a hard drive or solid state disk. To install the software, type:

install

from the A: directory. The installation batch program will copy the distribution files to the default P-CNC2 directory on the C: drive.

Read the RELEASE.DOC and README.DOC files for the latest additional information. The following files should be installed on the hard drive in sub-directory P-CNC2.

README.DOC	installation notes
RELEASE.DOC	latest release history
P-CNC3X.EXE	P-CNC application for 3-axis, or
P-CNC4X.EXE	Fourth axis version of P-CNC
OEM.EXE	OEM Setup Utility
OEM.F16	OEM setup file
OEM.SET	OEM parameter setup (generated file)
CNC.SET	CNC parameter setup (generated file)
JOG.SET	JOG parameter setup (generated file)
OEM.HLP	OEM help file
P-CNC.HLP	P-CNC help file
DEMO.CNC	demonstration NC program

The following files should be in the UTILS subdirectory on the distribution disk.

EX.EXE	Motor controller Exerciser
MCP.HLP	Data file for on-line help
CHECK.EXE	MCP-04 board diagnostics
MCA.TST	Check document A

MCB.TST	Check document B

The following files should be in the MCPLIB subdirectory on the distribution disk.

MCP.C	C library routines
MCP.H	C library function declarations
MCP04.LIB	MCP-04 compiled library

After checking the distribution disk using the 'dir' command, note the software version and dates of the P-CNC.EXE file below.

P-CNC File Date:________________ Version:___________

2.1.2. PC Boot-Up Configuration

The Personal Computer (PC) is configured using the “config.sys” and “autoexec.bat” text files located in the root directory. An environment variable “P-CNC” must be set to locate where the setup files are stored. Add the following line to the “autoexec.bat” file assuming P-CNC2 is the install directory.

SET P-CNC=C:\P-CNC2

P-CNC takes advantage of system resources that are loaded in these files. Include a PATH statement in the “autoexec.bat” file that specifies the path where P-CNC is installed. The example configuration files given below are a good starting point and can easily be modified using a text editor for a particular computer setup.

CONFIG.SYS

```
DEVICE=c:\dos\himem.sys
DOS=HIGH,UMB
SHELL=c:\dos\command.com c:\dos\ /E:512 /p
BREAK=ON
FILES=40
BUFFERS=50
STACKS=9,256
```

AUTOEXEC.BAT

```
@echo off
prompt $P$G
set COMSPEC=c:\dos\comand.com
set TEMP=c:\dos
set PATH=c:\dos;c:\p-cnc2\utils
set P-CNC=c:\p-cnc2
P-CNC3X
```

P-CNC uses extended (XMS) or expanded (EMS) memory when shelling out to another DOS executable program. This frees up conventional memory for the spawned program to execute. When expanded memory is not detected P-CNC will write to the TEMP directory specified in AUTOEXEC.BAT. P-CNC also uses extended memory, expanded memory and hard disk space for its virtual memory manager. The virtual memory size is set in the OEM Setup Utility and must be less than 1 gigabit total. The HIMEM.SYS driver allows access to the extended memory above 1 MB.

The computer BIOS setup can have a drastic effect on P-CNC performance. You should use Video ROM Shadow for faster screen display. The CPU settings should be optimized for speed. Generally the default BIOS CMOS settings are good enough for most computer systems.

2.1.3. P-CNC Program Execution

P-CNC may be invoked from any directory using the following format:

p-cnc3x [/F:<*nc-file*>]

The optional /F:<*nc-file*> can be used to automatically load a default program when the program initializes. P-CNC uses the default directory path saved in the LOAD panel as the base reference from which the NC-file will be searched. If the *nc-file* name is given and the file is found, this file will be loaded and parsed (syntax checked) before the main run screen is displayed. The NC program name of the loaded file will be shown at the top of the run execution panel. If the file with the given name is not found, or if the NC program contains syntax errors, appropriate messages will appear to indicate the problem.

P-CNC can be organized to meet the needs of several users that desire different default P-CNC setups and NC part program directories. In order for P-CNC to execute, you must either provide a DOS path or currently be in the same directory containing 'P-CNC3.EXE' or alternatively, the four axis version 'P-CNC4X.EXE'. The setup files must be located in the directory specified by environment variable P-CNC set in the 'autoexec.bat' file. This can be used to your advantage when considering how to organize P-CNC for multiple users or to provide different operating setups for different types of NC part programs. It also allows the P-CNC program to be stored in Flash memory and the setup files stored on the hard disk, thus, freeing up flash memory.

A NC part program may be located in any directory and P-CNC will find this directory if its path has been specified in the Load menu screen. The diagram shown in Figure 2-1 gives an example of how one may wish to organize P-CNC file directories on a hard disk.

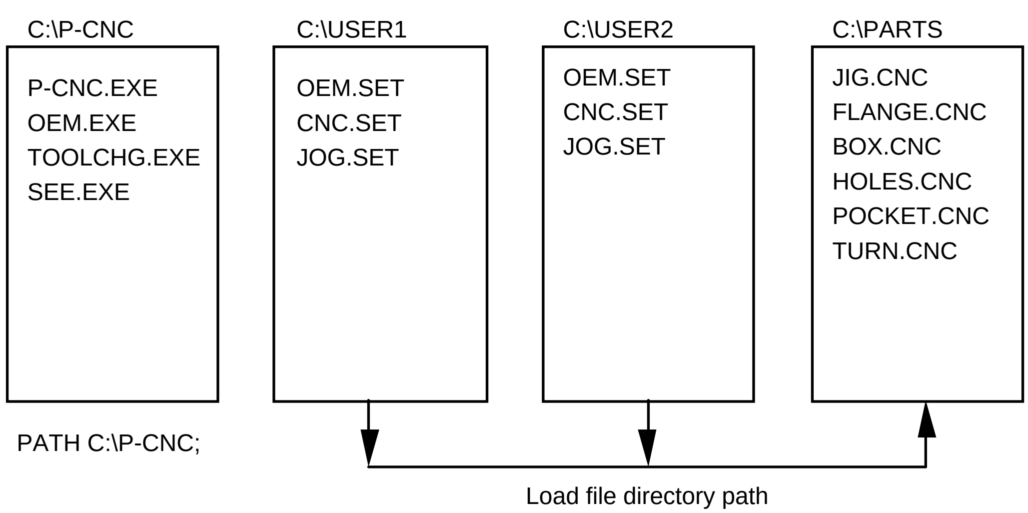


Figure 2-1. P-CNC Directory Organization

In this example, User1 or User2 would enter their respective directories and type "P-CNC". The DOS environment path would find P-CNC3X.EXE and start execution. (This path should be added to the "autoexec.bat" file in the root directory.) P-CNC then loads the setup files as specified by the environment variable P-CNC. Use the DOS command

    set P-CNC=c:\user1

The CNC.SET file contains the path to the PARTS directory if it was specified the last time P-CNC was invoked from this directory. P-CNC can also be invoked with the file name specified as a command argument. For example, assume you wish to load "DEFAULT.CNC". This can be accomplished by invoking P-CNC with the following syntax:

C:\> **p-cnc3x /f:default**

P-CNC would start with the NC file "DEFAULT.CNC" loaded due to the load directory path being "C:\PARTS".

2.2. HARDWARE INSTALLATION

P-CNC has many built-in features that should be used to get the maximum benefit from the software. The overall hardware features are shown with block diagrams in Figure 2-2.

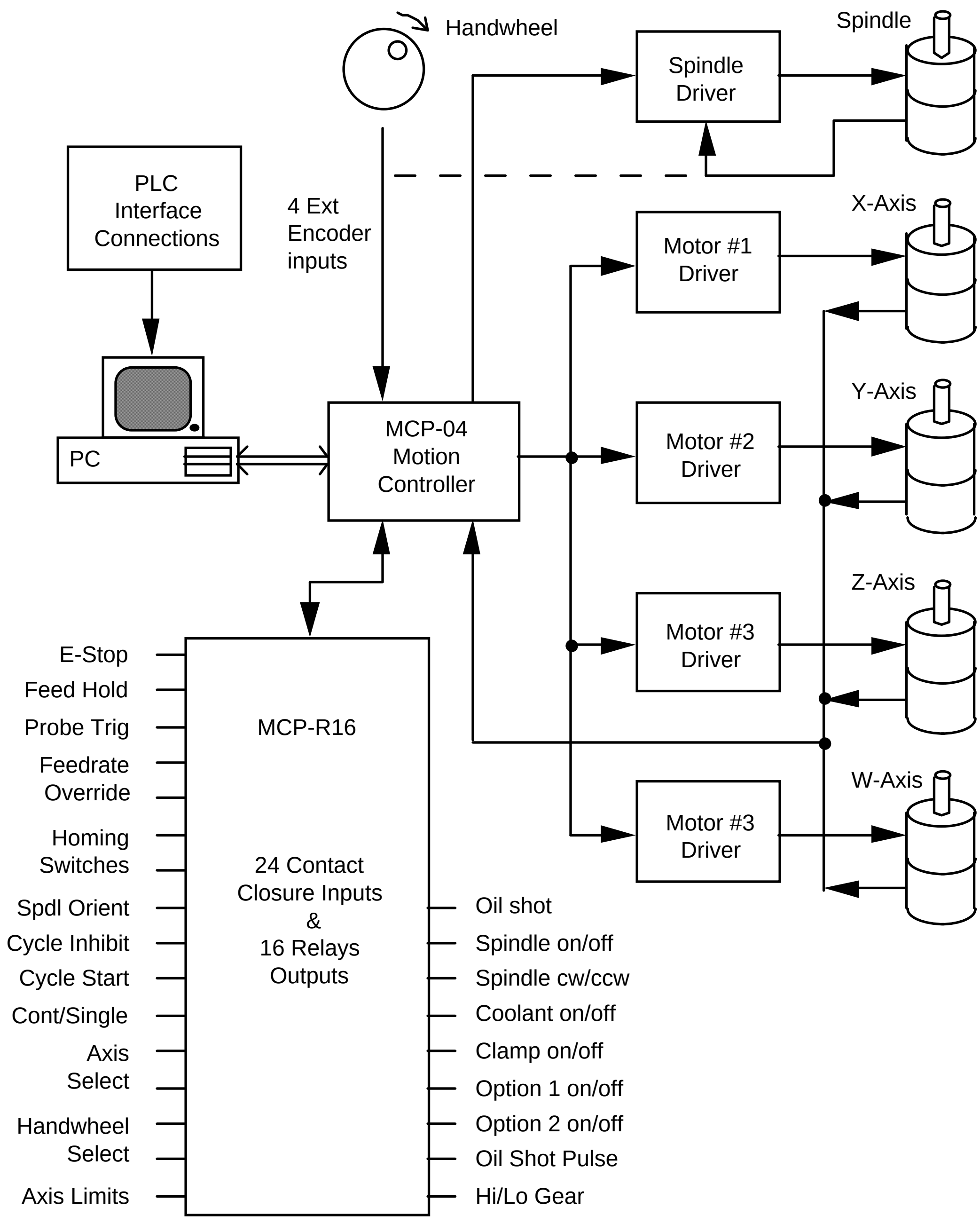


Figure 2-2. P-CNC System Configuration

Power Recommendations: In order to assure reliable operation of the P-CNC system in an industrial setting, it is important to pay particular attention to the power systems. A good system ground is of the utmost importance due to high level transients generated by switching devices and the power drivers controlling the servo motors. All subsystems should be tied to ground along a grounding bar located next to the machine. Ideally, a copper grounding stake located near the machine should be attached to the grounding bar via a wire braid. The 120 Vac supply line to the computer may need to be protected with a ferroresonant transformer. These devices provide line regulation and isolation for reliable operation of the computer subsystem. If power outages are of concern, an uninterruptible power system provides the added benefit of battery backup.

2.2.1. Configuring the MCP-04 Board

The MCP-04 Four Axis Motion Control board uses the PC's I/O addresses from 330 to 33B hexadecimal. A standard PC configuration will not conflict with this address space. If additional I/O boards are installed, verify that they do not share addresses with the MCP-04 board.

The only jumper configuration is for determining the limit switch polarity. When using the MCP-R16 I/O module, the limits must be configured with normally closed switches. In this way, a broken connection will always be detected. The MCP-04 board layout is given in Figure 2-3. The jumpers JP1, JP2, JP3, and JP4 assign polarity to the limit switches. Each jumper has a '+' sign indicating a connection to +5V. Place jumpers on the '+' side when limit switches are not installed. Jumper to the opposite side (closest to the to edge) for normal configurations using the MCP-R16 and normally closed limit switches.

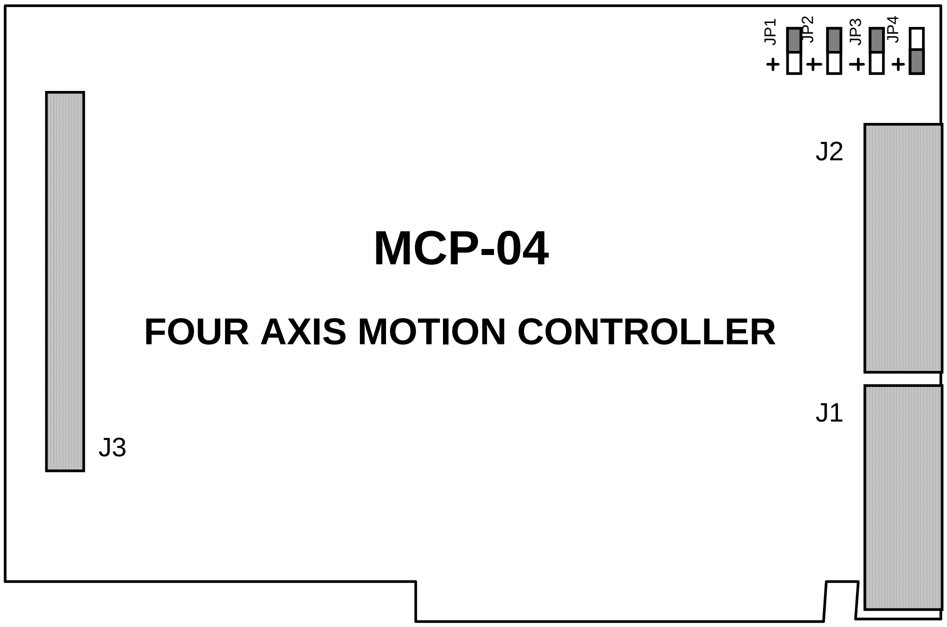


Figure 2-3. MCP-04 Board Layout

Connector J1 interfaces to the motor driver modules, J2 connects to the MCP-R16 I/O module, and J3 provides convenient break-out interconnections for axis encoders and up to four external encoders (used for other functions such as a handwheel and spindle speed measurement). P-CNC also makes use of the PC's parallel port for communication to a PLC using M, S, and T address strobes with BCD output and other control signals. The P-CNC developer can disable the operation of the parallel port if these features are not required or set a valid port address in the OEM Setup Utility.

The layout of the interconnect wiring should be well thought out for modularity and noise immunity. Provided the interface connections are made in accordance with industry standards, your P-CNC system will work reliably in industrial environments. The MCP-04 connector pin-outs given in Table 2-1 are used by the P-CNC package development system. Typically, axes assignments are XYZ for axis 1, 2, 3, respectively and C for the optional fourth axis. Connect the jogging handwheel to external encoder number one.

J1 26-Pin High Density D-Sub Connector

J1-1	**Command (Axis 1)**	**J1-10**	**Common (Axis 1)**	**J1-19**	**PWM Pulse (Axis 1)**
J1-2	**PWM Sign (Axis 1)**	**J1-11**	**Command (Axis 2)**	**J1-20**	**Common (Axis 2)**
J1-3	**PWM Pulse (Axis 2)**	**J1-12**	**PWM Sign (Axis 2)**	**J1-21**	**Command (Axis 3)**
J1-4	**Common (Axis 3)**	**J1-13**	**PWM Pulse (Axis 3)**	**J1-22**	**PWM Sign (Axis 3)**
J1-5	**Command (Axis 4)**	**J1-14**	**Common (Axis 4)**	**J1-23**	**PWM Pulse (Axis 4)**
J1-6	**PWM Sign (Axis 4)**	**J1-15**	**Command (Spindle)**	**J1-24**	**Common (Spindle)**
J1-7	**Enable Out (Axis 3)**	**J1-16**	**Enable Out (Axis 2)**	**J1-25**	**GND**
J1-8	**+5V Output**	**J1-17**	**Enable Out (Axis 4)**	**J1-26**	**Enable Out (Axis 1)**
J1-9	**GND**	**J1-18**	**+5V Output**		

J2 44-Pin High Density D-Sub Connector

J2-1	**OP-A0: M03/M05**	**J2-16**	**OP-A1: M04/M05**	**J2-31**	**OP-A2: M08/M09**
J2-2	**OP-A3: M10/M11**	**J2-17**	**OP-A4: M12/M13**	**J2-32**	**OP-A5: M14/M15**
J2-3	**OP-A6: M07**	**J2-18**	**OP-A7: M40/M41/42**	**J2-33**	**OP-B0: User 1**
J2-4	**OP-B1: User 2**	**J2-19**	**OP-B2: User 3**	**J2-34**	**OP-B3: User 4**
J2-5	**OP-B4: User 5**	**J2-20**	**OP-B5: User 6**	**J2-35**	**OP-B6: User 7**
J2-6	**OP-B7: User 8**	**J2-21**	**IP-L1: Ax1 Limit**	**J2-36**	**IP-L2: Ax2 Limit**
J2-7	**IP-L3: Ax3 Limit**	**J2-22**	**IP-L4: Ax4 Limit**	**J2-37**	**IP-A0: E-Stop**
J2-8	**IP-A1: Feed Hold**	**J2-23**	**IP-A2: Ext Fault**	**J2-38**	**IP-A3: Probe Trig**
J2-9	**IP-A4: Feed D0**	**J2-24**	**IP-A5: Feed D1**	**J2-39**	**IP-A6: Feed D2**
J2-10	**IP-A7: Feed D3**	**J2-25**	**IP-B0: Ax1 Home**	**J2-40**	**IP-B1: Ax2 Home**
J2-11	**IP-B2: Ax3 Home**	**J2-26**	**IP-B3: Ax4 Home**	**J2-41**	**IP-B4: Spdl Orient**
J2-12	**IP-B5: Cycle Inhibit**	**J2-27**	**IP-B6: Tool Chg1**	**J2-42**	**IP-B7: Tool Chg2**
J2-13	**IP-C0: Cycle Start**	**J2-28**	**IP-C1: Cont/Single**	**J2-43**	**IP-C2: Ax Sel DB0**
J2-14	**IP-C3: Ax Sel DB1**	**J2-29**	**IP-C4: Hdw Sel DB0**	**J2-44**	**IP-C5: Hdw Sel DB1**
J2-15	**IP-C6: Panel Spare**	**J2-30**	**IP-C7: Panel Spare**		

J3 50-Pin DIN Ribbon Connector

J3-1	**+5V Output**	**J3-2**	**Ch A (Axis 1)**
J3-3	**/Ch A (Axis 1)**	**J3-4**	**Ch B (Axis 1)**
J3-5	**/Ch B (Axis 1)**	**J3-6**	**/Ch I (Axis 1)**
J3-7	**Ch I (Axis 1)**	**J3-8**	**2.6 Vref Output**
J3-9	**GND**	**J3-10**	**+5V Output**
J3-11	**Ch A (Axis 2)**	**J3-12**	**/Ch A (Axis 2)**
J3-13	**Ch B (Axis 2)**	**J3-14**	**/Ch B (Axis 2)**
J3-15	**/Ch I (Axis 2)**	**J3-16**	**Ch I (Axis 2)**
J3-17	**2.6 Vref Output**	**J3-18**	**GND**
J3-19	**+5V Output**	**J3-20**	**Ch A (Axis 3)**
J3-21	**/Ch A (Axis 3)**	**J3-22**	**Ch B (Axis 3)**
J3-23	**/Ch B (Axis 3)**	**J3-24**	**/Ch I (Axis 3)**
J3-25	**Ch I (Axis 3)**	**J3-26**	**2.6 Vref Output**
J3-27	**GND**	**J3-28**	**+5V Output**
J3-29	**Ch A (Axis 4)**	**J3-30**	**/Ch A (Axis 4)**
J3-31	**Ch B (Axis 4)**	**J3-32**	**/Ch B (Axis 4)**
J3-33	**/Ch I (Axis 4)**	**J3-34**	**Ch I (Axis 4)**
J3-35	**2.6 Vref Output**	**J3-36**	**GND**
J3-37	**+5V Output**	**J3-38**	**Ch B (Ext. Encoder 1)**
J3-39	**Ch A (Ext. Encoder 1)**	**J3-40**	**GND**
J3-41	**+5V Output**	**J3-42**	**Ch B (Ext. Encoder 2)**
J3-43	**Ch A (Ext. Encoder 2)**	**J3-44**	**GND**
J3-45	**Ch B (Ext. Encoder 3)**	**J3-46**	**Ch A (Ext. Encoder 3)**
J3-47	**+5V Output**	**J3-48**	**Ch B (Ext. Encoder 4)**
J3-49	**Ch A (Ext. Encoder 4)**	**J3-50**	**GND**

Table 2-1. P-CNC Connector Pin-Outs, MCP-04 Board

2.2.2. Interface to I/O Module

The connector J2, on the MCP-04 board interconnects to the MCP-R16 I/O module, shown in Figure 2-4 below, using the 44 pin high density D-Sub cable. The cable is wired pin-to-pin from the male plug to the female receptacle. Screw terminal connections are provided for all inputs and outputs on the I/O module. Connect 24 Vdc to the terminals marked +, -, the negative terminal being referenced to the same ground as the PC's power supply. The 24 Vdc supply and the 120 Vac PC power should be switched on and off at the same time. Axis limit inputs are connected to the XYZW and ++++ terminals. The positive terminals are +24 Vdc outputs that connect to normally closed limit switches. The other side of the limit switch connects to the appropriate X, Y, Z, or W terminal. When limit switches are installed, JP1, JP2, JP3 and JP4 on the MCP-04 board must be configured to the ground side (opposite the '+' designator, i.e. closest to edge of PCB).

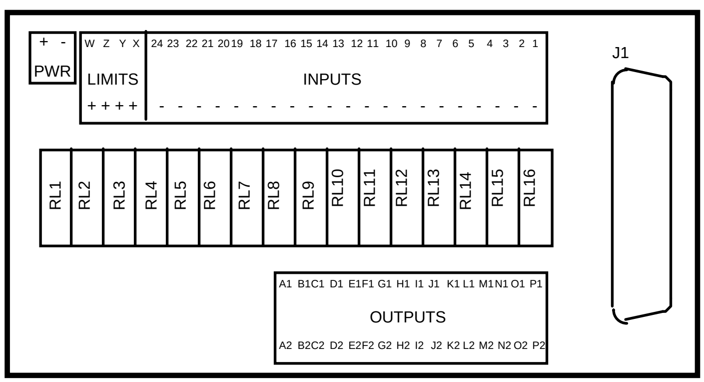


Figure 2-4 MCP-R16 I/O Module Layout

All switch inputs require contact closure at the MCP-R16 input terminals. No voltages should be supplied to the I/O module; the inputs are pulled up to 5 volts internally. The contact closure inputs connect to terminals 1 through 24. The other side of a switch is connected to the '-' negative terminal (GND). The assignment of the inputs is listed in Table 2-1. The relay outputs are routed to the terminals labeled A1/2 to P1/P2. The outputs are rated to 5 Amps for 24 Vdc 120 Vac resistive loads. DC loads are recommended for added reliability.

The two cables coming from J1 and J2 use the color codes described in Table 2-2. J1 is a 26-pin connector that goes to the motor drivers and spindle controller. J2 is a 44-pin connector that mates to the MCP-R16 I/O module. The color codes allow easier pin number identification during installation and check-out.

Pin No.	Color Code
1	Black
2	White
3	Red
4	Green
5	Orange
6	Blue
7	White/Black
8	Red/Black
9	Green/Black
10	Orange/Black
11	Blue/Black
12	Black/White
13	Red/White
14	Green/White
15	Blue/White
16	Black/Red
17	White/Red
18	Orange/Red
19	Blue/Red
20	Red/Green
21	Orange/Green
22	Black/White/Red
23	White/Black/Red
24	Red/Black/White
25	Green/Black/White
26	Orange/Black/White
27	Blue/Black/White
28	Black/Red/Green
29	White/Red/Green
30	Red/Black/Green
31	Green/Black/Orange
32	Orange/Black/Green
33	Blue/White/Orange
34	Black/White/Orange
35	White/Red/Orange
36	Orange/White/Blue
37	White/Red/Blue
38	Black/White/Green
39	White/Black/Green
40	Red/White/Green
41	Green/White/Blue
42	Orange/Red/Green
43	Blue/Red/Green
44	Black/White/Blue

Table 2-2. Cable Color Code Chart

2.3. INTERFACE TO EXTERNAL DEVICES

The interface to external devices is configured using the OEM Setup Utility in combination with the MCP-R16 I/O module. Rarely, is it necessary to add additional I/O or an external PLC. Expansion of the number of I/O points can be accomplished by adding an additional board in the PC or adding an external PLC. Expansion inside the PC has the added benefit of allowing the movement of controlled axis motors by utilizing the programming interface library. A PLC, on the other hand, can only control equipment connected to it. This type of consideration becomes important when a tool change program needs to be developed. Mektronix engineers can develop a custom tool change program for your system or you can write your own using a compiled software language that can build a DOS executable program.

Connections to the MCP-R16 I/O module are shown in Figure 2-4 on the following page. All switch inputs provide contact closures to the MCP-R16. **No voltages should be supplied to the inputs**; they are pulled up to 5 volts internally. 24 volts must be supplied to the +/- terminals on the MCP-R16 to operate the relay outputs. The outputs are reed relays that can be used with AC or DC loads with 24 Vdc recommended. The load current must be held to less than 5 amps at 120 Vac or for 24 Vdc supplies.

All the panel inputs should be used if at all possible. However, if the input is not connected it will default to the open condition which may be fine for some installations. For example, if the feed-rate override switch is not installed, connect jumpers from input numbers 6 and 8 if ground to provide 100% override.

Some of the inputs are optional and are for future expansion. Two panel inputs are reserved for custom applications. The two tool changer inputs are also reserved for custom tool changer executables. The first eight outputs are reserved for predefined M-codes. The next eight outputs (Port B) may be used for a tool changer program or any other custom program. When using Port B, you must keep track of the state of the outputs because there is no way to read the current condition of the outputs.

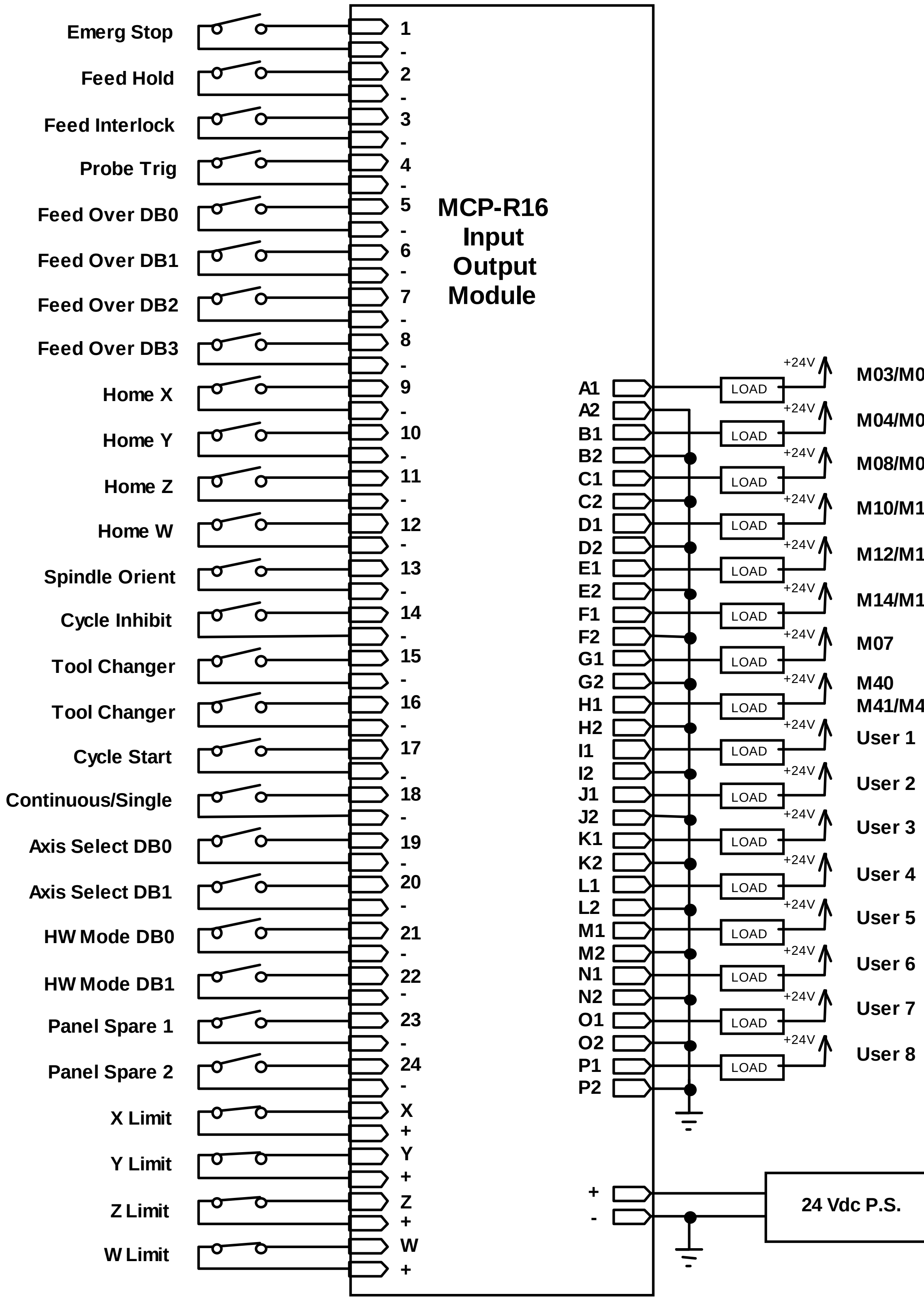


Figure 2-4 MCP-R16 I/O Module Layout

2.3.1. Relay Control Outputs

24 volts DC must be supplied to the +/- terminals on the MCP-R16 to operate the relay outputs. The outputs are reed relays that can be used with AC or DC loads with 24 Vdc recommended. The load current must be held to less than 5 Amps.

M-Codes control the dedicated outputs specified in Table 2-3. The MCP-04 J2 connector pin numbers and control bit names are listed. These pin numbers are available on the MCP-R16 I/O module as terminals A1/A2 through H1/H2.

Bit No.	Conn No.	Term. No.	Relay Closed	Relay Open
OP-A0	J2-1	A1, A2	M03	M05
OP-A1	J2-16	B1, B2	M04	M05
OP-A2	J2-31	C1, C2	M08	M09
OP-A3	J2-2	D1, D2	M10	M11
OP-A4	J2-17	E1, E2	M12	M13
OP-A5	J2-32	F1, F2	M14	M15
OP-A6	J2-3	G1, G2	M07	PULSED
OP-A7	J2-18	H1, H2	M42	M41 GEAR

Table 2-3. Relay Output Control Functions, MCP-04 Port A

Outputs OP-A4 (M12/M13) and OP-A5 (M14/M15) are special case outputs that are interpreted differently in P-CNC. Their outputs may be turned on/off during continuous profiles without causing profile interruptions. Otherwise, the output functions are treated identically as the other control outputs.

2.3.2. Spindle Control

The spindle is controlled using a combination of relay outputs and an analog output voltage to command a speed. Bit OP-A0 and bit OP-A1 together determine spindle operation at output terminals A and B as shown in Table 2-4.

FUNCTION	M-Code	Relay B	Relay A
SPINDLE OFF	M05	Open	Open
SPINDLE CW	M03	Open	Closed
SPINDLE CCW	M04	Closed	Open

Table 2-4. Spindle Control Outputs

A 0-10 volt analog drive signal is provided at J1-15 for controlling the speed of the spindle motor. The P-CNC developer can set the rpm per volt ratio and the speed at which to change from low gear to high gear. An output is provided on the MCP-R16 I/O module that indicates which speed range is currently commanded. Once the commanded RPM exceeds 10 volts the output will switch to the high gear rpm/volt setting defined in the OEM Setup. The output will never be allowed to exceed the high-end spindle speed ratio at the 10 volt output. A velocity controlled spindle drive that accepts a velocity input is required to make use of this feature.

The spindle control should also make use of the spindle CW/CCW relay outputs provided at terminals A1/A2 and B1/B2 as described in Table 2-3. The direction outputs must be used for bi-directional spindle control.

2.3.3. Homing Signals

The homing switches provide the control direction while performing a homing sequence. The homing sequence consists of the following two stages:

1) the axis moves in the direction specified by the control bit (on => opposite) at the OEM Setup homing velocity. The first stage ends as soon as the control bit changes state.
2) the axis then moves in the reverse direction at very slow speed until it backs off of the switch.

3) The axis continues to move and stops when the index pulse is detected. At the end of this stage, the actual position is set to zero or to the defined position specification in the OEM Setup utility.

The homing operation is performed in the Jog screen and must be executed before running a NC program. If power is lost and then reapplied, the current position will be defined as zero. It is important to remember that the OEM homing velocity in combination with the state of the home switch defines the homing direction. It is normally sufficient to place the home switch at one end of axis travel. If you want the axis to home in the center of axis travel you can use an arrangement similar to Fig. 2-5.

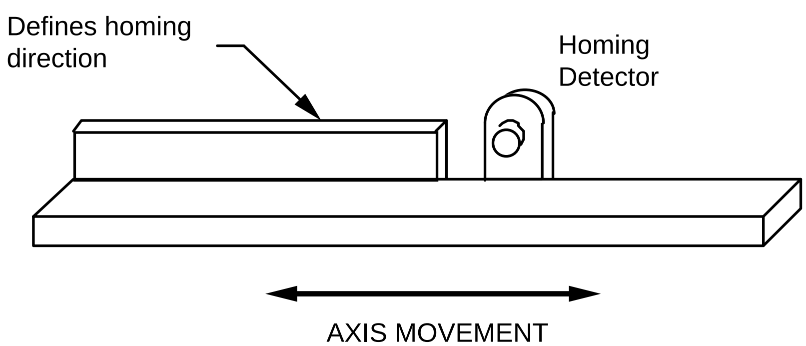


Figure 2-5. Homing Implementation

The homing switch must also be located relative to the motor encoder index pulse. After the home switch is detected, the axes stop and move in the positive direction at a very slow speed. It is important that the index pulse, which occurs once every rotation of the motor, be far enough away from the homing switch to provide repeatable operation. One does not want the homing operation to miss the intended index by the amount of one revolution.

2.3.4. Axis Limit Inputs

When a Limit is triggered, the controller automatically goes into Initialize mode and outputs zero command voltage to the motor drivers. In order to clear the emergency flag, the limit condition must be removed. The MCP-04 has the capability to override the limit condition. P-CNC will notify the operator of a limit condition and will request that the operator hold the

Cycle Start button in until the limit switch is released. Every time the cycle start switch is pushed, the axis direction reverses. There is one limit input provided for each axis which should be triggered at both ends of travel for safety using a N.C. switch. The MCP-04 Limit inputs must be asserted for at least 1 μsec in order to trigger.

2.3.5. Front Panel Switches

The feed rate override interface uses a 16 position binary switch to derive the commanded speed during interpolation moves and handwheel jogging. The feed rate override switch does not affect rapid positioning. Table 2-5 lists the correspondence between the switch, the feed rate percent and the handwheel multiplier.

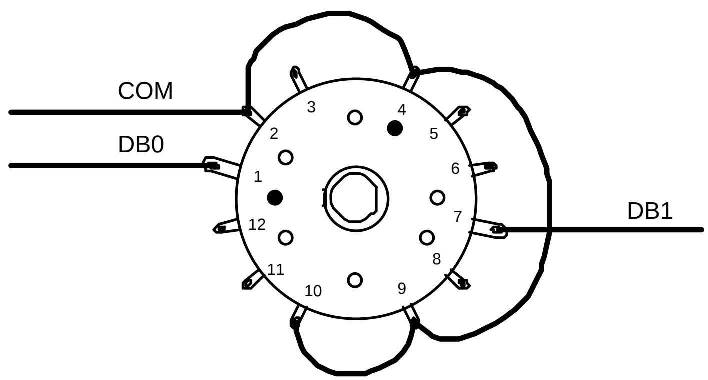
Binary Input	Feed/Vel Percent
1111	150
1110	140
1101	130
1100	120
1011	110
1010	100
1001	90
1000	80
0111	70
0110	60
0101	50
0100	40
0011	30
0010	20
0001	10
0000	0

Table 2-5. Override Switch Ratio Table

The feed hold switch is connected to I/O module and provides immediate deceleration to zero velocity during trajectories when enabled. The trajectory can be resumed while in the pause condition by disabling the feed hold input. A feed rate override switch setting of 0% is functionally equivalent to pause without the display warning. Therefore, it is the operator's responsibility to set the correct feed rate override and verify the switch is not set to 0% for correct operation. No warning is given while the feed rate override switch is set to 0 percent.

A possible feed rate override switch is a Grayhill Series 26 coded rotary switch (26ASD22-01-1-AJN). The feed hold switch could be a simple on/off single pole switch. However, it would be convenient to provide a LED indicator on the pause switch to offer visual feedback when in the feed hold is switched on.

The axis select and handwheel mode select switches need to be in the form of a 2 bit binary output. A two deck, four position switch can be configured as a binary switch as shown in Figure 2-6. Contact closure between common and the data bit represents a binary 1. The axes are selected as X (00), Y (01), Z(10), and W (11). The handwheel selections are .0001 (00), .001 (01), slow jog (10), and fast jog (11).



2-Bit Binary Configuration of Grayhill 71BD30-01-2-AJN

Figure 2-6. Axis Select and Rapid Override Switch

2.3.6. Parallel Port Definitions

The parallel port provides BCD outputs and strobes for the M, S and T addresses in a NC program and a manual mode indicator. The OEM Setup Utility allows these features to be disabled or to set a valid I/O port address. If a printer is required, an additional parallel port can be installed or the serial RS232 port may be used. The parallel port pin-out definitions are given in Table 2-6.

Description	Type	DB-25	Notes
Error Code Bit 3	Input	10	Normally high
Error Code Bit 2	Input	12	Normally high
Error Code Bit 1	Input	13	Normally high
Error Code Bit 0	Input	15	Normally high
Common	Output	18	Ground
Finish Signal	Input	11	Normally high
Common	Output	19	Ground
BCD Bit 7	Output	9	Data bit
BCD Bit 6	Output	8	Data bit
BCD Bit 5	Output	7	Data bit
BCD Bit 4	Output	6	Data bit
BCD Bit 3	Output	5	Data bit
BCD Bit 2	Output	4	Data bit
BCD Bit 1	Output	3	Data bit
BCD Bit 0	Output	2	Data bit
T Address Strobe	Output	17	Active low
S Address Strobe	Output	16	Active low
M Address Strobe	Output	14	Active low
Auto/Manual Mode	Output	1	Manual (Jog) - low
	Output	20	Ground
	Output	21	Ground
	Output	22	Ground
	Output	23	Ground
	Output	24	Ground
	Output	25	Ground

Table 2-6. Parallel Port Pin-out Definitions

The parallel port outputs may be used to provide additional information to an external programmable logic controller (PLC). The PLC can decode the M, S and T addresses, send a feed hold signal to the MCP-04 motion controller and perform its logic functions. To resume execution the PLC releases the cycle inhibit line. This feature is useful for complicated tool changers and to provide additional M-code functionality. All valid M, S and T-code values are output to the BCD port regardless of whether they are supported in the P-CNC program. The T and M address strobe signal timing is given in Figure 2-7. Following the generated T or M strobe signal, P-CNC checks to see if the cycle input line is asserted. P-CNC will stay in the hold condition until it is released or the OEM Execution Time-out has elapsed. An Execution Time-out will halt execution and display a warning message to the operator.

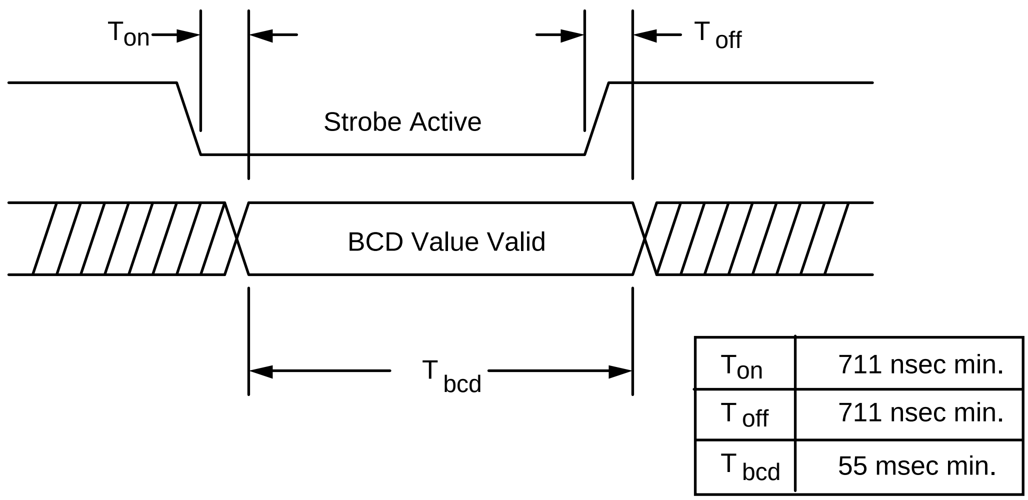


Figure 2-7. Strobe and BCD Value Timing Diagram

The Auto/Manual output indicator is asserted low (Manual) when the operator enters the Jog menu. Otherwise, the indicator is normally high for Automatic mode.

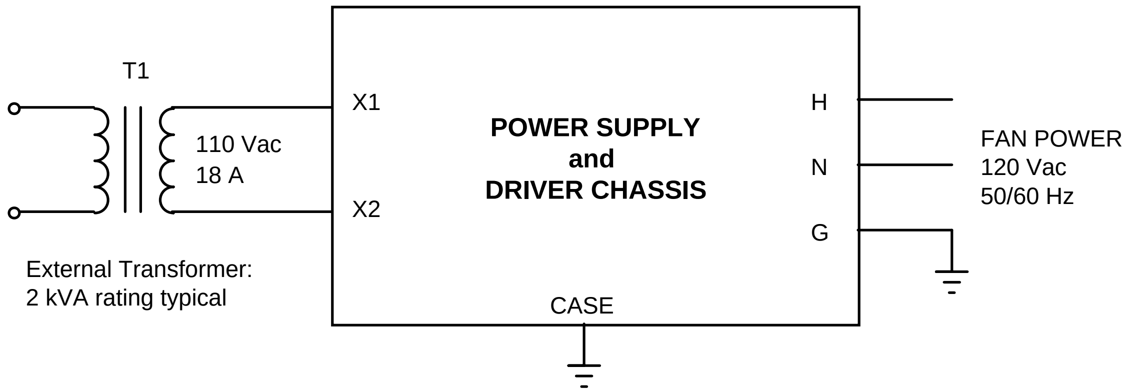
2.3.7. Motors and Drivers

The following motors and associated equipment are required to drive the axes:

- Motors with incremental position encoders
- Motor drivers with power supply
- Power supply for the motor drivers
- Optional spindle motor with velocity feedback amplifier.

The motor drivers must be suitable for driving the selected motors. A current mode (torque) or velocity mode motor driver may be used. It is also possible to use P-CNC with a velocity mode drive by connecting the velocity feedback (tachometer) directly to the motor driver. The driver may use either of the command output formats from the MCP-04 boards; ±10V analog or pulse-width-modulation (PWM). The ±10V command signal will interface directly to the motor driver without external components. Most driver manufacturers provide a differential command input. Connect the differential HI input to the analog ±10V command line at connector J1 and the differential LO input to the common at the same connector. The ±10V command output can source up to 50 mA, allowing the use of cable lengths up to about 25 feet (using 24 AWG stranded wire) provided the input impedance to the motor driver is greater than 10 kΩ and the cable capacitance is less than 300 pF.

The power supply must meet the specification requirements of the motor drivers. An isolation transformer is used to provide a 110 Vac single phase to the power supply rectifier bridge. The resulting bus voltage is around 155 Vdc and must never exceed 180 Vdc when using the Mektronix DR-10A20 and 15A30 motor drivers. Three signal lines are used to connect to the Mektronix motor drivers; the ±10V command and common and the drive enable signal. The drive enable signal is a TTL output that is low (ground) when disabling the drives. This signal keeps the drives in their correct state during power on and off conditions. Otherwise, it is likely that the axes will move under these conditions. Figure 2-8 below shows the typical connections for the drivers supplied by Mektronix.



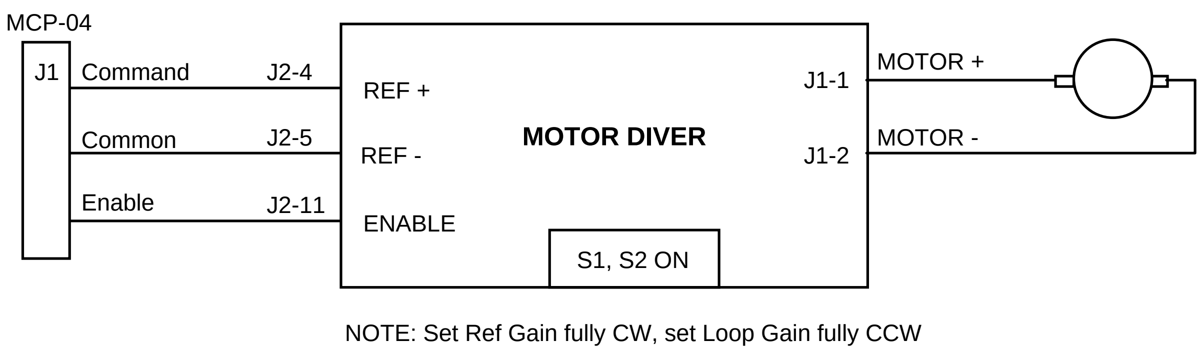


Figure 2-8. Mektronix Motor Driver Configuration

The Mektronix MCP-3A6 four axis PWM drivers are designed for a direct connect to the MCP-04 board. In applications that need less than 6 Amp motor current, this is an obvious choice due to its low cost and reliability in a small DIN-rail mounted module.

2.3.8. Position Encoders

The axis position encoders and the handwheel encoder must output two phase channels in quadrature. They may have either TTL or differential line driver outputs. (The handwheel encoder must use single line TTL encoders.) Differential transmission has the advantage of greater immunity to noise interference than single-ended transmission. For single-ended TTL signals jumper the 2.6 Vref output to the complementary encoder inputs inside the connector shell.

DB-9 Pin No.	Encoder Function
1	+5 Volts Out
2	/Ch A Input
3	/Ch B Input
4	Ch I Input
5	Ground
6	Ch A Input
7	Ch B Input
8	/Ch I Input
9	Vref Output

Table 2-7. DB-9 Motor Encoder Pin-outs

DB-15 Pin No.	Encoder Function
1	+5 Volts Out
2	Ch A, Hdw #1
3	+5 Volts Out
4	Ch A, Hdw #2
5	Ch B, Hdw #3
6	+5 Volts Out
7	Ch A, Spindle
8	Not Used
9	Ch B, Hdw #1
10	Ground
11	Ch B, Hdw #2
12	Ground
13	Ch A, Hdw #3
14	Ch B, Spindle
15	Ground

Table 2-8. DB-15 Handwheel and Spindle Pin-outs

An incremental position encoder produces Channel A and Channel B square waves are offset by 90 degrees, making it possible to determine direction and to increase the resolution by a factor of four. Hence the term quadrature encoder, which produces 4000 counts per revolution for a 1000 line rotary encoder. The required resolution and accuracy of the encoder will depend on the accuracy of your machine. The MCP-04 board can position to ±1 count of the encoder. Check the mechanical configuration and the accuracy specifications of the shaft encoder. Select an encoder resolution that exceeds the desired accuracy of your machine. A good rule of thumb is to select an encoder resolution about 10 times the mechanical accuracy. Another way to select the motor encoder is to set the maximum rapid speed to give 127 quadrature encoder counts per sampling rate. This will give the best resolution possible for the position feedback.

P-CNC requires the axis resolutions to be equal for axes that need to profile in circular interpolation. Select an encoder and mechanical transmission that results in the same number of encoder counts per length traveled. The handwheel encoder resolution is not critical and may be scaled in the OEM Setup Utility.

The handwheel encoder is typically 100 pulses or 400 quadrature counts per revolution. A detent handwheel will also provide precise positioning. The handwheel select switch provides two positioning modes and two velocity modes. The switch selects .0001, .001, low and high velocity modes. A detent handwheel provides 100 detents per revolution and the operator can select the multiplier to precisely position the machine tool. The velocity mode uses the feed rate override switch to move at varying speeds. The velocity mode is useful for quickly locating part zero.

2.3.9. Tool Changer Control

P-CNC outputs a strobe signal and the BCD tool number value to the parallel port if enabled. This output may be decoded by a PLC to perform a tool change sequence. The cycle inhibit signal must be held low during this time to keep P-CNC from continuing to execute. P-CNC also allows a custom tool change program to be called that executes as a child process within P-CNC. If a tool changer is not used, do not type in anything in the OEM Setup Utility under item Tool. P-CNC calls the tool changer program specified in the OEM Setup Utility with the new tool number and old tool number passed as arguments. The call is equivalent to issuing from the DOS prompt:

c:\p-cnc> **toolchg 12 6**

for changing from tool number 6 to 12. The first argument is the new tool number and the second is the old tool number. The first time a tool is specified the old tool number will be zero and can signify the need to perform a turret homing sequence in the tool change program. When the tool number is specified as zero, the tool change program should return the current tool back into it's correct holder. P-CNC does not call the tool change program until a T address is specified within a NC program or commanded in the Jogging menu. A M06 code should only be added for manual tool change prompts.

If the tool changer program is issued an invalid tool number, the program exits with an error code. The source code for the default tool changer program is given in file "toolchg.c". Use this as an example to write your own tool change program in C or QuickBASIC. You can use the Mektronix Programming Interface Libraries to build your program. It is very important that the MCP-04 board's configuration not be changed nor any of its settings. As an example, if the axis positions are moved and not returned to their original place, P-CNC will not continue to run correctly after a tool change operation. Two tool changer inputs are reserved and any of the 8 user outputs may be used for tool changer operations. An additional 24 bits of I/O is available as an option if needed.

2.4. OEM SETUP UTILITY

The OEM Setup Utility provides system integrators a way to configure P-CNC for a particular machine and end-user. This utility should only be used by qualified technicians that understand the machine specifications and the detailed operation of the MCP-04 board and P-CNC application program.

The OEM Setup Utility can be invoked with the following options.

oem [/e] [/m] [/s:<*setup-file*>] [/a:<*ascii-file*>]

Switches:	/e	change to English system
	/m	change to metric system
	/s:<*setup-file*>	load/save substitute file name for OEM.SET
	/a:<*ascii-file*>	generate documentation (ascii) text file

Normally, the OEM setup program will be invoked without any of the switches described above. If the OEM setup file cannot be found or it is not in the correct format, you are given the choice to load the default system parameters. The **/e** and **/m** options are required only when changing from the previous measurement system. The only difference between the two is that distance is described in either inches (imperial) or millimeters (metric). The **/s**:<filename> option allows other filenames to be loaded and saved instead of the default OEM.SET binary file. For system integrators retrofitting several different types of machines, it is convenient to store each setup file (∗.set) with a descriptive project name. Remember to change the name back to “oem.set” before distributing the setup file to the customer. The **/.a**:<filename> option produces an ascii (text) file with all the current OEM setups. Use this option to keep records of the factory settings used for different machines. You can examine this file with any text editor or send the file to a printer for a hard copy.

Run the OEM setup utility by entering the correct directory and typing "OEM" (other command line arguments are optional). The following command line initiates the OEM Setup Utility with switches for metric input and to store the settings in a file named “mill1.txt”.

oem /m /a:mill1.txt

The following command line runs the OEM Setup Utility and uses the name mill1.set as the setup file.

oem /s:mill1.set

The OEM main screen is shown in Figure 2-9 with the Axis Specification option highlighted.

Figure 2-9. OEM Setup Utility Screen

To immediately activate any of the setup options, simply type its label number. To select an option, type the first letter of the option or use the cursor keys to highlight it. A carriage return [CR] will activate the selected option.

2.4.1. Setting Axis Specifications

Hit F1 function key to activate the Axis Parameter setup panel. Each axis is selected using the numbers ***1*** through ***4***. Cursor to the parameter you want to change and hit Enter to begin typing the new value. The input editor allows for inserting and overwriting the existing text. Finish by hitting Enter again to record the value. If an error is detected a warning will be generated and the value will not be modified. If you decide not to change the original value hit the [Esc] key. Although the displayed resolution of the setup parameters is fixed, the internal storage allocation is for 32-bit floating point numbers. Therefore, display truncation of the parameter values does not change the actual values stored with the input editor.

Each axis parameter must be set to the physical specifications of the machine being controlled. Once set by the system developer, these settings should not need to be changed. The following describes how these parameters should be set.

(1) Axis Address: (linear, rotary, not used)

The axis may be configured with the following axis addresses: X, Y, Z, W, C and not used. The C-Axis specification converts to degrees otherwise, the specification is for inches or millimeters. If an axis is not used you should disable it by selecting the not used option.

(2) Axis Resolution: (counts per inch or mm)

This parameter is set based on the encoder specification. The default is 5,000 counts per inch. To calculate this value, divide the encoder resolution (cnt/rev) by the lead screw pitch (inch/rev). Axis resolutions must be the same for axes using circular interpolation.

> **Note:** The MCP-04 board interfaces to quadrature incremental encoders. For this type of encoder, the number of counts is 4 times the number of slots in the encoder wheel.

(3) Backlash Compensation: (inch or mm)

P-CNC keeps track of the direction of motion for each axis. When a reversal occurs, the lead screw is adjusted by the amount specified. Backlash compensation is applied during rapid and profiling moves. After entering a value for backlash, test the amount by repeating a back and forth movement while using a dial gage to assure repeatability. Also, verify that the rapid acceleration is slow enough to keep the axis from moving off the lead when coming to a stop. Keep in mind that it is always better to eliminate backlash using anti-backlash nuts than to resort to electronic methods.

(4) Range Maximum: (inch or mm)

This is the maximum absolute travel position allowed for the selected axis. The maximum range of the axis is relative to the machine home position and is not affected by changes in the work coordinate system set by the user.

(5) Range Minimum: (inch or mm)

This is the minimum absolute travel position allowed for the selected axis. The minimum range of the axis is relative to the machine home position and is not affected by changes in the work coordinate system set by the user.

(6) End-point Error: (inch or mm)

The required precision for the selected axis at the programmed end-point of a move must meet this specification before execution will continue. A warning will be displayed when the End-point Precision specification has not been satisfied.

(7) Lag Trigger: (inch or mm)

This specification determines the maximum following error allowed during motion before an error message is initiated. The maximum following error should be the worst case error at the highest rapid speed, since the error is likely to increase with an increase in velocity. The purpose of this specification is to detect collisions and is not an accuracy measurement.

(8) Rapid Velocity: (inch/sec or mm/sec)

Rapid positioning (G00) is carried out using trapezoidal profile control mode. Each axis moves independently to the final position without regard to the other axes. The maximum velocity for each axis is set by the Rapid Velocity parameter and can not be changed within P-CNC. Possible values range from 1,000 to 127,000 encoder counts per second (assuming an update rate of 1 kHz).

(9) Rapid Acceleration: (inch/sec^2 or mm/sec^2)

The rapid acceleration (G00) is defined for each axis as a linear rate of change in velocity. This value should be adjusted to obtain smooth acceleration and deceleration in rapid and incremental jogging. Possible values range from 3.91×10^3 to 128×10^6 encoder counts per sec^2 (assuming an update rate of 1 kHz).

(10) Home Velocity: (inch/sec or mm/sec)

This value is used when automatic homing to switches is enabled and during recover from software range limits. Positive and negative values may be used to define the home direction. The home input logic level is also used in determining the initial home direction. If the home input is high, the velocity specified in OEM is complemented.

(11) Home Offset: (inch or mm)

The home location may be set to any desired position using this specification. Normally, the home location is set to zero.

(12) Filter Gain: (0 to 63.75)

This is the filter gain and should be kept as high as possible without causing instability. The user input is rounded to the nearest possible setting on the MCP-04 board. The Exerciser program command 'tune_filter' assists in determining the correct filter parameters experimentally. Please consult the online help facility in the Exerciser by typing 'help' from the command line for a more detailed discussion.

(13) Filter Zero: (0. to 0.960375)

This defines the zero for the digital filter. When the zero approaches 1, the system will become overly damped and sluggish. Set the zero to obtain maximum stiffness (smaller value) without any oscillations of the servo motor. The typical range for the filter zero is between 0.8 to 0.95. The user input is rounded to the nearest possible setting on the MCP-04 board.

(14) Filter Pole: (-0.9060375 to 0.)

This defines the pole for the digital filter. The pole does not normally need to be adjusted. Increasing the value to -.9 will increase the high frequency stabilization. The user input is rounded to the nearest possible setting on the MCP-04 board.

2.4.2. Miscellaneous Specifications

If Miscellaneous Specifications is selected from the OEM main screen, the panel is displayed as shown in Figure 2-10. The values shown in this figure are the default values set when an OEM.SET file is first produced. Use the cursor keys (↑↓←→) to select a parameter and the [SPACE] bar allow you to toggle the Homing and Specs setting.

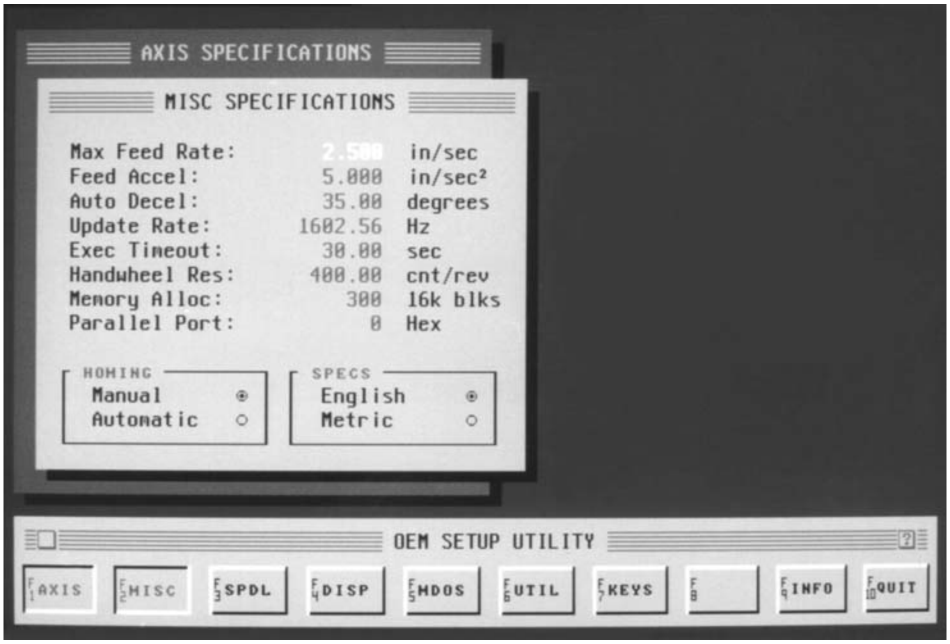


Figure 2-10. Miscellaneous System Specifications

Hit [F2] to enter the Miscellaneous Specification panel and then use the cursor keys and input editor to configure the following parameters.

(1) Max Feed Rate: (inch/sec or mm/sec)

The maximum feed rate parameter specifies the largest value for the F address register while executing a NC program. If a user program specifies a value that exceeds the specified maximum feed rate, the Max Feed Rate setting will be in effect. The Max Feed Rate parameter also specifies the feed and rapid positioning velocity when running in Dry Run mode. The actual programmed feed rates and rapid velocity settings are ignored when the Dry Run option is set ON in the CNC Setup menu.

(2) Feed Acceleration: (inch/sec^2 or mm/sec^2)

The acceleration to be used for all interpolation modes is specified as Feed Accel. The feed acceleration is along the profile specified in linear, circular and spline interpolation. Feed rate override, pause and exact stop commands also use this parameter setting. P-CNC will automatically reduce feed-rates around small radius corners so that the acceleration does not exceed this setting.

(3) Auto Deceleration: (degrees)

Specifies the maximum angle before inserting an auto exact stop in a profile. When the vector angle of two profile segments exceeds this value, the axes will decelerate before continuing with the programmed path.

(4) Update Rate: (Hz)

The update rate is the rate at which interpolation occurs on all three axes of the MCP-04 board. This value should be kept at the default 1 kHz setting unless there is a good reason to change its value. Possible settings range from 500 Hz to 2 kHz.

(5) Execution Time-out: (sec)

During homing and BCD strobe sequences, the estimated time to complete the task is calculated. The execution time-out value will be added to the estimated time before an error message will be issued to the operator. The execution time-out setting is used for other tasks that require a positive response.

(6) Handwheel Resolution: (cnt/rev)

The handwheel resolution parameter specifies the number of encoder counts (4x pulse/rev). This value is used for positioning handwheels that have mechanical detents. When in the handwheel increment mode, each graduated dial mark indicates a relative position depending on the handwheel select switch. An inexpensive encoder can be used for velocity jogging. Make sure to test the full range of handwheel jogging speeds with the handwheel select switch and feed rate override switch to assure that the handwheel operates satisfactory.

(7) Virtual Memory: (16k blocks)

The virtual memory manager requests XMS, EXP, EXT and half the available hard disk space, in that order. The virtual memory is allocated in 16k byte blocks. So, the number entered in the OEM Setup must reflect the number of memory blocks, not the total memory size required. The default setting of 50 memory blocks (16k bytes each) provides 819,200 bytes of storage for the NC text and internal structures. The default setting is a good typical setting for most computers. If you anticipate extremely large part program files and have the necessary hardware, you should increase this setting to about double the size of your largest part program.

(8) Parallel Port: (Hex)

P-CNC's capabilities can be expanded by using a spare parallel port on the host computer. The address of the parallel port must be entered as a hexadecimal number. To determine what the correct address is, you can use the MSD.EXE utility that comes with DOS 6.0. The three most common addresses are:

Parallel Port	Address (Hex)
LPT1:	378
LPT2:	278
Mono/Printer Adapter:	3BC

Addresses less than 100H will disable P-CNC output to the parallel port.

(9) Home to Switch: (Auto/Manual)

The homing sequence is enabled using the automatic home specification. Manual homing is accomplished using a MDI editor. When homing switches are installed, automatic homing should be enabled. You select between the two options by highlighting the Homing title and hitting the SPACE bar to toggle between the two options. NC programs are not allowed to run without first homing the machine either manually or automatically.

(10) Measurement Specs: (in or mm)

The default measurement and OEM input specification is determined by this switch. Changes will not be effective until the OEM utility is restarted.

2.4.3. Spindle Specifications

(1) Spindle Max Lo-Gear: (rpm)

The spindle speed is controlled by an analog output signal that is proportional to the rpm. The analog output signal can vary from 0 to 10 volts. Set the Spindle Max Lo-Gear parameter to the speed (rpm) that pproduces a 10 volt signal. I the spindle speed is commanded to a higher value, the spindle gear output will go high and the ratio will be converted to the Spindle Max Hi-Gear setting described next. Set both the Lo-Gear and Hi-Gear specifications to the same value when only one speed range is used. A zero setting is invalid and will cause a run-time execution error.

(2) Spindle Max Hi-Gear: (rpm)

The spindle speed is controlled by an analog output signal that is proportional to the rpm. The analog output signal can vary from 0 to 10 volts. Set the Spindle Max Hi-Gear parameter to the speed (rpm) that produces a 10 volt signal. This ratio is used by P-CNC when the commanded spindle speed exceeds the Spindle Max Lo-Gear setting described previously. If only one gear ratio is being used, set this parameter to the proper ratio and enter the same value for the Spindle Max Lo-Gear specification. A zero setting is invalid and will cause a run-time execution error.

(3) Spindle Orient Speed: (rpm)

Set the speed that the spindle should run when the spindle is to orient itself to a input marker. The speed should be set as slow as possible with reliable operation.

NOTE: Laser applications are supported when the spindle jog resolution is specified less than 1. Since the laser output is specified in percent vs. rpm, a value of .01 gives 1% increments. The display is changed to laser settings, M03 is PWR, M04 is ON and M05 is OFF.

2.4.4. Display Specifications

(1) Border Color: (toggle selections)

Set the border color to one of the options by hitting the space bar to make selections.

(2) Text Color: (toggle selections)

The text background color is set for the NC program execution panel. The various options will be used only in the Run screen's NC program panel.

(3) Relay Names: (RL3 - RL8)

The relay outputs can be given a name that appears in the Run Status Panel. Names are restricted to a maximum of five characters in length.

(4) File Extension: (DOS extension)

The file extension may specified differently from the default (*.CNC) for loading part programs. It is recommended that the default be used unless there is justification for changing the extension. As with all DOS files, the extension is from one to three letters long and may be either lower or upper case. Do not include the dot (.) or any other extra characters.

2.4.5. DOS Specifications

(1) On-line Editor:

Your favorite text editor may be used to make changes to the NC program that is currently loaded. Type in the editor's file name with an arguments (/options) in the space provided. It is not a good idea to use a word processor to create and edit your NC program unless the file can be saved in text only format. Files created by a word processor contain special format information which will not be recognized by the P-CNC interpreter.

(2) Tool Changer Program:

Type in the name of the executable file to be used by P-CNC when a tool change is requested. If you do not have a tool changer program, leave this field blank. Custom tool changer programs can be easily developed for most machine tools as a software add-on utility. Nothing should follow the program name (i.e. tool number arguments are automatically inserted).

(3) M-code DOS Escapes:

M90 through M95 are miscellaneous codes that can be used to perform DOS shells to other executable files or commands. M90 - M95 M-codes allow an easy way to expand the capabilities of P-CNC to include more sophisticated I/O. It is not advisable to modify the status of the MCP-04 board configured with P-CNC because these changes will not be known by the P-CNC program. Type in the DOS executable as if they were typed directly from the DOS prompt and include any slash options required.

The DOS Shell M-codes may also pass specified address values to the child program. Any valid address (A-Z) may be specified using the `&' control character. For example, the following M90 example calls the program pallet with the current feed rate and W address:

M90: pallet &W &F

which turns into,

pallet 10.125 20

at the DOS command line (assuming the W10.125 and F20 addresses are already set).

2.4.6. DOS Utilities

Five DOS utilities can be configured to run in P-CNC. Each utility is configured for Menu Text, Command Line, Arguments, and Initial Directory. If no program name is defined, P-CNC executes a DOS shell command line prompt.

The default settings provide access to DOS manuals, the MCP-04 Exerciser, a sample calculator, the DOS prompt, and the DOSShell program. A CAD/CAM program could also be configured to enhance the capabilities of the P-CNC system.

2.4.7. Keyboard Macro Definitions

Keyboard macros allow the assignment of multiple keystrokes to a single function key or potentially a discrete switch. The keystrokes are assigned as though the operator had typed them individually. The keyboard macros become effective while in any main screen such as Jog or Run. They have no effect during other sub-menu operations. The first keystroke is usually an ESC key so that the P-CNC Run screen is reached regardless of the current screen status. Then a sequence of keystrokes can be referenced starting from the Run screen. ALT

+ (F1-F10) function keys are assigned for up to ten keystrokes. The key codes are entered as hexadecimal values consisting of the scan code as the Most Significant Byte (MSB) and the ASCII code as the Least Significant Byte (LSB). Reference Table 2-9 given below for a list of the most common key codes.

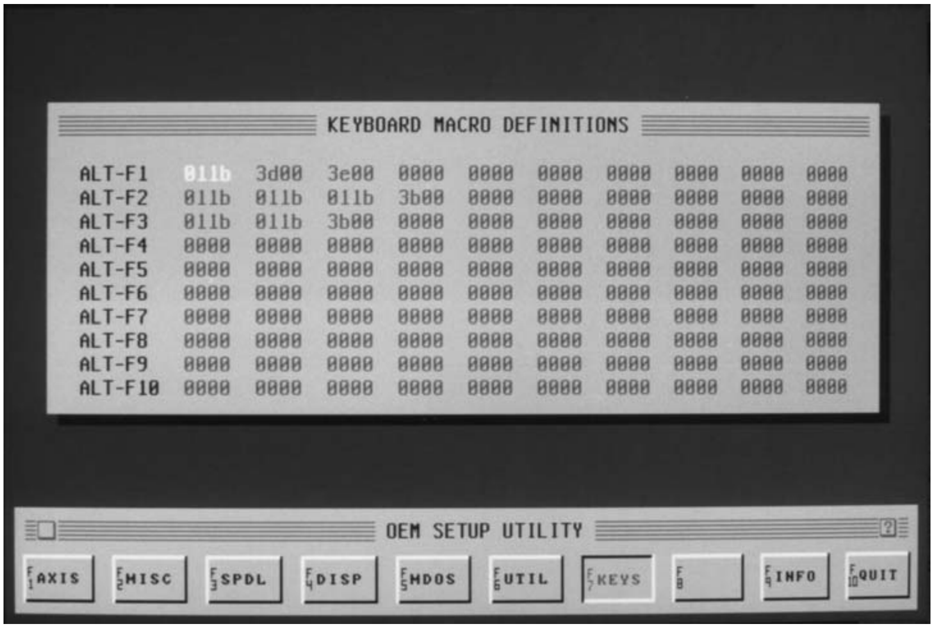


Figure 2-11. Keyboard Macro Definition Assignment Panel

Keystroke	Code	Keystroke	Code
INSERT	5200	5	0635
PLUS	0D2B	6	0736
MINUS	0C2D	7	0837
SPACE	3920	8	0938
ESC	011B	9	0A39
PGDN	5100	A	1E41
PGUP	4900	B	3042
HOME	4700	C	2E43
END	4F00	D	2044
BS	0E08	E	1245
ENTER	1C0D	F	2146
UP ARROW	4800	G	2247
RT ARROW	4D00	H	2348
LT ARROW	4B00	I	1749
DN ARROW	5000	J	244A
PERIOD	342E	K	254B
F1	3B00	L	264B
F2	3C00	M	324D
F3	3D00	N	314E
F4	3E00	O	184F
F5	3F00	P	1950
F6	4000	Q	1051
F7	4100	R	1352
F8	4200	S	1F53
F9	4300	T	1454
F10	4400	U	1655
0	0B30	V	2F56
1	0231	W	1157
2	0332	X	2D58
3	0433	Y	1559
4	0534	Z	2C5A

Table 2-9. List of Common Keyboard Codes

2.4.8. OEM Information

The current OEM licensing information is displayed and may be modified by a qualified installer. The P-CNC software is licensed for a single MCP-04 board with the given serial board number. The title, company, address and phone number may be modified from within the information panel.

P-CNC Version 2.06

2.4.9. Utility Programs

The utilities described in this section can be used to assist in system debugging and verification. The Check utility verifies that the computer is communicating properly with the installed Mektronix MCP-04 control board. The Exerciser program allows the integrator to interactively send commands to the MCP-04 board. The exerciser can also monitor the axis positions, controller status and current I/O conditions. The programming interface library is provided for developing custom tool changer programs and other specialized functions. The high-level programming library allows the tool change program to have more capabilities than if an external PLC was used. However, an external PLC can also be used if required by interfacing to the PC's parallel port.

2.5. MCP-04 LOW-LEVEL SYSTEM UTILITIES

The utilities described in this section can be used to assist in system debugging and verification. The Check utility verifies that the computer is communicating properly with the installed Mektronix MCP-04 control board. The Exerciser program allows the integrator to interactively send commands to the MCP-04 board. The exerciser can also monitor the axis positions, controller status and current I/O conditions. The programming interface library is provided for developing custom tool changer programs and other specialized utilities. Using the high-level programming library allows the tool changer program to have more capabilities than using an external PLC. However, an external PLC can also be used if required by interfacing to the PC’s parallel port.

2.5.1 Check Board Utility

The Check program is used to verify that the MCP-04 board is working properly and that the communication to and from the PC is reliable. IMPORTANT: Disable the motor drives before operating this utility, otherwise, the axis motors will move during the test. Follow the following procedure to verify that the MCP-04 board is working correctly. Four axis boards will need the MCINIT environment set to "4:330;" either in the AUTOEXE.BAT or from the command line. Type in the following command:

set MCINIT=4:330;

Change to the directory where 'check.exe' is installed. Run 'check.exe' from this directory. The program first comes up with a screen that detects the assignment and the number of axes detected. The number of axes defaults to four but can be configured for more using a DOS variable called MCINIT. Type 'y' to accept and continue with the program.

From the main menu of Check, type '1' to enter the Diagnostic tests. The first axis to test is axis one. Type 'a' to automatically test all the board registers for the axis. All the register names should change from lower case to upper case. If any register blinks than an error was detected. The "status" register will give an error if a limit or E-stop condition is detected. Type 'n' to go on to the next axis. Repeat the same procedure for all the installed axes and return to the main menu. Type '0' to return to the DOS prompt.

When you test a new computer a more stringent test should be run at least once. Select '2' to run the Burn-in test. The test will run the same but 100 times longer. You should notice when the test reaches the "status" the count will decrement down from 100 at a periodic rate. If the count does not decrement to zero an error was detected. Return to the main menu and type '0' to get back to the DOS prompt.

2.5.2 Command Exerciser Utility

Initial System Checkout

This section describes how to close the control loop around a servo motor and how to adjust the digital compensator using an experimental method. A general procedure for establishing the proper feedback polarity is also given.

Configuring P-CNC

The control loop may be closed after assembling all the necessary components in the control system. Make cables (as per instructions given in Section 2) to interface the motor drivers and the incremental quadrature encoders to the MCP-04 board. Connect these cables to the appropriate places when power is off. Disconnect the motor shafts from any mechanical transmission so that they are free to turn continuously. The control loop is closed when the position feedback is present from the quadrature encoders. Follow the procedure given below to verify that the control loop is functional.

Step 1. During installation make sure power is off to all equipment and then turn the power on to only the personal computer. Turn power on the motor drivers and adjust out any amplifier input offset so that the motors remain stationary.

Step 2. Run the Exerciser and turn the monitor on at the dot prompt by typing:

ex
. monitor on

The monitor screen will display information on all axes configured in the DOS environment variable. At this point, all axes should be in "INIT" mode and the status should display E0H.

Step 3. Turn the motors/encoders with your hand and verify that all the position encoders are working by watching the "act_pos" change on the display.

Step 4. To establish the correct polarity for closed-loop control, output a small positive voltage command to the motor drivers by issuing either

. **motor_com = 83**
83 (or)
. **pwm_com = 3**
3

depending on which command output format is being used. If the motor does not turn, you should increase the output command slightly. The motor should turn in the positive direction. Verify this by observing the "act_pos" display increasing in value. If the actual position is decreasing in value, then the encoder CHA and CHB inputs need to be reversed.

Step 5. The final step is to begin servoing the motor by entering position control mode. There are four parameters that can be varied for each digital compensator; the Gain, Pole, Zero, and Sample Frequency. Next, before closing the control loop, set the filter Gain to a minimal value and the Zero to a large value for each axis.

. **gain = 2**
2.0
. **zero = .90**
0.89843750

Now the position loop may be closed typing the following from the Exerciser.

. **enter_ctl_mode**
control mode entered: (axis 1)

CAUTION

There is the possibility that the position feedback is positive which could cause the motor to run away when closing the control loop. Make sure that if this condition does exist, it will not cause any damage.

At this point there are two possibilities:

I. The motor runs away after any small disturbance and continues to turn at full speed. This indicates that the position feedback is positive and should be reversed. This condition is easily corrected by switching the Phase A and the Phase B encoder outputs.

II. The motor remains in the same position even after a disturbance. If you turn the motor shaft and let go, the motor will return to the same position. This indicates that the control loop has negative position feedback, as required.

At this point, the control loop gain is very low and the motor will have little restoring torque. Gradually increase the gain by repeatedly commanding the Gain register to a higher value. Send the Exerciser command:

. **gain = <N>**

where N is the decimal value of gain.

If the motor begins to vibrate, reduce the gain slightly. The default value for gain is 16 decimal when the controller is first powered or issued a software reset. The objective is to increase the gain as high as possible without causing oscillations. The Exerciser utility 'tune_filter' can be used to assist in selecting the best filter parameters. Type

. **tf**

and then F9 to begin.

While the motor is stepping back and forth, the filter parameters Gain, Pole, and Zero can be changed for the best response. The pole term (POLE) does not significantly affect the system response unless the system has high frequency resonance. Initially leave POLE to the default value. Start with a low gain, (GAIN < 10) and increase until the motor begins to oscillate and then reduce GAIN to a safe margin. The Zero may be adjusted upward to give more damping, if necessary. The goal is for the motors to jump as fast as possible from one position to the next without oscillating while at stand-still. Type F10 to return back to the Exerciser prompt.

The default mode upon applying power to the MCP-04 board is Initialize (Init) mode. While in this mode, the user should program all the necessary registers before executing a control mode. Motor command outputs can be issued while in the Initialize mode to check for correct open-loop operation. The following procedure tests the analog and PWM output pins and should be performed while the motor drivers are turned off. The ±10V analog command signals are available at connector J1.

. **init**	put in Initialize mode
initialize mode: (axis 1)	
. **motor_com = ff**	outputs 10 volts to ±10 V command pin

FFH	
. **motor_com = 0**	outputs -10 volts to ±10 V command pin
00H	
. **motor_com = 80**	outputs 0 volts to ±10 V command pin
80H	
. **pwm_com = 50**	50% duty cycle to PWM pulse pin
50	and low level to PWM sign pin
. **pwm_com = -50**	50% duty cycle to PWM pulse pin
-50	and high level to PWM sign pin

The user should set all compensation parameters prior to entering a control mode. All control modes use some part of the digital compensator which must be set to the values determined in Section 3.2.

. **gain = N**
. **zero = N**
. **sample_freq = N**

Where N is user selectable. Once the digital compensator parameters are set, Position Control mode may be entered. Put each axis in Initialize mode and turn power on to the motor drivers. Enter the values in the OEM setup utility.

Position Control Mode

Position Control mode servos on the current commanded position so that the controller tries to maintain zero error between the 'com_pos' and 'act_pos' registers. When entering from Initialize mode the command position is set to the actual position. The following gives an example of how to command a step position change to the controller.

. **init**	enter Initialize mode
initialize mode: (axis 1)	
. **act_pos = 0**	specify current position as zero
0	
. **enter_ctl_mode**	enter Position Control
control mode entered: (axis 1)	

. **?act_pos**	find current position
0	
. **com_pos=200**	move 200 counts as fast as possible
200	
. **?act_pos**	check new position
200	

All the other control modes are executed from Position Control mode. The provided software does not allow any of these modes to be executed directly from Initialize mode.

Trapezoidal Profile Control is used to profile to a final position at a specified velocity. The motion profile uses the acceleration register to ramp up to the commanded maximum velocity and to ramp down to the final position. Use the example below to initiate a velocity profile that looks like a trapezoid.

. **init**	enter Initialize mode
initialize mode: (axis 1)	
. **act_pos = 0**	specify current position as zero
0	
. **enter_ctl_mode**	positions control mode entered
control mode entered: (axis 1)	
. **accel = .02**	specify a small acceleration
0.01953125	
. **max_vel = 10**	command maximum velocity
10	
. **final_pos = 200000**	final position set to +200,000 counts
200000	
. **go_tp_ctl**	execute Trapezoidal Profile Control
trapezoidal profile control: (axis 1)	
. **?status**	check status (profile flag set)
D0H	repeat until motion complete
. **?status**	trajectory complete (position control)
C0H	
. **?act_pos**	get actual position
200000	

The external Ports A and B can be tested using the Exerciser as shown below.

. **pa = 55**	output 55H to Port A
55H	
. **?pb**	read in 8-bit value from Port B
00H	
. **?pa**	read in 8-bit value from Port A
00H	

It is also possible to read in an additional quadrature encoder besides axis encoders on the MCP-04 board. Encoder input 1 is the normal handwheel input for the P-CNC software.

. **?e1**	read external encoder
200445	
. **e1 = 0**	clear external encoder count
0	
. **?e1**	read external encoder
0	

The External DAC outputs the spindle or laser voltage from the MCP-04 board. The DAC output is normally configured for 0 to 10 volt operation where 0H corresponds to 0 volts. To send an 8-bit hexadecimal number to the DAC, use the following commands.

. **da = 0**	set external DAC to 0 volts
00H	
. **ext_dac = 80**	set external DAC to +5 volts
FFH	
. **ext_dac = FF**	set external DAC to +10 volts
00H	

The limit inputs can be tested by connecting external switches as previously described. The following procedure can be used to verify proper operation.

. **enter_ctl_mode**	enter position control mode
. control mode entered: (axis 1)	

. **clr_emergency**	clear emergency flags
emergency flags of axis 1 cleared	
. **echo long**	set echo format to *long*
. **?status**	get current status

status = E0H: no limit
no stop
not in initialize mode
not in trapezoidal profile
commutator count quadrature
commutator 3-phase
pwm sign reversal off

. **int_vel = -20**	set integral velocity
command integral velocity (axis 1): -20	
. **go_iv_ctl**	begin velocity profiling
integral velocity control: (axis 1)	

While the axis is profiling in Integral Velocity Control mode, trigger the corresponding Limit input. The axis should immediately be disabled while the axis enters Initialize mode. Release the Limit input and continue with the procedure.

. **?status**	get current status

status = E0H: limit triggered
no stop
in initialize mode
not in trapezoidal profile
commutator count quadrature
commutator 3-phase
pwm sign reversal off

. **clr_emergency**	clear emergency flags
emergency flags of axis 1 cleared	
. **enter_ctl_mode**	enter position control mode
. control mode entered: (axis 1)	
. **echo short**	set echo format to *short*

Exerciser Commands

The Exerciser is an interactive tool that allows commands to be issued to the MCP-04 motion controller boards. The AXIS command selects which global axis to which commands are sent, since only one axis can be interrogated at a time. The ECHO command determines the output display, in response to a user command, in either *long*, *short* or *off* format. Commands can easily be repeated by pressing the [F3] key and then the [CR] (carriage return). Any DOS executable may also be invoked without leaving the Exerciser via a temporary DOS escape.

Commands to the Exerciser can be issued directly from the keyboard under the Exerciser prompt ('.') or from an ASCII text file. Cases (upper or lower) are irrelevant. Namely, a command may be in all upper cases, all lower cases, or a combination of upper and lower cases.

A Exerciser command uses one of the following two formats:

(1) . ? <variable>
(2) . <variable> = <value>

The registers and I/O ports on MCP-04 boards and certain system parameters are referred by *variables* in the Exerciser. A variable represents a piece of storage whose value can be set and/or retrieved. For example, the digital filter parameter *gain* is a variable. Thus, it can be referenced by the first two command formats. Try the following example:

. **gain = 21.25**	set GAIN to 21.25
21.25	echo from Exerciser
. **?gain**	what is GAIN ?
21.25	echo from Exerciser
. **gain = 6.50**	set GAIN to 56.50
6.50	echo from Exerciser
. **?gain**	what is GAIN
6.50	echo from Exerciser

While commands in general are for setting or getting register settings, other commands perform auxiliary functions and have unique formats. For example:

. **reset**	reset current axis (axis 1)
reset: (axis 1)	echo from Exerciser
. **echo long**	set echo format to long

An important notion in the Exerciser is the *current axis*, which is set to global axis 1 when the Exerciser gets started. The current axis can be determined at any time by command:

. **?axis**

To select another axis, use:

. **axis =** <global axis number>

Most operations in the Exerciser apply to the current axis, or the board that contains the current axis.

Details on each command (syntax and semantics) are explained in the reference pages as well as in an on-line help facility. To start the help facility inside the Exerciser, type either

. **help**

or

. **help** <topic>

The former is menu driven in which topics are grouped according to their functionality. The latter simply prints user selected <topic> to the screen.

The MCP-04 Exerciser contains a monitor, which when activated, displays on the screen critical parameters and activities of the system in real time. Experience has shown that it is an extremely useful debugging tool.

The monitor can be activated by typing at the Exerciser prompt

. monitor on

The display shown in Figure 2-12 will then appear at the top portion of your PC's screen.

```
                 MCP-04 Motion Controller Monitor

Axis   Mode   Status   Com_pos   Act_pos   Error   Final_pos   Inputs Outputs

 1   CTL_MODE   C0H     334455    334453       2       900       04      00
 2     INIT     E0H         77   1111234  ++++++         0       00      00
 3    IV_CTL    C0H       1122      1119       3       600       50      00
```

```
. ? gain
gain (axis 2) = 15.25
.
```

Figure 2-12. Exerciser Monitor Screen Display

Each axis is shown in one row with the current axis in reverse video. For each axis, the following system information is displayed:

Axis	Global axis number
Status	Status register
Mode	Control mode
Com_pos	Command position register
Act_pos	Actual position register
Error	difference between command and actual positions
Final_pos	Final position register (for trapezoidal profile control)

If the actual error is greater than +32,767, a number of +'s will be shown on the Error column. If the actual error is less than -32,768, a number of -'s will be shown.

For each I/O port on a MCP-04, the following information is displayed:

Input	Input ports for A, B and C
Output	Output ports for A, B, and C

Exerciser commands can be issued as usual from the lower portion of the screen. The monitor can be deactivated (turned off) by command

. monitor off

Three echo modes are provided: *off*, *short* and *long*. The default mode is *short*. *Off* turns off all echoes to the screen. *Short* echoes the command briefly (usually the value of a variable involved in the command). *Long* echoes detailed information about each command executed. The following is an example.

. **? gain**	echo is *short*
16.00	
. **echo off**	set echo to *off*
. **? gain**	
. **echo long**	set echo to *long*
. **? gain**	
gain (axis 2): 16.00	

The echo to a 'set' command (command of the form <variable> = <value>) is the actual setting in effect. For example:

. **gain = 23.341**	the resolution is 0.25
23.50	nearest possible setting
? gain	
23.50	

Recall that the specified value is always *rounded* to the nearest possible setting.

Command files, stored as an ASCII text containing a series of Exerciser commands may be *executed* inside the Exerciser by issuing the EXECUTE command.

. **execute** <command file>

An ASCII text file may also be executed repeatedly by optionally specifying the number of iterations after the file name.

. **execute** <command file> [<number of iterations>]

The effect of executing a command file is the same as if the commands in the file were typed manually from the keyboard. Caution must be taken so that each command move will have enough time to complete before the next command is issued. The WAIT command in the Exerciser can be employed for this specific purpose.

An Exerciser command file may be also be executed directly from the DOS shell by typing

x> **exercise** <command file>

The commands in <command file> will be interpreted by the Exerciser. The control returns to DOS once the end of the command file has been reached. The last exerciser command issued can be repeated by hitting F3 at the Exerciser dot prompt.

Each Exerciser command has an equivalent 2-letter short hand version. This is very convenient if you issue a lot of commands through the Exerciser. For example, *ap* is used for ACT_POS and *cm* is used for command ENTER_CTL_MODE. Reference the on-line help to become familiar with short hand versions of the Exerciser commands.

2.5.3 Programming Interface Library

The MCP-04 C Interface Library is located in the UTILS directory on the distribution disk. It allows the user to develop his/her own utility application programs in C. The MCP04.LIB library was compiled in the Large memory model. Check the directory files for additional programming information or contact Mektronix Technology for assistance.

Programming Hints

Data Type Conventions:

velocity, acceleration	float (32 bits)
filter parameters	float (32 bits)
position	long (32 bits)
ports, status	unsigned char (8 bits)
offset, pwm_com	int (16 bits)
others	unsigned int (16 bits)

MCP04.H:

File MCP.H on the distribution disk contains all necessary type declarations for functions defined in MCP04.C. Therefore, it is recommended to include MCP04.H in every module that calls functions in this library.

Return Values of 'Set' functions:

The return value of a 'set' function reflects its actual setting in effect, which may be different from the input value provided by the user as it may be out of valid range and/or subject to the rounding rules.

List of Functions

Control Mode functions

value = get_com_pos(axis) set_com_pos(axis, value)	get/set *command position* $-8{,}388{,}608 \le value \le 8{,}388{,}607$
value = get_act_pos(axis) set_act_pos(axis value) clr_act_pos(axis)	get/set/clear *actual position* $-8{,}388{,}608 \le value \le 8{,}388{,}607$
value = get_gain(axis) set_gain(axis, value)	get/set *gain* of filter $0 \le value \le 63.75$
value = get_zero(axis) set_zero(axis, value)	get/set *zero* of filter $0 \le value < 1$
value = get_pole(axis) set_pole(axis, value)	get /set *pole* of filter $-1 < value \le 0$
set_sample_freq(axis, value, md) value = get_sample_timer(axis)	set *sample frequency* of filter get *sample timer* value
value = get_accel(axis) set_accel(axis, value)	get/set *command acceleration* $0 \le value < 128$ (cnts/sample time^2)
value = get_max_vel(axis) set_max_vel(axis, value)	get/set *maximum velocity* $0 \le value \le 127$ (cnts/sample time)
value = get_final_pos(axis) set_final_pos(axis, value)	get/set *final position* $-8388608 \le value \le 8388607$ (counts)
value = get_prop_vel(axis) set_prop_vel(axis, value)	get/set *proportional velocity* $-2048 \le value < 2048$ (cnts/samp time)

value = get_int_vel(axis) set_int_vel(axis, value)	get/set *integral velocity* $-128 \leq$ value ≤ 127 (cnts/sample time)
value = get_act_vel(axis)	get *actual velocity*
value = get_ctl_mode(axis)	get *control mode*
go_tp_ctl(axis)	go *trapezoidal profile* control
go_pv_ctl(axis)	go *proportional velocity* control
go_iv_ctl(axis)	go *integral velocity* control
enter_ctl_mode(axis)	enter default *position control*
reset(axis)	software *reset*
init(axis)	put controller in *Initialize* mode

Configuration Functions

value = get_status(axis) set_config(axis, value)	get current *status* set configuration *value*
set_bipolar(axis) set_unipolar(axis)	± command voltage format + command voltage format
open_comm_loop(axis) close_comm_loop(axis)	commutator *open-loop* control commutator *closed-loop* control
check_ring(axis, ring , x, y, np) check_comm(axis, ring, off, ma)	verify *comm. ring* parameters verify *commutator* constraints
config_ports (board, a, b, c)	*configure ports* as input or output
value = get_ring(axis) set_ring(axis, value)	get/set commutator *ring* $0 \leq$ value ≤ 127
value = get_x_reg(axis) set_x_reg(axis, value)	get/set commutator *x register* $0 \leq$ value ≤ 127
value = get_y_reg(axis) set_y_reg(axis, value)	get/set commutator *y register* $0 \leq$ value ≤ 127
value = get_offset(axis) set_offset(axis, value)	get/set commutator *offset* $-128 \leq$ value ≤ 127
value = get_max_adv(axis) set_max_adv(axis, value)	get/set *max. commutator phase advance* $0 \leq$ value ≤ 127
set_vel_timer(axis, value)	set commutator *velocity timer* $0 \leq$ value ≤ 255
value = naxis() value = nboard()	*number of axis* in the system *number of boards* in the system

value = board_type(axis)	*type of board* containing axis
value = board_num(axis)	*board number* containing axis
value = axis_num(axis)	in-board axis number
value = get_clock_freq(axis)	get *clock frequency* for axis

Miscellaneous Functions

value = get_motor_com(axis) set_motor_com(axis, value)	get/set *motor command* $0 \leq$ value $\leq$ FFH
value = get_pwm_com(axis) set_pwm_com(axis, value)	PWM command port duty cycle $-100 \leq$ value ≤ 100
align(axis)	perform commutator alignment
clr_emergency(axis)	*clear emergency* flags
home(axis, board, port, bit, iv, acc)	perform *homing* sequence
value = read_port_a(board) write_port_a(board, value)	read from *Port A* write to *Port A* (8-bits)
value = read_port_b(board) write_port_b(board, value)	read from *Port B* write to *Port B* (8-bits)
value = read_port_c(board) write_port_c(board, value)	read from *Port C* write to *Port C* (4-bits)
read_ext_encoder(board) clr_ext_encoder (board)	read *external encoder* (16 bit integer) clear *external encoder* counts
write_ext_dac(board, value)	write to *external dac* (8 bits)
sync(board)	*synchronize* HCTL-1100's on board
mc_init()	establish system communication

Section 3

P-CNC OPERATION

If the system has never been set up, first run the OEM Setup Utility described in Section 2.4. After configuring the machine parameters to the desired specifications, save the configuration to produce the file OEM.SET. P-CNC will automatically set default specifications for the jog and system setup if the JOG.SET and CNC.SET files are not found in the working directory. *(NOTE: CNC.SET must be initialized with valid numbers from within P-CNC)*. Once configured, the OEM setup file will not need to be changed. Backup copies should be stored in a safe place to guard against accidental erasure or unauthorized modification. The OEM.EXE setup utility should be accessible only to authorized personnel that understand the details of the machine design. This is the motive for providing a separate setup utility outside the P-CNC operating environment.

In normal circumstances, you should never need to terminate the P-CNC program. All utilities and DOS shells are provided in the Utility panel. Quitting P-CNC is required to restart P-CNC with a new OEM Setup configuration. Type CTRL-Q to quit P-CNC and return to the DOS prompt. Change the OEM settings and invoke P-CNC with its 3-axis or 4-axis version. Only qualified technicians and resellers should attempt to modify the OEM configuration.

All the interface screens are made up of panels. On the top of each panel is a title bar that may have a ☐ symbol in the left corner. This box signifies that using the ESC key will close the panel. The question mark symbol in the right corner indicates that help information is available by typing the '?' character on the keyboard.

The cursor keys are used to make selections within a panel. Blue highlighted letters allow immediate access to the field by typing the letter. Selections may be toggled using the SPACE bar or edited using the ENTER key. Function key buttons located at the bottom of the screen labeled F1–F10 are used to navigate to other P-CNC functions. The abbreviated names are descriptive of the operation to be performed when the function key is pressed.

The Run screen shown in Figure 3-1 consists of the Program panel, position window and status window. Once a program is loaded, it will be displayed in the Program panel with the file name as a heading. The position window displays the current position and destination for each axis. The status window keeps track of the current modal settings for each G-code group and the miscellaneous outputs. You are allowed to enter the status panel by using the CTRL key. The status items will show their quick key letters in blue highlight. Typing the quick key will immediately transfer you to that item for editing. You may also move to the item using the cursor keys. Some selections may be toggled using the SPACE bar or edited by hitting the ENTER key. Hit the CTRL key again to exit the status window area.

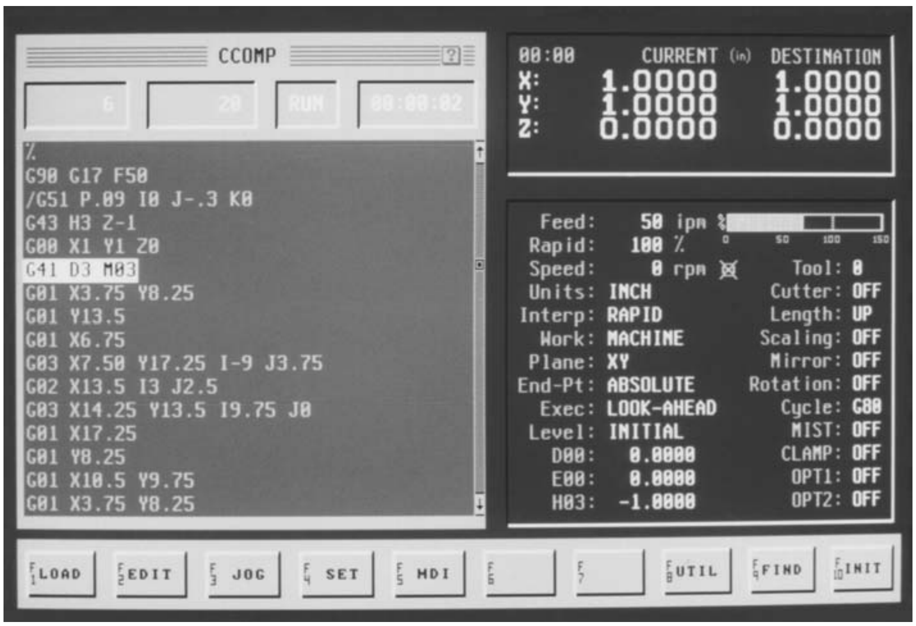


Figure 3-1. P-CNC Run Screen

3.1. USING ON-LINE HELP

Additional help information is available when the '?' symbol is displayed in the upper right corner of a panel. The help utility displays an index on the left and the selected topic on the right as shown in Figure 3.2. Use the TAB and the Shift-TAB keys to select different index headings. Each heading may have additional subheadings that may be displayed by typing the '+' key. Use the '-' key to reduce the expanded headings. Hit the ENTER key to go to and display the selected topic.

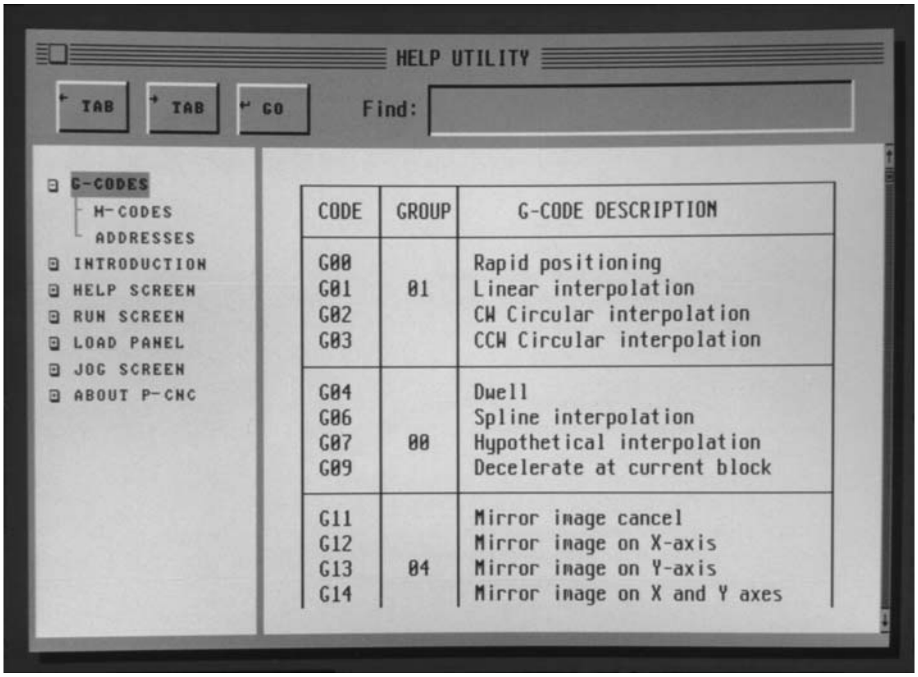


Figure 3-2. Help Utility Screen

Scrolling the help text is accomplished by using the cursor keys. The UP, DOWN, PGUP and PGDN keys are allowed. You can also search the help file for a particular character string by using the find feature. Type ‘F’ and enter the character match that you are looking for and hit ENTER. If a match is found it will be displayed in the help page. To repeat the search hit ENTER again.

3.2. LOADING AND EDITING NC PROGRAMS

When Load (F1) is activated from the Run screen, all the ∗.CNC files in the default directory are displayed as shown in Figure 3-3. The file in the upper left hand corner will initially be selected with inverse video. Use the cursor keys to highlight the selection and then ENTER to load the program.

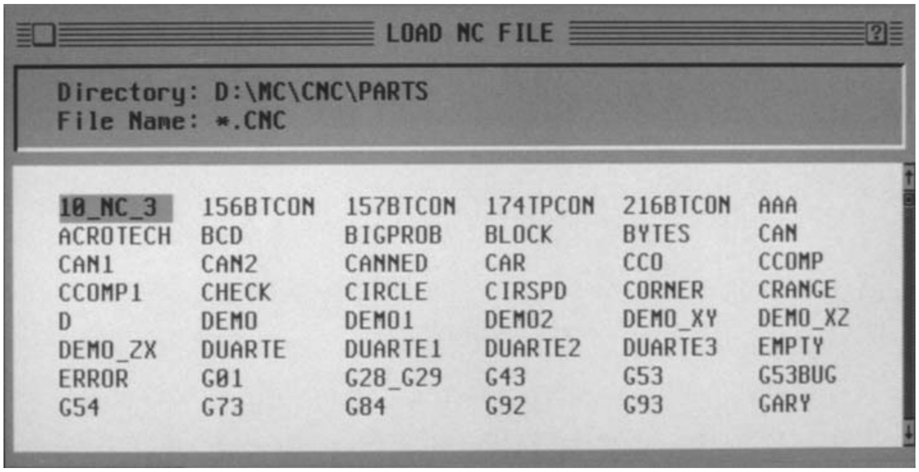


Figure 3-3. Loading a NC Program

The default file extension is specified in the OEM Setup utility, but is normally *.CNC. The load panel defaults to the last visited directory for listing NC programs. Hit the 'D' key to change the default directory. Type in the desired directory and hit ENTER to save the change. You may manually type in the file to be loaded by hitting 'F' and typing in the file name. The file extension is not required. Once the file is loaded, you will automatically return to the Run screen with the file name displayed in the title bar.

NC files may be downloaded from a remote server, a floppy disk or from the internal hard drive. When connected to a network, use the network drive letter in the Load panel to gain access. Networks should provide one-way, "on demand", communication when requested from P-CNC. It is best not to run large NC programs from the floppy drive due to its slow access speed. First copy the program to the parts directory on the internal hard drive using the DOS shell utility or the 'copy' command from a DOS prompt. The DOS shell utility is available from the Utility panel. When at a DOS prompt, type EXIT to return to the P-CNC Run screen.

If the NC file has errors, a warning will be displayed and a listing will be given with the problem blocks delineated by line number. If the program could not be loaded, first make sure that there is a percent sign at the beginning and at the end of the NC file. Very large files may also require increasing the virtual memory allocation in the OEM Utility.

You can create a new file with a specified name using the 'F' key in the Load panel. P-CNC will recognize that the file does not exist and will ask if you wish to create a new file. The editor will be started with the new file name that was created. The text editor that you install in P-CNC must accept the command line argument <*filename*> for this feature to work properly.

P-CNC may be configured to call any text editor of your choice. The default editor is usually the one that was supplied with DOS. Note that the editor selected must store a file in plain text format and should be able to load large NC files. P-CNC converts all text to uppercase when displayed in the Program panel and is therefore case insensitive. There is no advantage to packing command codes without spaces since P-CNC reconstructs the NC program into its own internal format. Block sequence (line) numbers are not required unless they provide a label for a subprogram jump.

When Edit (F2) is activated from the Run Screen, the text editor specified the OEM Setup Utility will be invoked with the current NC program. You may now edit your program with this text editor. As soon as you exit the editor, P-CNC reloads the newly edited program and parses it again. If no syntax error is found, the program will go back to the Run screen with the program name displayed at the top of the Program panel. If the specified editor cannot be found, an error message will occur and control goes back to the Run screen.

3.3. EXECUTING NC PROGRAMS

With P-CNC, a program may be *simulated* or *run* depending on the setting in the Plot panel. No physical activities are generated on the MCP-04 board when in Simulation mode, which is helpful when debugging your NC program. The current mode setting is displayed in the Program panel window. The differences between Simulation and Run are given below:

- No physical motion/actuation is generated with Simulation. All activities are performed in computer memory. Therefore, no physical damage may result from Simulation.
- When the system is in Run mode, the displayed current position always corresponds to the physical position of the tool. Clearly, this can not be the case with Simulation. It is important to remember that at the beginning of each simulation, the current position is always set to the actual (physical) position of the machine tool.

- Simulation does not generate output signals for external devices, nor does it read any external input signals besides the Feed Hold.
- Simulation checks that each profile point is within the minimum and maximum position ranges, while Run only checks the end-points. The operator should execute the NC program in Simulation mode to check that points along circular arcs do not exceed the work space limits.

The Program panel is located on the left half of the Run screen and displays the program currently being executed by P-CNC. The name of the file that contains this program is shown in the panel title bar. The program is displayed "as is" with all letters converted to upper case. Empty lines as well as comments are all preserved. Due to the limitation of the size of the window, characters beyond the right edge of the window are not shown. The program text currently being interpreted is displayed in inverse video. The window scrolls automatically whenever necessary. If a break-point has been set by the operator, the break-point line number is given in the upper right window of the execution panel. The current SIM/RUN mode setting is displayed in the upper panel window.

The right half of the screen displays the position outputs and the run time status of the program currently being interpreted. The status of all ON/OFF switches, modal commands, current tool number, current interpolation, and feed and speeds are updated continuously.

3.3.1. Run Modes

The Single/Continuous selector switch determines how NC programs are executed. In Step mode, one block of the NC program is interpreted and executed each time the Cycle Start button is pushed. Blocks that have been interpreted but not yet executed are displayed with a highlighted bar to the left of the line.

While operating in the Continuous mode, P-CNC behaves according to Group 15 (G61, G64). The default look-ahead mode scans ahead and assembles small segments into a continuous motion. It stops assembling the sequences when any of the following occurs:

- a sharp corner is detected
- an exact stop command (G09)
- a non-interpolation G-code

- other addresses except (M12-M15)

P-CNC then generates the combined tool movement before proceeding to process the interruption. From the user's point of view, if a smooth motion composed of a collection of blocks is desired, these blocks should contain only dimension words (X, Y and Z), interpolation words (I, J, K and R) and interpolation G codes (G01-G03). Option 1 and Option 2 control outputs (M12 - M15) may also be turned on and off during continuous profiling. Within the block, Option-on commands are handled first, then motion commands (G01, G02, and G03) and then Option-off commands, regardless of the order in which they appear.

Execution may be gracefully interrupted by hitting any key during continuous run. The program will stop at the next opportunity without causing feed rate blemishes in the part. Use the Feed Hold button to cause an immediate controlled stop during profiles. Once the Feed Hold button is detected, the feed rate will ramp down to zero velocity and wait for the operator to release the switch or type ESC to terminate the execution.

During interpolation modes (G01-G03), the feed rate may be adjusted from 0% to 150% in 10% increments (reference Table 2-6). Once a new feed rate is selected, P-CNC accelerates to the new adjusted feed rate. The operator must be sure the feed rate override switch is not in the 0% setting for execution to proceed. No warning is given when the feed rate override is selected to be zero percent.

It is important to consider when a program is using the default parameters. These parameters are initialized when loading or restarting a NC Program and after the INIT [F10] command is given within the Run screen. Otherwise, the current modal registers remain in effect.

3.3.2. Browsing and Searching

After a NC program is loaded you may examine the program using the cursor keys. The UP/DOWN keys move one program block at a time. The HOME key returns to the last executed block. The END key sends you to the end of the program or to a set break-point. Break-points are set by placing the highlight at the desired program block and typing SHIFT+END on the numeric keypad. To remove the break-point repeat the same procedure. The break-point line number is displayed in a window at the top of the NC panel and the program block is displayed in yellow.

You are allowed to start execution at any block location you move the cursor to in the program. This can be dangerous since modal conditions may require executing in the correct sequence. Attempting to skip blocks while in cutter compensation is very difficult since blocks are buffered before execution.

WARNING: If the browser is used, be sure to return to the last executed line of code using the Home key before re-executing the NC program. Execution from anywhere in a program is allowed using the current register status (i.e. P-CNC does not read through unexecuted program blocks to obtain the status).

The F10 'INIT" key is used to re-initialize the NC program to the default settings and clear all existing modal settings in effect. This is required after an emergency stop and is performed automatically when loading a NC program.

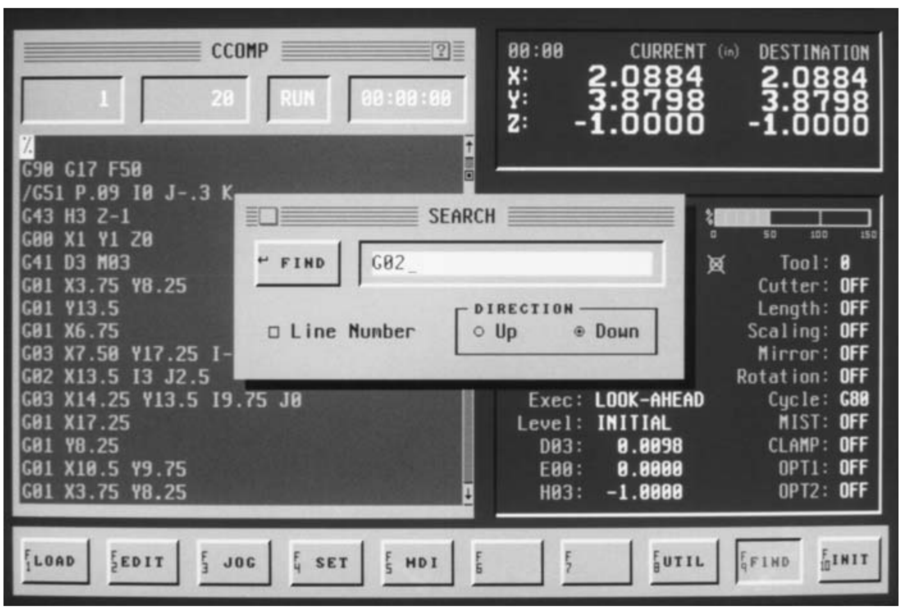


Figure 3-4. Run Screen with Search Panel

Searching the loaded NC file is easy using the Find (F9) function key. The Search panel shown below, allows searching for matching text strings or a specified line number. Use the TAB key to move to each option in the Search panel and the SPACE bar to toggle each option. The options include searching Up and Down the file or to a specified line number. Type either the matching text string or the line number in the Find field and hit ENTER to begin the search. You may repeat the find by hitting the ENTER key again.

3.3.3. MDI Execution

Enter MDI mode by using function key F5. The bottom of the screen changes to the input editor allowing commands to be entered and executed. Multiple blocks of code can be entered by using a semicolon (;) to terminate each block. Do not use a semicolon at the end of the input line. Hit the ENTER key to begin execution.

There are some logical differences between MDI and program execution due to the nature of its intended use. The MDI buffer will not accept sequence addresses `N' or `O' since M47 and M98 jump to the actual NC program loaded in memory. M47 will bring the execution cursor to the beginning of the NC program and return to the MDI editor. M98, call sub-program, will jump to the sequence number specified in the NC program and start executing until a M99, return from sub-program, block is encountered. Exact stop mode, G61, is not effective while executing multiple blocks from the MDI buffer. Type ESC to quit the MDI editor.

3.4. UTILITY PROGRAMS

P-CNC is configured with five utility programs. You must use the OEM Setup utility to change the configuration. The OEM help is available within the utility by typing ‘?’ from the main screen. When no program is specified, P-CNC defaults to a DOS prompt. In order to return to P-CNC you must type ***“exit”***. A DOS manual, the MCP-04 board Exerciser, a calculator, the DOS Shell interface and the DOS prompt are all pre-configured to run from the Utility panel. Large programs can still run from memory since the P-CNC application is reduced to less than 5 Kbytes and stored in a temporary file. Other useful programs may be a CAD/CAM program, virus protection (MSAV) and a backup utility (MSBACKUP).

3.5. JOGGING AND HOMING

The Jog screen may be configured to display only the control panels that you would like to see using the function keys. The panels do not need to be activated for their associated keys to be effective. The Jog screen is where you home the machine, define positions, and perform incremental, velocity and handwheel jogging of selected axes. Manual control of relay outputs is provided in the Relay and Spindle control panels.

The Axis panel provides both incremental and velocity jogging control. Axes are selected for jogging using the axis select switch. Subsequent jogging and homing operations are effective only for the axis selected using this switch. The selected axis is displayed in inverse video in the panel entitled "Command Position". The F10 key is used to toggle the axis display to report actual position, command position or position lag. Figure 3-5 shows the Jog screen with the Relay, Spindle and Axis Control panels activated.

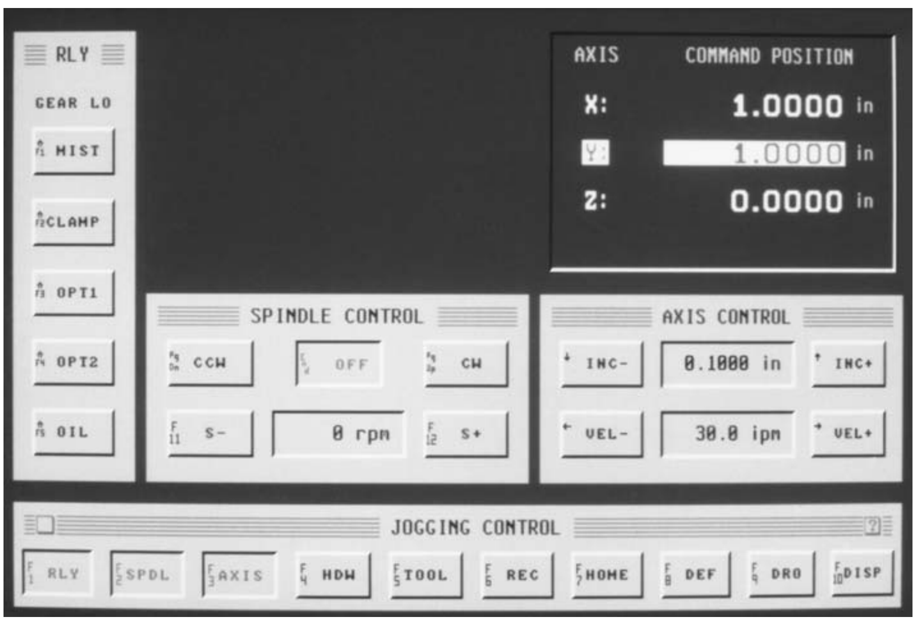


Figure 3-5. Jogging Control Display

3.5.1. Setting Velocity and Increment

These operations are used to set the velocity and increment used while jogging with the cursor keys. The left and right cursor keys are associated with velocity jogging and the up and down cursor keys are associated with point to point (increment) jogging. Velocity defines the jog velocity while jogging in velocity mode, and Increment determines the distance each axis travels while jogging in increment mode. Type [V] to edit the velocity of the selected axis. Type [I] to edit the jogging increment size for the selected axis.

3.5.2. Jogging Axes

This operation allows your machine tool to jog according to specified jog parameter settings. P-CNC supports two jogging modes: *velocity jogging* and *incremental jogging*. In Velocity Jogging mode, selected axes move with predetermined velocities while the left/right cursor key is depressed. In the Incremental Jogging mode, selected axes, upon each activation, start to move the designated increment distance simultaneously. Each axis travels the distance specified by the Increment setting. The rapid positioning (G00) parameters, rapid velocity and acceleration are in effect during incremental jogging.

The cursor keys are employed to control jogging on axes.

[←→]: Initiates velocity jog mode on selected axes. The velocity used is identical to that indicated in the jog parameter settings. The selected axis starts to move immediately in either direction once the key is pressed and remains in effect until the key is released. The feed rate override switch will further adjust the jog velocity.

[↑↓]: Each selected axis travels the increment displacement relative to the current position. The selected axis moves with each key stroke commanded. If the system is currently in velocity jog mode, P-CNC first terminates velocity jogging and then an increment position will be commanded. P-CNC waits for your command after the final position has been reached.

3.5.3. Handwheel Jogging

The handwheel option [F4] allows the selected axes to be manually jogged with a control handwheel. Four modes of operation are available using the handwheel mode switch. Two incremental modes (.0001" and .001") and two velocity modes (slow and fast) are provided. The velocity modes are useful for quickly touching off a part to set part zero.

The feed rate override switch may be used to adjust the speed ratio between the handwheel and the motor axes. The ratio ranges from 0.039 to 17 times the handwheel encoder pulses (refer to Table 2-6). The default setting when the feed rate override switch is not used is a 2.25 multiplier value. Select [F4] again to terminate handwheel jogging and return to the Jog screen.

3.5.4. Defining Axis Positions

Defining axis positions to establish user work offsets can be accomplished in the Jog menu. This is equivalent to a G92 or G54-G59 but done manually at the current axis position. Use the F8 key to enter the Define Position sub-menu. All axes are initially selected, but they may be deselected using the F1-F4 keys. Function keys are provided to:

[F5]: zero selected axes
[F6]: clear axis offsets
[F7]: MDI the new positions
[XYZ]: quickly MDI a single axis

3.5.5. Homing the Machine

Homing the machine tool must be performed before a NC program can be executed. Only one axis is required to be homed and is only required after shutting down the computer. The act of homing establishes the machine coordinates. All range errors are referenced to the machine coordinates. The axes are selected and deselected using the F1-F4 function keys. Initially, the F7 function key in the Run screen provides a direct path to the Home screen. Homing can be performed within the Jog screen at any time.

Multiple axes may be homed at the same time using the automatic home. When automatic homing is not enabled in the OEM Setup, the operator must manually enter the machine coordinates. The automatic homing sequence consists of the following two phases.

> In phase 1, all selected axes start to move simultaneously towards the origin. The velocity of each individual axis is equal to the setting in the OEM Setup. Each axis stops as soon as the home switch for that axis is triggered. Phase 2 is entered only if all selected axes have completed phase 1 and stopped.
>
> In phase 2, each selected axis moves in the direction opposite to that specified in the OEM Setup Utility. The axes move at slow speed and seek the encoder index pulse. If the encoder does not have an index pulse, phase 2 will not be executed. Once phase 2 is complete, the axis will be set to home zero.

3.5.6. Manual Switch and Spindle Control

The Relay panel is used to manually turn outputs on and off using the key combination Shift + (F1-F5). The Relay panel is displayed using the F1 function key. The status of these switches are subsequently altered by either a NC program (M codes) or by switch toggling while in the Jog menu. It is important to remember that P-CNC does not reset these switches (to OFF) at the beginning of each program execution. The control signals are provided so that the MCP-04 can interface to external devices such as a coolant pump.

The Spindle panel allows the speed to be specified by entering the desired rpm by hitting the F2 function key. The F11 and F12 function keys decrement and increment the spindle speed, respectively. The spindle can be turned on CCW using the PgDn key and turned on CW using the PgUp key. The END key turns the spindle relays CW and CCW both off. The F11 and F12 keys are also effective while in the Run menu.

Type [F5] to manually command the tool changer from within the Jog screen. A sub-panel appears where you can use the [±] keys to increment the tool to the next position or type ‘T’ to MDI any valid tool number. To close the Tool panel, type F5 again.

3.5.7. Digital Read-Out (DRO) Mode

The Digital Read-Out (DRO) mode shown in Figure 3-6 is entered from the Jog screen using the F9 function key. The actual axis positions are displayed in large numbers with simulated LED axis enable lights. To operate a small mill using hand cranks, first disable the servo drives using the XYZ keys. The simulated LED light will toggle off and on by hitting the axis [XYZ] letter keys. The axis positions may be preset using the 'P' hot key and typing in the value. Use ESC to return to the Jog screen which will re-enable the axis drives.

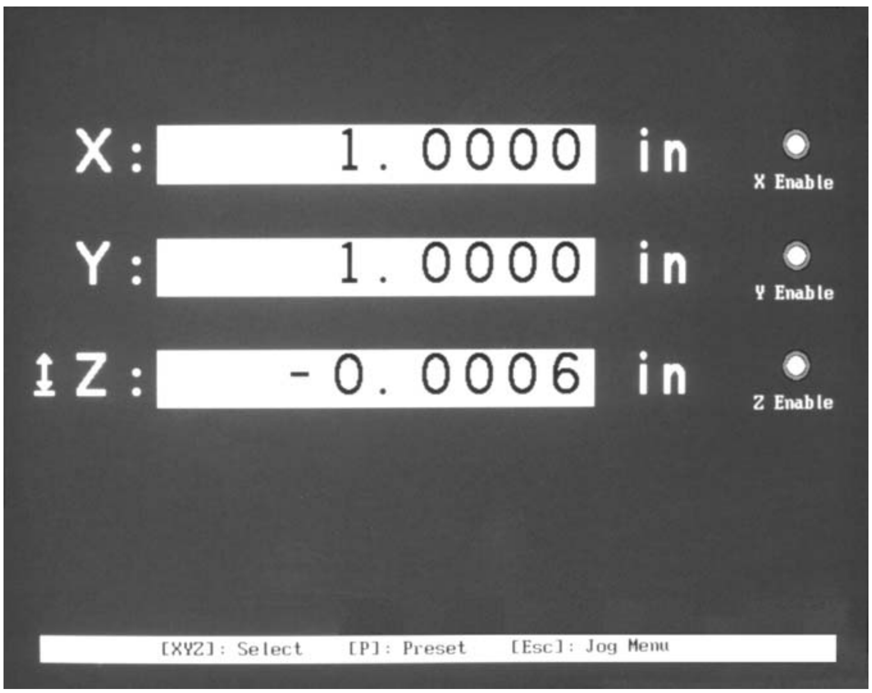


Figure 3-6. DRO Mode Screen Display

3.6. CNC SETUPS

The Setup screen is used to define the default programming modes, configure background plotting, define work coordinates, tool offsets and display I/O diagnostics. The operator can set Dry Run for part check-out, enable the Slash block skip symbol and the M01 optional stop command. The Run mode can also be changed to a Simulation mode which provides background plotting without moving the tool.

3.6.1. Default Programming Modes

The CNC operator is allowed to set certain default programming modes whenever a NC program is initialized. The modal commands will still change when commanded within a NC program. The default modes may be set to either metric or inch and incremental or absolute programming. The default plane may be selected as XY, YZ, or ZX.

The Default Mode panel provides toggles to enable Dry Run, Slash Enable and Optional Stop. When Dry Run is selected all the programmed feed rates and the rapid positioning speed run at the maximum feed rate. The feed rate override switch is ineffective while in Dry Run. This mode is used to test a new program in air, providing slower rapid moves and faster cutting speeds to reduce overall execution time. When a Slash is present at the beginning of a block, that block will be skipped if the Slash Enable is checked. The optional stop M-code (M01) is ignored when the Optional Stop is disabled in the Default Mode panel. When enabled, the NC program pauses after the block containing M01 and becomes equivalent to the M00 program stop M-code.

3.6.2 Background Plotting

A simulated execution of the NC program is possible by selecting Simulation in the Plot panel. SIM or RUN is displayed in a window at the top of the Program panel. During run execution the position commands are stored for later retrieval for plotting. The same code is used as the real motion to provide an exact representation of the programmed path. As a diagnostic tool, you are allowed to store actual positions from the shaft encoders during motion profiles using the ALT-D key combination. This will verify the accuracy of the actual tool path at the programmed feed rate. This diagnostic should normally not be used during actual production because it does use a small amount of CPU time to work in the background.

3.6.3 Work & Tool Offsets

User offsets are stored in the Setup screen using the Work Coordinates (F3) panel, the D Reg (F4) panel, E Reg (F5) panel, and the H Reg (F6) panel. The D registers are the tool radius offsets used with cutter compensation programming. The E registers are reserved for future use. The H registers are used to apply tool offsets perpendicular to the selected plane. The

Work Coordinates are used to set predefined offsets to the machine coordinates and may be applied at any position in the work space.

To edit an entry, highlight the desired register using the cursor keys and hit ENTER to begin editing the value. The registers can be automatically set to the current axis position by using the INSERT key. The automatic insert feature, along with jogging controls, allows the operator to quickly set new fixture offsets into the work coordinate table.

The units displayed are determined by the setting in the Default Programming modes panel described previously. Refer to G40, G41 and G42 commands for a description on cutter compensation and G43, G44, and G49 commands for additional information on using tool height offsets. Refer to G92, and G53-G59 commands for additional information on using work coordinate systems in a NC program. It is also possible to preset these registers using the G10 command and the probe calibration cycles.

3.6.4. Default Programming Modes

The F1 function key opens the Default Modes panel, use the [Cursor] keys to select the desired option and the space bar to change selections. All possible settings will be toggled each time the space bar is pressed. The following discusses each option and the possible settings:

(1) Dry Run (ON/OFF)

If Dry Run is selected ON, all the programmed feed rates and the rapid positioning speed run at the Max Feed Rate parameter specified in the OEM Utility. The feed rate override switch is also ineffective. This mode is used to test a program, providing slower rapid moves, to allow the operator enough time to hit the emergency stop, and faster cutting speeds, so that the test does not take an inordinate amount of time.

(2) Slash Enable (ON/OFF)

When a slash is present at the beginning of a block, that block will be skipped if the slash option is ENABLED. The operator can disable the Slash option using this setting so that slashes have no effect when present in a NC program.

(3) Optional Stop (ON/OFF)

The optional stop M-code, M01, in a NC program checks to see if the Optional Stop setting is ENABLED or DISABLED. When DISABLED, the interpreter will ignore M01 as if it never appeared. When ENABLED, the NC program pauses after the block being executed and is, therefore, equivalent to the Program Stop, M00 code.

(4) Measurement (ENGLISH/METRIC)

The space bar on the keyboard is used to switch the initial default input mode between IMPERIAL and METRIC. This setting can be overwritten in a NC program by G70 or G71 (G20 or G21, Fanuc). The current setting of this specification is used in the P-CNC Setup menu to determine which units to use for data input.

(5) End-Point (ABSOLUTE/INCREMENTAL)

Toggling the space bar on the keyboard will switch the initial default programming mode between ABSOLUTE and INCREMENTAL. This setting can be overwritten in a NC program by G90 or G91.

(5) Plane Selection (XY, ZX or YZ)

The space bar on the keyboard is used to select the initial default 2-D plane. This setting can be overwritten in the NC program by G17, G18 or G19.

3.6.5. Plot Setup

The Plot panel is used to configure the background plotting while in simulation mode. The plot is initiated by hitting the F6 function key.

(1) Simulation Mode: (ON/OFF)

The simulation mode is used during Run execution to verify a program and to do background plotting of the tool path.

(2) Plot Size: (in)

The Plot Max View should be set to a value that will show the complete range of the tool path when plotted to the screen in Simulation mode. After interpolating an axis motion, use the [F9] key to plot the history of the tool path. The Plot Max View setting can be adjusted so that the complete path is shown on the screen.

(3) Zoom Increment:

While in the plot display, the view can be expanded or reduced using the [+] and [-] keys, respectively. The Zoom Increment is the amount of the expansion or reduction as a multiplier of the current view. So, a value of 0.1 will provide a 10% expansion or reduction of the current display each time the [+] or [-] key is pressed.

(4) XYZ Origin: (in)

The XYZ Origin is defined as the lower left corner of the tool path plot display. Both positive and negative values are valid.

3.6.6. Information and I/O Diagnostics

The F9 key displays the P-CNC Information panel. The panel displays the MCP-04 registration serial number, the current P-CNC environment variable setting, the status of the virtual memory manager, and the CPU speed rating.

The I/O Diagnostics panel is displayed with the F10 key and is used to verify that input switches are working correctly and that all relay outputs are properly set. The panel also provides the status of axis limits and control mode. When axes are initialized the motor commands are set to output zero volts and the drives are disabled. This condition occurs if an E-stop or limit is triggered. The I/O Diagnostics panel is displayed and numbered the same as the actual I/O module MCP-R16.

Section 4

PROGRAMMING WITH P-CNC

P-CNC is designed to be compatible with G and M code (EIA Standard RS-274D) NC programs. When interfacing with a CAM program, the post-processor should be configured to the specifications given in this section. P-CNC emulates many of the features of the GE/Fanuc series controllers and their specific interpretation of the RS-274D standard.

4.1. BASIC PROGRAMMING COMPONENTS

A valid P-CNC program consists of a *program leader*, a *main program*, an optional number of *sub-programs* and a *program trailer*. The main program and subprograms consist of sequences of *blocks*, which are a set of command lines given to a NC interpretation system for operating the machine. According to these commands, the machine tool is directed along trajectories while external equipment is controlled. The trajectories are referenced to either the machine coordinates or arbitrary work coordinate systems setup by the user.

Figure 4-1 shows the overview of a P-CNC program. The leader and trailer are identified with a '%' (percent) sign. The leader must be the first character of a NC program. All text following the leader on the same line is ignored by the interpreter and is typically used for a program description or a part identifier. A NC program is stored as a (ASCII) text file. Only one program can be loaded into P-CNC at a time. In other words, the main program and all sub-programs must be contained in one text file for P-CNC to execute correctly.

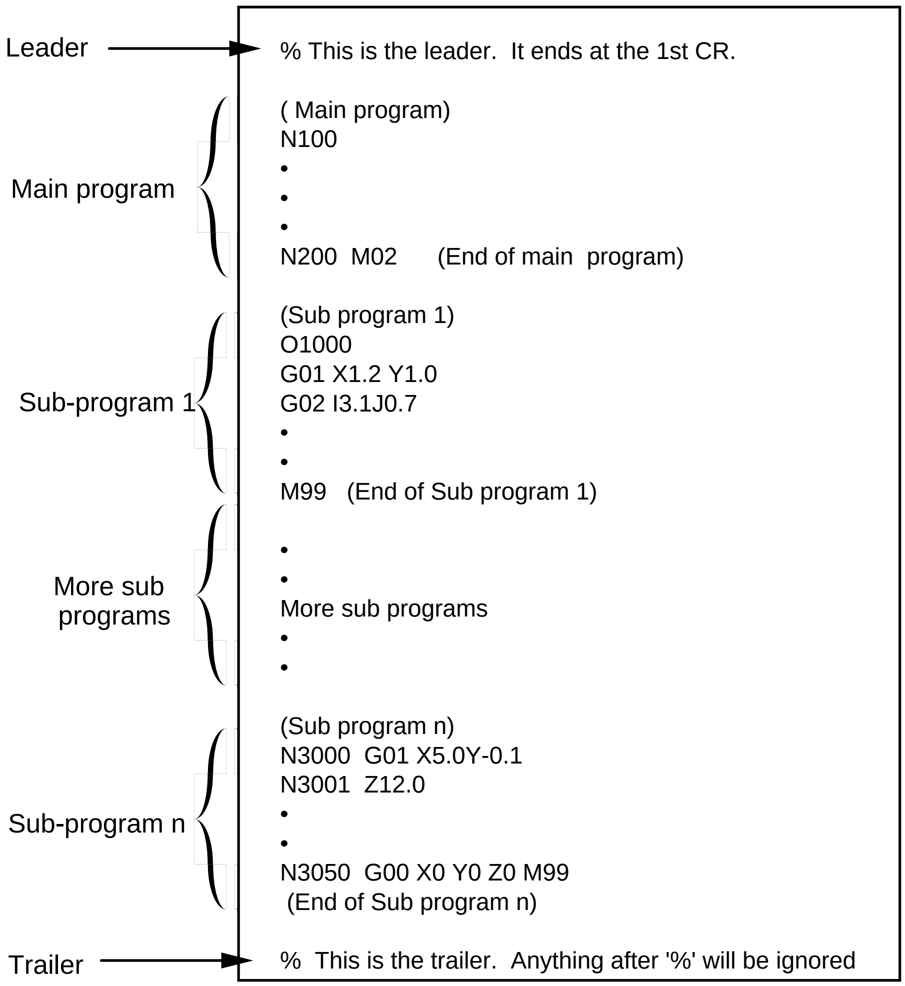


Figure 4-1. P-CNC Program Format

4.1.1. Leader

The leader of a P-CNC program consists of a '%' and a segment of text including everything before the next [CR]. Nothing before the first '%' in the text file is allowed. Text within the leader may be used to identify the NC program as shown in Figure 4-2.

```
% This program processes part # 13345 of Project 9981.↵
```

Figure 4-2. The Program Leader

It is clear from the definition that the Leader always takes the first line of your NC program.

4.1.2. Program Comments

The text between the *control out* character, '(' (left parenthesis), and the *control in* character ')' (right parenthesis) is considered a comment and will not be interpreted. This allows you to write comments in multiple lines as long as you put everything between the open and closing parentheses as shown below in Figure 4-3.

```
( This is a one line comment to be ignored)

N100  G01  X0.5 Y1.0  (comments can be inserted in a line) Z0.8

(This is the first line of a two line comment.
This is the second line of a two line comment. )
```

Figure 4-3. Comments in a NC Program

4.1.3. Blocks, Words and Addresses

Any main program or sub-program is composed of *blocks*. One block is separated from another by the *end of block* symbol, which is the new line character in P-CNC. This manual uses [CR] to denote the "end of block" symbol.

A *sequence number* (N followed by a number between 1 and 99999) may be optionally given at the beginning of a block for identification purposes. Each sequence number must be unique. It is not recommended to have a sequence number for each program block unless it is used to identify a feature or sub-program location. Although they are not required to appear in the program in a particular order, it is helpful that they be specified in ascending order in the program structure.

A block consists of several *words*. A word is composed of a character address followed by a numerical value as shown below:

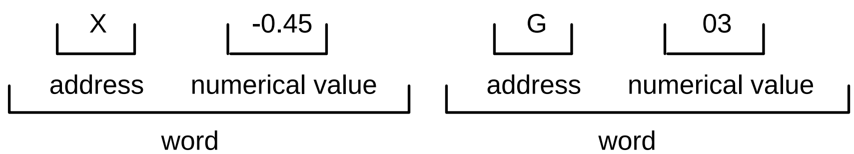


Figure 4-4. Word Structure

An *address* is a letter indicating the meaning of the following numerical value. The summary given in Table 4-1 shows all the addresses and their data types. *(NOTE: Not all of the addresses are currently employed by P-CNC but will become effective in future versions of the software.)*

Address	Type	Function
A	REAL	Angular dimension -polar coordinates
B	REAL	Multi-purpose designator
C	REAL	Fourth axis rotary dimension
D	INT	Radial offset lookup designator
E	INT	Variable lookup designator
F	REAL	Feed function
G	INT	Preparatory function
H	INT	Length offset lookup designator
I	REAL	Interpolation parameter parallel to X
J	REAL	Interpolation parameter parallel to Y
K	REAL	Interpolation parameter parallel to Z
L	INT	Number of loops
M	INT	Miscellaneous function
N	INT	Sequence number
O	INT	Sequence number for macros
P	REAL	Multi-purpose designator
Q	REAL	Multi-purpose designator
R	REAL	Radius or R-point
S	INT	Spindle speed function
T	INT	Tool function
U	REAL	Multi-purpose designator
V	REAL	Multi-purpose designator
W	REAL	Fourth axis linear dimension
X	REAL	Primary X-motion dimension
Y	REAL	Primary Y-motion dimension
Z	REAL	Primary Z-motion dimension

Table 4-1. Addresses Employed by P-CNC

Except words with addresses N and O (i.e., sequence numbers and program numbers), which must appear at the beginning of the block, there is generally no requirement for the order of words within a block. Codes are processed in a particular order based on their associated activity. For clarity and compatibility, Figure 4-5 shows the recommended order of words within a block.

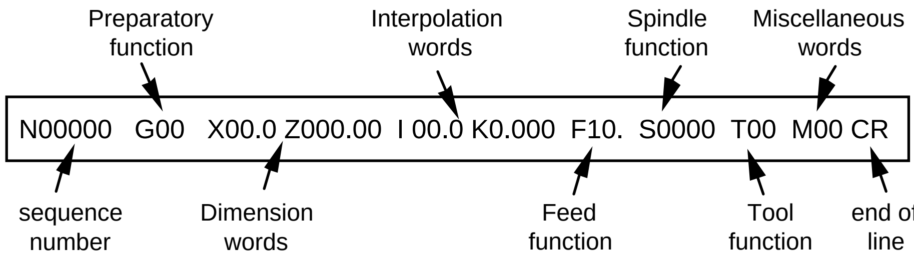


Figure 4-5. Recommended Order of Addresses within a Block

Addresses can be specified in either upper or lower cases. Leading zeros of integer numerical values may be omitted. Blanks and tabs embedded in the text of a block are ignored by the P-CNC interpreter unless they are placed in the middle of a numerical value in which case a syntax error will occur. Any portion of the block in the above figure can be omitted. So it is possible to specify an empty block. This flexibility may be used to improve your program's readability.

A block starting with '/' (forward slash) may be skipped at run time depending on the current setting in the Default Mode specification. Figure 4-6 below shows an example of a block containing a forward slash.

```
/ N201G00 X 0.5  (Skipped at run time)
```

Figure 4-6. Slash Option in P-CNC

4.1.4. Main Program and Sub-programs

The main program immediately follows the leader. Only one main program is allowed in a NC program. Unless a sub-program call (M98) or return from sub-program (M99) is encountered, blocks in a NC program are interpreted (executed) in order of their appearance (not in order of the sequence numbers associated with them).

A number of sub-programs may be defined following the main program. Each sub-program must start with a unique sequence number. Although addresses N and O are interchangeable, it is recommended for clarity and compatibility that N be used for sequence numbers.

Certain fixed sequences of NC code may be frequently used in many programs. These sequences of codes can be extracted to form sub-programs in order to simplify programming. Sub-programs may be called in the main program and/or another sub-program. The maximum number of nested sub-program calls in P-CNC is 16. A sub-program is identified by a sequence number: a word with address N followed by a unique positive integer.

4.2. PROGRAMMING FUNCTIONS SUPPORTED

This section contains descriptions of how the supported G-codes, M-codes, tool and spindle functions operate using P-CNC. This section serves as a reference for NC program development and the specific implementation used by Mektronix in support of the EIA RS-274D standard.

4.2.1. Preparatory Functions (G Codes)

A non-negative integer between 0 and 99 following address G determines the basic meaning of the related block. According to their functionality, G codes are partitioned into several operational groups. At any given moment, only one G code within the same group can be in effect. If more than one G code of the same group is specified in the same block, the last one is effective. Most G code groups will issue an operator warning and terminate program execution when this occurs.

G codes can also be divided into two categories. If its function affects only the block within which it appears, it is called a *"one shot"* G code. On the other hand, if the function is retained until canceled or superseded by a subsequent G code of the same group, it is called a *"modal"* G code. One-shot G codes are associated with Group 00. The following example illustrates the modal G code concept for Group 01.

```
G00  X2.0 Y1.0      (rapid positioning)
Z-1.0               (rapid positioning: G00 still effective)
G01  X1.0 Y-1.0     (linear interpolation: G00 no longer in effect)
Z0                  (linear interpolation: G01 still effective)
```

The following G codes are currently supported by P-CNC. A syntax error occurs if an unsupported G code is specified.

G Code	Group	Address for
◆ G00	01	Rapid positioning (point to point)
G01		Linear interpolation
G02		Circular/helical interpolation CW
G03		Circular/helical interpolation CCW
G04	00	Dwell
G05		*unassigned*
G06		Spline interpolation
G07		Hypothetical axis interpolation
G08		unassigned
G09		Exact Stop
G10		Preset D and H offset tables
G11	04	Mirror image cancel
G12		Mirror image on primary axis (X-axis)
G13		Mirror image on secondary axis (Y-axis)
G14		Mirror image on both axes (XY-axes)
◆ G17	02	X-Y plane select
◆ G18		Z-X plane select
◆ G19		Y-Z plane select
G25	00	Probe calibration cycles
G28		Return to machine coordinate origin
G29		Return from machine coordinate origin
◆ G40	07	Cutter radius compensation cancel
G41		Cutter radius compensation left
G42		Cutter radius compensation right
G43	08	Tool length compensation +
G44		Tool length compensation -
◆ G49		Tool length compensation cancel
◆ G50	11	Scaling cancel
G51		Scaling
G52	00	Shift current coordinate system

G Code	Group	Address for
◆ G53	14	Machine coordinate system selected
G54		Work coordinate frame 1 selection
G55		Work coordinate frame 2 selection
G56		Work coordinate frame 3 selection
G57		Work coordinate frame 4 selection
G58		Work coordinate frame 5 selection
G59		Work coordinate frame 6 selection
G60	15	unassigned
G61		Exact stop mode (block by block)
G62-G63		*unassigned*
◆ G64		Continuous mode (look-ahead)
G65-G69	12	*reserved for macro programming*
◆ G70	06	Inch input
◆ G71		Metric input
G73	09	Peck drilling cycle
G74		Counter tapping cycle
◆ G80		Canned cycle cancel
G81		Drilling cycle, spot boring
G82		Drilling cycle, counter boring
G83		Peck drilling cycle
G84		Tapping cycle
G85		Boring cycle
G86		Boring cycle
G87		Back boring cycle
G88		Boring cycle
G89		Boring cycle
◆ G90	03	Absolute command
◆ G91		Incremental command
G92	00	Work coordinates change
◆ G93	05	Inverse time feed
◆ G94		Feed rate per minute
◆ G95		Feed rate per revolution
G96	13	Constant surface speed
◆ G97		Revolutions per minute
◆ G98	10	Cycle initial level return
G99		Cycle reference level return

◆ - may be selected as default within a group

Table 4-2. G Codes Supported by P-CNC

4.2.2. Miscellaneous Functions (M Codes)

M codes are used for ON/OFF control of machine switches and to specify execution control of CNC programs. According to their functionality, they are divided into several groups.

- M_{on}: contains M codes that turn on switches.
- M_{off}: contains M codes that turn off switches.
- M_{stop}: M codes that terminate or pause NC program execution.
- M_{DOS}: M codes for accessing DOS services.
- M_{jump}: M codes that alter the sequential execution of the NC program.

M_{on} codes are processed before G-codes, which are processed before M_{off} codes within a block. M_{stop}, M_{DOS}, and M_{jump} codes are then processed sequentially after all M_{off} codes within a block.

P-CNC currently supports the M-codes listed in Table 4-3.

M Code	Description
M00	Program stop
M01	Optional program stop
M02	Program end
M03	Spindle on CW (RL1 ON)
M04	Spindle on CCW (RL2 ON
M05	Spindle off (RL1,RL2 OFF)
M06	Manual tool change
M07	Oil Shot (RL7 PULSED)
M08	Coolant on (RL3 ON)
M09	Coolant off (RL3 OFF)
M10	Clamp on (RL4 ON)
M11	Clamp off (RL4 OFF)
M12	Option1 on (RL5 ON)
M13	Option1 off (RL5 OFF)
M14	Option2 on (RL6 ON)
M15	Option2 off (RL6 OFF)
M16	Relay Paddr on (RL9-RL16)
M17	Relay Paddr off (RL9-RL16)
M18	All outputs off (RL1-R16)
M19	Spindle orient
*M40	Auto speed gear change
M41	Manual speed low gear (RL8 OFF)
M42	Manual speed high gear (RL8 ON)
M47	Return to program start
*M48	Enable feed rate override
M49	Bypass feed rate override
M90	User defined DOS call 1
M91	User defined DOS call 2
M92	User defined DOS call 3
M93	User defined DOS call 4
M94	User defined DOS call 5
M95	User defined DOS call 6
M96	User defined DOS call 7
M97	User defined DOS call 8
M98	Call sub-program
M99	Return from sub-program

Table 4-3. M Codes Supported by P-CNC

4.2.3. Tool Function (T)

Each tool should be assigned a number so that it can be specified in your program. The desired tool number is specified in your program by attaching it to address T (for example T01, T03). Many different types of tool changer programs have been developed by Mektronix. An example tool magazine with positions 0 to 7 is shown in Figure 4-7.

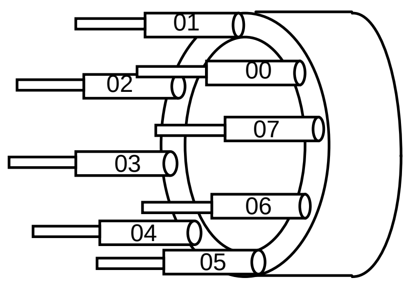


Figure 4-7. Tool Magazine

If the programmed tool number differs from the current tool number, a tool change will be performed by P-CNC. Automatic tool change is processed before G-codes and M-codes within the same block. A tool change can also be commanded within the Run screen or the Jog screen. When P-CNC first boots-up, the tool position is set to T00 signifying that no tool has been selected. No automatic home is initiated upon starting P-CNC. A turret homing sequence is initiated after commanding the first tool number.

If it is desired to perform a manual tool change, use M06 as described in Section 4.8. The machine can be positioned for an automatic tool change with G28, machine reference zero return and G29, return from machine reference zero. Each machine will be configured differently for handling the tool changer. Please consult your Mektronix representative as to the particulars for your installation.

4.2.4. Spindle Function (S)

The desired spindle speed can be specified in your program by attaching it to address S. An integer value is expected in units of revolutions per minute (rpm). The maximum spindle speed is determined by the OEM Setup Utility. The spindle can be turned on clockwise, counter clockwise or off through M codes in the program (Section 4.8.2) or in a jog session. A zero (0 rpm) spindle speed is assumed at the beginning of program execution.

In addition to spindle speed control, P-CNC provides two output signals to command the spindle operation. One of the signal lines indicates the current spindle direction (CW or CCW). The other tells whether the spindle is on or off (the former does not make sense if the latter is off). These two signals may be used to operate a fixed speed spindle motor. A spindle low and high gear signal is also available for increasing the spindle speed range. See Section 2.3.5 for more information regarding interfacing the spindle functions to P-CNC.

4.3. INTERPOLATION FUNCTIONS

P-CNC provides linear, spline, and helical interpolation of up to four axes at a time with circular interpolation on any two. A hypothetical axis may be specified to produce sinusoidal curves on a specified plane. The maximum feed rate for interpolation may not exceed the value set in the OEM Setup Utility. Rapid and interpolation positioning G codes are described next.

4.3.1. Rapid Positioning (G00)

Format:

G00 X___ Y___ Z___

G00 commands rapid point to point positioning of all axes. The rapid velocity may be adjusted with a rapid override setting in the Status window. The tool moves to a certain (absolute) position, or to a specific distance (incremental) from the current position at a predefined rapid traverse rate. The traverse distance is given by dimensional functions X, Y, Z and W (or C if defined as a rotary axis).

G00 employs simultaneous trapezoidal profiling on X, Y and Z axes. The maximum velocity and acceleration are set in the OEM Setup Utility. All axes will start the move simultaneously, however, they probably will not stop at the same time.

The following illustrates the process of trapezoidal profiling. T_a is the time needed to accelerate and decelerate the tool. The tool movement will be in a constant velocity once the commanded maximum velocity has been reached.

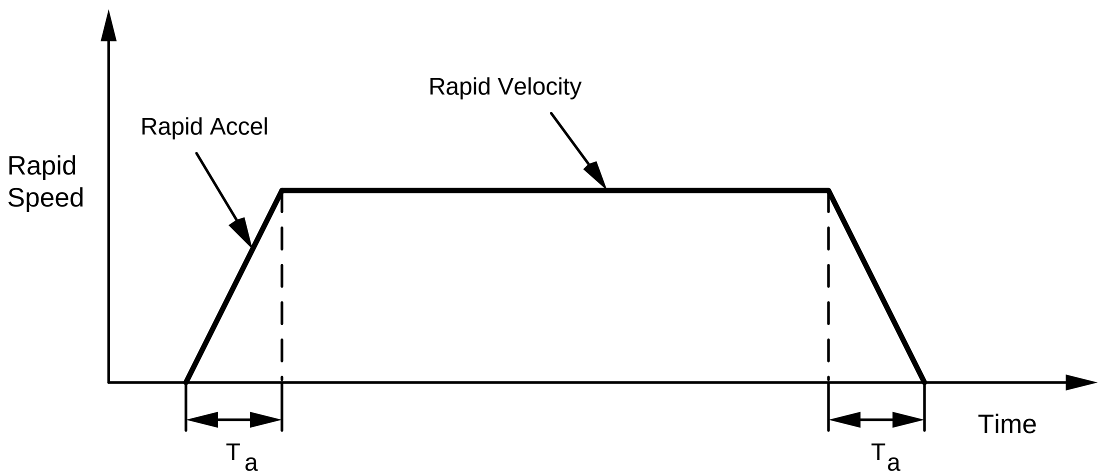


Figure 4-8. Trapezoidal Profile Control (G00)

Example:

G00 X1.0 Y2.0

4.3.2. Linear Interpolation (G01)

Format:

G01 X___ Y___ Z___ F___

G01 specifies linear interpolation in up to four axes and produces a straight line by accelerating to a constant velocity set by the feed rate address. The value of dimension functions, in conjunction with the current status of G90/G91 (incremental or absolute, respectively), define the distance the tool travels. The feed rate given by the F address is measured along the path trajectory. If F is not commanded, the default feed rate (0) will be used which will cause a run time error. The actual feed rate along each axis may be calculated according to the following:

$$F_x = \Delta x / d \cdot F \quad F_y = \Delta y / d \cdot F \text{ and } \quad F_z = \Delta z / d \cdot F$$

where F is the feed rate given with the F function, Δx, Δy and Δz are distances the tool transverses along X, Y and Z direction, respectively and

$$d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$$

Example:

G01 X2.0 Y3.0 Z-1.0 F3.0

4.3.3. Circular Interpolation (G02, G03)

Format:

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_\_\_ \; Y\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; J\_\_\_ \end{Bmatrix} F\_\_\_ \qquad \text{(X-Y plane)}$$

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_\_\_ \; Z\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; J\_\_\_ \end{Bmatrix} F\_\_\_ \qquad \text{(Z-X plane)}$$

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y\_\_\_ \; Z\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; J\_\_\_ \end{Bmatrix} F\_\_\_ \qquad \text{(Y-Z plane)}$$

G02 and G03 specify a circle or part of a circle on a plane selected by G17 (X-Y), G18 (Z-X) or G19 (Y-Z). When a NC program is started, the initial plane selected is as defined in the Default Modes menu. The cutting direction is given by the different G codes: G02 indicates clockwise (CW) direction; G03 indicates counter clockwise (CCW) direction. The feed rate is specified as the tangential velocity along the trajectory.

Two of the dimension functions (X, Y, Z) are used to specify the end points, which are expressed as an absolute or incremental value according to current G90/G91 settings. In G90 mode, numerical values given with dimension functions are the absolute end point position in the coordinate system. If G91 is effective, dimension functions only give the incremental distance from the current start point.

Two of the interpolation functions (I, J, K) may be used to specify the radius and the center of the circle. I, J and K are always the signed distance from the start point to the arc center in

X, Y and Z directions, respectively, regardless of the current status of G90/G91. The case of any I, J, K, equal to zero the I0, J0, and K0 can be omitted.

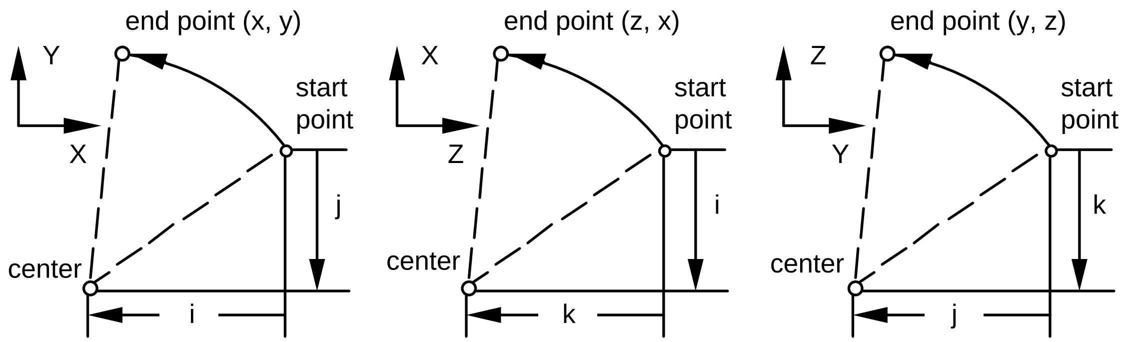


Figure 4-9. Specify the Center of the Circle with I, J and/or K

The radius can also be given with address R instead of specifying the center using I, J or K. In this case, two arcs with the same start and end points are considered. One arc is less than 180° whereas the other is greater than 180°. When the arc exceeding 180° is commanded, the radius must be specified with a negative value. In Figure 4-10, a negative value with address R specifies arc (1), a positive value specifies the shorter arc (2).

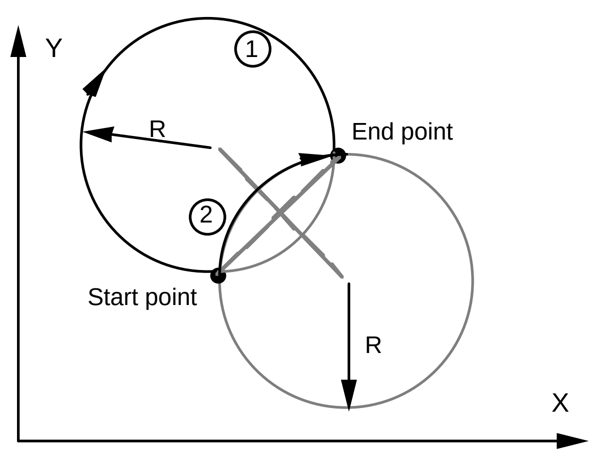


Figure 4-10. Specify the Radius and Center with R

If the current R is non-zero, the given X, Y, Z and R define the desired arc trajectory. When R appears simultaneously with I, J or K in the same block, R takes precedence. If the end point is located at the same position as the starting point, the motion is determined by the following rule: if the center/radius is given by I, J or K (R omitted), a full circle (arc of 360°) is specified; if the radius is given by R (that is, R is non-zero), no move is to be commanded.

Example:

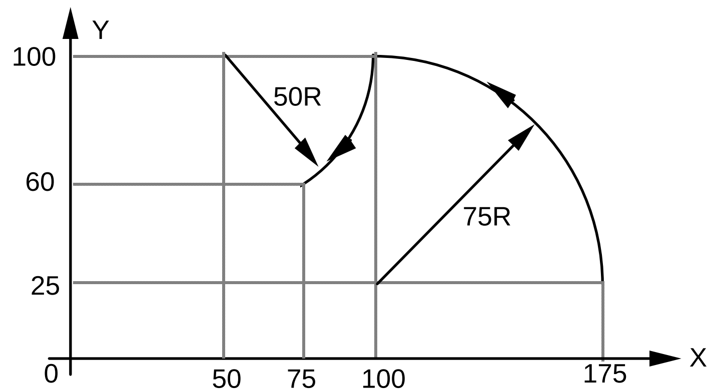


Figure 4-11. Example Trajectory using G02 and G03

Assuming the current tool position is x = 175.0 and y = 25.0, the above tool path can be programmed as follows:

- In absolute mode

  G03 X100.0 Y100.0 I-75.0
  G02 X75.0 Y60.0 I-50.0

or

  G03 X100.0 Y100.0 R75.0
  G02 X75.0 Y60.0 R50.0

- In incremental mode

  G03 X-75.0 Y75.0 I-75.0
  G02 X-25.0 Y-40.0 I-50.0

or

  G03 X-75.0 Y75.0 R75.0
  G02 X-25.0 Y-40.0 R50.0

4.3.4. Helical Interpolation (G02, G03)

Format:

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_\_\_ \ Y\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \ J\_\_\_ \end{Bmatrix} Z\_\_\_ \ F\_\_\_ \qquad \text{(X-Y plane)}$$

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_\_\_ \ Z\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \ J\_\_\_ \end{Bmatrix} Y\_\_\_ \ F\_\_\_ \qquad \text{(Z-X plane)}$$

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y\_\_\_ \ Z\_\_\_ \begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \ J\_\_\_ \end{Bmatrix} X\_\_\_ \ F\_\_\_ \qquad \text{(Y-Z plane)}$$

Helical interpolation is specified by providing a linear motion parameter with circular interpolation parameters. The effective feed rate is applied to the circular arc while the linear motion is adjusted to coincide with the circular end point. Therefore, the feed rate of the linear axis is as follows:

$$F \times \frac{\text{length of linear move}}{\text{length of circular arc}}$$

Use this equation to make sure that the feed rate of the linear axis does not exceed any velocity limits.

4.3.5. Hypothetical Axis Interpolation (G07)

Format:

G07 α 0	set α axis as hypothetical axis
G07 α 1	hypothetical axis cancel

Sine interpolation can be performed by keeping one of the axes in circular interpolation stationary while moving the third axis linearly. G07 is used to command which axis is the hypothetical axis. Specify α as the optional address of one of the controlled axes (X, Y, Z). After an axis has been set to hypothetical, normal motion will be inhibited but the time display will occur as though the axis was profiling. Hypothetical axes are ignored while in canned cycle operations. The following program example serves to clarify the effect of G07 commanding a hypothetical axis.

Example:

G07 Y0 F40	(y-axis is specified as the hypothetical axis)
G17 G03 X0 Y1 J.5 Z1	(y-axis does not move)
G01 Y-1	(profile time computed, no move)
G07 X1	(cancel hypothetical axis)

4.3.6. Spline Interpolation (G06)

Format:

G06 X___ Y___ Z___ F___	1st via point and feed rate specified
X___ Y___ Z___	2nd via point
X___ Y___ Z___	3rd via point
∘ ∘ ∘	
∘ ∘ ∘	
∘ ∘ ∘	
X___ Y___ Z___	nth via point
G01 X___ Y___ Z___	cancel G06 and rapid to XYZ

Figure 4-12 illustrates how the via points are used to define a curve. The start point is the current position when G06 is first specified. The following via points are specified while in G06 mode. The end point is the last via point specified before being canceled as a "one-shot" command. This is usually accomplished by specifying a Group 01 interpolation to exit the spline.

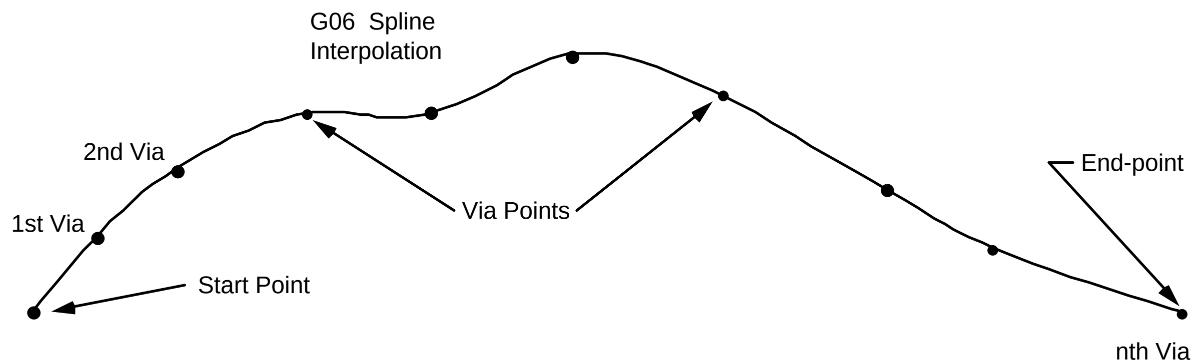


Figure 4-12. Spline Interpolation

Spline interpolation is not associated with Group 01 modal command interpolations. It is part of the “one-shot” command group due to its requirement to process many blocks of XYZ coordinates that represent the curves via points. The via points are the positions where the curve must traverse during interpolation. After the last via point is specified, the complete spline is calculated and executed. Spline interpolation is canceled when either a Group 01 interpolation (G00 - G03) is specified or a non-motion block is given.

The spline algorithm uses a polynomial curve to specify the trajectory. Spline interpolation is good for generating smooth curves, but is not useful for moving along straight lines. Knowing when to use spline interpolation can greatly reduce program size and produce smooth surfaces. When using a CAM program, write a post processor that will take spline curves and generate the via points instead of a multiple of straight line segments.

4.4. FEED FUNCTIONS

Feed rate for spline, linear and circular interpolation are commanded with a value following the F address. The rapid traverse (G00) speed is set in the OEM setup and is not affected by the feed rate specification. The feed rate may be specified in feed per minute, feed per revolution, or inverse time. The feed rate can further be adjusted during profiles using the feed rate override switch. A feed interlock input is provided to make sure that the spindle motor is turned ON and turning at the programmed speed.

4.4.1. Feed Rate (F)

The F address is used to specify the feed rate, which is the velocity of the tool relative to the work piece along the tool path as shown in Figure 4-13.

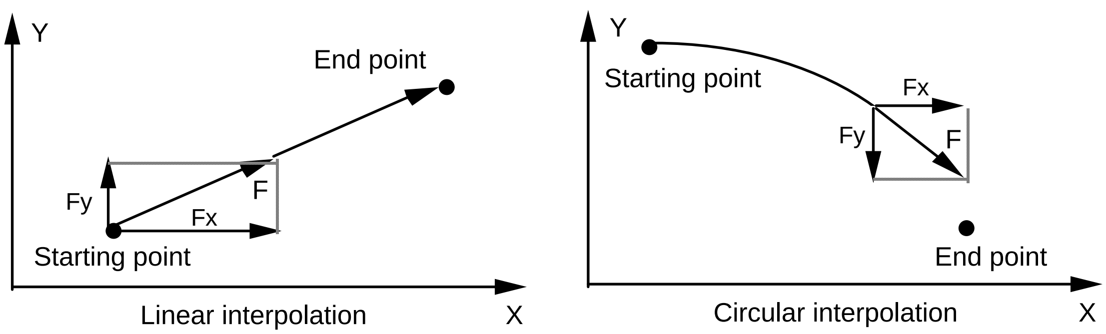


Figure 4-13. Feed Rate Specification

The numerical value following the F address must be greater than zero (0) and can not exceed the maximum feed rate specification. The data is modal so it will not be changed until a new F specification is made. A zero feed rate is assumed whenever execution of a program is initiated.

The feed rate specified in a NC program may be overridden while profiling, using the optional feed rate override switch. The feed rate override setting is effective for all feed rate specifications (G93, G94, and G95). A 4-bit binary switch provides adjustments from 0% to 150% in 10% increments. The operator is cautioned not to leave the switch in the 0% setting due to no special warnings being given when running a NC program.

4.4.2. Feed Specifications (G93, G94, G95)

G94 specifies feed rate in inches per minute or millimeters per minute and is the default. G95 specifies feed rate as inches or millimeters per spindle revolution. G93 specifies feed rate in inverse time. The default setting at the time when a NC program starts execution is given in the P-CNC Setup (Section 3.6.1). The following explains each of the three possible settings for specifying the feed rate.

4.4.2.1. G94 (Feed per minute)

With the per minute feed mode, G94, the trajectory feed rate per minute is directly commanded by the numerical value following the F address. G94 is modal and stays in effect until G93 or G95 is commanded. The feed rate specification is graphically depicted in Figure 4-14.

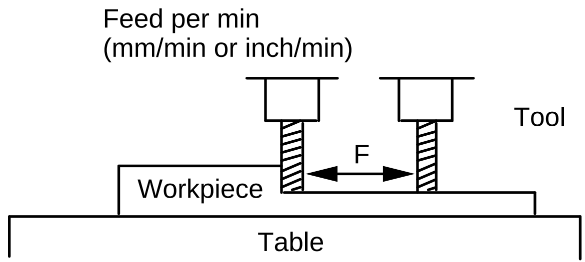


Figure 4-14. Feed Rate Specified in Inch (mm) per Minute

4.4.2.2. G95 (Feed per revolution)

The feed rate can be specified in feed rate per revolution using G95. The numerical value following the F address is the feed of the tool per spindle revolution. The calculation is based on the commanded speed of the spindle and does not account for the actual spindle position due to the lack of position feedback. The graphical representation of this feed rate mode is shown in Figure 4-15.

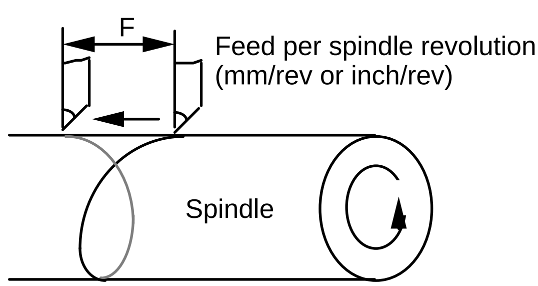


Figure 4-15. Feed Rate Specified in Inch(mm) per Spindle Resolution

4.4.2.3. G93 (Inverse time)

The inverse time specification for feed rate is commanded with G93. The numerical value following the F address is the actual feed rate in terms of speed, divided by the distance to travel or arc radius. This produces units of inverse time which may be helpful to think of as the inverse of the time to travel the specified distance or proportional to the arc radius.

For linear interpolation (G01) the inverse time (IT) is calculated:

$$IT = \frac{\text{Feed Rate}}{\text{Distance}}$$

For circular interpolation (G02, G03) the inverse time (IT) is calculated:

$$IT = \frac{\text{Feed Rate}}{\text{Arc Radius}}$$

4.4.3. Dwell (G04)

Format:

G04 P___	(P value referenced to feed spec)
G94 G04 P___	(P value in seconds)
G95 G04 P___	(P value in revolutions)

A G04 word followed by a P word in a block is used to program the dwell duration. The numerical value P is interpreted as floating point. The feed rate specification currently in effect will determine how the P word is evaluated. When commanded with a per minute mode (G94), the dwell is specified in seconds. When in a per rotation mode (G95), the dwell is specified in spindle revolutions. If no P word is given, the program execution pauses and will not resume until operator intervention.

4.5. COORDINATE SYSTEMS

P-CNC uses only two coordinate systems to evaluate motion trajectories, a machine coordinate frame and a work coordinate frame. The machine coordinates are set by either homing switches or defined manually in the Jog screen of P-CNC. These coordinates are based on actual hardware counters and will be lost when the host PC is turned off. The work coordinate frame is based on software offsets that are referenced to the machine coordinates and can be set by the user with various G-codes in a program. P-CNC defaults to the machine coordinate frame (G53) each time a NC program is executed from the beginning. Several work coordinate systems can be predefined in the P-CNC Setup panel configuration for G54 - G59.

4.5.1. Work Coordinates (G92, G53-G59)

There are no work coordinates set when a program is loaded, which implies machine coordinates. A work coordinate frame is setup within a program by either using the G92 command or the G54 to G59 commands. A G53 command can be programmed to clear work coordinate offsets and return to the machine coordinate frame. Six preset working coordinate frames can be selected by using a G54 to G59 command. The working coordinate frames are referenced to the machine coordinates. A convenient way of setting work coordinate systems is to use the Insert key in the Work Coordinate panel after positioning to a known coordinate.

A G92 command sets the current position to the coordinates specified, in either increment or absolute mode. In absolute mode, the current position is always set the same as the XYZ registers. In incremental mode, the current position will be set to the relative distance of the XYZ registers plus the current position. G92 may also be used to preload the tool (T) register to a desired value. No machine operation is initiated.

Example:

```
(set position to X1, Y2, Z3)
G90 G92 X1 Y2 Z3
```

4.5.2. Shift Current Coordinate System (G52)

Format:

G52 X___ Y___ Z___ (not all coordinates necessary)

G52 is a "one-shot" command that shifts the current coordinate frame origin by the amount specified. G52 works similarly to G92 except that the command is relative to the coordinate origin instead of the current position. All coordinate systems including the machine coordinate system can be modified by this command. The shift amount can be specified as an absolute or an increment value. The shifted coordinates may be canceled by either specifying

G52 X0 Y0 Z0

or commanding a coordinate system using (G53 - G59, G92).

4.5.3. Probe Calibration Routine (G25)

Format:

G25 [S1 -> S6] [XYZW] [Q]
G25 [D1 -> D99] [XYZW] [Q]
G25 [E1 -> E99] [XYZW] [Q]
G25 [H1 -> H99] [XYZW] [Q]

where,

S - specifies work coordinates to modify
D - register to modify
E - register to modify
H - register to modify
XYZW - any one specifies axis to move and the move distance
Q - tolerance or window in which to expect probe trigger

The probe calibration routine is used to automatically set registers to the probe trigger location. Only one register type (S, D, E, or H) may be specified in a given block. The axis to be moved is commanded using X, Y, Z, or W. The Q value specifies the tolerance of the specified move that generates an operator error when exceeded. If a Q value is not specified, the tolerance defaults to ¬0.5 inches. After a trigger has occurred the axis rapids back to its original position and stores the probed location in the specified register. The following example stores into the H11 register the probed location along the Z-axis.

Example:

(trigger expected at -10 ¬2 inches)
G91 G25 H11 Z-10 Q2

4.5.3. Probe Calibration Routine (G25)

Format:

G25 [S1 -> S6] [XYZW] [Q]
G25 [D1 -> D99] [XYZW] [Q]
G25 [E1 -> E99] [XYZW] [Q]
G25 [H1 -> H99] [XYZW] [Q]

where,

S - specifies work coordinates to modify
D - register to modify
E - register to modify
H - register to modify
XYZW - any one specifies axis to move and the move distance
Q - tolerance or window in which to expect probe trigger

The probe calibration routine is used to automatically set registers to the probe trigger location. Only one register type (S, D, E, or H) may be specified on a given block. The axis to be moved is commanded using X, Y, Z, or W. The Q value specifies the tolerance of the specified move that generates an operator error when exceeded. If a Q value is not specified, the tolerance defaults to ±0.5 inches. After a trigger has occurred the axis rapids back to its original position and stores the probed location in the specified register. The following example stores into the H11 register the probed location along the Z-axis.

G91 G25 H11 Z-10 Q2 (trigger expected at -10 ±2 inches)

4.5.4. Automatic Machine Zero Return (G28, G29)

Format:

G28 X___ Y___ Z___ (XYZ specify intermediate point)
G29 X___ Y___ Z___ (XYZ specify final point)

The G28 command specifies an automatic return to machine reference zero via an intermediate point. Along with G28, an intermediate point can be given in absolute or incremental modes referenced to the current coordinate frame so the machine tool first reaches the intermediate point before going on to machine reference zero.

G29 is the return from machine reference zero command that moves to the intermediate point specified with G28, and goes on to the optional final point specified in the G29 block. For incremental programming, the command value specifies the incremental value from the intermediate point. If X, Y, and Z are not present in the G29 block, the end-point is equivalent to the intermediate point. However, if any position address is given in the G29 block, the current register settings for X, Y, Z will be used as the end-point. Both G28 and G29 move at the rapid traverse speed while executing the desired moves and may be used for tool changing operations when the machine zero corresponds to the tool change position. An application example is given below and in Figure 4-16.

```
G91
G28 Y4          (Path A to B to C)
T05 G04 F3      (Tool change at C)
G29 X7 Y6       (Path C to B to D)
```

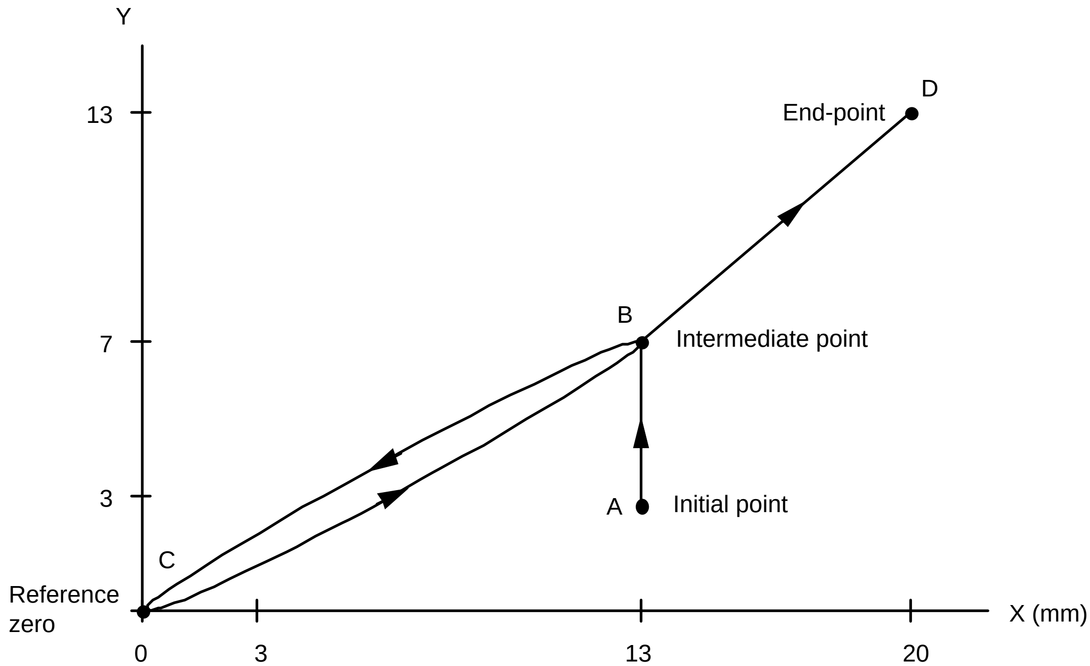


Figure 4-16. Application Example of G28 and G29

4.5.5. Mirror Image (G11, G12, G13, G14)

Format:

G11	(Mirror Image Cancel)
G12	(Mirror Image on X Axis)
G13	(Mirror Image on Y Axis)
G14	(Mirror Image on X and Y Axes)

Before setting Mirror Image, set the reflection point on the axis as absolute zero. This can be accomplished using G92 or G54 through G59. Images are mirrored across the zero boundary.

4.5.6. Plane Selection (G17, G18, G19)

Used to set the operating 2-D plane for G-code commands. G17, G18 and G19 select XY plane, ZX plane, and YZ plane, respectively. The default plane at the time when a CNC program starts to execute is given in the Default Modes setup (Section 3.5.2).

4.5.7. Absolute and Incremental Programming (G90, G91)

G90 sets the current programming mode in which the input data is given in absolute dimensions, whereas G91 sets that to incremental dimensions. The default setting at the time when a NC program starts its execution is given in the Default Modes setup (Section 3.5.2).

4.5.8. Inch and Metric Conversion (G70, G71)

G70 sets the current input mode to programming in which input data is specified in inch units, whereas G71 sets it to metric. The default setting at the time when a NC program starts execution is given in the Default Modes setup (Section 3.5.2).

4.6. COMPENSATION FUNCTIONS

P-CNC supports the compensation functions of tool radius, tool length and coordinate scaling. These functions simplify programming by automatically compensating for tool dimensions or when a cutter path needs to be scaled proportionally.

4.6.1. Tool Radius Compensation (G40, G41, G42)

Format:

G41 G42 D___	(where D ranges from D01 to D16)
G40	cancel cutter compensation

The programmed path is offset in the plane selected either to the left or right by the amount specified in the D register panel. There are 99 D registers available which may be stored in the offset tables. The D00 register specifies a zero offset and cannot be changed. G41 applies the offset to the left and G42 applies the offset to the right of the programmed path. The offset calculation is calculated in the selected plane (G17, G18, and G19). An axis not in the selected plane has no offset. For example, if the current plane is set to XY then the Z-axis will not be offset. Cutter compensation works with all Group 01 interpolations (G00 - G03).

The example program given below demonstrates how to use cutter compensation to adjust the programmed path for the tool diameter. First, the plane is selected and the tool is moved to the start position. Make sure that the approach angle of the first move, after cutter compensation is turned on, will not interfere with the part geometry. Then cutter compensation is applied to the left by the amount specified in the D code in line 4. In this case, the D3 code has been previously set to a value of 0.25 inch. The first move positions the cutter at a right angle and to the left of the first line segment at a distance of 0.25 inches. Lines 5 though 15 specify the consecutive motion blocks to be compensated. Cutter compensation is canceled in sequence number 15. The programmed path and compensated path are shown in Figure 4-17 as a solid line and dashed line, respectively.

```
 1:  % CCOMP Demonstration Program
 2:  G17 F50
 3:  G00 X1 Y1 Z1
 4:  G41 D3
 5:  G01 X3.75 Y8.25 Z0
 6:  G01 Y13.5
 7:  G01 X6.75
 8:  G03 X7.50 Y17.25 I-9 J3.75
 9:  G02 X13.5 I3 J2.25
10:  G03 X14.25 Y13.5 I9.75 J0
11:  G01 X17.25
12:  G01 Y8.25
13:  G01 X10.5 Y9.75
14:  G01 X3.75 Y8.25
15:  G01 X1 Y1 G40
16:  M02
17:  %
```

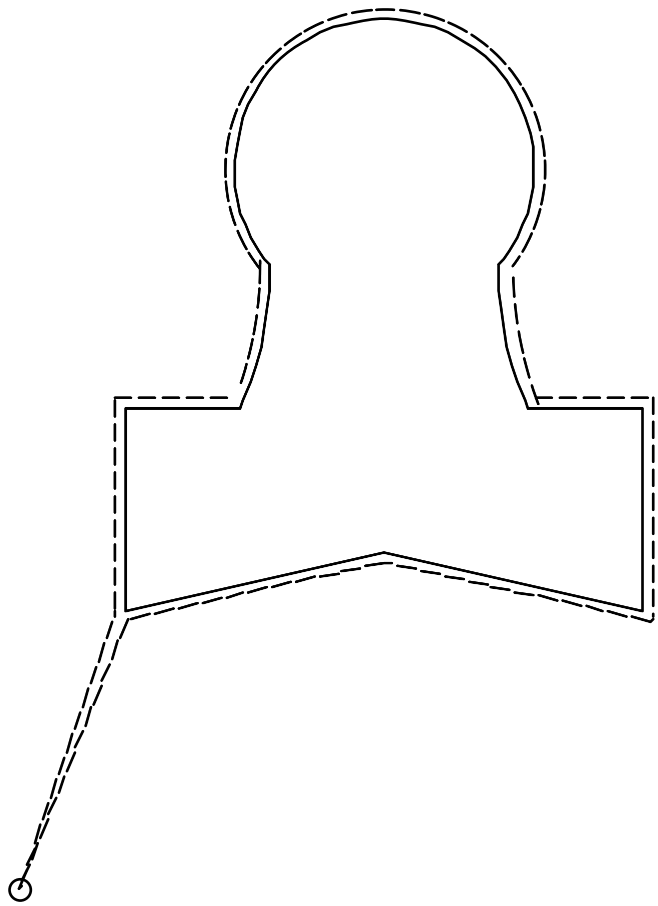


Figure 4-17. Cutter Compensated Tool Path Example

After commanding cutter compensation left or right, all compensated paths should be programmed consecutively (i.e. each block must contain a motion move). Once a non-motion block is interpreted, cutter compensation is temporarily canceled and waits to start up again after the next motion block. Cutter compensation looks ahead one motion block before commanding the previous move. No warning is given if the tool path is not physically possible with given tool diameter. The start-up block must not contain a circular move. Up to five perpendicular axis moves are allowed before returning to motion moves in the selected plane.

It is possible to call a sub-program while cutter compensation is on, by keeping the M98 miscellaneous code within the same block as the Group 01 interpolation command. All other preparatory and miscellaneous codes should be placed in the same block as motion moves, so that cutter compensation is not automatically canceled.

4.6.2. Tool Length Offset (G43, G44, G49)

Format:

G43 G44	H___	(where H ranges from H01 to H16)
G49		(cancel tool length offset)

G43 and G44 are used to set the tool length offset perpendicular to the XY plane. This function is used to compensate different tools without the need to change the NC program. G43 adds the H code offset and G44 subtracts the H code offset from the Z-axis current position. Both G43 and G44 are modal G codes and remain in effect until another G code in the same group is specified. Z-axis motion occurs to the new offset position immediately after a command is executed. The offsets are pre-stored in the H-register table within the P-CNC Setup menu. G49 cancels tool length compensation which returns the Z-axis to its uncompensated position.

4.6.3. Preset D and H Registers (G10)

There are 99 available D and H registers used for storing tool radius and length offsets, respectively. D and H offsets can be stored using the following variation of the G10 command.

G10 H__ P__
G10 D__ P__
G10 D__ H__ P__

D, H:	Specify which register is to be modified.
P:	The value to be stored

As you can see, it is possible to store the same value in both register tables in the same block. G10 is a "one-shot" command and therefore, cannot be specified in the same block as another Group 00 command.

4.6.4. Scaling (G50, G51)

Scaling is used to reduce the programmed path proportionally about a center reference point in all three axes. Scaling is commanded with G51 using the following parameter list:

G51 I__ J__ K__ P __

I, J, K: Specify scaling center in absolute coordinates.
P: Specifies scaling factor.

While G51 is in effect, subsequent moves are scaled by the scale factor, P, in relation to the scaling center, I, J, K. The user must not specify scaling parameters in the same block with other G-codes that use an I, J, K or P parameter. It is good practice to keep G51 within its own block, for clarity of intent when writing CNC code.

If I, J, K is omitted, the scaling center will be assumed to be at absolute coordinates 0, 0, 0. If P is not specified, P is assumed 1.0 and G51 has no effect. The scaling factor is common to all axes. Scaling has no effect on tool length offsets and for Z-axis movement in the case of canned cycles. Scaling is canceled by G50.

4.7. CANNED CYCLES

Canned cycles simplify programming by combining a sequence of operations into a single G code. Canned cycles are used for drilling, boring, and tapping operations.

A canned cycle has a positioning plane and a drilling axis. The positioning plane is determined by G17, G18, and G19. The drilling axis is perpendicular to the selected positioning plane. Generally, a canned cycle consists of a sequence of the following six operations:

Operation 1: Positioning in selected plane (XY for G17)
Operation 2: Rapid traverse to R point
Operation 3: Drilling
Operation 4: Operation at bottom of hole
Operation 5: Retraction to R point
Operation 6: Rapid traverse to the initial point (G99)

Figure 4-18 shows a graphical representation of the six operations used in canned cycles.

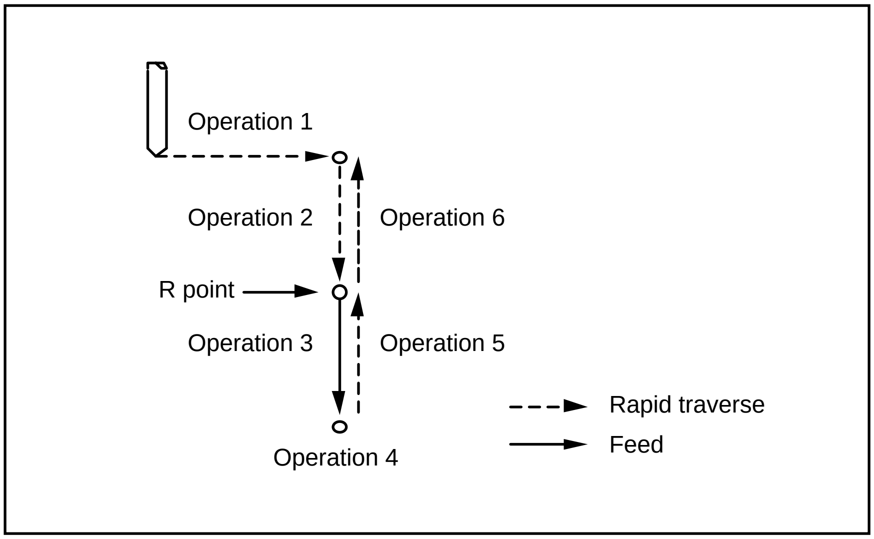


Figure 4-18 Canned Cycle Operation

Drilling data is specified in a single block using the following format.

G___ X___ Y___ Z___ R___ Q___ P___ U___ F___ L___

where:

G:	Specifies G73, G74, G76, G81-G89.
XYZ:	Specifies initial level: hole position in positioning plane and the bottom of the hole for the drilling axis.
R:	Specifies R point (distance from the initial level if G91).
Q:	Specifies the cut-in value with G73 and G83 or the shift value with G76 and G87 (always specified with incremental value).
P:	Specifies dwell time at the bottom of the hole in seconds.
U:	Retract amount in high-speed peck drilling.
F:	Specifies feed rate.
L:	Specifies the number of repetitions of complete canned cycle. When L is not specified, L = 1 is assumed.

Canned cycles are modal to G code Group 09 and remain in effect until canceled with G80 or a Group 01 (G01-G03) interpolation code. Once the drilling data has been specified it is retained until another canned cycle G-code (could be the same G-code) is specified or when canned cycles are canceled. In other words, the drilling data can only be specified in the same block that contains the G code. From that point on, the drilling position is all that is required to perform the same cycle at another location. The drilling cycle does not start until a [XYZ] address is specified. Sub-program calls using the L address will not affect the canned cycle repetition value.

The number of repetitions specified with the L address must be given in the command block to be effective. L is assumed one (1), if not present in a canned cycle block. The feed rate specified with the F address is retained even if the canned cycle is canceled. Repetitions can have a very different effect depending on the data format G90 or G91. When in absolute mode, the same hole will be drilled 'L' times and when in incremental mode, the distance specified in the positioning plane will be incremented each 'L' repetition. Figure 4-19 illustrates an example of repeating the relative drilling position when in incremental positioning mode.

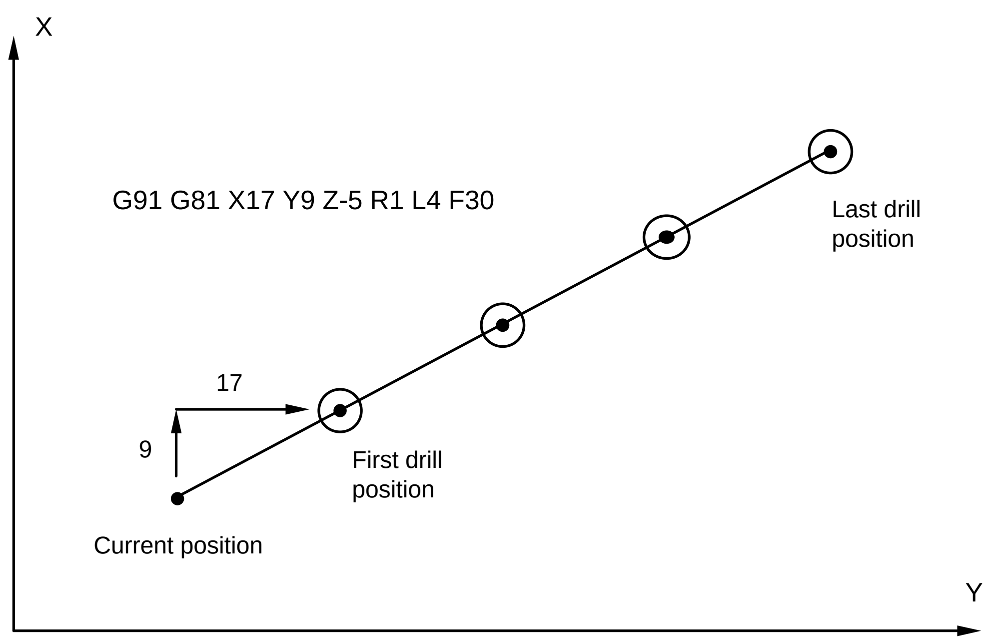


Figure 4-19 Example of Incremental Drilling

4.7.1. Return Point Level (G98, G99)

Besides the drilling mode selected, canned cycles are dependent on the data format (G90, G91) and the return point level (G98, G99). When specifying canned cycle parameters while in incremental mode (G91), relative distances are assumed from the current position. The default return point level is G98; the initial level at the beginning of canned cycles.

G98 and G99 are modal commands that specify the return point level during Operation 6 of a canned cycle. G98 specifies the return point to be the initial level and G99 specifies the return point to be the point specified by the R address ("R point"). The return from the bottom of the hole to the R point or initial level is performed at the rapid traverse rate Use G99 for the first drilling and G98 for the last drilling to return to the initial level.

4.7.2. High-speed Peck Drilling Cycle (G73)

Figure 4-20 shows G73 drilling cycle for both G98 and G99 modes. The high speed peck drilling cycle provides a cut-in depth of (q) and a retract amount (u). Due to the quick retraction speed, efficient machining of deep holes is performed. The Q address specifies the cut-in value, q, between each retraction and the U address specifies the retract amount. The operation continues until the Z depth has been reached. Retraction is performed at the rapid traverse rate.

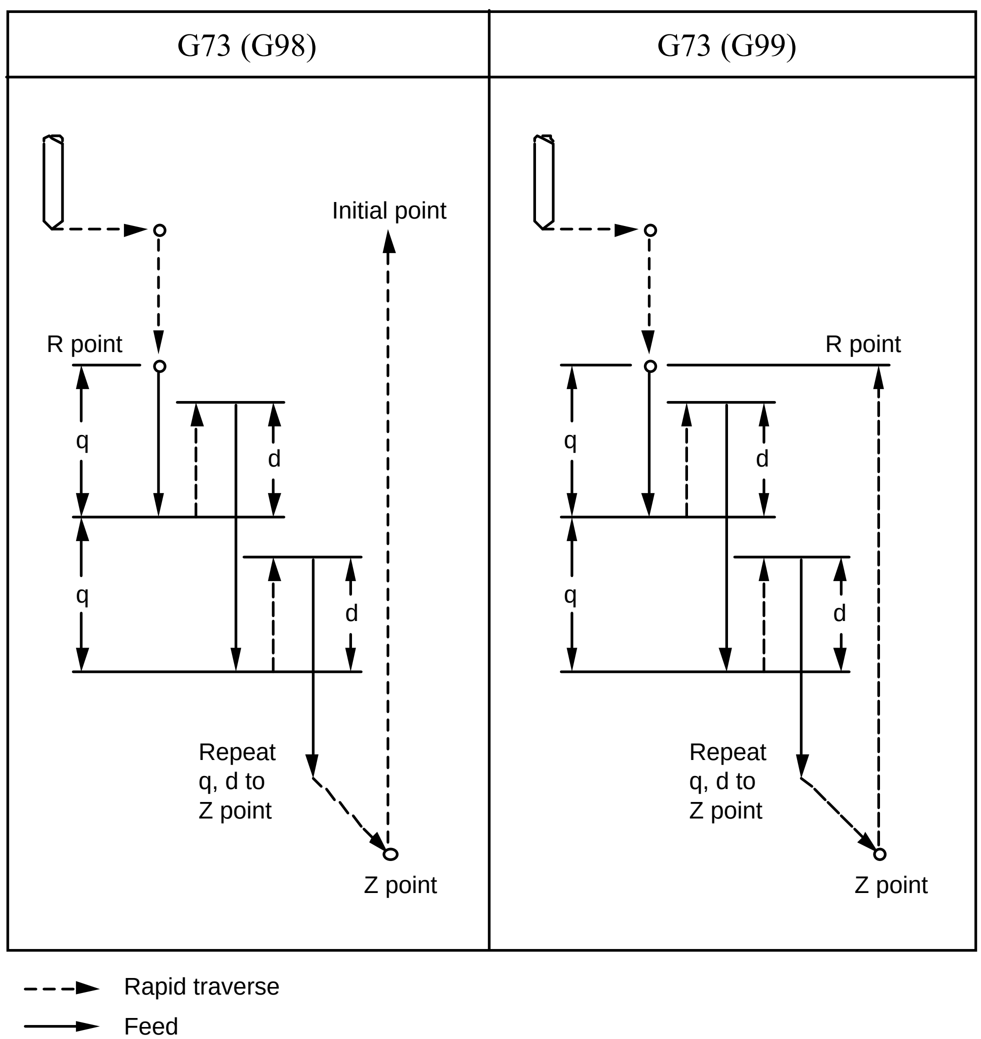


Figure 4-20 High-speed Peck Drilling Cycle (G73)

4.7.3. Left-hand Tapping Cycle (G74)

Figure 4-21 shows G74 tapping cycle for both G98 and G99 modes. The spindle must be set ON in the CCW direction before executing G74. This cycle turns the spindle CW while retracting from tapped hole. CAUTION: Accurate spindle position is not guaranteed since the spindle velocity is voltage controlled without position feedback. A flexible chuck coupling is required for tapping cycles.

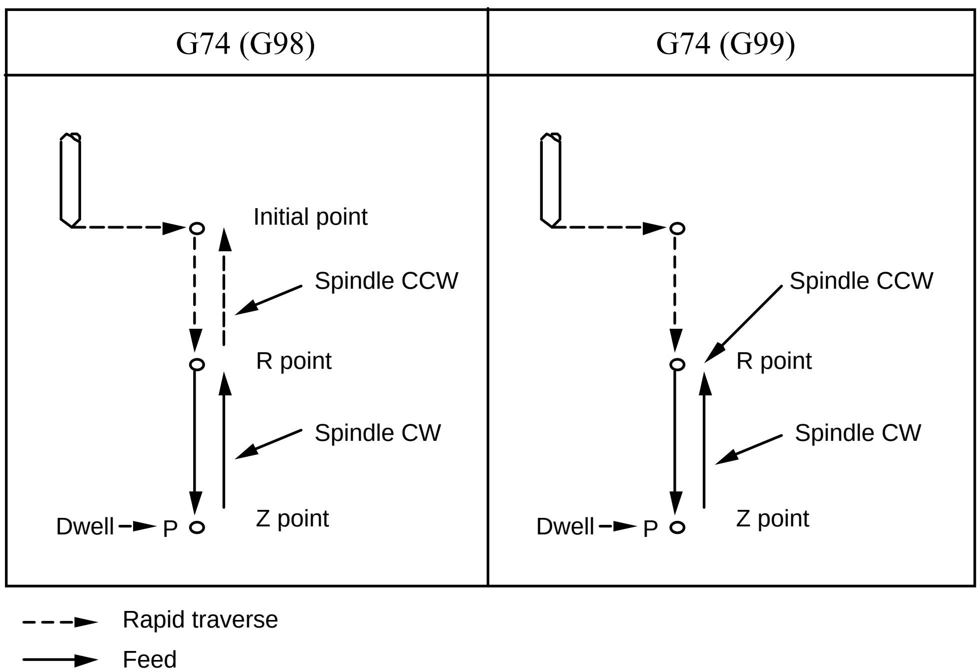


Figure 4-21 Left-handed Tapping Cycle (G74)

4.7.4. Spot and Counter Boring Drilling Cycles (G81, G82)

G81 and G82 are the same except that G82 provides a dwell, specified by the P address, at the bottom of the hole. A dwell at the bottom of the hole improves the hole depth precision in blind hole drilling. The retraction from the Z-depth is at rapid. Figure 4-22 shows the operation of the spot and counter boring drilling cycles for both G98 and G99 modes.

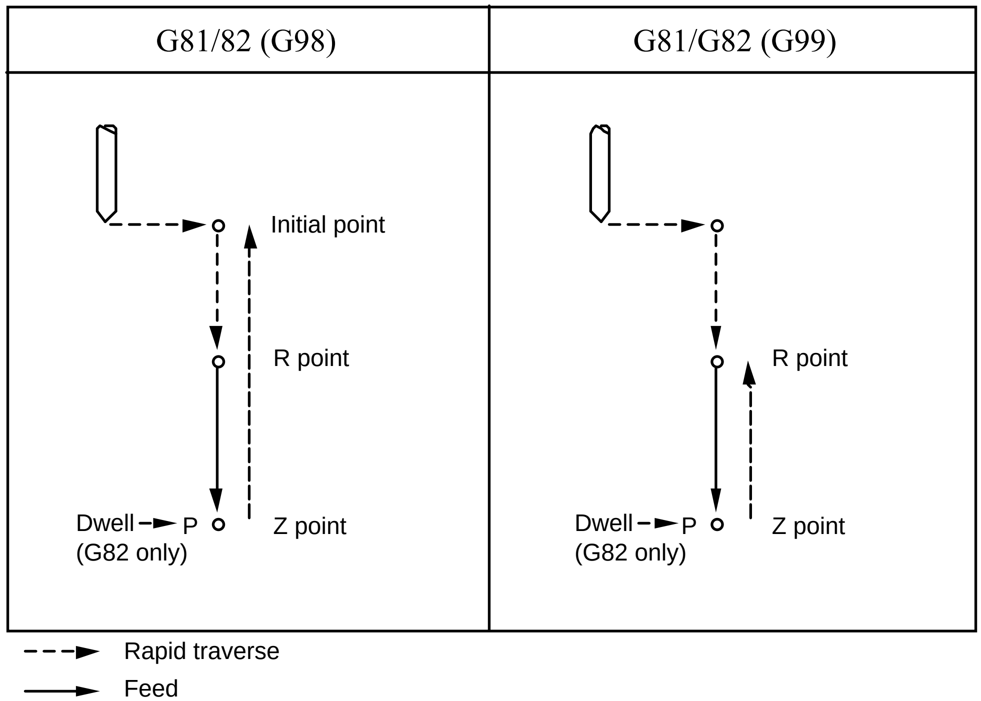


Figure 4-22 Spot and Counter Boring Drilling Cycles (G81, G82)

4.7.5. Peck Drilling Cycle (G83)

Figure 4-23 shows the peck drilling cycle for both G98 and G99 modes. The standard peck drilling cycle is similar to G73 with retraction to the R-point instead of the U distance. The Q address specifies the cut-in distance as an incremental value. After the first cut-in, the tool will retract to the R point and then rapid back into the hole to the drill retract distance above the cut-in depth. The final cut-in will be at the specified feed rate. The retraction is to the R point each cycle of the cut-in at the rapid traverse rate.

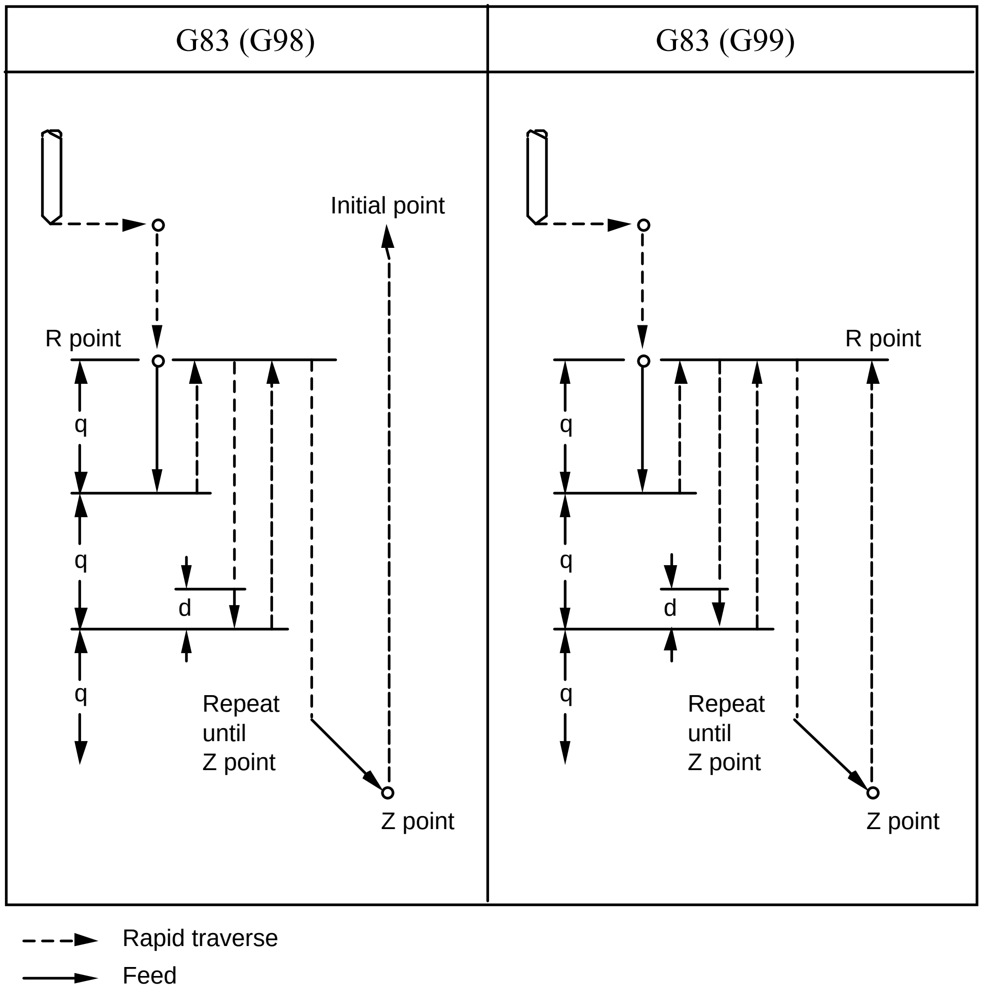


Figure 4-23 Peck Drilling Cycle (G83)

4.7.6. Tapping Cycle (G84)

Figure 4-24 shows the operation of the tapping cycle for both G98 and G99 modes. Tapping cycles require a flexible chuck coupling. The spindle must be set ON in the CW direction before executing G84. The spindle is reversed at the bottom of the hole and the tapping cycle performed. CAUTION: Accurate spindle position is not guaranteed since the spindle velocity is voltage controlled without position feedback. A flexible chuck coupling is required for tapping cycles.

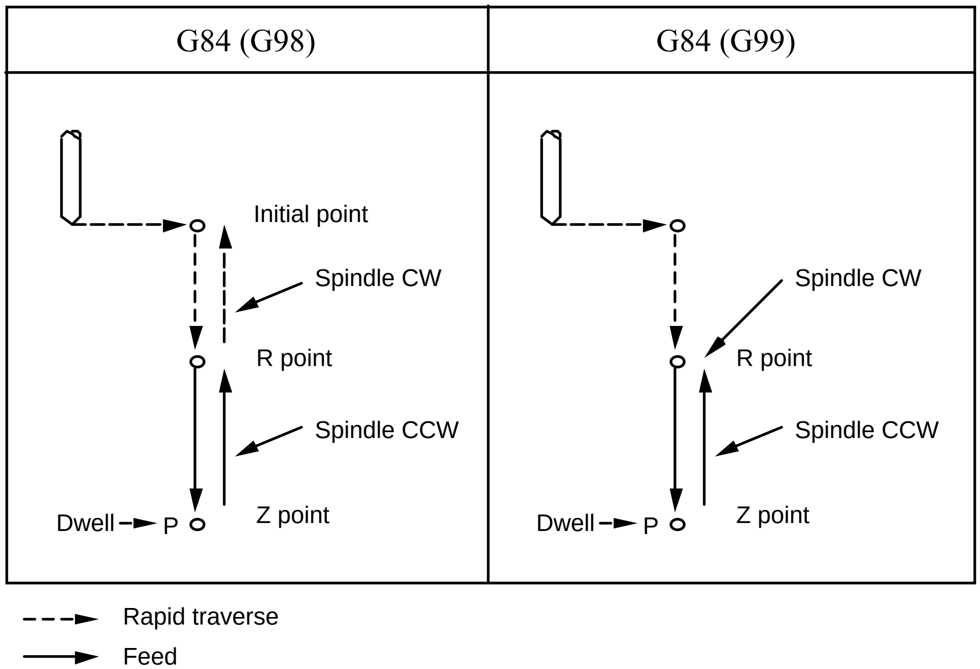


Figure 4-24 Tapping Cycle (G84)

4.7.7. Boring Cycle (G85)

Figure 4-25 shows G85 boring cycle for both G98 and G99 modes. This cycle is the same as G84, but the spindle is not reversed at the bottom of the hole.

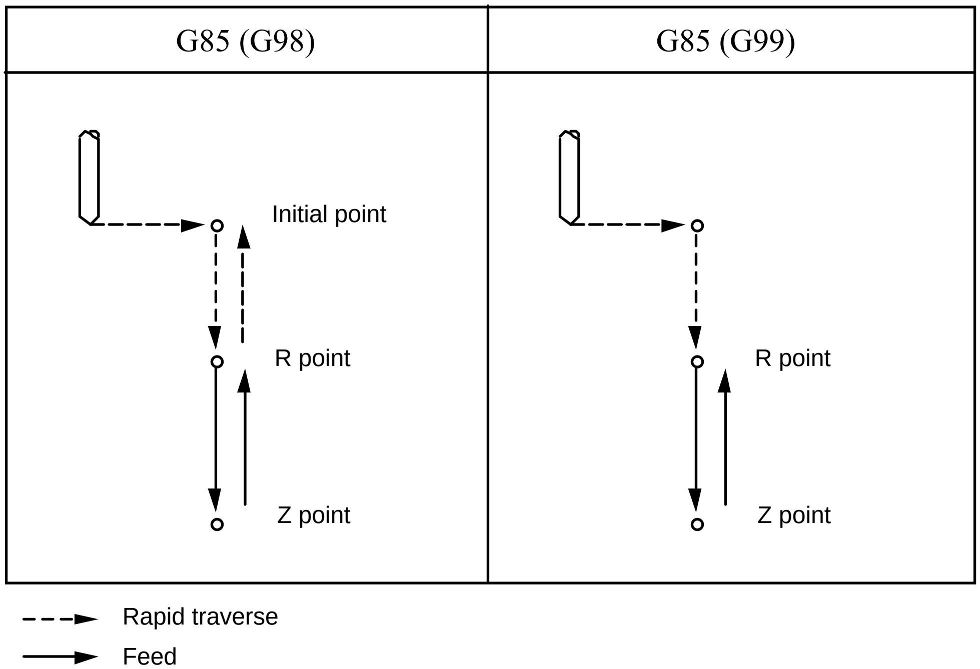


Figure 4-25 Boring Cycle (G85)

4.7.8. Boring Cycle (G86)

Figure 4-26 shows G86 boring cycle for both G98 and G99 modes. This is the same as G81, but the spindle stops at the bottom of the hole and is retracted at the rapid traverse rate.

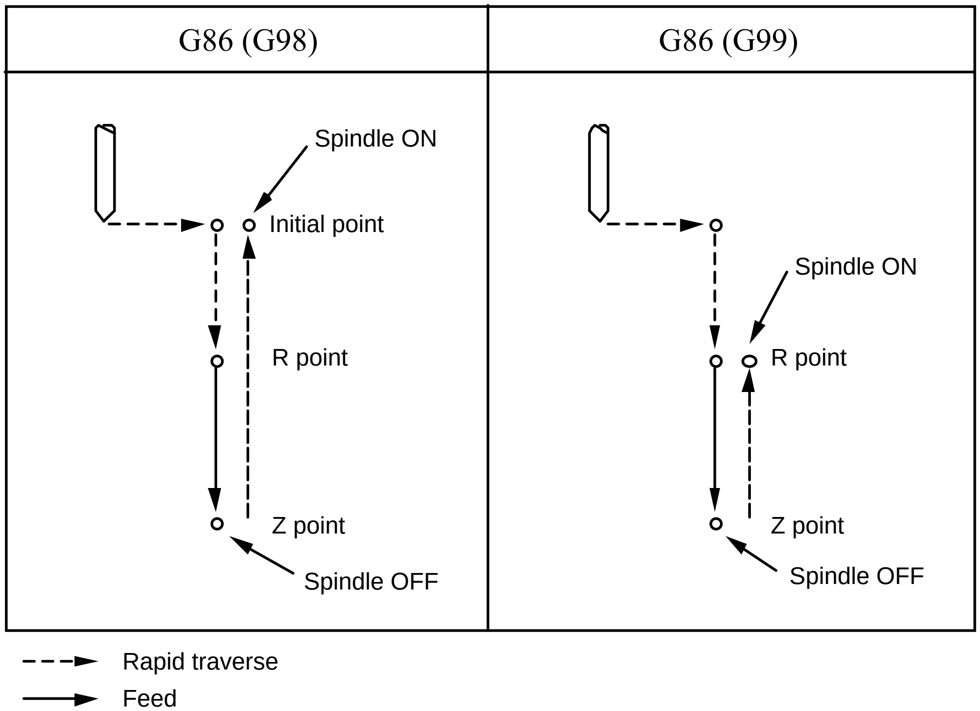


Figure 4-26 Boring Cycle (G86)

4.7.9. Back Boring Cycle (G87)

Figure 4-27 shows G87 boring cycle for both G98 and G99 modes. At the bottom of the hole the spindle stops and the system enters the manual mode. The tool then can be retracted manually. When machining is restarted, the tool returns to the R or initial level point according to G98 or G99. The spindle then rotates CW and execution of the NC program is continued.

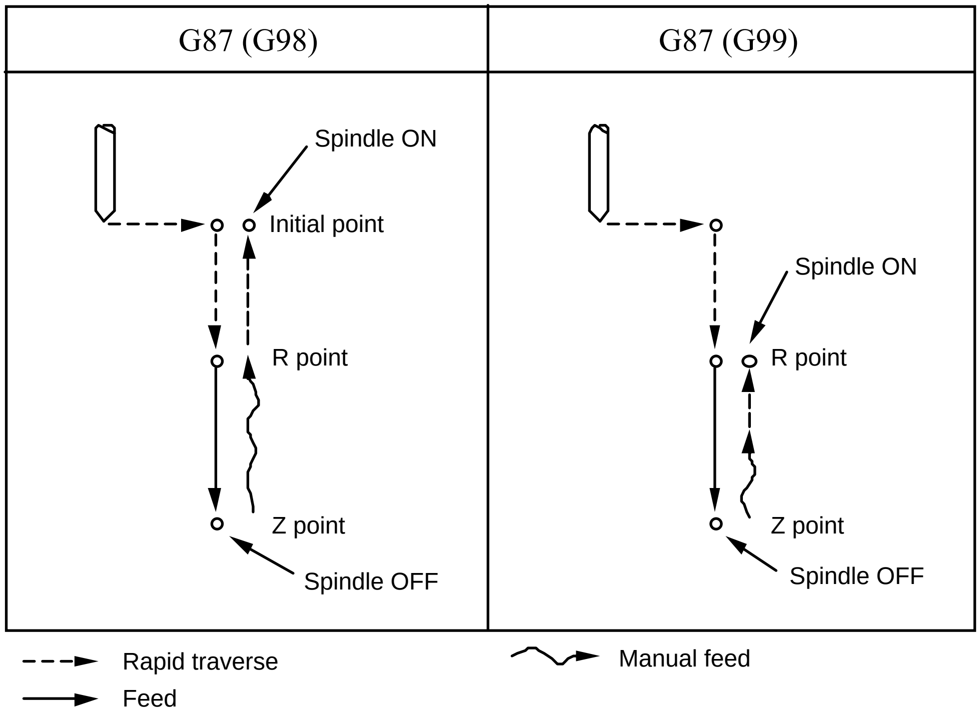


Figure 4-27 Back Boring Cycle (G87)

4.7.10. Boring Cycle (G88)

Figure 4-28 shows G88 boring cycle for both G98 and G99 modes. This is the same as G87 except that the spindle stops after dwelling *P* seconds at the bottom of the hole.

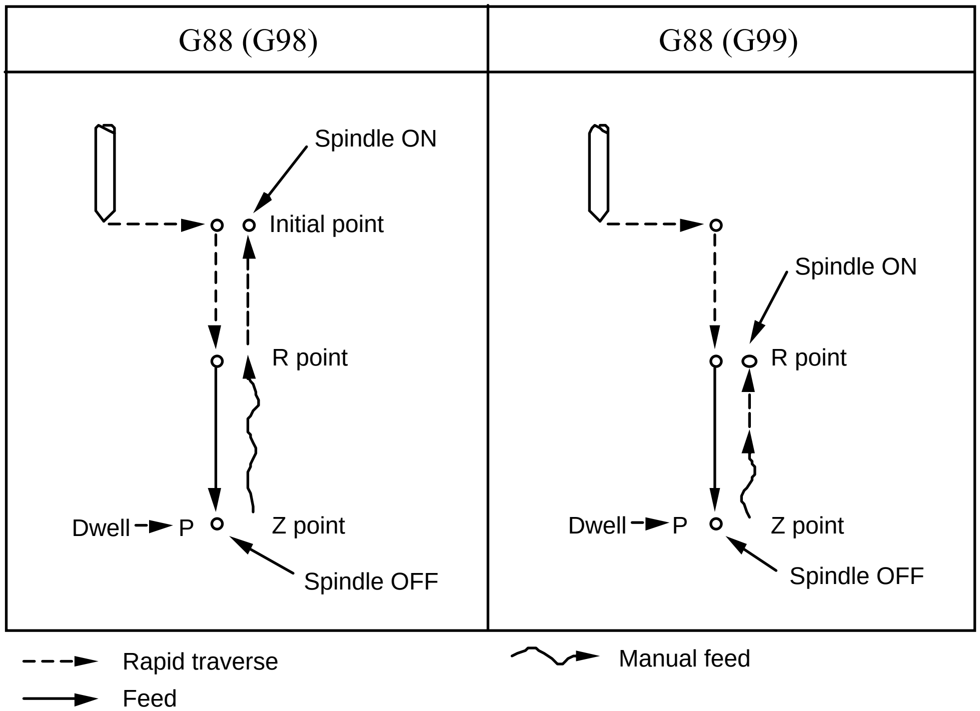


Figure 4-28 Boring Cycle (G88)

4.7.11. Boring Cycle (G89)

Figure 4-29 shows G89 boring cycle for both G98 and G99 modes. This is the same as G85 except dwell is performed at the bottom of the hole.

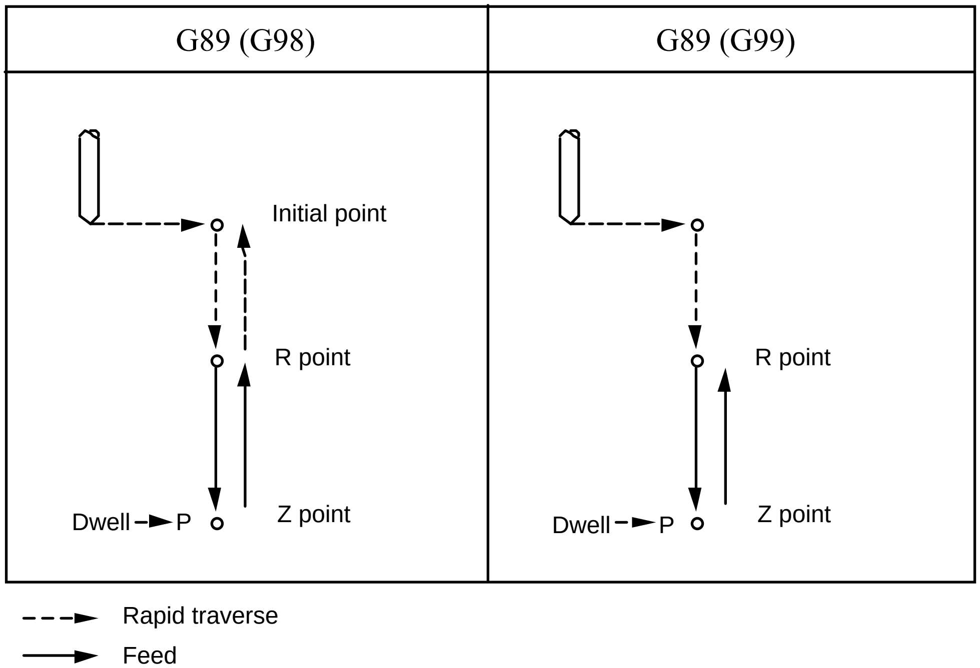


Figure 4-29 Boring Cycle (G89)

4.8. MISCELLANEOUS FUNCTIONS (M-CODES)

Miscellaneous functions are used to effect program execution, control external devices and support DOS calls from within P-CNC.

4.8.1. Program Execution (M00 - M02, M06, M47, M98, M99)

Program execution may be controlled using M-codes to pause between blocks of code, jump to sub-programs and terminate the NC program. These features assist in developing efficient code and allow operator intervention using different types of pause functions.

4.8.1.1. M00, M01, M02 (Program stop)

M00 and M02 are treated equivalent and used to terminate program execution. A P-CNC program containing one or more sub-programs must use M00 or M02 to terminate the main program. It is recommended that M02 be used to terminate the main program even if no sub-program is employed. The execution of the NC program will be terminated once a block containing M00 or M02 is encountered whether or not there is any code following this block.

M01 is equivalent to M00 but is enabled or disabled by the setting in the P-CNC SETUP menu. When disabled, M01 is ignored.

4.8.1.2. M06 (Manual tool change)

M06 is equivalent to G04 but displays a message to perform a manual tool change. This command should be used when the operator needs to be alerted to change the tool specified by the T address. When T is present in the same block, it will be interpreted before the M06 alarm and message display.

4.8.1.3 M47 (Return to Program Start)

M47 simply jumps to the beginning of the NC program and continues execution from the first block of code. This feature is likely to cause the NC program to run continuously in a repetitive fashion.

4.8.1.4. M98, M99 (Sub-programs)

Any valid NC block can be included in a sub-program body as illustrated in Figure 4-30. If M00 or M02 appears in a sub-program before M99, the execution of your P-CNC program will terminate right after this block is executed, even if there are still unexecuted blocks in the calling program. A sub-program must be ended with a block containing M99. M99 need not be specified in a block by itself.

```
N1000   (sub-program designator)
•
•                 (sub-program body)
•
M99               (end of sub-program)
```

Figure 4-30. A Sub-program

Sub-programs are invoked by M98 and must be identified by the program number following address P. For example, the following CNC code will cause a sub-program designated by 1000 to be executed completely before the next block is interpreted.

M98 P1000

A sub-program can be executed repeatedly by just one call along with the loop (L) function. The positive integer associated with an L address indicates the number of repetitions the called sub-program is to be executed. For example, the following CNC code

M98 P1000 L3

causes the sub-program designated by 1000 to be executed three times before it returns to the caller. Therefore, an omitted L function in the previous example is equivalent to that in which L1 is specified. Activities involved in executing the example program (Figure 4-31) are:

(1) the main program calls sub-program (Sub 1) designated by N1000 at N200. Sub 1 will be executed three times before it returns to the main program (1'). The main program resumes execution on the block immediately following N200.

(2) Sub 1 calls sub-program Sub 2 designated by O10 at N2000. The call will cause Sub 2 to be executed twice. Control returns to the block following N2000 (2').

(3) Sub 1 calls Sub 2 again at N3000. This time, Sub 2 is executed only once and then returns to the caller (3').

(4) the main program calls Sub 2 at N300. Sub 2 returns to the main program (4') after one time execution.

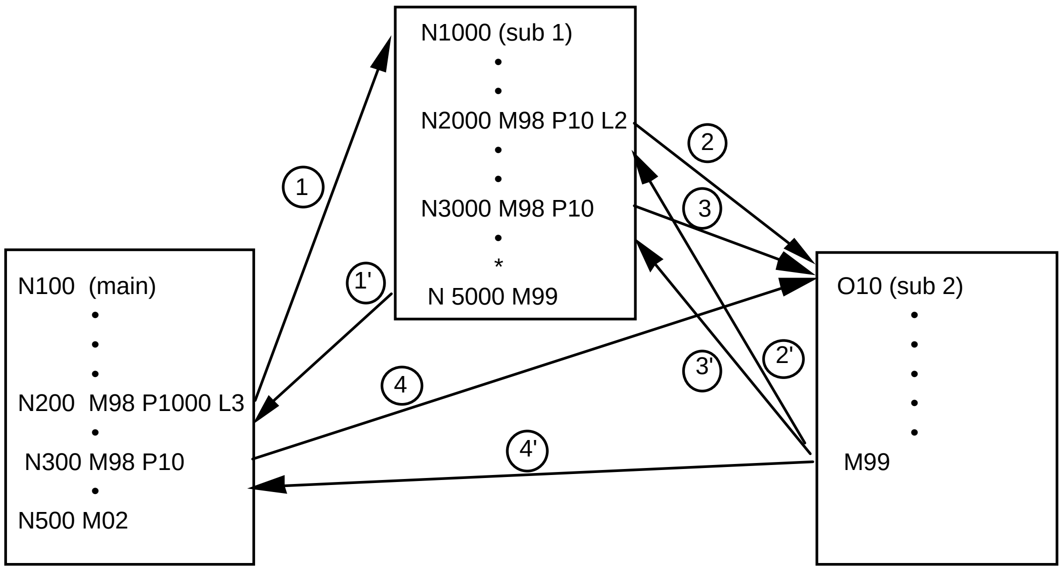


Figure 4-31. An Example Program

4.8.2. Spindle Control (M03 - M05)

Spindle on/off and direction control are commanded with three M codes (M03 - M05). M03 turns the spindle ON clockwise. M04 turns the spindle ON counter clockwise. M05 turns the spindle OFF. Recall that the spindle speed is set by the spindle address (S). Despite the fact that the spindle speed may be controlled, P-CNC provides these two outputs for fixed speed spindle motors. The M40, M41, and M42 M-codes may also be used to switch between high and low gears. The high/low output at relay H can also be automatically switched based on the current spindle speed command.

4.8.2.1. M03 - M05 (Spindle ON/OFF)

M03 turns the spindle on clockwise. M04 turns the spindle on counter clockwise. M05 turns the spindle off. Recall the spindle speed is set by the spindle function (S). Despite the fact that the spindle speed may be controlled, P-CNC provides two signals indicating the current status of the spindle. These two signals are sufficient for operating a fixed speed spindle motor.

4.8.2.2. M13 - M14 (Coolant and Spindle ON)

M13 and M14 are provided for the convenience of turning the coolant and spindle on with the same M code. In this regard, M13 and M14 are redundant and need not be used when writing CNC programs.

M13 turns the coolant on and the spindle on in the CW direction. M14 turns the coolant on and the spindle on in the CCW direction.

4.8.3. ON/OFF Control Outputs (M08 - M11, M50 - M53)

M08 turns the coolant on and M09 turns the coolant off. M10 and M11 codes are exactly the same as M08 and M09, respectively, except they are designated to turn the clamp on and off. The switches may also be operated manually during a jog session. When P-CNC is first loaded, all outputs are switched off. No automatic reset (turn off) is performed when a NC program starts to execute. Subsequent operations on a switch, programmed or not, determine the state of the outputs. If a run-time error occurs during execution of a NC program, all outputs are automatically turned off.

Option 1 and Option 2 outputs are treated differently than M08 and M09, when executing a sequence of blocks of continuous profiles. M50, M51, M52, and M53 are stored in the look-ahead buffer as is G01, G02 and G03. This feature allows on/off control during continuous profiles without breaking up the profile sequence. Within a block, the sequence of execution is still M_{on}, motion, and then M_{off} regardless of the order in which they appear. M50 and M51 codes (Option 1 on/off), and M52 and M53 codes (Option 2 on/off), are user designated. Use the OEM Setup Utility to change the display to appropriate titles.

4.8.4. Bypass Feed Rate Override (M48, M49)

M49 commands P-CNC to ignore the feed rate override switch and to run all feeds at 100 percent. M48 cancels M49 which is the state of the default configuration.

4.8.5. DOS Shell Execution (M90 - M95)

M90 through M95 execute DOS shells to the operating system. When commanded in a NC program, P-CNC halts execution to perform the DOS shell and resumes execution after the shell has been completed. M90 through M95 descriptions must be typed in at the P-CNC SETUP menu for the desired activity. Execution of other programs or DOS commands are examples of possible uses for these M codes. This feature allows maximum flexibility for third party enhancements to P-CNC.