GE Intelligent Platforms

Programmable Control Products

Series 90 PLC*
Axis Positioning Module (APM)

Programmer's Manual

GFK–0664A *January 2010*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note: Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Intelligent Platforms assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Intelligent Platforms makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

Technical Support

If you purchased this product through an Authorized Channel Partner, please contact them directly.

General Contact Information

Online Technical Support and GlobalCare: www.ge-ip.com/support

Additional information: www.ge-ip.com

Technical Support

If you have technical problems that cannot be resolved with the information in this guide, please contact us by telephone or email, or on the web at www.ge-ip.com/support

Americas

Online Technical Support: www.ge-ip.com/support

Phone: 1-800-433-2682

International Americas Direct Dial: 1-434-978-5100

Technical Support Email: support.ip@ge.com

Customer Care Email: customercare.ip@ge.com

Primary language of support: English

Europe, the Middle East, and Africa

Online Technical Support: www.ge-ip.com/support

Phone: +800-1-433-2682

Technical Support Email: support.emea.ip@ge.com

Customer Care Email: customercare.emea.ip@ge.com

Primary languages of support: English, French, German, Italian, Czech, Spanish

Asia Pacific

Online Technical Support: www.ge-ip.com/support

Phone: +86-400-820-8208

+86-21-3217-4826 (India, Indonesia, and Pakistan)

Technical Support Email: support.cn.ip@ge.com (China)
support.jp.ip@ge.com (Japan)
support.in.ip@ge.com (remaining Asia customers)

Customer Care Email: customercare.apo.ip@ge.com
customercare.cn.ip@ge.com (China)

Content of This Manual

This manual describes how to program the Series 90 Programmable Controller Axis Positioning Modules. Included in Chapter 2 is a complete description of the Motion Programmer Software which is used to create Motion Programs. In addition, Chapter 3 provides several practical examples including combinations of Motion Commands. The Appendices provide detailed descriptions of the Motion Commands as well as the Status and Control data that are transferred automatically each sweep from the PLC to the APM.

Chapter 1. **Introduction to Motion Programming:** This chapter is an overview of both the Motion Programmer Software and the Program Zero Editor (which is part of the Logicmaster 90 Configuration software). Operation of the Program Zero Editor is provided in GFK–0840, *The APM30 Standard Mode User's Manual*, but the appendices and some of the program examples in this manual apply to it as well.

Chapter 2. **The Motion Programmer:** This chapter is a comprehensive description of the Motion Programmer Software.

Chapter 3. **Motion Program Examples:** This chapter includes several examples describing how to combine Motion Commands to produce desired effects.

Appendix A. **The APM Motion Commands:** This appendix includes all the Motion Commands with detailed descriptions.

Appendix B. **The APM Discrete Commands (%Q):** This appendix includes all the Discrete Commands with detailed descriptions.

Appendix C. **The APM Immediate Commands (%AQ):** This appendix includes all the Immediate Commands with detailed descriptions.

Appendix D. **The APM Status Bits (%I):** This appendix includes all the Status Bits with detailed descriptions.

Appendix E. **The APM Status Words (%AI):** This appendix includes all the Status Words with detailed descriptions. APM Error Codes are listed here.

Related Publications

GFK–0840 *Series 90™–30 Programmable Controllers Axis Positioning Module (APM30) – Standard Mode User's Manual*

GFK–0781 *Series 90™–30 Programmable Controllers Axis Positioning Module (APM30) – Follower Mode User's Manual*

GFK–0466 *Logicmaster™ 90 Series 90™–30 and 90™–20 Programming Software User's Manual*

GFK–0265 *Logicmaster™ 90 Programming Software Reference Manual*

GFK–0356 *Series 90™–30 Programmable Controller Installation and Operation Manual*

At GE Intelligent Platforms, we strive to produce quality technical documentation. After you have used this manual, please take a few mooments to complete and return the Reader's Comment Card located on the next page.

Contents

Chapter 1 Introduction to APM Programming 1-1

Introduction to APM Programming 1-1
The Motion Programmer 1-2
Instruction Format 1-3
The Program Zero Editor 1-4

Chapter 2 Motion Programmer 2-1

Section 1: Starting-Up the Motion Programmer 2-2

Installing the Motion Programmer Software 2-2
Starting-Up the Motion Programmer 2-4
Creating and Selecting a Folder 2-4
Creating an APM File 2-7
The Motion Programmer Main Screen 2-8
Motion Programmer Functions 2-8
Message Line 2-9
Command Line 2-9
Status Line Definition 2-9
Help and Special Function Keys 2-10
Establishing Communications Between the APM and Programmer 2-11

Section 2. Creating and Editing Programs and Subroutines 2-12

Displaying a Program 2-12
Functions Accessed from Display Mode 2-13
Additional Status Fields 2-13
Program Edit Fields 2-14
Moving the Cursor 2-15
APM Motion Command List 2-15
Inserting Lines in a Program or Subroutine 2-16
Making Changes to a Program or Subroutine 2-17
Deleting a Line 2-18
Checking Program or Subroutine Syntax 2-18
Renumbering Block Numbers 2-19
Using the GoTo Function 2-19
Size Limitations 2-20

Section 3. Status Functions 2-21

Section 4. Using APM Files and Folders 2-22

APM File Functions (F5) 2-23
Selecting and Creating APM Files 2-23
Deleting an APM File 2-25
Selecting and Configuring APM Programs and Subroutines 2-26
Copying APM Files 2-28
Folder Functions (F8) 2-29
Selecting and Creating a Folder 2-30
Deleting a Folder 2-31
Backing-Up All APM Files in a Folder 2-32
Restoring All APM Files in a Folder 2-33
Clearing All APM Files in a Folder 2-34
Locking and Unlocking Current Program Folder 2-35
Copying All APM Files in a Folder 2-36

Section 5. Setting-Up the Programmer Ports 2-37

Setting-Up the Printer Serial Port 2-37
Setting the Programmer Mode 2-39
Setting-Up APM Communications 2-40
The Select SNP Connections Screen 2-40
The PLC Communications Port Setup Screen 2-41

Section 6. Loading and Storing APM Files 2-43

Loading from the APM to the Programmer 2-43
Storing from the Programmer to the APM 2-45
Verifying the APM with the Programmer 2-47
Clearing APM Memory 2-48
EEPROM Operations 2-50

Section 7. Printing APM Files 2-52

Setting-Up Printer Parameters 2-52
Selecting the Screen Print Destination 2-54
Printing Programs and Subroutines 2-55
Pausing the Print Operation 2-56

Chapter 3 Motion Program Examples 3-1

Calculating Move Distances 3-1
Single Pmove 3-3
Cmove Followed by Pmove 3-5
Multiple Cmoves 3-7
Using the Jump Command 3-8

Contents

Appendix A APM Motion Commands A-1

Acceleration A-2
Block A-2
Call Subroutine A-3
Cmove A-3
Dwell A-4
Jump A-4
Load Parameter A-5
Null A-5
Pmove A-6
Velocity A-7
Wait A-7

Appendix B APM Discrete Commands (%Q) B-1

Abort All Moves B-2
Clear Error B-2
CTL09-CTL12 Output Controls B-3
Enable Drive B-3
Execute Motion Program 0-10 ..... B-4
Feedhold (On Transition) B-4
Feedhold (Off Transition) B-5
Find Home B-5
Jog Minus B-7
Jog Plus B-8
Reset Strobe Flag B-9

Appendix C APM Immediate Commands (%AQ) C-1

Force D/A Output C-2
Integrator Time Constant C-2
Jog Acceleration C-3
Jog Velocity C-3
Load Parameter Immediate C-4
Move At Velocity C-4
Position Increment C-5
Position Loop Time Constant C-5
Rate Override C-6
Set Position C-6
Velocity Feedforward C-7

Appendix D APM Status Bits (%1) D-1

Axis Enabled D-2
Drive Enabled D-2
Error D-2
Faceplate Input Status CTL 01-08 ..... D-3
In Error Limit D-3
In Zone D-4
Moving D-4
PLC Control Active D-4
Position Strobe D-5
Position Valid D-5
Program Active D-6

Appendix E APM Status Words (%AI) E-1

Actual Position E-2
Actual Velocity E-2
Analog Input E-2
Command Block Number E-2
Commanded Position E-2
Commanded Velocity E-2
Position Error E-2
Status Code E-2
Strobe Position E-3
The Status Code Word E-3

Contents

Figure 1-1. Example Motion Programmer Editor Screen 1-3
Figure 1-2. Example Program Zero Editor Screen 1-4
Figure 2-1. Logicmaster 90 Startup Screen 2-4
Figure 2-2. Startup Folder Selection Screen 2-5
Figure 2-3. Startup File Creation Screen 2-7
Figure 2-4. Motion Programmer Main Screen 2-8
Figure 2-5. Display Mode - Null Program 2-13
Figure 2-6. Display Mode - Example Instructions 2-14
Figure 2-7. Status and Functions Screen 2-21
Figure 2-8. APM File Functions Screen 2-23
Figure 2-9. Select or Create APM File Screen 2-24
Figure 2-10. Delete APM File Screen 2-25
Figure 2-11. The Select Motion Program Screen 2-26
Figure 2-12. COPY APM File Screen 2-28
Figure 2-13. APM Folder Functions Screen 2-29
Figure 2-14. Select or Create Folder Screen 2-30
Figure 2-15. Delete Folder Screen 2-31
Figure 2-16. Backup Folder Screen 2-32
Figure 2-17. Restore Folder Screen 2-33
Figure 2-18. Clear Folder Screen 2-34
Figure 2-19. Lock or Unlock Current Folder Screen 2-35
Figure 2-20. Copy Folder Screen 2-36
Figure 2-21. The Setup Functions Screen 2-37
Figure 2-22. The Printer Port Setup Screen 2-38
Figure 2-23. The Set Programmer Mode Screen 2-39
Figure 2-24. The Select APM Connections Screen 2-41
Figure 2-25. APM Communications Port Setup Screen 2-42
Figure 2-26. Utilities Functions Screen 2-43
Figure 2-27. Load from APM to Programmer Screen 2-44
Figure 2-28. Store from Programmer to APM Screen 2-45
Figure 2-29. Verify APM with Programmer Screen 2-47
Figure 2-30. Miscompares on the Verify Screen 2-48
Figure 2-31. Clear APM User Memory Screen 2-49
Figure 2-32. EEPROM Operations Screen 2-50
Figure 2-33. Print Functions Screen 2-52
Figure 2-34. Setup Printer Parameters Screen 2-53
Figure 2-35. Select Screen Print Destination Screen 2-54
Figure 2-36. Print Motion Program Screen 2-55
Figure 3-1. Single Pmove Profile 3-3
Figure 3-2. Cmove followed by Pmove Profile 3-5
Figure 3-3. Multiple Cmoves Profile 3-7
Figure 3-4. Jump Command Condition Testing 3-8
Figure E-1. Status Code Organization E-3

Contents

Table 2-1. The APM Motion Command List 2-15
Table 3-1. Formulas for Calculating Linear Move Distances 3-1
Table 3-2. Formulas for Calculating S-Curve Move Distances 3-2
Table B-1. %Q Discrete Commands B-1
Table C-1. Immediate Commands Using the 6-Byte Format C-1
Table D-1. %I Status Bits D-1
Table E-1. %AI Status Words (1 Axis) E-1
Table E-2. %AI Status Words (2 Axis) E-1
Table E-3. Status Word Error Codes E-4
Table E-3. Status Word Error Codes (Continued) E-5

Chapter 1 Introduction to APM Programming

This chapter is an overview of Series 90 APM Motion Programming. Both the Motion Programmer software package and the Program Zero editor (in the Logicmaster configuration software package) are discussed.

Introduction to APM Programming

A Motion Program can be created for the APM30 by either of two methods:

- **The Motion Programmer.** The Motion Programmer provides the capability of writing English-language motion programs, storing the programs on disks, and downloading the programs to the APM30 as desired.
- **The Program Zero Editor.** The Logicmaster Configuration package contains an editor for creating a short (20 commands maximum) program defined as Program Zero. The commands are entered in an English-language format similar to the Motion Programmer. No ladder programming is needed.

Chapter 2 of this manual, The Motion Programmer, describes how to use the Motion Programmer software. The Program Zero Editor is part of the Logicmaster configuration software. It's operation is described in GFK-0707, *Series 90-30 Axis Positioning Module (APM30) - Standard Mode User's Manual*, and GFK-0781, *Series 90-30 Axis Positioning Module (APM30) - Follower Mode User's Manual*

The information contained in Chapter 3, Motion Program Examples, and in Appendices A-E, are applicable to both programming packages.

The Motion Programmer

Motion programs are created primarily using the Motion Programmer. The Motion Programmer is a powerful, English-language programmer for the APM30. It can be executed as a stand-alone package or from the Logicmaster Startup Menu. Programs and Subroutines are developed by selecting commands using function keys and then entering values into Operand fields to complete the instruction. The commands which can be programmed in the Motion Programmer are listed below.

Command	Description
ACCEL	Acceleration
CALL	Call Subroutine
CMOVE	Continuous Move
DWELL	Dwell
JUMP	Jump
LOAD	Load Parameter
NULL	Causes no action
PMOVE	Positioning Move
VELOCITY	Velocity
WAIT	Wait

Features of the Motion Programmer

- Create and Edit Programs and Subroutines
- Setup the Programmer Ports
- Load and Store Program files between the Motion Programmer and APM30
- Print APM30 Files

Features of the Program/Subroutine Editor

- Insert Program Lines
- Edit Program Lines
- Delete Program Line
- Check Program Syntax
- Renumber Block Numbers
- GOTO Function

The Figure 1-1 illustrates the editor screen with sample instructions.

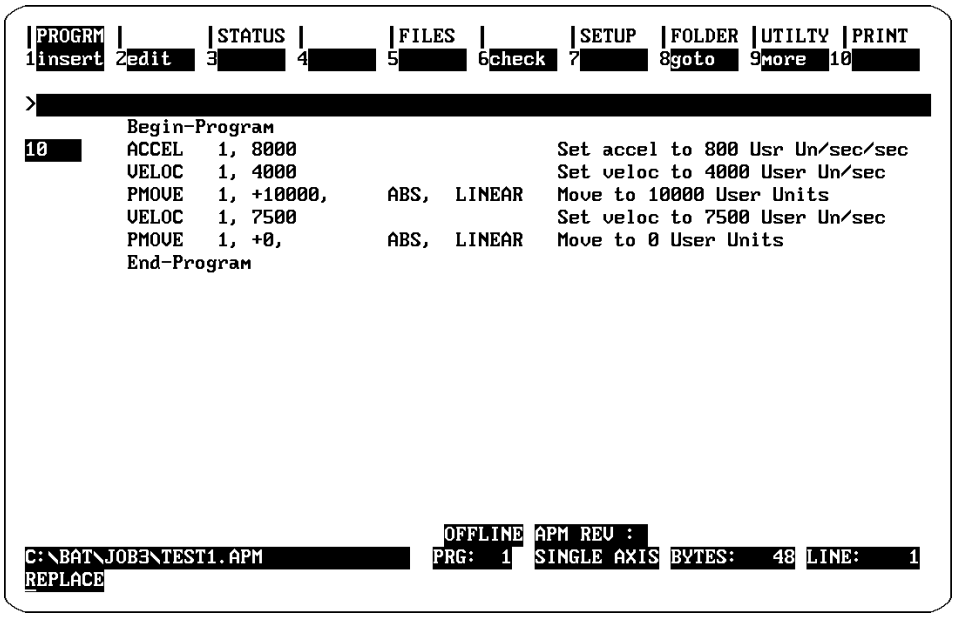


Figure 1-1. Example Motion Programmer Editor Screen

Instruction Format

A program or subroutine instruction consists of a command and associated data describing the command. The command and data are entered into areas on the program editor screen called fields. These fields are described below. From left to right the fields are:

Block Number Field. This field is *optional.* The Block Number field permits the entry of a number to designate the beginning of a block of program instructions. It is used as a place for *Jump* commands to go to and to provide a synchronization point for 2-axis programs.

Synchronous Block Specifier Field. This field is *optional.* This field appears only in multi-axis programs. An "X" specifies a block as a sync block. A space indicates a standard block.

Command Name Field. This field is *required.* It is an English-language designation for the command.

Operand Fields. Any operand field displayed is *required.* When you enter a Command name, the required Operand fields will be displayed with their default value. The number of Operand fields displayed varies from one to four depending on the Command name. Operand field 1 is reserved for the axis number. The content of the other 3 fields varies depending on the command.

Comments Field. This field is *optional.* It allows approximately 30 characters of comment on the same line as the Motion Command. Additional full 80-character line comments can also be added.

The Program Zero Editor

The Program Zero Editor allows you to program a short (20 command maximum) program without having to use the Motion Programmer. Program Zero is stored in PLC non-volatile memory when the Configuration file (created in the Logicmaster 90 configuration package) is stored to the PLC-CPU. Program Zero is transferred to the APM whenever the PLC transitions from STOP to RUN. The commands which can be programmed using the Program Zero Programmer are listed below.

Command	Description
ACCEL	Acceleration
BLOCK	AssignsBlock Number
CMOVE	ContinuousMove
DWELL	Dwell
JUMP	Jump
LOAD	Load Parameter
NULL	Causes no action
PMOVE	PositioningMove
VELOCITY	Velocity
WAIT	Wait

The figure below illustrates the editor screen with sample instructions.

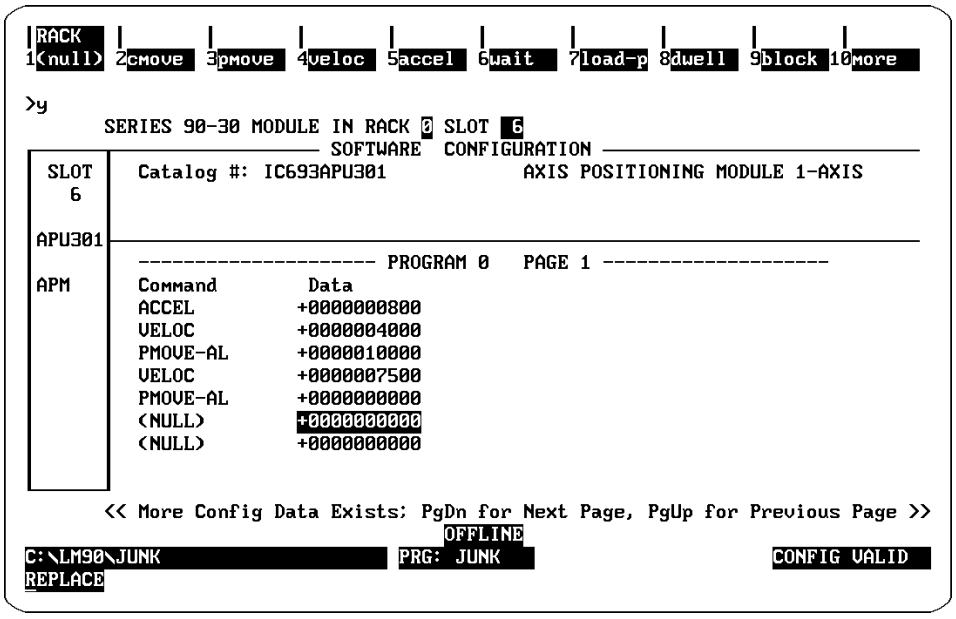


Figure 1-2. Example Program Zero Editor Screen

Instruction Format

A Program Zero instruction consists of a command and associated data describing the command. The command and data are entered into areas on the program editor screen called fields. These fields are described below. From left to right the fields are:

Command Name Field. It is an English-language designation for the command. To enter a command, position the cursor on a command name field then press the desired **Function** key (F1-F10). Most of the commands fall into groups such as CMOVE, PMOVE, VELOC, ACCEL, etc. The variations of these commands can be displayed after you press the **Function** key for the particular group. To display the JUMP command, first press the **F10 more** Function key.

For example, if you want to program a CMOVE-IS-P (Continuous Move, Incremental, S-Curve, using a parameter), first move to a command field, then press the *cmove* function key. This causes CMOVE-AL to be displayed in the command name field of the screen. Press the **Tab** key to cycle through all the variations of the command.

Operand Fields. Paired with each command is a data field,. and, if you are configuring a 2-axis APM, an axis number field. In the data field you enter either a signed integer or the number of an APM parameter, as appropriate for the configured command.

Chapter 2 Motion Programmer

The Motion Programmer software package is a powerful, English-language programmer for the APM. Motion programs are developed using English-language commands selected through function keys. This reduces programming time and makes the programs you create easy to follow. The APM Motion Programmer operates much like the other packages in Logicmaster 90 Programming Software.

This chapter describes how to use all the features of the Motion Programmer. The chapter is divided into the following sections.

- Section 1. Starting-Up the Motion Programmer
- Section 2. Creating and Editing Programs and Subroutines
- Section 3. Status Functions
- Section 4. Using APM Files and Folders
- Section 5. Setting-Up the Programmer Ports
- Section 6. Loading and Storing APM Files
- Section 7. Printing APM Files

Note

The purpose of this chapter is to familiarize you with the procedures for entering motion commands and with the other editor functions. For a detailed description of how each command works, etc. refer to Appendix A, The APM Motion Commands.

Section 1: Starting-Up the Motion Programmer

The Motion Programmer can be executed as a stand-alone package or as part of the Logicmaster 90 Programming Software.

This section covers the following main topics.

- Installing the Motion Programmer Software
- Starting-Up the Motion Programmer
- The Motion Programmer Main Screen (Function and Status Line Descriptions)

Installing the Motion Programmer Software

Before starting the installation procedure, you may wish to create backups for the current CONFIG.SYS file and AUTOEXEC.BAT file. The installation software will modify these files for you automatically if you so choose.

To install the Motion Programmer Software:

1. Insert the Motion Programmer software diskette into Drive A or another drive if desired.

2. From the prompt of the drive containing the Motion Programmer diskette, type.

   ```
   A:\> install
   ```

3. A screen appears prompting you to enter the destination hard drive for the Motion Programmer software. Enter the drive letter (or use the default drive that is provided) and press **Enter**.

4. If this is the first installation of the software, a screen for registering the software appears. This screen contains prompts for your name, company, address, and software serial number. Fill in this information.

   Note

   The serial number for your software is located on the back of diskette number 1.

   After you have entered the information, press **Enter**.

5. A screen for confirming the registration information appears next. If the information you entered is correct, press **Enter**. If it is not, press **Esc** to correct any information. If you pressed **Enter** the data is then written onto the master distribution disk.

6. The Copyright screen then appears. Press **Enter** to continue.

7. The AUTOEXEC.BAT and CONFIG.SYS modification screen appears next. Press **Y** if you want the Install program to automatically modify these files for optimum opera-

tion of the Motion Programmer software. Press **N** if you want to modify the files yourself.

8. If you pressed Y, the Install program will create an LM90 directory on the hard drive you specified, and write the Motion Programmer software to it.

   If you pressed **N**, so you could modify the files yourself, a screen will appear prompting you to make the modifications shown below after installing the software. A confirm prompt also appears at the bottom of this screen which permits you to change your mind and have the Install program modify the AUTOEXEC.BAT and the CONFIG.SYS files. Press **Y** for automatic update or press **N** if you still want to modify them yourself. In either case, the Motion Programmer files will be installed on your hard disk at this time.

   To the AUTOEXEC.BAT file, add the following to the *path* line.

   ```
   (Drive ID):\LM90
   ```

   The Drive ID is the letter corresponding to the hard disk drive where the Motion Programmer software is installed.

   To the CONFIG.SYS file, make the following entries.

   ```
   FILES = 20
   BUFFERS = 15
   ```

9. After the Install program writes all the files to the destination drive, the Motion Programmer software is installed. If you elected to modify the AUTOEXEC.BAT and CONFIG.SYS files yourself, do so now.

10. Press **Ctrl-Alt-Delete** to reboot your computer and activate the new AUTOEXEC.BAT and CONFIG.SYS files.

Starting-Up the Motion Programmer

After you have installed The Motion Programmer software, you are ready to start it up. The startup procedure is the same whether you are using The Motion Programmer as a stand-alone package or as a part of the Logicmaster 90 software. First, you must access the Logicmaster 90 Software Startup screen.

To access the Logicmaster 90 Software Startup screen:

At the DOS prompt, type **LM90**.

The startup screen will appear as shown below.

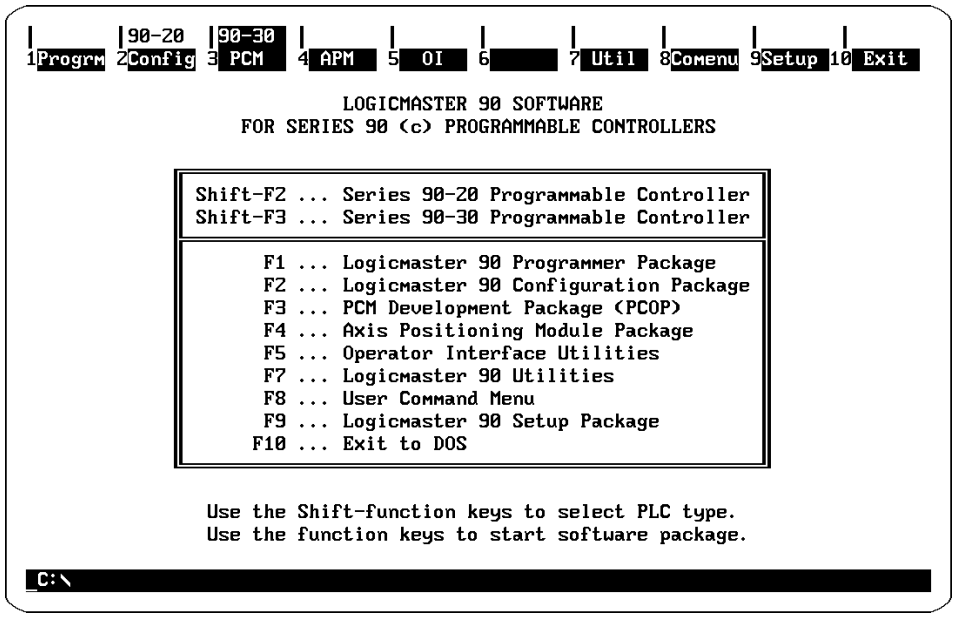


Figure 2-1. Logicmaster 90 Startup Screen

To start-up the Motion Programmer:

1. Select the appropriate PLC type by pressing **Shift-(F3)** for the Series 90-30 Programmable Controller.

2. Press (**F4)** , Axis Positioning Module Package.

Creating and Selecting a Folder

The first screen that appears after initiating the startup procedure is the Create and Select Folder Screen. From this screen you can select an existing folder or create a new one. This screen is shown in Figure 2-2.

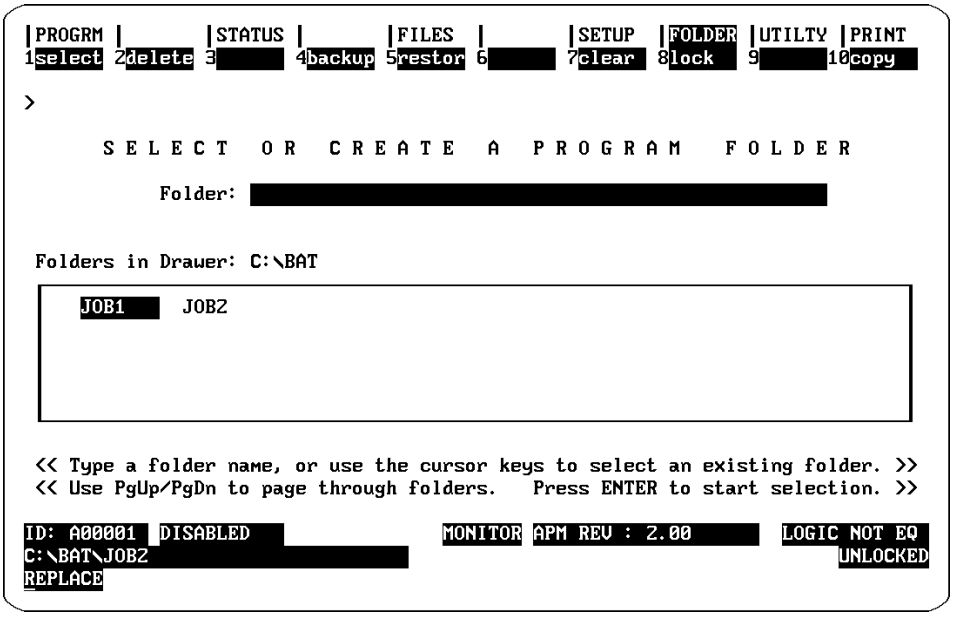


Figure 2-2. Startup Folder Selection Screen

A folder is the next to the highest of a four level hierarchy for organizing your work. It is the highest level that can be created or selected within the Motion Programmer and can contain PLC ladder programs, Configuration files, and APM files. The hierarchy is as follows.

Level	Name
First	Drawer
Second	Folder
Third	APM File
Fourth	Programs and Subroutines

Detailed descriptions of the hierarchy are found in Section 4. Using Folders and Files.

Note

The folders displayed on this screen correspond to the product type specified for the Motion Programmer. If the product type is the Series 90-30, then only Series 90-30 folders will be displayed. If the product type is the 90-70, then only Series 90-70 folders will be displayed. If no folders exist for either package, a message such as "No folders exist" will be displayed in the folder window.

To continue the start-up procedure, a folder must exist and be selected or you must create a new folder if none exists.

To create a Folder:

1. Type-in a new folder name in the folder field. The folder name can be no longer than 7 characters, otherwise use the same DOS conventions as used for naming sub-directories.

2. Press Enter.

3. A prompt will appear to confirm the creation of a new folder. Press "Y" to confirm. (Confirming the operation will create a new folder and it will be the selected folder). The APM File Selection screen will then appear.

If the create folder procedure is successful, the newly created folder becomes the currently selected folder.

If the create folder procedure is not successful, an appropriate error message will be displayed. You can either try again or exit the Motion Programmer by pressing the **Esc** key.

To select an existing Folder:

1. Type in an existing folder name or use the **Cursor** keys and the **PgUp/PgDn** keys to position the cursor on the desired folder in the boxed in area.

2. Press **Enter**.

If the folder selected already contains files, the APM File Selection screen is bypassed and the Main Motion Programmer screen appears. The Motion Programmer automatically selects a file and program / subroutine for you based on the following rules.

First choice: The APM file and program / subroutine last edited by the user as long as the currently selected folder matches the previously selected folder.

Second choice: The first alphabetical APM file in the folder. In this case the default program selection is Program 1.

Creating an APM File

The create folder procedure above will cause the APM File Selection screen to appear as shown below.

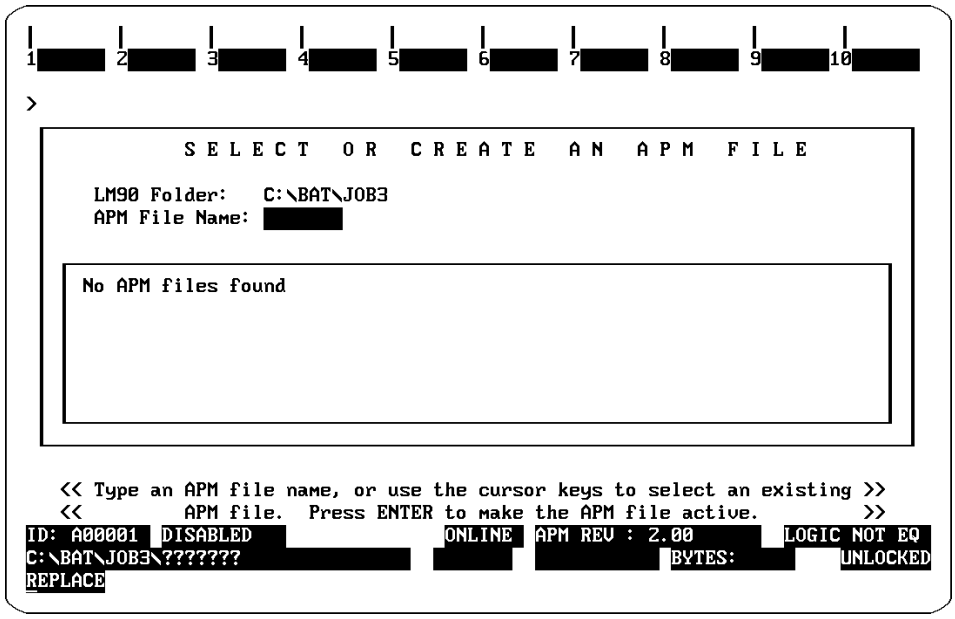


Figure 2-3. Startup File Creation Screen

To continue the start-up procedure you must create an APM File.

To create an APM File:

1. Type in a new file name (maximum of 7 characters) in the APM File Name field.
2. Press **Enter**.
3. A prompt is displayed to confirm the create APM file operation. Press **"Y"** to confirm.

If the create file procedure is successful, the Main Screen is displayed. The newly created file becomes the currently selected file and the Motion Programmer automatically selects Program 1 as the currently selected program.

If the create file procedure is not successful, an appropriate error message will be displayed. You can either try again or proceed to the Main Screen by pressing the **Esc** key.

The Motion Programmer Main Screen

The Main Screen as shown below provides access to all primary Motion Programmer functions. Simply press the key that corresponds to the desired function.

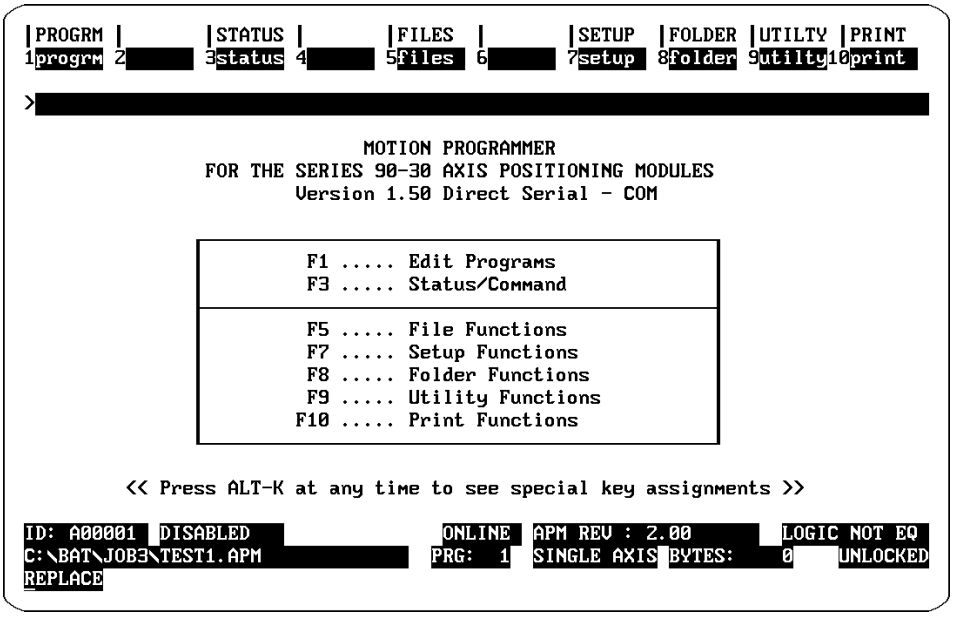


Figure 2-4. Motion Programmer Main Screen

Motion Programmer Functions

The Motion Programmer functions are summarized below.

(F1) Edit Programs. This is where you edit motion control programs and subroutines.

(F3) Status/Command. This function allows you to monitor the APM Status bits and Status words.

(F5) APM File Functions. These functions allow you to select a different file, create a new file, and select a particular program or subroutine to edit. In addition, you can copy files and clear Programs and Subroutines. Program configuration for axis type, acceleration mode, positioning mode, and fixed-point offsets is also performed here.

(F7) Setup Functions. These functions allow you to set up the printer and communications ports and select between Offline, Monitor, and Online for the Motion Programmer.

(F8) Folder Functions. These functions allow you to select, create, or delete a folder as well as backup, restore, clear and copy APM files.

(F9) Utility Functions. These functions allow you to load, store, and verify an APM file, clear APM memory, and read, write, and verify APM EEPROM.

(F10) Print Functions. These functions allow you to setup printer parameters, select print destination, and print screens, programs, and subroutines.

Message Line

Near the top of the screen below the function keys is the message line. This line is where user prompts and system messages appear.

Command Line

The Command Line is identified by the ">" symbol in the upper left-hand corner of the screen. It is used for the GoTo and Renumber functions in the Program Editor.

Status Line Definition

There are three lines at the bottom of the screen which contain information about the status of the Motion Programmer. Status line information may differ somewhat depending on the screen you are in. Many of these screens, however, contain common fields which will always be displayed. Both common and unique fields are described here.

When the programmer mode is Off-Line, the status lines show information relating to the selected drawer, folder, file, and program / subroutine, the programmer mode, and the keyboard setup. When the programmer mode is Monitor or On-Line, the status of communications between the programmer and APM, and the logic equality between the selected file and the file in the APM are also displayed.

Status Line 1

SNP ID Field. It shows the SNP ID (if any) of the connected APM. This field is known as an *MOC* field because it appears only when The Motion Programmer is:

- In Monitor or Online modes, and
- Connected with the APM

Enabled/Disabled/InvalidField. This MOC field shows the Enabled / Disabled state of the APM. Invalid APM is displayed if connected to a non-APM device. For example, if you are connected directly to the PLC instead of the APM, Invalid APM will be displayed.

No Communications Field. This field appears when The Motion Programmer is in either Monitor or On-line mode, and the APM is *not* communicating with the programmer. It will appear in place of the SNP ID field and the Enabled / Disabled field.

Programmer Mode Field. This field is displayed at all times. It shows whether the mode is Offline, Monitor, or On-Line.

Software Revision Field. This field displays the software revision level of the APM.

Logic Equality Field. This an MOC field. It shows whether the program logic on disk is equal to the logic in the APM.

Status Line 2

Selected File Field. This field is displayed at all times. It shows the current APM file selected for The Motion Programmer. In folder screens, only the selected folder is displayed.

Prg/Sub Number Field. This field is displayed on many different status line configurations, but not all of them. It shows the selected program or subroutine number.

Prg/Sub Type Field. This field is displayed on many different status line configurations, but not all of them. It shows the selected program or subroutine axis type (whether it is single or multi-axis).

Bytes. This field displays the total of the number of bytes used by the selected program / subroutine. Displayed only on some screens.

Line Number. This field keeps track of the current line you are on within the selected program / subroutine. Displayed in the editor only.

Folder Locked/Unlocked Field. Displays the current folder locked or unlocked status. Only displayed in folder and file functions. When locked, no writes are allowed to the folder.

Status Line 3

Insert/Replace. Displays whether the keyboard is in the insert text or the replace text mode.

CAPS. Displays the keyboard Caps Lock status. May be toggled on and off by pressing the Caps Lock key. Default state is off.

NUM. Displays the keyboard Num Lock status. May be toggled on and off by pressing the Num Lock key. Default state is off.

Help and Special Function Keys

There are a number of special function keys available throughout The Motion Programmer. These keys, which are listed below, can be viewed on screen by pressing **Alt-K**.

Function	Key	Function	Key
Abort	Alt-A	Exit Package	Ctrl-Break
Clear Field	Alt-C	Zoom Out	Esc
Change Programmer Mode	Alt-M	Previous Command Line	Ctrl-Home
Toggle Status Area	Alt-E	Next Command Line	Ctrl-End
List Directory Files	Alt-L	Cursor Left Within Field	Ctrl
Print Screen	Alt-P	Cursor Right Within Field	Ctrl
Help	Alt-H	Change / Incr Field Contents	Tab
Key Help	Alt-K	Change / Decr Field contents	Shift-Tab
Command Help	Alt-I	Accept Field Contents	Enter
Start Teach Mode	Alt-T	Display Last System Error	Ctrl-E
Stop Teach Mode	Alt-Q		
Playback File (n = 0 - 9)	Alt-N		

Establishing Communications Between the APM and Programmer

Some Motion Programmer operations require the Motion Programmer to be physically connected and communicating with the APM. Refer to Section 5, Setting-Up the Ports. to establish communications between the Motion Programmer and the APM.

Section 2. Creating and Editing Programs and Subroutines

This section describes the features of the Motion Program Editor. The editor allows you to display and edit a program and provides a number of other functions to make motion programming easier. Topics covered in this section are:

- Displaying a Program
- Inserting Program Lines
- Making Changes to a Line
- Deleting a Line
- Checking Program Syntax
- Renumbering Block Numbers
- Using the GOTO Function
- APM Command List
- Size Limitations

Displaying a Program

Press **(F1) Edit Programs** at the Main screen to enter the Motion Programmer Editor. The editor is in Display Mode at this time, and the currently selected program or subroutine is displayed. The currently selected file and program or subroutine are identified in Status Line 2.

To Display a Different Program or Subroutine:

1. If you want to work with a program or subroutine in a different folder or file, you must go to the File functions, press **Shift (F5)** or Folder functions, press **Shift (F8),** and select the desired folder, file, and program or subroutine, then re-enter the editor, press **Shift (F1)**.

2. If you want to work with a different program or subroutine in the same folder and file, press **(F8) goto**, enter the program or subroutine number in the Command Line, then press Enter.

If you have not yet entered any lines in a program, it is known as a null program. A null program is displayed in Figure 2-5.

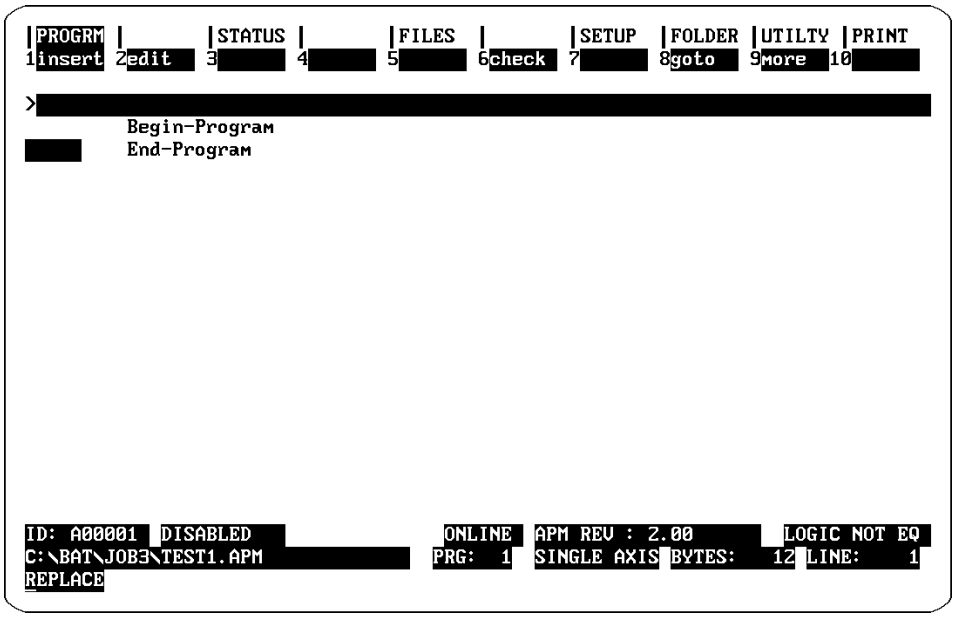


Figure 2-5. Display Mode - Null Program

Functions Accessed from Display Mode

The editor's main functions can be accessed only when the editor is in Display Mode. The functions are listed below with their associated function keys.

(F1) Insert Line(s). Accesses Insert Mode. New lines can be added until the user exits Insert Mode by pressing the **Esc** key.

(F2) Edit Line(s). Accesses Edit Mode. Allows the edit of one or more lines within the displayed screen.

(F6) Check. Checks the structure of the displayed program or subroutine for problems.

(F8) Goto. Goes to the block number, line number, program, or subroutine which you specify.

(F9) More. Provides access to other edit functions described below.

> **(F1) Select.** Selects line or lines to be acted upon by delete or renumber.
>
> **(F6) Delete Line(s).** Deletes the selected line(s) of the displayed program / subroutine or line.
>
> **(F7) Renum.** Renumbers the Block Numbers of the displayed program or subroutine.

Additional Status Fields

There are two additional status fields displayed when you are in the Motion Program Editor. Both fields are in Status Line 2.

Bytes. This field keeps a running total of the number of bytes used by the selected program / subroutine.

Line Number. This field keeps track of the current line you are on within the selected program / subroutine.

Program Edit Fields

The area on the screen between the command line and the status lines is the instruction area. Begin and End Program or Subroutine tags appear at the top and bottom of the instruction area. All motion program instructions are entered between the Begin and End tags. A screen with example instructions is shown below.

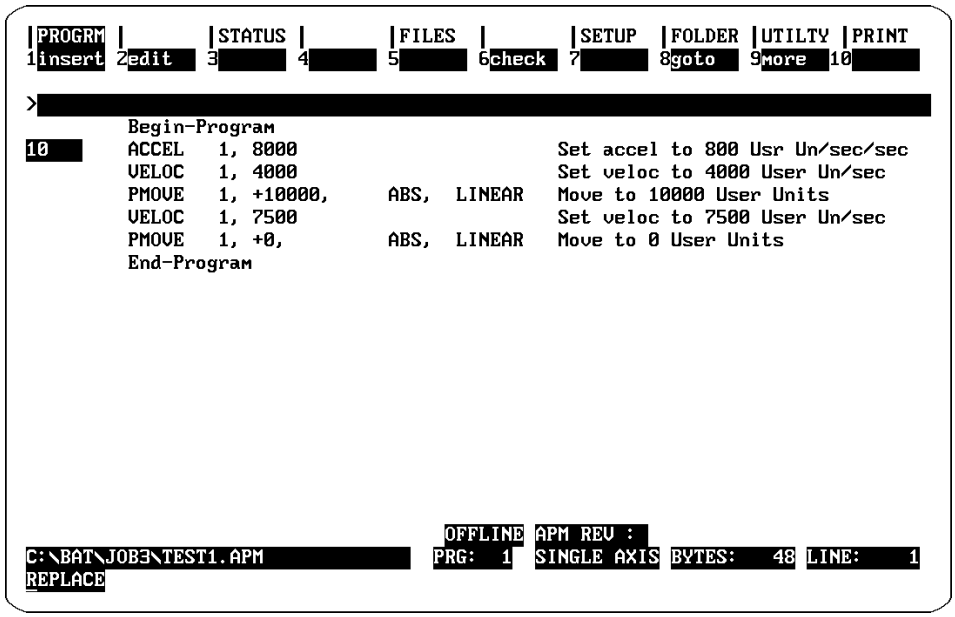


Figure 2-6. Display Mode - Example Instructions

A program or subroutine instruction consists of a command and associated data describing the command. The command and data are entered into areas on the program editor screen called fields. These fields are described below. From left to right the fields are:

Block Number Field. This field is *optional.* The Block Number field permits the entry of a number to designate the beginning of a block of program instructions. It is used as a place for Jump commands to go to and for synchronizing 2-axis programs.

Sync Block Specification Field. This field is *optional.* This field appears only in multi-axis programs on lines with a block number. An "X" in this field specifies a sync block; otherwise the block is a standard block.

Command Name Field. This field is *required.* It is an English language designation for the command.

Operand Fields. Any operand field displayed is *required.* When you enter a Command name, the required Operand fields will be displayed with its default value. The number of Operand fields displayed vary from one to four depending on the Command name. Operand field 1 is reserved for the axis number. The content of the other 3 fields vary depending on the command.

Comments Field. This field is *optional.* It allows approximately 30 characters of comment for each program instruction. Separate comment lines of 80 characters each can also be added.

Moving the Cursor

While in Display Mode you are allowed to move vertically through a displayed program. The cursor will always be positioned on the first field of the selected line. The cursor can be moved horizontally from field to field in Insert and Edit Modes.

- **Up Arrow** - Moves cursor up one line.
- **Down Arrow** - Moves cursor down one line.
- **PgUp** - Moves cursor up one page.
- **PgDn** - Moves cursor down one page.
- **Home** - Moves cursor to beginning of Program or Subroutine.
- **End** - Moves the cursor to the end of Program or Subroutine.

APM Motion Command List

The following list includes all Motion Commands available in the Motion Programmer. The next few pages in this section will describe how to enter these commands in a Program or Subroutine.

Table 2-1. The APM Motion Command List

CommandName Field	Operand1	Operand2	Operand3	Operand4
Acceleration (ACCEL)	Axis # (1, 2)	USER UNITS / sec / sec (1 – 134,217,727)	n / a	n / a
Call Subroutine (CALL)	Axis # (1, 2)	Subroutine Number (1-40)	n / a	n / a
Continuous Move (CMOVE)	Axis # (1, 2)	USER UNITS (–8,388,608...+8,388,607) or Parameter Number (0 – 255)	Positioning Mode (ABS, INCR)	Acceleration Mode (LINEAR, SCURVE)
Dwell (DWELL)	Axis # (1, 2)	Milliseconds (0–65,535)	n / a	n / a
Jump (JUMP)	n / a	CTL # CTL01–CTL12	Block Number (1–65,535)	n / a
Load Parameter (LOAD)	n / a	Parameter Number (0 – 255)	Parameter Value (Depends on Parameter Type)	n / a
Null (NULL)	n / a	n / a	n / a	n / a
Positioning Move (PMOVE)	Axis # (1, 2)	USER UNITS (–8,388,608...+8,388,607) or Parameter Number (0 – 255)	Positioning Mode (ABS, INCR)	Acceleration Mode (LINEAR, SCURVE)
Velocity (VELOC)	Axis # (1, 2)	USER UNITS / sec (1 – 8,388,607), or Parameter Number (0 – 255)	n / a	n / a
Wait (WAIT)	Axis # (1, 2)	CTL # (CTL01 – CTL12)	n / a	n / a

Inserting Lines in a Program or Subroutine

When you are programming a new program or subroutine or if you want to add lines to an existing program, you must enter the Insert Line Mode. The procedure below describes how to enter Insert Line Mode and insert a new line.

While connected and in the Online or Monitor mode, Motion Programmer allows insertions only up to the maximum program memory size of the currently communicating APM. This limit is for the sum of all 10 programs and subroutines. To insert lines beyond this limit, either change to Offline mode or connect to an APM with more program memory.

To enter Insert Mode:

1. If you are inserting lines in a null program / subroutine, Press the **(F1) insert** key.
2. If you are inserting a line(s) to an existing program, determine the insertion point of the new line(s), position the cursor on the line *below* the insertion point. Then press the **(F1) insert** key.

The function keys displayed at the top of the screen will change to reflect only the current options available. As you enter Insert Mode, the cursor remains on the Block Number.

To add an APM Command line(s):

1. Enter a Block Number if desired (1– 65,535). This field is optional. It is recommended to make Block Numbers sequential for fastest program execution, but it is not required. Block Numbers must be unique.
2. Move the cursor to the Command Name field by pressing the **Enter** or **Right Arrow** key.
3. Enter the Command by pressing the appropriate function key or by typing it in and pressing **Enter**.
4. This will cause the applicable operand fields for the command to be displayed. Some commands have only two operands such as the *Velocity* command, others have as many as four operands such as the *Move* commands.
   - Default attributes for each operand will be displayed when possible.
   - Operands requiring input will display a “?”.
   - Enter an appropriate value for each Operand by typing it in or, using the **Tab** key to display the desired selection or, when possible, by selecting the appropriate function key and pressing **Enter**.
   - Press **Enter** to bypass an Operand.
   - For more information on valid ranges for operands see Table 2-1.

Note

You can set default attributes for Acceleration Mode (Linear, Scurve), Positioning Mode (Absolute, Incremental), and Fixed Point formats in the Program and Subroutine Selection screens of the File functions.

5. If you wish to enter a comment, move the cursor to the Comment field and type in your comment.

6. At this point you have three options.

   - Press the **Esc** key to accept the instruction and return to Display Mode.
   - Press the **Enter** or **Down Arrow** key to accept the instruction and to open another line below the previously inserted line and remain in Insert Mode.
   - Press the **Up Arrow** key to accept the instruction and to open a line above the previously entered line and remain in Insert Mode.

Note

The adding or editing of a line may be aborted at any time by pressing the Alt / A key sequence. If you have not changed any field in the line, aborting will return you to Display Mode. If you have made changes, aborting will cause a prompt to be displayed asking you to confirm the abort. If you confirm, the line is cleared and the cursor returns to the beginning of the line. If you do not confirm, the cursor remains at its current position.

To add a Comment line(s):

You can add a separate comment line of up to 80 characters by typing a semicolon followed by the comment.

Making Changes to a Program or Subroutine

To make changes to an existing line or lines of a program or subroutine you must enter the Edit Mode. Most Edit Mode operations are the same as for Insert Mode, and the screens are the same except that a line within the program or subroutine is not opened in Edit Mode as it is in Insert Mode.

To enter the Edit Mode:

1. Position the cursor on the line to edit.
2. Press the **F2 edit** key.

To change an entry in a line:

1. Position the cursor on the field you want to change.
2. Type in or select the new entry in the same manner as for Insert Mode.

3. At this point you have three options.
   - Press the **Esc** key to accept the change and return to Display Mode.
   - Press the **Enter** or **Down Arrow** key to accept the change and to move to the line below and remain in Edit Mode.
   - Press the **Up Arrow** key to accept the change and to move to the line above and remain in Edit Mode.

Deleting a Line

The Delete Line function is executed in Display Mode only. There are two basic ways to delete a line.

To delete *one* line at a time:

1. Position the cursor on the line to be deleted.
2. Press the **Alt-D** key sequence. A confirmation prompt will be displayed. If you enter a Y, the line will be deleted. If you enter an N, the function will be aborted.

To delete *multiple* lines:

1. Press the **(F9) more** key so that the **(F1) select** key is displayed.
2. Position the cursor on one boundary of the lines you want to delete.
3. Press the **(F1) select** key.
4. Move the cursor down or up to include all lines you want to delete. The lines you select will be shown in reverse video.
5. Press the **Alt-D** key sequence or the **(F6) delete** key. A confirmation prompt will be displayed. If you enter a Y, the lines will be deleted. If you enter an N, the function will be aborted.

Checking Program or Subroutine Syntax

Each instruction you enter is checked to prevent you from entering invalid data. The Check Syntax function provides a check of the structure of the overall program or subroutine. The Check Syntax function tests the following conditions.

- Jump to unknown Block Number (Fatal)
- Call to an undefined subroutine (Fatal)
- Recursive subroutine call (Fatal)
- Block Number out of sequence (Warning)

To execute the Check Syntax function:

1. Press **(F6) check** when in Display Mode.
2. These checks are also made when you exit Display Mode to go to the editor's Main Menu by pressing the **Esc** key.

In either case of executing the Check Syntax function, a message will appear if a structural problem is found. The message will state the type of syntax error and the corresponding line number. Press any key to continue.

Renumbering Block Numbers

You can renumber the Block Numbers within selected lines. You can also specify the increment value of the Block Numbers.

To renumber Block Numbers:

1. Press the **(F9) more** key so that the **(F1) select** key is displayed.
2. Position the cursor on one boundary of the Block Numbers you want to renumber.
3. Press the **(F1) select** key.
4. Move the cursor down or up to include all Block Numbers you want to renumber.
5. Press the **(F7) renum** key. This will renumber all Block Numbers within the selected lines by an increment of 10.

   Or, first place an increment value in the command line before pressing the **(F7) renum** key.

Using the GoTo Function

You may go to a specific *blocknumber* or *line number* within the current program or subroutine using the GoTo function. You can also use this function to select a different *program* or *subroutine* within the current APM File.

To use the GoTo function:

1. Enter the GoTo destination in the command line.

   Block numbers are *not* preceded by a symbol.
   Line numbers are preceded by the "#" symbol.
   Program numbers are preceded by the "%" symbol.
   Subroutine numbers are preceded by the "$" symbol.

2. Press the **(F8) goto** key.

If the Block Number is found, the cursor will be placed at the beginning of the line containing the Block Number. If the Block Number is not found, the cursor will not move and an error message will be displayed. If the line number is greater than the number of lines in the program, the cursor will move to the last line of the program.

Size Limitations

When the Motion Programmer is On-Line and communicating with an APM, it will not allow a file to grow larger than the program memory limit of the APM. The file size is the sum of all 10 program and 40 subroutine sizes and can be seen in the Store screen under Utilities. For Release 1 APMs, this limit is 16 K Bytes. For Release 2, the limit is 14 K Bytes.

If the Motion Programmer is Off-Line or not communicating, APM memory limitations will not apply to generating a program. However, limits will apply when an attempt to store the file is made.

Section 3. Status Functions

The status functions allow you to monitor APM Status bits (%I data) and Status words (%AI data).

To access the Status Function screen:

1. For the screen to be active, The Motion Programmer must be in the On-Line or Monitor mode and communicating with the APM. See Section 5. Loading and Storing APM Files, for instructions.

2. Press the **(F3) status** key from the Main screen.

The following screen will appear.

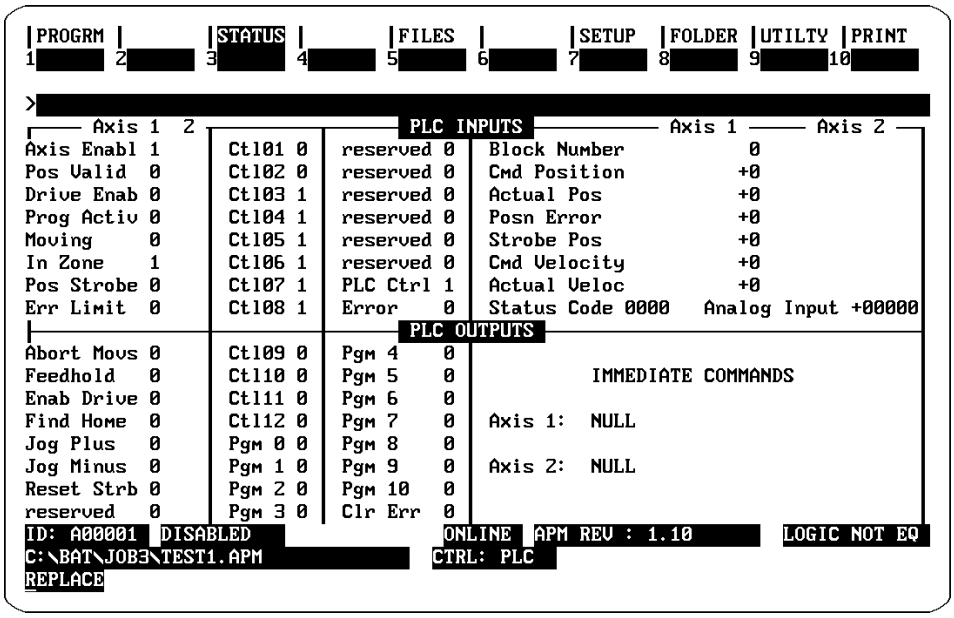


Figure 2-7. Status and Functions Screen

This screen is divided into two main sections. The top section is the Status section which displays the Status bits and Status words. The bottom section is the Control section which displays the Discrete commands and Immediate commands. Axis 2 data is displayed when the APM is a 2-axis model.

Upon entering this screen, the PLC will be in control as indicated in the CTRL field in Status Line 2.

Section 4. Using APM Files and Folders

This section describes the features available in the APM File Functions **(F5)** and the Folder Functions **(F8)**. The topics in this section are organized as follows.

- Hierarchy of Folders and Files
- APM File Functions
- Folder Functions

Hierarchy of Folders and Files

The Motion Programmer permits a hierarchical storage of APM Programs and Subroutines to help you organize your work. APM Programs and Subroutines are stored in APM Files. APM Files are stored within Folders. And Folders are stored within Drawers. This storage hierarchy is described below.

Drawer	Highest level DOS directory for storing Logicmaster Folders. The default Drawer is \LM90. New Drawers must be created outside the Motion Programmer using the DOS Make Directory (mkdir) command. For example, **C:\> mkdir motor1**
Folder	DOS subdirectory within a Drawer used for organizing files specific to an application., must be created by the Logicmaster 90 software or the Motion Programmer software. APM Files as well as Logicmaster program and configuration files can all exist in the same Folder.
APM File	DOS File which contains APM programs, subroutines, and comments. Created only in the Motion Programmer. APM Program and Subroutine Files have an .APM extension; Comment Files have a .CMT extension..
APM Programs and Subroutines	There are 10 APM Programs numbered 1-10 and 40 APM Subroutines numbered 1-40 which always exist within each APM File. The Programs and Subroutines are empty until edited.

APM File Functions (F5)

The APM File functions allow you to select, create, copy, and delete an APM file and select and configure APM programs or subroutines. These functions are accessed by pressing **(F5) files** or **(Shift-F5) FILES** from the Motion Programmer Main Screen. You can also access these functions from any other screen where **FILES** is displayed as a function key. The APM File Functions screen is shown below.

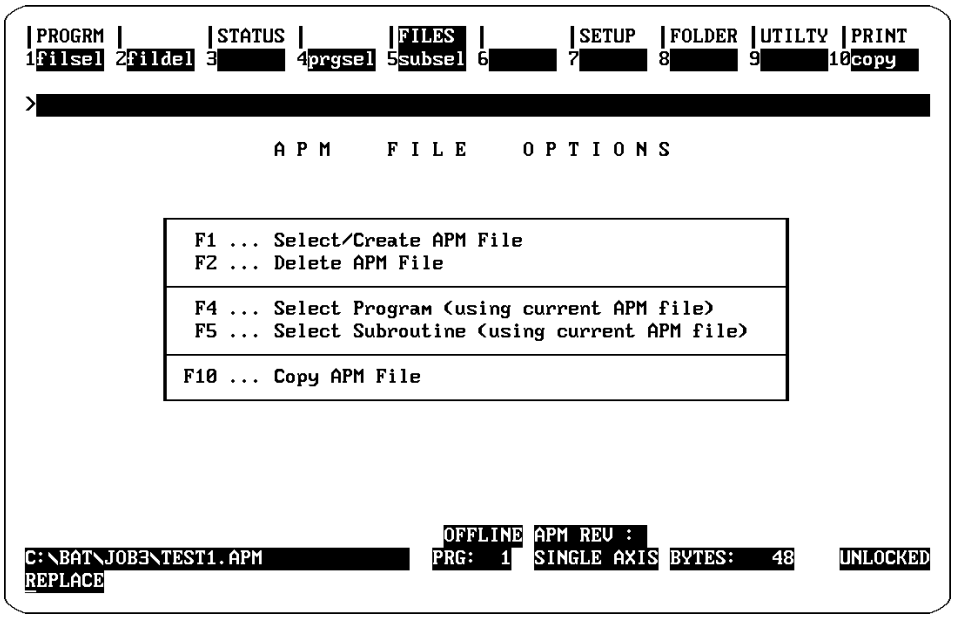


Figure 2-8. APM File Functions Screen

Selecting and Creating APM Files

APM file selecting and creating is performed from the APM File Select or Create Screen as shown below. To access this screen press the **(F1) filsel** key from the APM File Options screen.

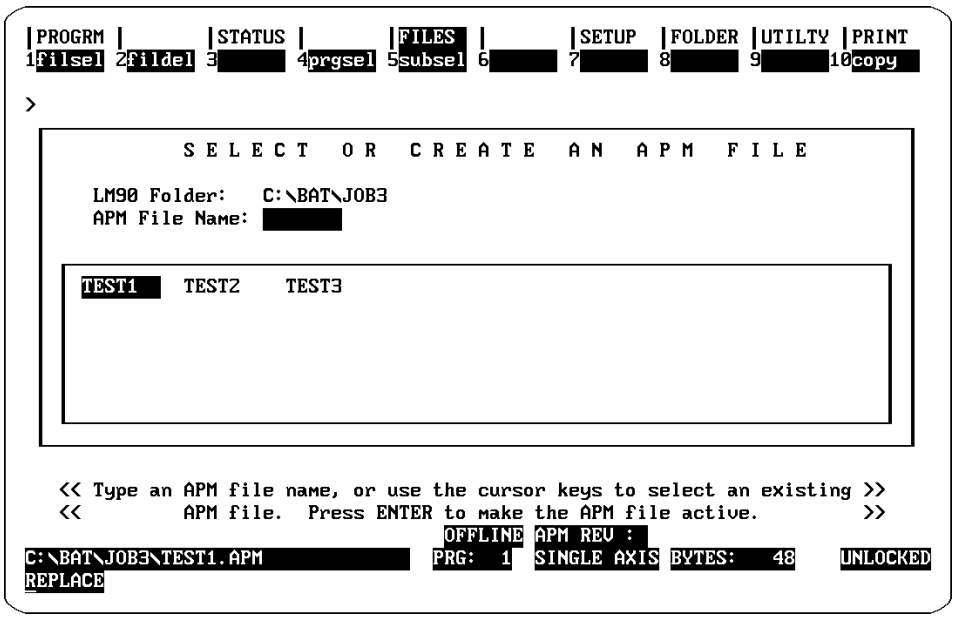


Figure 2-9. Select or Create APM File Screen

Note

1. If you have not selected an APM file and then try to exit this screen, the message "WARNING: No APM file has been selected" will be displayed before exiting. You will be prevented from entering Motion Programmer functions that require an APM file to be selected.
2. If you want to create or select a file in an existing drawer or folder other than the currently selected one, you must go to the Select or Create Folder screen in the Folder functions.

Creating an APM File

Creating an APM file is done from the Select or Create APM File Screen which is accessed as explained above.

To create an APM file:

1. Type in a new file name (maximum of 7 characters) in the APM File Name field.
2. Press **Enter**.

If the create file procedure is successful, the message, "APM file selected," is displayed on the message line and the status lines are updated to reflect the new file name. The Motion Programmer automatically selects Program 1 as the currently selected program.

If the create file procedure is not successful, an error message will be displayed.

Selecting an APM File

Selecting an APM file is also performed from the Select or Create APM File Screen which is accessed as described earlier.

To select an APM file:

1. Type in an existing file name in the file name field or use the **Cursor** keys and the **PgUp/PgDn** keys to position the cursor on the desired file in the boxed-in area.
2. Press **Enter**.

If the select file procedure is successful, the message, "APM file selected," will be displayed on the message line and the status lines will be updated to reflect the new file name. The Motion Programmer automatically selects Program 1 as the currently selected program.

If the select file procedure is not successful, an error message will be displayed. You can either try selecting the same file again or another file.

Once a file has been successfully selected, you can return to the APM File Options screen by pressing the Esc key or you can move directly to another File function by pressing the appropriate function key.

Deleting an APM File

You can delete APM files from the Delete APM File screen. Enter the Delete APM File screen by pressing **(F2) fildel** from the APM File Options menu. The Delete APM File screen will appear.

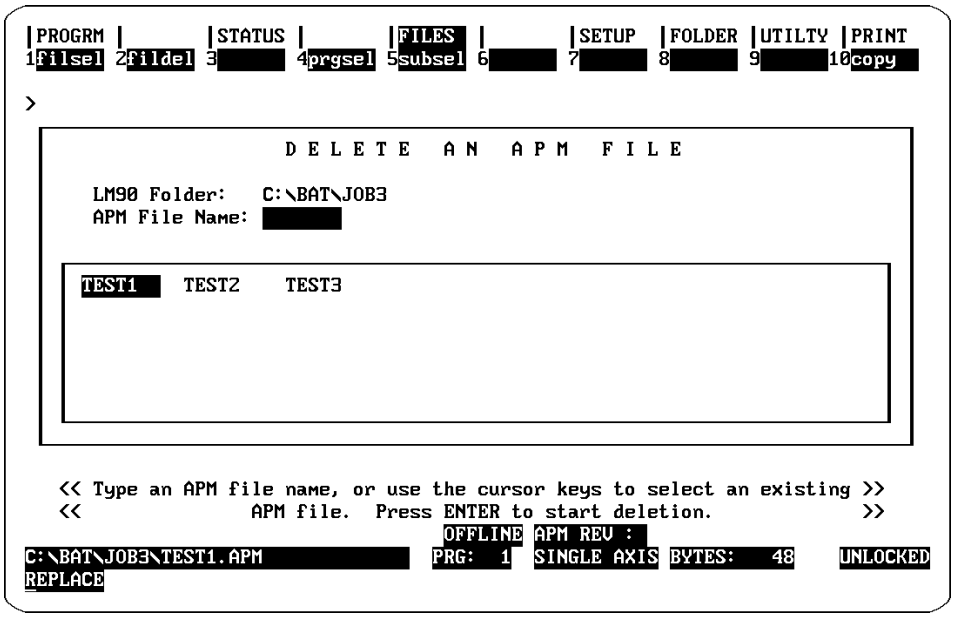


Figure 2-10. Delete APM File Screen

To delete an APM file:

1. First, make sure the file you want to delete is *not* the currently selected file. The currently selected file can not be deleted.
2. Position the cursor on the APM File to be deleted. Or type in the name of the file to be deleted in the APM File field.
3. Press **Enter**.

Selecting and Configuring APM Programs and Subroutines

After you have selected a APM File you can select a Program or Subroutine to edit. The Program and Subroutine Selection screens also contain fields for axis number, acceleration mode, positioning mode, and fixed point formats.

The Program or Subroutine Selection screens are accessed from either the APM File Options screen or the other file functions screens by pressing **(F4) prgsel** or **(F5) subsel**. The two screens are nearly identical and are discussed below with any differences noted. The Program Selection screen is shown below.

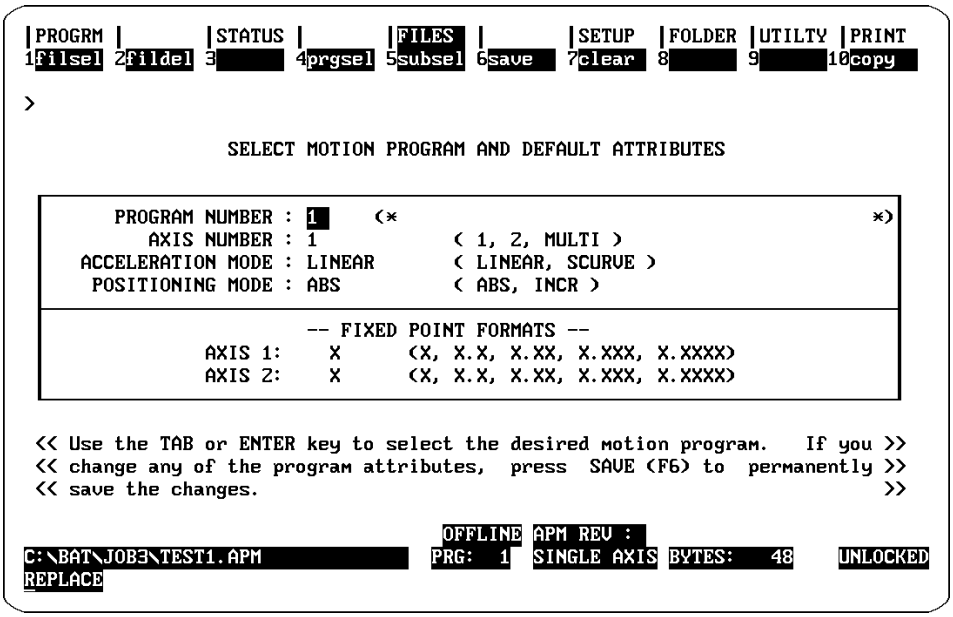


Figure 2-11. The Select Motion Program Screen

Below is a description of each field that appears on the Program and Subroutine Selection screens.

Program/Subroutine Number Field. This specifies the Program Number (1-10) or the Subroutine Number (1-40) to be selected.

Comment Field. This field allows you to enter a small comment for the program specified in the Program / Subroutine Number field. It is located beside the Program Number field and is updated when you change the program / subroutine number.

Axis Number Field. (Programs Only) This field specifies the axis number for a program. The options are 1, 2, and Multi.

- The entry, 1, designates a single-axis program using Axis 1.
- The entry, 2, designates a single axis program using Axis 2.
- The entry, Multi, designates a two-axis implementation in which a single program controls both axes.

Type Field. (Subroutines Only) This field designates whether the subroutine will be a single or multi-axis subroutine. A multi-axis subroutine can only be called from multi-axis programs. A single-axis subroutine can only be called by single-axis programs.

Acceleration Mode Field. This field specifies the default acceleration mode, Linear or S-Curve.

Positioning Mode Field. This field specifies the default positioning mode, Absolute or Incremental.

Fixed Point Format Field. The fixed point offset is a user convenience which applies to commands specifying position, velocity, and acceleration. A program can have different fixed point offsets for each axis in a 2-axis module. Fixed point offsets do not apply to the *Dwell* command, which is specified in milliseconds, nor does it apply to parameters (P0 ... P255) since parameters can be used for more than one axis. The examples below show what values would appear on the screen for a fixed point offset of x.xx.

- Enter 20399; result on the screen is 20339.00
- Enter 2033.9; result on the screen is 2033.90
- Enter 20.339; result on the screen is 20.33
- Enter 20.33; result on the screen is 20.33

If your servo system has a gearing ratio of 1000 counts = 1 mm, you might want to select a fixed point format of x.xxx. This would allow you to program 1.000 for a distance of 1 mm.

If your servo system has a gearing ratio of 100 counts = 1 inch, you might want to select a fixed point format of x.xx. This would allow you to program 1.00 for a distance of 1 inch.

To select a program or subroutine:

1. From the Select Program or Subroutine screen, use the **Cursor** keys to position the cursor on the program or subroutine number field.

2. Step through the program or subroutine numbers using the **Tab** key until you find the desired program or subroutine number. Each time you press the **Tab** key, the displayed program or subroutine is selected and the attribute fields will be updated.

   Or, type in the desired number in the Program or Subroutine field, and press **Enter** to select. The attribute fields will then be updated.

To set default attributes for a program or subroutine:

1. From the Select Program or Subroutine screen, use the **Cursor** keys position the cursor on the appropriate attribute field.

2. Step through the options for the fields using the **Tab** key until the desired option is displayed. Or, type in the desired option in the field. You may also press **Alt-A** to restore the fields to their previous values.

3. Press **(F6) save**, to save the attributes.

Copying APM Files

You can copy individual APM Files within the same folder using this function. Access the Copy APM File screen by pressing **(F10) copy**. The Copy APM File screen appears below.

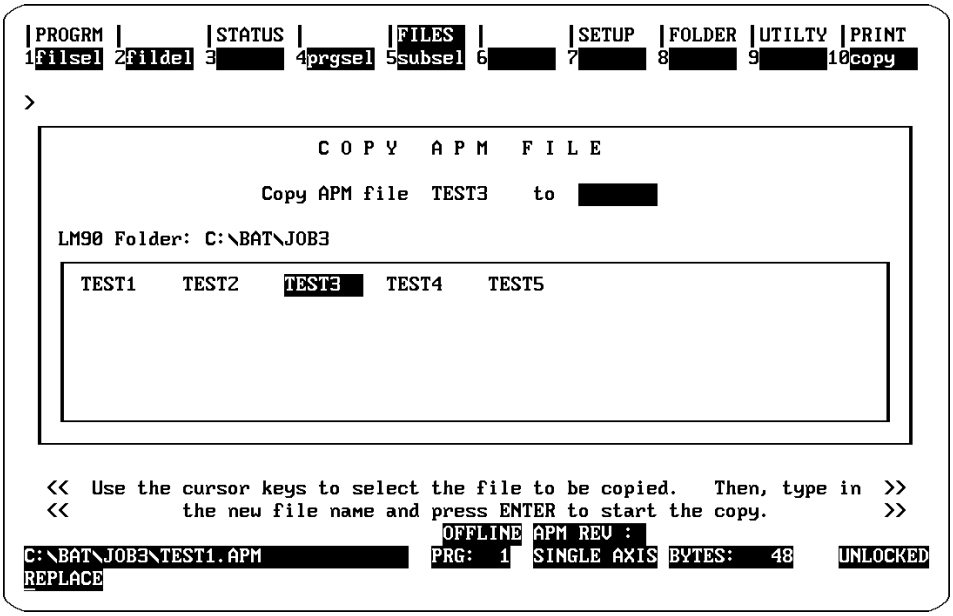


Figure 2-12. COPY APM File Screen

To copy an APM File:

1. From the Copy APM File screen, select the file you want to copy by positioning the cursor on the desired file in the boxed-in area on the screen.
2. Type in the name of the destination file in the copy-to field.
3. Press **Enter**.

The message “Copy complete” is displayed if the copy operation is successful.

Folder Functions (F8)

The Folder functions allow you to select, create, and delete a folder and to backup, restore, clear, and copy APM files. These functions are accessed by pressing **(F8) folder** or **(Shift-F8) FOLDER** from the Motion Programmer Main Screen. The APM Folder Functions screen is shown below.

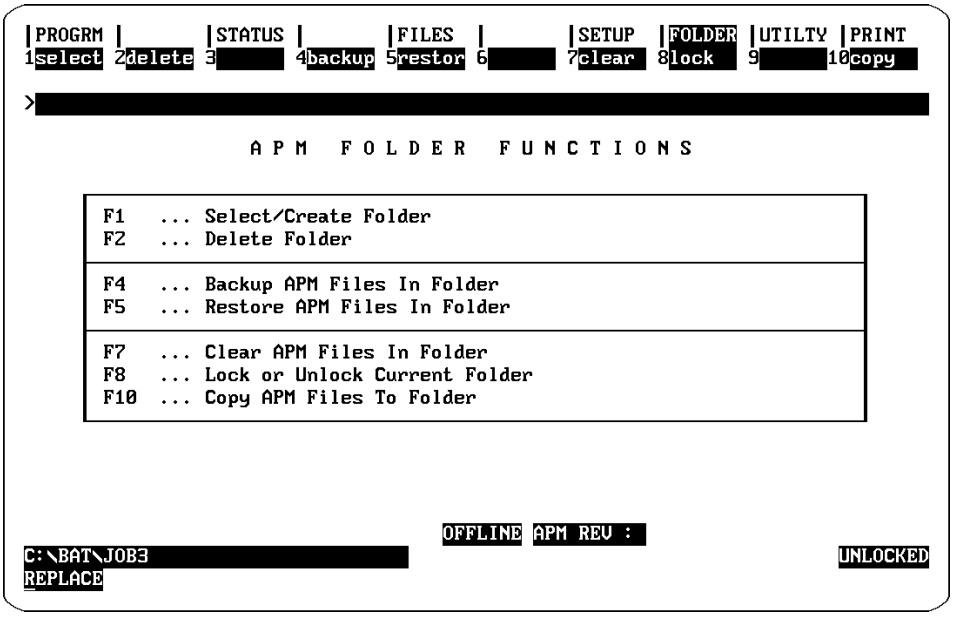


Figure 2-13. APM Folder Functions Screen

The folder functions in the Motion Programmer are derived from the Logicmaster 90 folder functions. Some of the folder functions will manipulate only APM files (backup, restore, clear, and copy APM files). This feature is included to protect you from inadvertently destroying Logicmaster 90 files that are contained in the same folder, and also to save time by avoiding the copying of unnecessary files.

Selecting and Creating a Folder

Press **F1 select** to access the Select or Create Folder screen. The screen appears below.

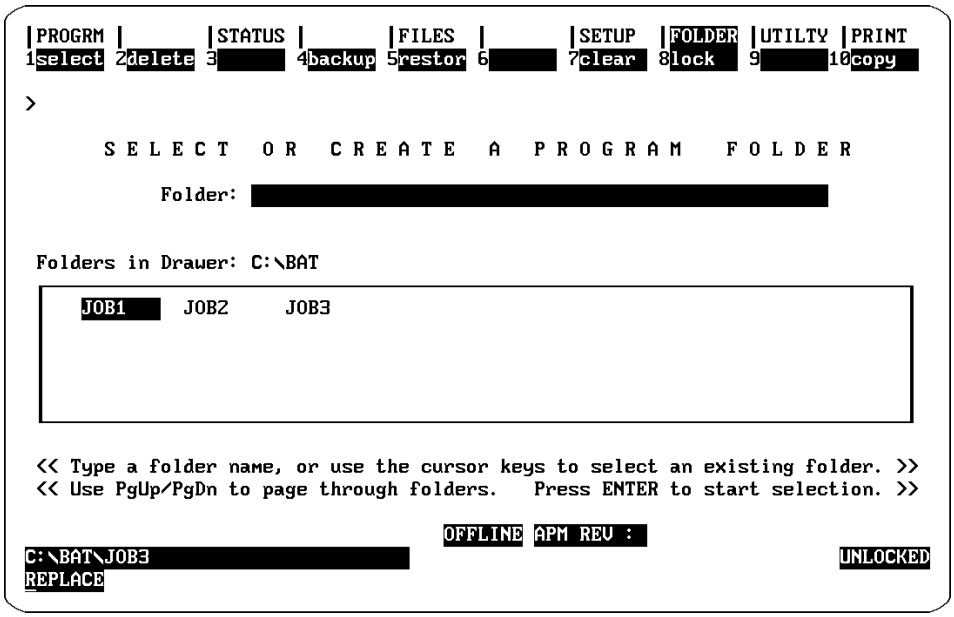


Figure 2-14. Select or Create Folder Screen

Creating a Folder

Folders are created from the Select or Create Folder screen accessed as explained above.

To create a folder:

1. Type in a new folder name (maximum of 7 characters) in the Folder Name field. Or, if you want to create Folder in another Drawer, specify the complete path. For example, **C:\motor1\testa.**
2. Press **Enter**.

After you create a folder then exit the folder functions, the software will automatically take you to the APM File Options screen, so you can create an APM file to work with.

Selecting a Folder

Folders are selected from the Select or Create Folder screen accessed as explained above.

1. Type in an existing folder name in the folder name field or use the **Cursor** keys and the **PgUp/PgDn** keys to position the cursor on the desired folder in the boxed-in area.
2. Press **Enter**.
3. If the folder exists, the message “Folder found; back up its contents?” will appear. Enter “Y” if you desire to backup the folder.

If the select folder procedure is successful, the message, "APM folder selected," will be displayed on the message line and the status lines will be updated to reflect the new folder name.

If the select folder procedure is not successful, an error message will be displayed. You can either try selecting the same folder again or another folder.

Deleting a Folder

You may delete any folder other than the currently selected one as follows.

Enter the Delete Folder screen by pressing **(F2) delete** from the APM File Options menu. The Delete Folder screen will appear.

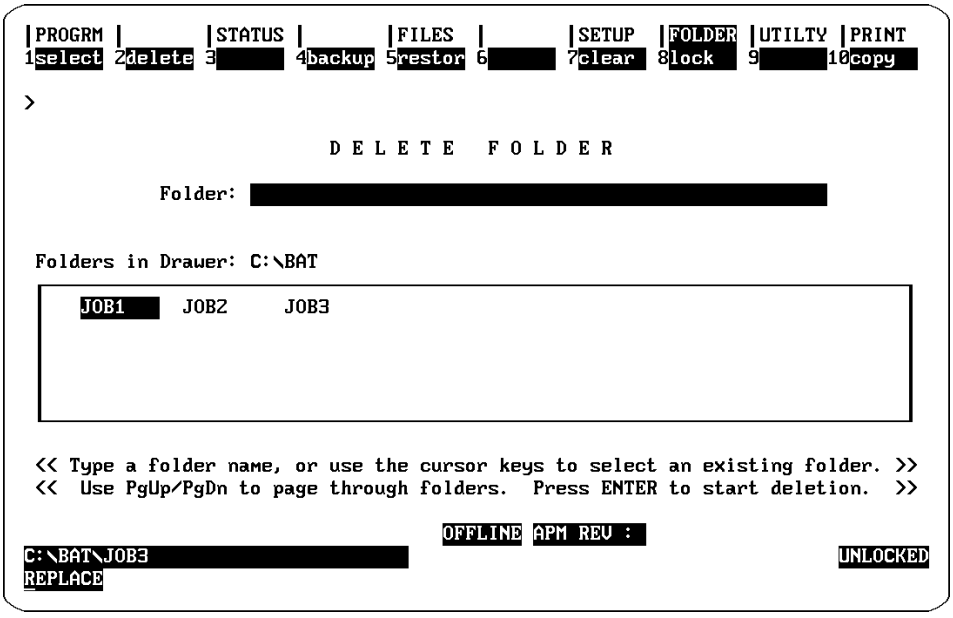


Figure 2-15. Delete Folder Screen

To delete a folder:

1. Position the cursor on the folder to be deleted. Note that the current folder cannot be deleted. Or type in the folder to be deleted in the folder field.

2. Press **Enter**. If the folder contains no files, the folder will be deleted. If the folder contains one or more APM files, the message, "Folder contains data; continue delete? (Y / N)" appears. Pressing "N" will abort the function, and pressing "Y" will delete all files in the folder and the folder itself. Or, if the folder contains Logicmaster 90 files, the message "LM90 files exist in the folder; deletion not allowed." appears. (You must go to the Folder functions in the Logicmaster 90 Programmer package to delete folders containing Logicmaster 90 files).

Backing-Up All APM Files in a Folder

All APM files in the selected folder can be backed up. To enter the Backup screen, press **(F4) backup** from the main Folder Functions screen or any of the other Folder sub-screens. The Backup screen is shown below.

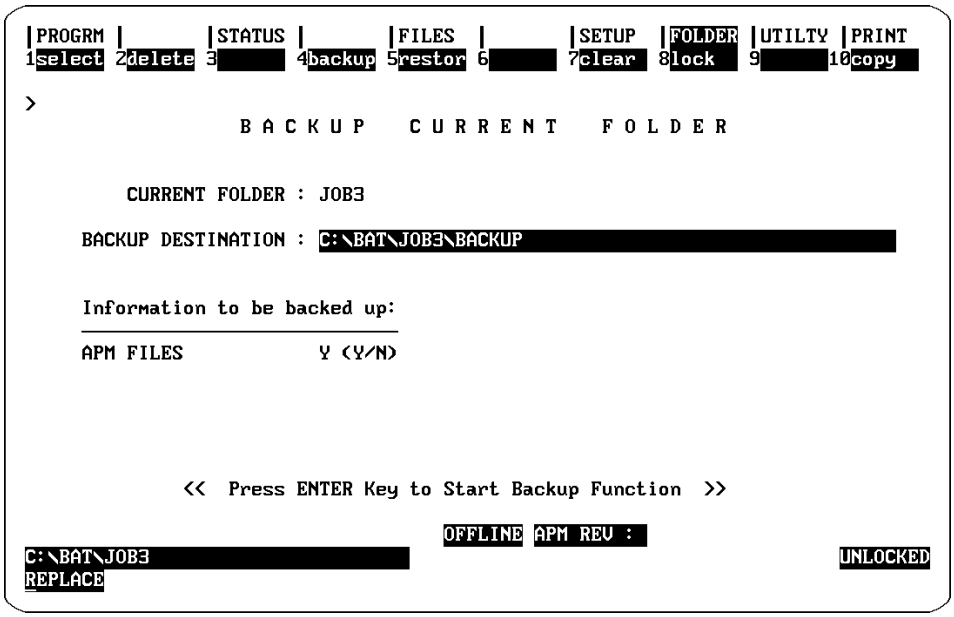


Figure 2-16. Backup Folder Screen

The default backup directory is a subdirectory under the folder subdirectory selected. You may choose your own backup directory by typing in the desired path in the backup destination field.

To backup the files of the selected folder:

1. Use the default destination or enter your own backup destination..
2. Press **Enter**.

Restoring All APM Files in a Folder

A folder which has been backed up can be restored through the Restore Folder screen. To enter the Restore screen, press **(F5) restore** from the main Folder Functions screen or any of the other Folder sub-screens. The Restore screen is shown below.

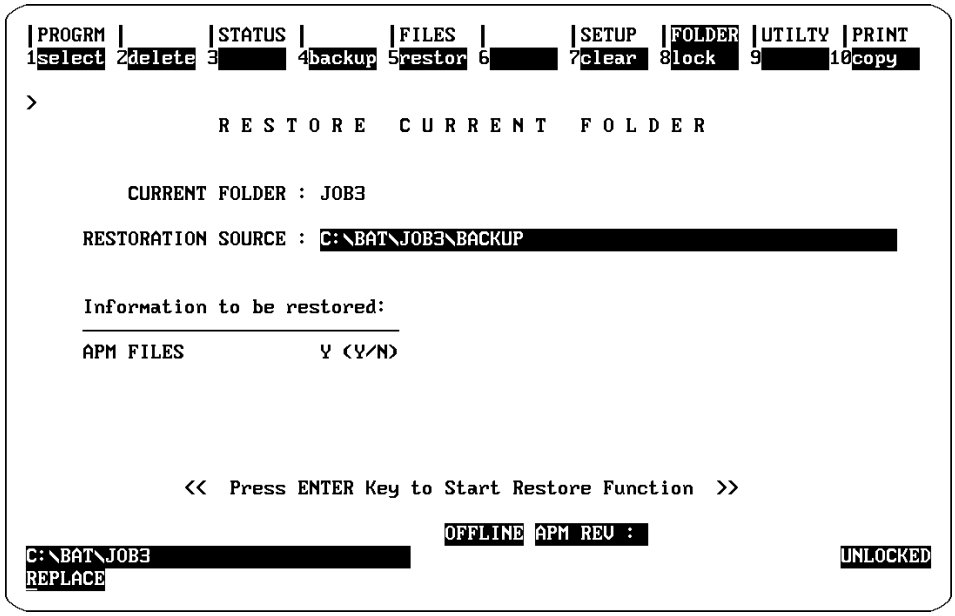


Figure 2-17. Restore Folder Screen

When the Restore Folder screen is displayed. The restoration source field will contain the path to the directory where the select folder was backed up.

To restore the files of the selected folder:

1. Use the default restoration source, or type in your own source directory.
2. Press **Enter** to restore the APM files.

Clearing All APM Files in a Folder

All APM files in the selected folder can be deleted through the Clear Folder screen. To enter the Clear Folder screen, press **(F7) clear** from the main Folder Functions screen or any of the other Folder sub-screens. The Clear Folder screen is shown below.

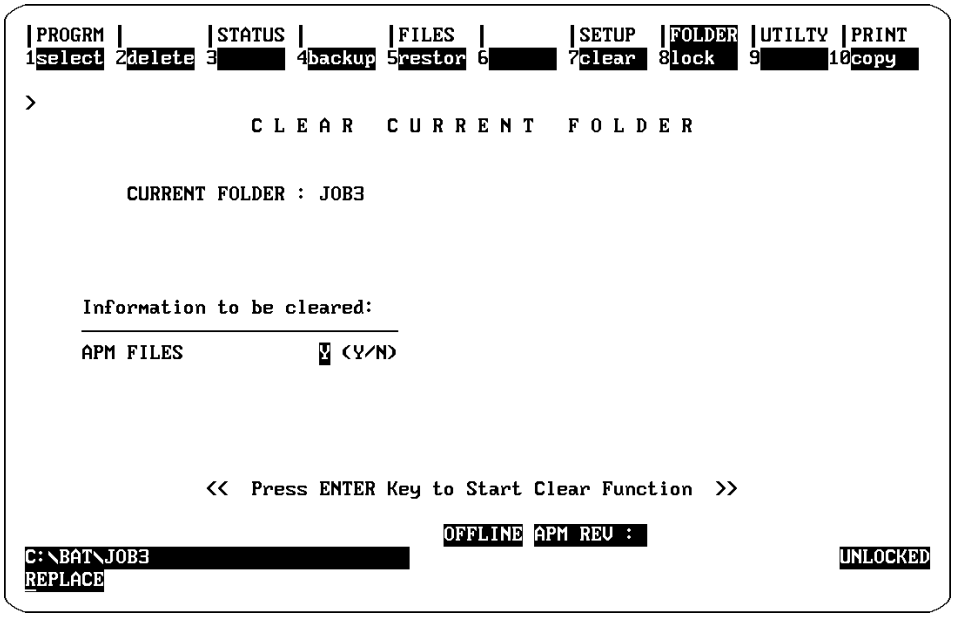


Figure 2-18. Clear Folder Screen

To clear files in a folder:

From the Clear Folder screen press **Enter**.

The following message will be displayed: "Program folder contains data; continue clear? (Y / N)" If you enter N, the function will be aborted. If you enter Y, the currently selected APM file will be closed, and then all the APM files will be deleted.

Locking and Unlocking Current Program Folder

Locking a program folder protects its files against accidental alteration. To change the locked status of the current program folder, select **(F8) lock** from the Program Folder Functions menu. The Lock or Unlock Current Folder screen appears.

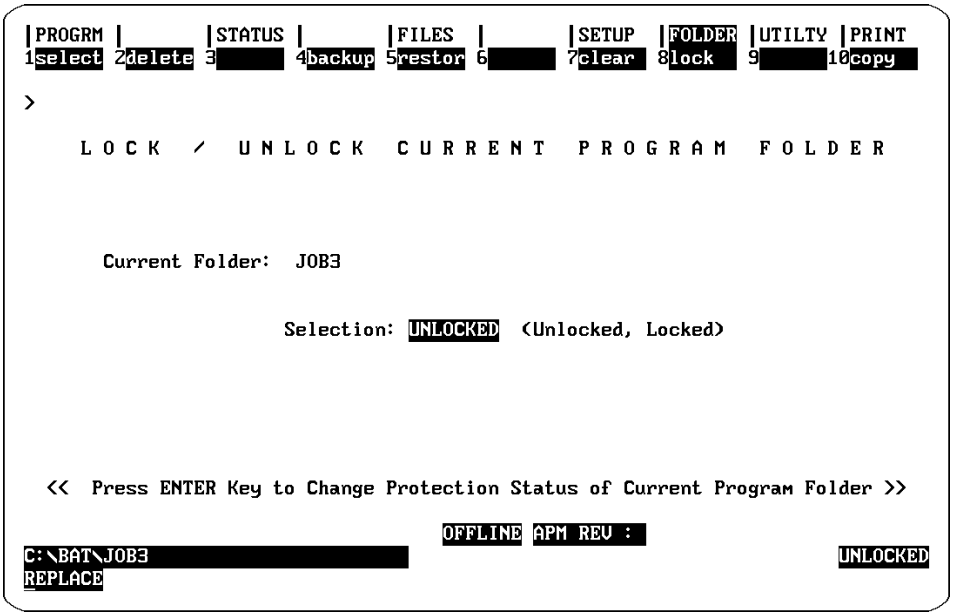


Figure 2-19. Lock or Unlock Current Folder Screen

To change the locked or unlocked status:

1. Use the **Tab** key to change locked or unlocked status.
2. Press **Enter**. The new access description will appear in the lower right corner of the screen.

Note

The lock function applies not only to APM files, but also ladder program and configuration files.

Program folders on write-protected floppy diskettes are automatically locked. You must remove the write-protect tab and unlock the folder using this function.

Copying All APM Files in a Folder

All APM files in a source folder can be copied to the selected folder through the Copy Folder screen. To enter the Copy Folder screen, press (F10) copy from the main Folder Functions screen or any of the other Folder sub-screens. The Copy Folder screen appears.

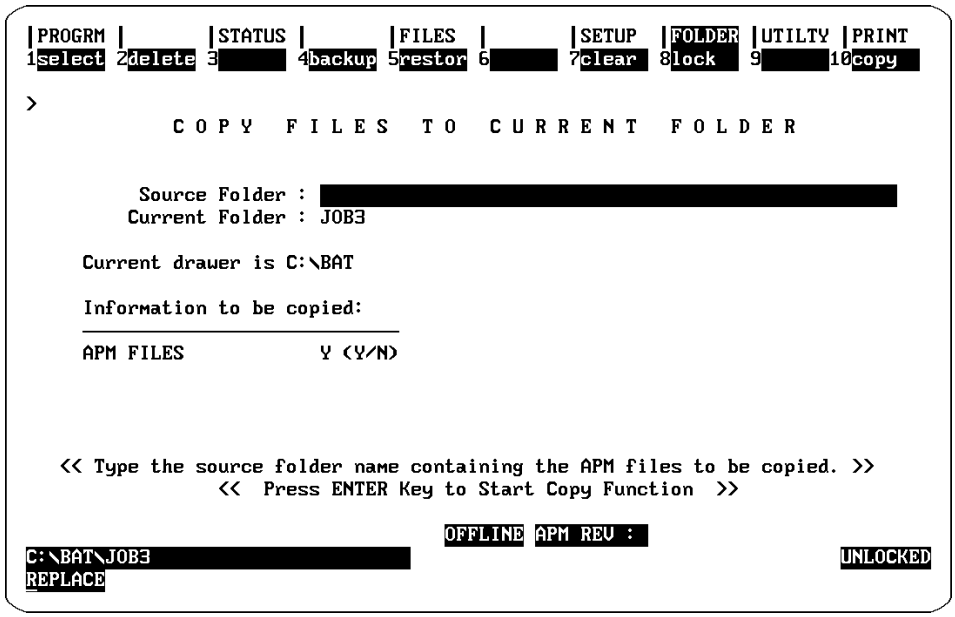


Figure 2-20. Copy Folder Screen

To copy files from one folder to another:

1. Before entering the Copy Folder screen, select the folder *to* which you wish to copy the files. Then enter the Copy Folder screen by pressing the **(F10) copy** key.

2. Enter the path name of the source folder in the source folder field.

   Example:C:\LM90\JOB2 if JOB2 is in a different drawer than the current folder.

   Example: JOB2 if JOB2 is in the same drawer as the current folder.

3. Press **Enter** to copy the APM files to the current folder.

Section 5. Setting-Up the Programmer Ports

This section describes the Setup Functions. The Setup Functions are used to configure the programmer's ports for connection to a printer or the APM and to select the programmer mode. The following topics are covered in this section.

- Setting the ProgrammerMode(Offline / Monitor / Online)
- Setting-Up the Printer Serial Port
- Setting-Up APM Communications

To access the Setup Screen:

From the Main Menu, press the **(F7) setup** or **Shift-(F7) SETUP** key. The Setup functions are available from any other screen in which the **SETUP** key is displayed.

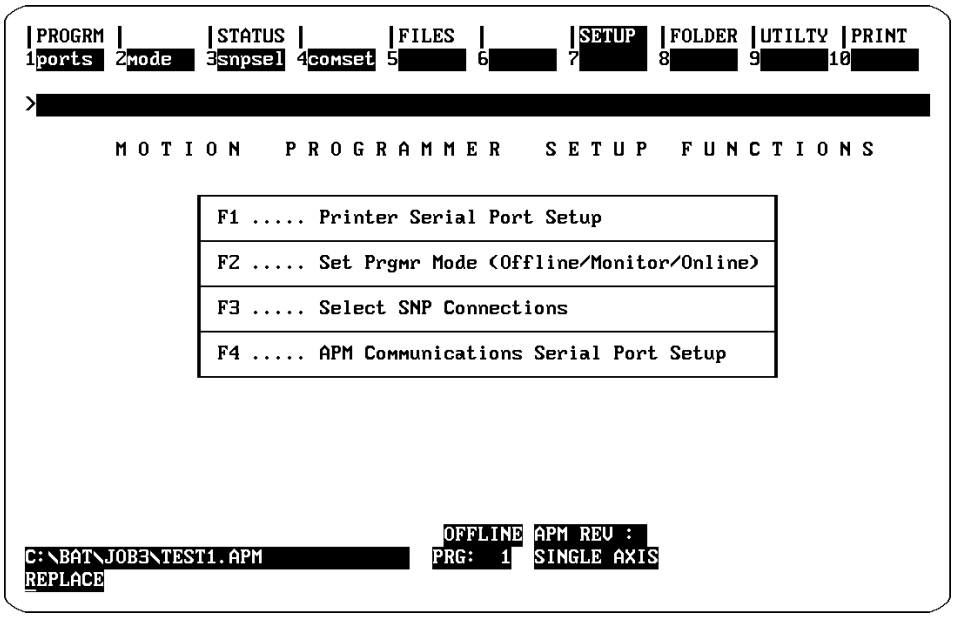


Figure 2-21. The Setup Functions Screen

Setting-Up the Printer Serial Port

The printer serial port is configured in the Printer Serial Port Setup screen.

To access the Printer Port Setup Screen:

Press the **(F1) ports** key from the Motion Programmer Setup Functions screen.

The following screen will appear. You can always return to the Main Setup screen by pressing the **Esc** key.

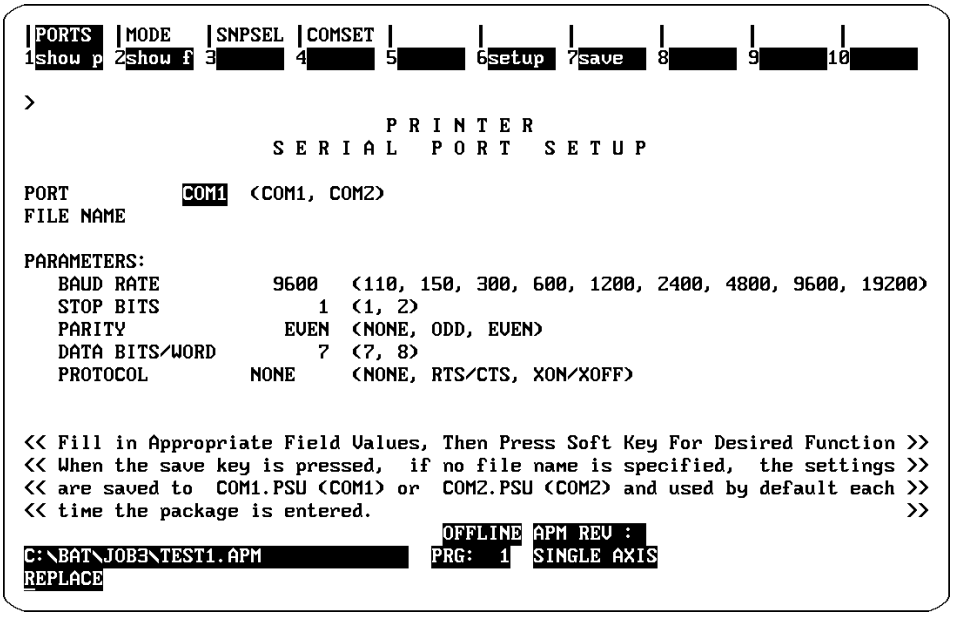


Figure 2-22. The Printer Port Setup Screen

From this screen, you can change printer parameters, save the current print settings to a file, or recall print settings from a file. The screen will contain default values the first time it is displayed.

Note

One of the COM ports (1 or 2) is used to communicate with the APM. The communications port is selected and configured in the PLC Communications Serial Port Setup screen. The Printer Port Setup screen displays the settings of this port, but you cannot change the settings.

To show the current printer port settings:

1. Select COM1 or COM2 in the port field.
2. Press the **(F1) show p** key.

To show the stored printer port settings:

1. Enter COM1, COM2, or the name of the file in the file name field.
2. Press the **(F2) show f** key.

To setup the printer port:

1. Position the cursor on the field you want to change.
2. Enter the desired value in the field or press the **Tab** key to step through the selections.
3. Press the **(F6) setup** key.

To save the port settings:

1. To save the settings to the default file name, make sure the file name field is blank and press the **(F7) save** key. This will save COM1 settings to %COM1.PSU, and COM2 settings to %COM2.PSU. These files are saved to the directory that contains the Motion Programmer software.

2. If you want to specify the file name, type in the desired file name in the file name field and press the **(F7) save** key.

Setting the Programmer Mode

The programmer mode can be changed between Offline, Monitor, and Online.

To access the Set Programmer Mode Screen:

Press the **(F2) mode** key from the Motion Programmer Setup Functions screen.

The following screen will appear. You can always return to the Main Setup screen by pressing the **Esc** key.

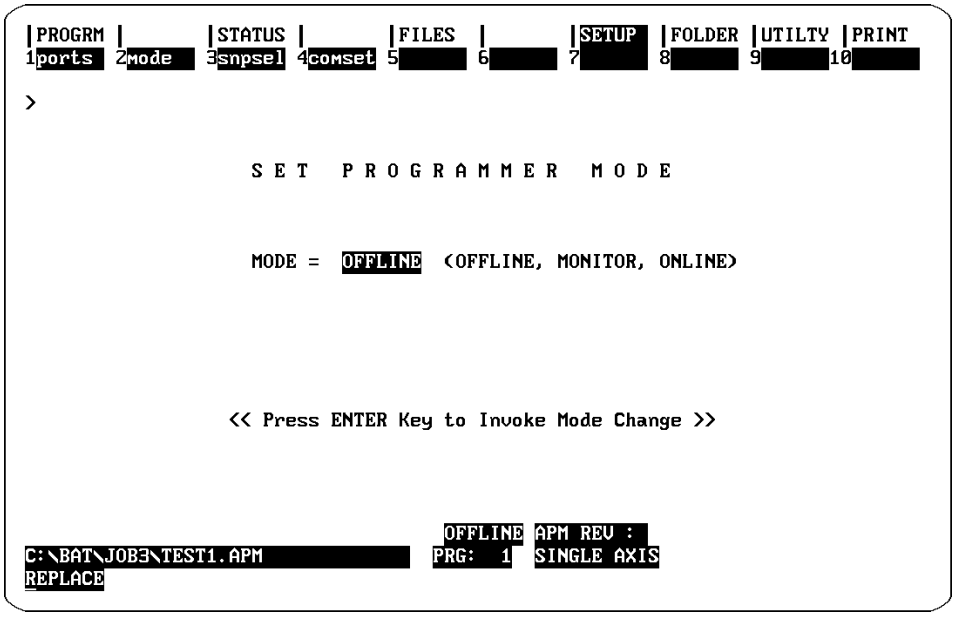


Figure 2-23. The Set Programmer Mode Screen

To change the Programmer Mode:

1. Press the **Tab** key to toggle the Programmer Mode to the desired setting.

2. Press the **Enter** key to accept the change.

Setting-Up APM Communications

To setup communications between the Motion Programmer and the APM, you will need to complete the following steps.

1. Connect the serial cable from the computer running the Motion Programmer software to the serial port on the APM.

Note

The connection between the programming computer and the APM30 is typically made from the RS-232 port of the computer through an RS-232 to RS-485 / RS-422 converter to the SNP Serial Communications Connector on the APM. A Miniconverter Kit (IC690ACC901) which includes a converter and 6 ft cable is available for this purpose. This is the same cable and converter that is used to connect to the SNP port on the CPU of the PLC.

If you are making your own cable, refer to GFK-0707, *The APM30 Quick Reference and Installation Guide*, for the pin-outs of the serial communications connector.

2. Go to the SNP Connections screen and select Direct or Multidrop communications. If you are using multidrop communications, you must select an SNP ID.

3. Go to the PLC Communications Serial Port Setup screen to set the communications parameters to match that of the APM Serial Communications port which is configured in the Logicmaster 90-30 configuration software.

4. Make sure the Motion Programmer is in the Online or Monitor mode by pressing Alt-M. Note that when communications are established, the Enabled / Disabled field and the SNPID field will appear.

The Select SNP Connections Screen

The Select SNP Connections screen is used in Multidrop applications. SNP is the proprietary communications protocol used by the APM. From this screen you select the SNP ID of the APM to which you want to communicate. You may also create a list of the SNP IDs you are using with a 6-character description for each.

The settings for these screens are saved to separate files. The procedures for using the show port, show file, setup, and save functions are the same as for the printer setup described above.

To access the Select SNP Connections Screen:

Press the **(F3) snpsel** key from the Motion Programmer Setup Functions screen.

The following screen will appear. You can always return to the Main Setup screen by pressing the **Esc** key.

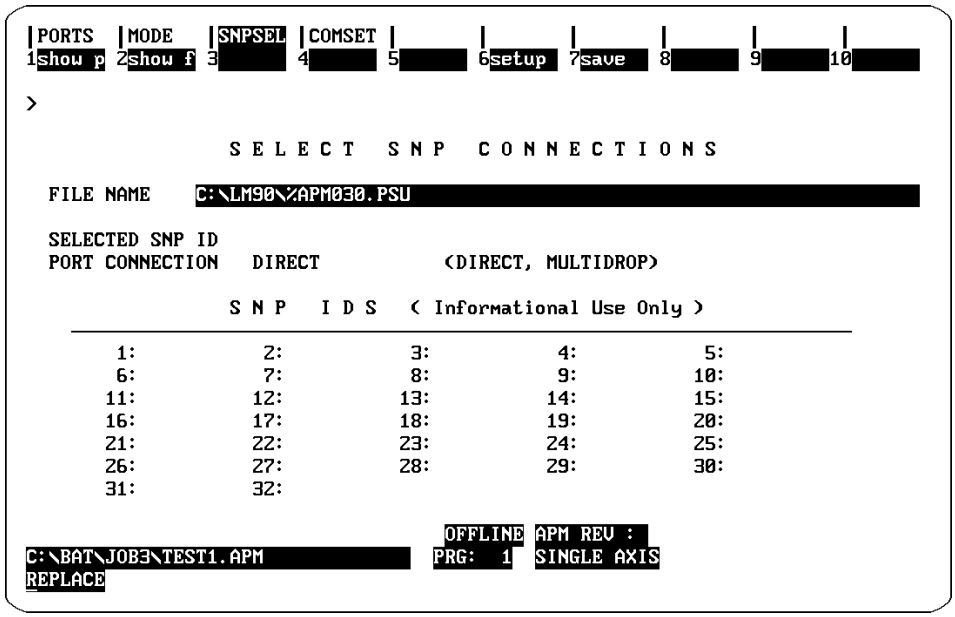


Figure 2-24. The Select APM Connections Screen

The fields on this screen are described below.

File Name. This is the name of the file containing the SNP connection information. If you do not specify a file name, the default file name will be %APM030.PSU. If no path is specified, the current default directory will be used. If no extension is specified, the default .PSU will be used.

Selected SNP ID. This name you can enter to identify the APM you want to communicate with. This field is applicable only when multidrop communications are selected.

Port Connection. This sets-up the port for direct (point-to-point) connection or multidrop connection.

SNP IDs. The SNP ID field allows you to enter a 6-character description for the SNP IDs which you defined for multidrop communications. This field is used only to create a handy list of multidrop connections. It is not used by the Motion Programmer software.

The PLC Communications Port Setup Screen

In the Communications Port Setup screen you select and configure the port used to communicate with the APM.

To access the PLC Communications Port Setup Screen:

Press the **(F4) comset** key from the Motion Programmer Setup Functions screen.

The following screen will appear. You can always return to the Main Setup screen by pressing the **Esc** key.

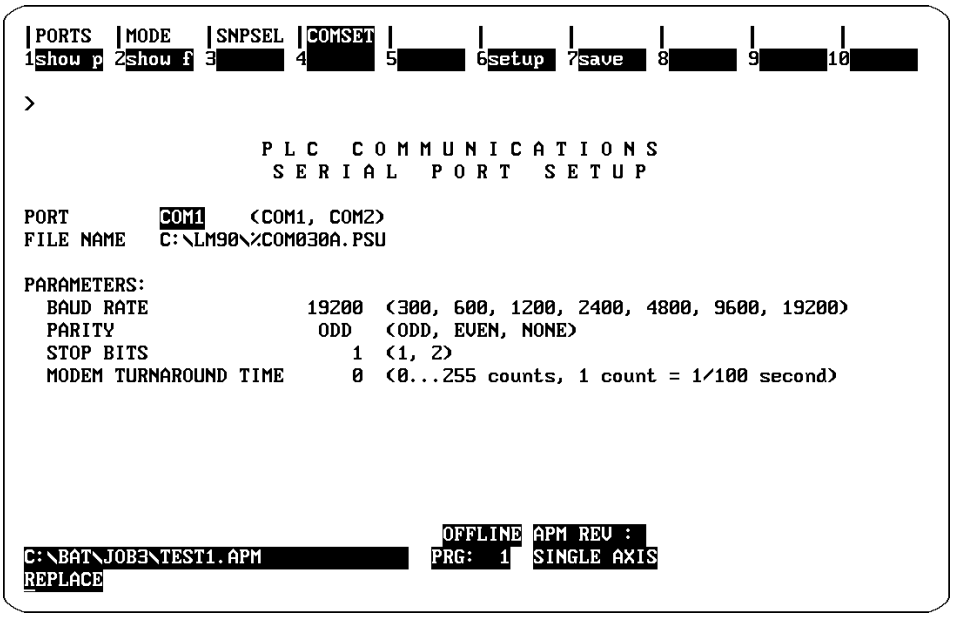


Figure 2-25. APM Communications Port Setup Screen

Follow the same procedures for using the show port, show file, setup, and save functions as for the printer port setup described earlier.

The default file name will be %COM030.PSU.

Section 6. Loading and Storing APM Files

This section describes the functions available in the Utilities Functions screen. These functions are similar to the utilities functions in the Logicmaster 90 Programmer. Topics covered are:

- Loading from the APM to the Programmer
- Storing from the Programmer to the APM
- Verifying the APM with the Programmer
- Clearing APM memory.
- Read / Write / Verify EEPROM

To access the Utilities Functions screen:

Press the **(F9) utilty** key from the Main screen.

You can execute all the utilities functions from this screen.

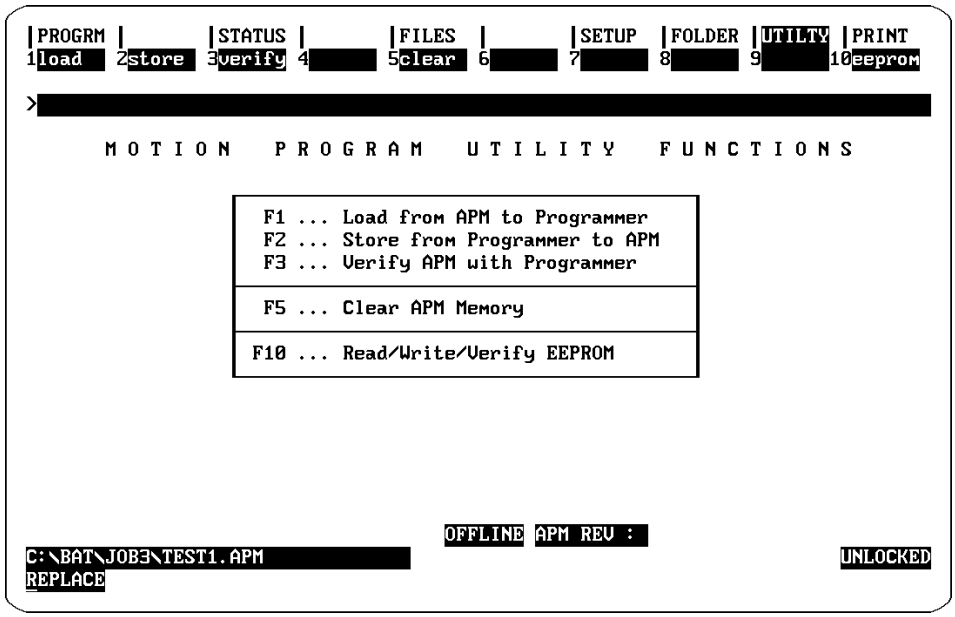


Figure 2-26. Utilities Functions Screen

Loading from the APM to the Programmer

The load function is executed from the Load From APM to Programmer screen.

To access the Load From APM to Programmer Screen:

Press the **(F1) load** key from the Motion Programmer Utility Functions screen.

The following screen will appear. You can always return to the Utility Functions screen by pressing the **Esc** key.

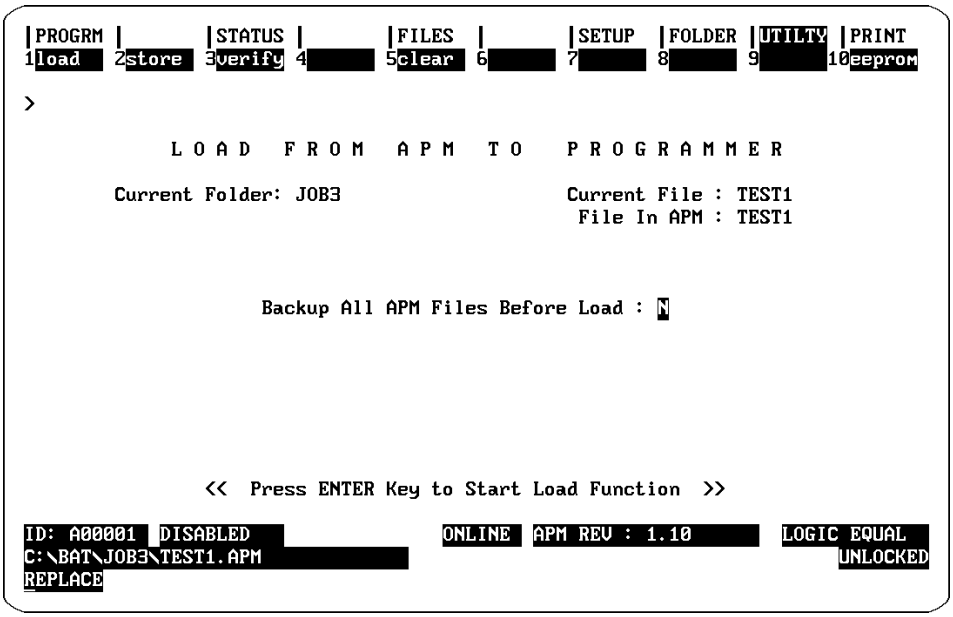


Figure 2-27. Load from APM to Programmer Screen

To load a file from the APM to the programmer:

1. Before entering the Load from APM to Programmer screen, ensure the following.
   - Make sure communications are established and the programmer is in either the Monitor or Online mode. This is done in the Programmer Setup Functions, or by pressing the **Alt-M** key combination. The PLC-CPU may be running and the APM may be executing a motion program.
   - The file in the APM must be loaded to either a Motion Programmer file of the same name or a file named TEMP.APM. Make sure the Current File field meets this requirement.
2. In the Load from APM to Programmer screen, set the Backup All APM Files field to Y or N as desired.
3. Press the **Enter** key to begin. An error message will be displayed if the conditions above are not met. Once the load begins, the prompt "Current file will be overwritten-continue? (Y / N)". Entering a "Y" will start the load ("N" will abort the load). If you elected to backup files, this is performed. Finally, the load function will begin. If you wish to abort the load, press the **Alt-A** key combination.

If an error occurs during the load, an error message will be displayed, but the selected file will not be affected. The Logic Equality Status Field will display "LOGIC NOT EQ".

If the load finishes successfully, a "Load complete" message will be displayed, and the Logic Equality Status Field will display "LOGIC EQUAL".

Storing from the Programmer to the APM

The Store function is executed from the Store from Programmer to APM screen. This function stores a file to the APM RAM memory, and, if selected, to the APM EEPROM.

To access the Store from Programmer to APM Screen:

Press the **(F2) store** key from the Motion Programmer Utility Functions screen.

The following screen will appear. You can always return to the Utility Functions screen by pressing the **Esc** key.

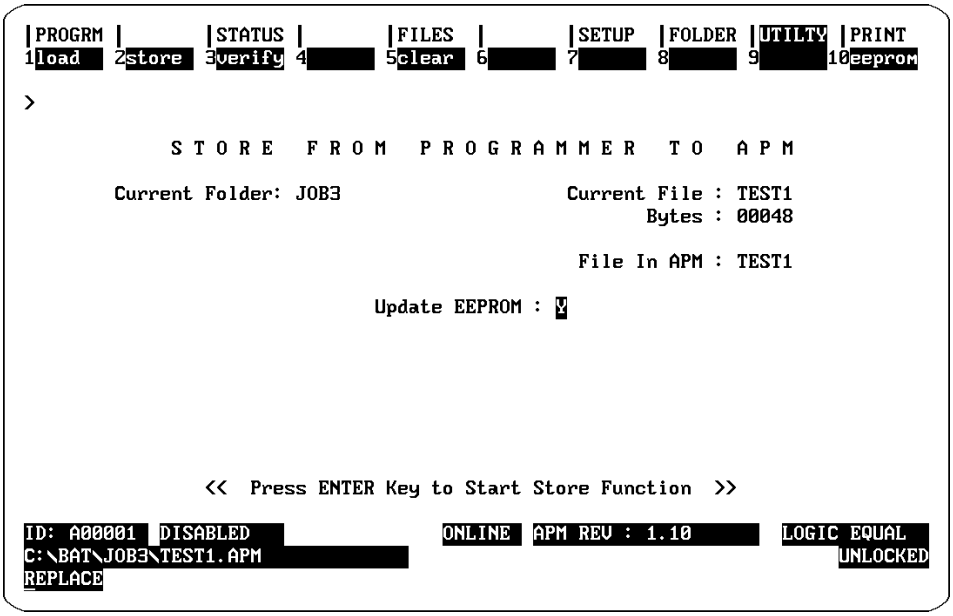


Figure 2-28. Store from Programmer to APM Screen

To store a file from the Programmer to the APM:

1. Before storing a file do the following.
   - Establish communications with the APM; the Motion Programmer must be in the On-Line or Monitor mode.
   - Make sure the APM is disabled. This can be determined after communications is established with the APM by checking the Enabled/Disabled Field in Status Line 1.

Note

The APM is always disabled when the PLC is Stopped. The APM can be disabled while the PLC is running by setting the *Drive Enabled* %I bit to 0 and making sure the APM is *not* jogging. (Use operator switches or the Logicmaster Programmer Status Tables to clear the *Enable Drive* %Q command bit).

   - If you have not selected the file you want to store to the APM, go to the File and Folder Functions and select it.

2. From the Store screen, select whether you want to update the APM EEPROM, Y or N.

3. Press the **Enter** key. If the file name you selected is the same as the file name in the APM, the store will execute. If the file names do not match, the following message will be displayed, "File name in the APM does not match file to be stored; continue? (Y / N)" will be displayed. If the user answers "N" to the prompt, a message such as "Store aborted" will be displayed. However, if the user answers "Y", then the store will continue as if the names matched.

The store function will first check to see whether there is enough APM user RAM to hold all the programs and subroutines. If there is not enough user memory, an error message will appear. If there is enough APM user RAM, then a confirmation prompt to continue the store will be displayed. If you reply N, the store will abort; if you reply Y, the store will continue. After this point you may still abort the function by pressing the **Alt-A key** sequence.

Note

Release 1 APMs have 16 K bytes of program memory.

Release 2 APMs have 14 Kbytes of program memory; 2 Kbytes have been separated for other uses.

Any errors which occur after this point cause an error message to be displayed and the APM to be cleared. The logic equality field is updated to "LOGIC NOT EQ".

If the store finishes successfully, then the "Store complete" message will be displayed. The File in APM field will be updated to reflect the file just stored, and the logic equality field is updated to "LOGIC EQUAL".

Verifying the APM with the Programmer

The Verify function compares the current file in the APM with the file in the Programmer. This function is executed from the Verify APM with Programmer screen.

To access the Verify APM with Programmer Screen:

Press the **(F3) verify** key from the Motion Programmer Utility Functions screen.

The following screen will appear. You can always return to the Utility Functions screen by pressing the **Esc** key.

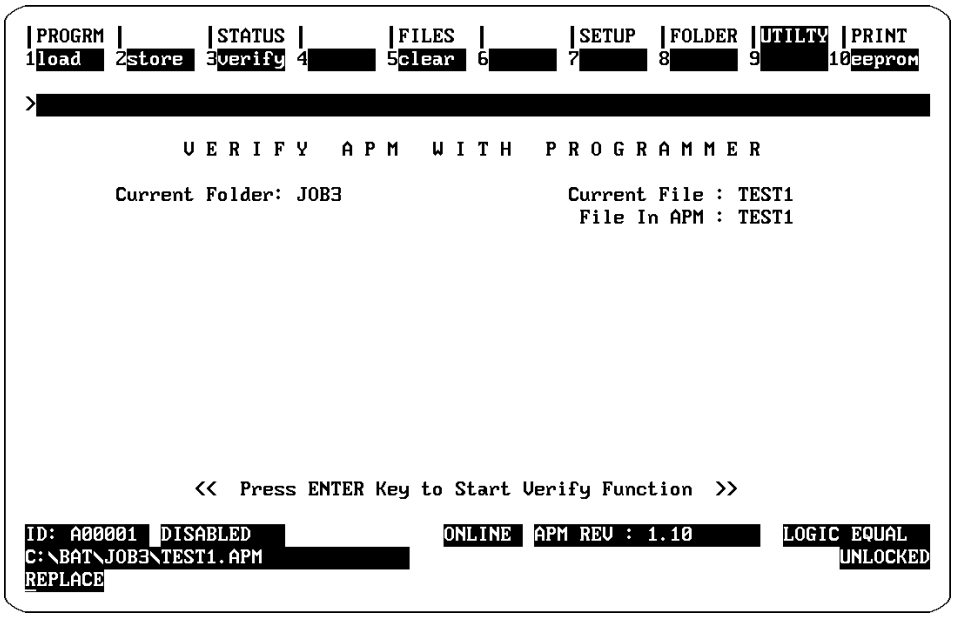


Figure 2-29. Verify APM with Programmer Screen

To verify the file in the APM with the current file in the Programmer:

1. Establish communications with the APM; the Motion Programmer must be in the On-Line or Monitor mode.

2. From the Verify APM with Programmer Screen press the **Enter** key.

The Programmer will begin comparing the programs and subroutines in the current file with the programs and subroutines in the APM. You can press the **Alt-A** key combination to abort.

If no errors are found, the message “No miscompares detected” is displayed and the Logic Equality field will be updated to “LOGIC EQUAL”. If an error is detected, the miscompares will be displayed on the screen and the Logic Equality field will be updated to “LOGIC NOT EQ”.

The following screen shows how miscompares are displayed.

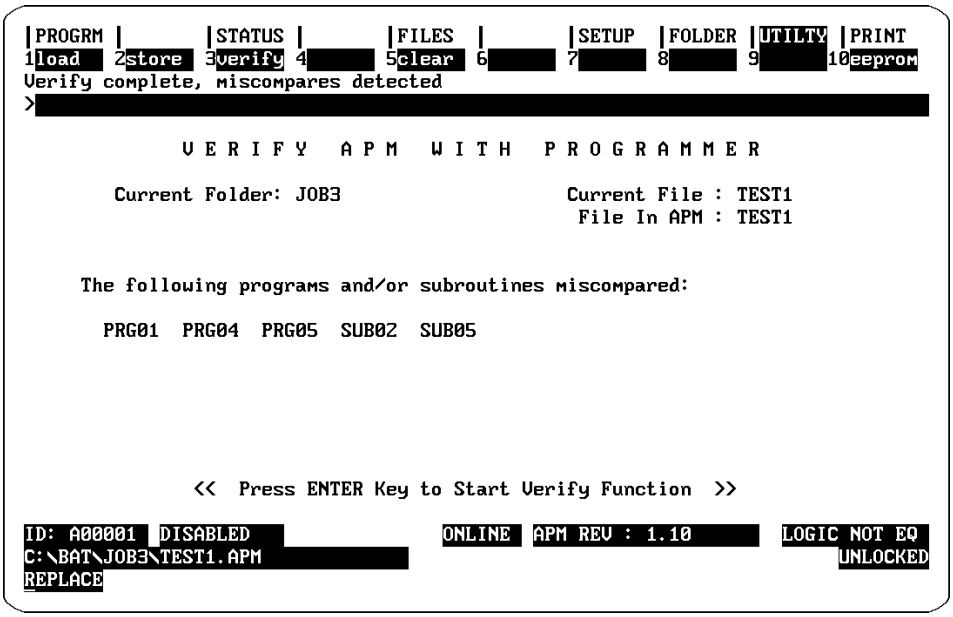


Figure 2-30. Miscompares on the Verify Screen

Clearing APM Memory

The Clear function clears the user RAM in the APM. This function is executed from the Clear APM User Memory screen.

To access the Clear APM User Memory Screen:

Press the **(F5) clear** key from the Motion Programmer Utility Functions screen.

The following screen will appear. You can always return to the Utility Functions screen by pressing the **Esc** key.

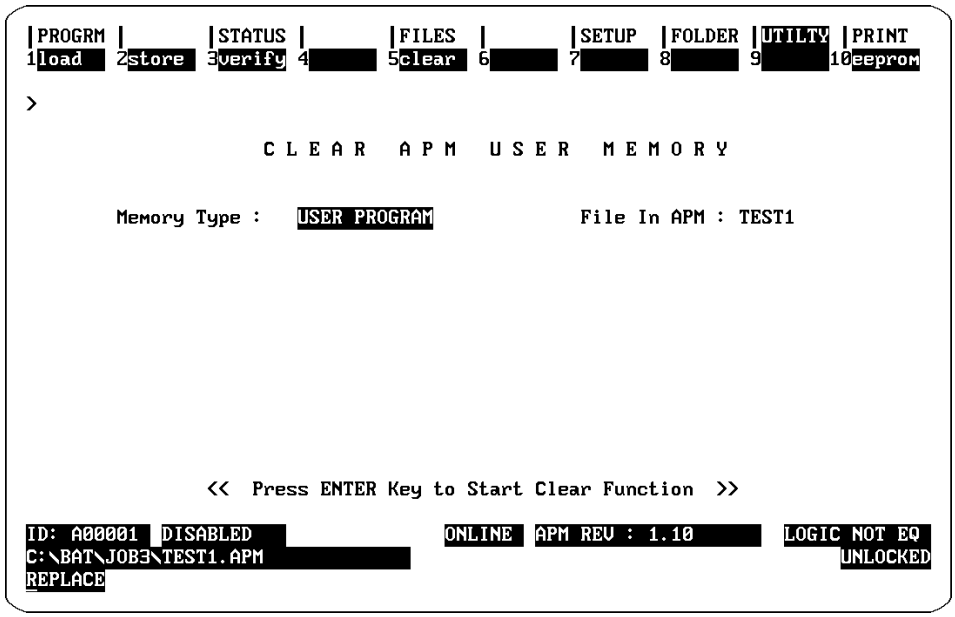


Figure 2-31. Clear APM User Memory Screen

To clear APM user memory:

1. Establish communications with the APM; the Motion Programmer must be in the On-Line or Monitor mode.

2. Make sure the APM is disabled. This can be determined by checking the Enabled / Disabled Field in Status Line 1.

3. From the Clear APM User Memory Screen select the desired CLEAR operation in the Memory Type field. Use the **Tab** key to toggle through the selections. The selections are described below.

   User Program: Clears all 10 programs and 40 subroutines from APM memory.
   Parameter Table: Sets all parameters to zero.
   All: Clears all 10 programs and 40 subroutines and sets all parameters to zero.

Note

> The APM is always disabled when the PLC is Stopped. The APM can be disabled while the PLC is running by setting the *Drive Enabled* %I bit to 0 and making sure the APM is *not* jogging. (Use operator switches or the Logicmaster Programmer Status Tables to clear the *Enable Drive* %Q command bit).

4. Press the **Enter** key. (The Motion Programmer must be in the On-Line or Monitor mode).

Before the Clear Function begins, a confirmation prompt will be displayed. If you reply "N", the function will abort. If you reply "Y", the operation will continue.

EEPROM Operations

EEPROM memory is permanent memory on the APM to which you can store files. Use this memory as you would if it were battery-backed storage. EEPROM Operations consist of:

- Reading contents of EEPROM into user RAM.
- Writing contents of user RAM into EEPROM.
- Verifying the EEPROM with the user RAM.

To access the EEPROM Operations screen:

Press the **(F10) eeprom** key from the Motion Programmer Utility Functions screen.

The following screen will appear. You can always return to the Utility Functions screen by pressing the **Esc** key.

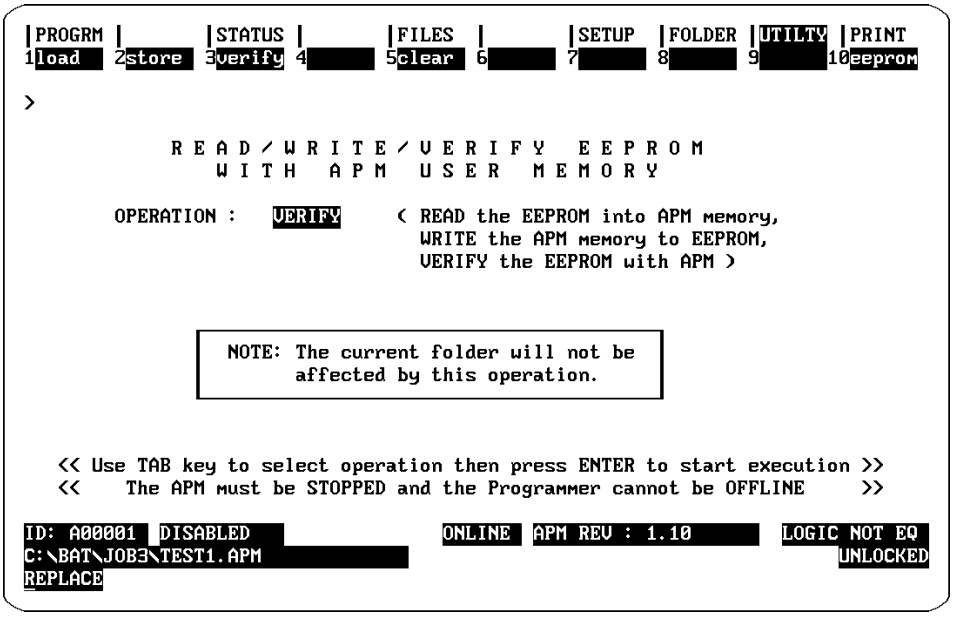


Figure 2-32. EEPROM Operations Screen

To execute an EEPROM Operation:

1. Before entering the EEPROM Operations screen, ensure the following.
   - Establish communications with the APM; the Motion Programmer must be in the On-Line or Monitor mode.
   - Make sure the APM is disabled. This can be determined by checking the Enabled / Disabled Field in Status Line 1.

Note

The APM is always disabled when the PLC is Stopped. The APM can be disabled while the PLC is running by setting the *Drive Enabled* %I bit to 0 and making sure the APM is *not* jogging. (Use operator switches or the Logicmaster Programmer Status Tables to clear the *Enable Drive* %Q command bit).

2. Select the desired EEPROM operation in the Operation field. Use the **Tab** key to toggle through the selections.

3. Press the **Enter** key.

Read EEPROM into APM User RAM. If the Operation field is set to READ, the APM Programmer will initiate a request to copy all the data from the EEPROM into the APM RAM memory.

Write APM User RAM into EEPROM. If the Operation field is set to WRITE, the APM Programmer will initiate a request to copy all the data from the APM RAM memory not the EEPROM.

Verify EEPROM with APM User RAM. If the Operation field is set to VERIFY, the APM Programmer will initiate a request to compare all the data from the EEPROM with the APM RAM memory.

Section 7. Printing APM Files

This section describes the functions available in the Print Functions screen. Topics covered are:

- Setting-Up Printer Parameters
- Selecting the Screen Print Destination
- Printing Programs and Subroutines
- Saving the Printer Setup and Screen Print Destination

To access the Print Functions screen:

Press the **(F10) print** key from the Main screen.

You can execute all the print functions from this screen.

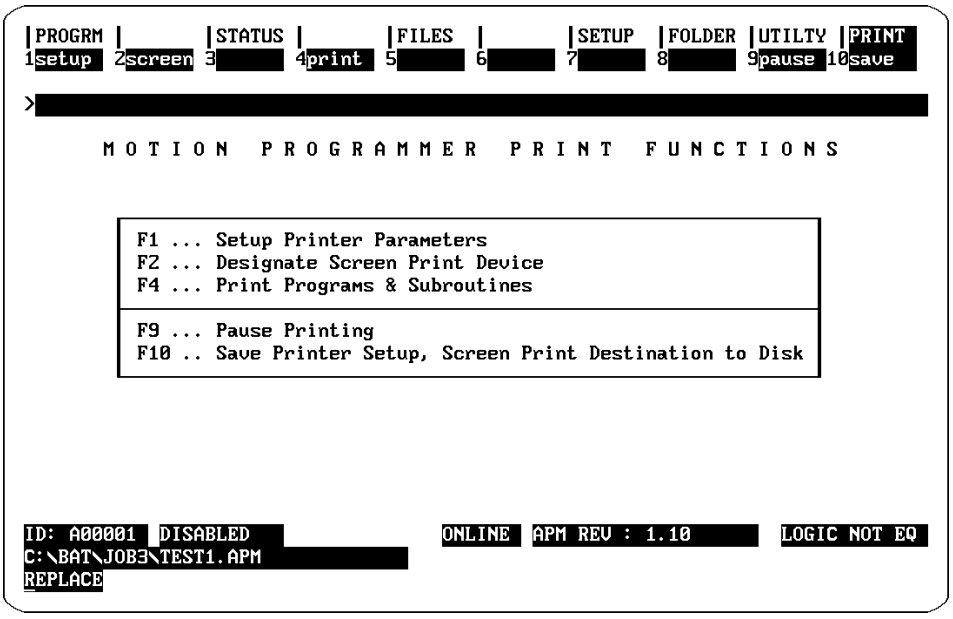


Figure 2-33. Print Functions Screen

Setting-Up Printer Parameters

Before you print a program, you may need to setup or change the current printer parameters. This is done in the Setup Printer Parameters screen.

To access the Setup Printer Parameters screen:

Press the **(F1) setup** key from the Motion Programmer Print Functions screen.

The following screen will appear. You can always return to the Print Functions screen by pressing the **Esc** key.

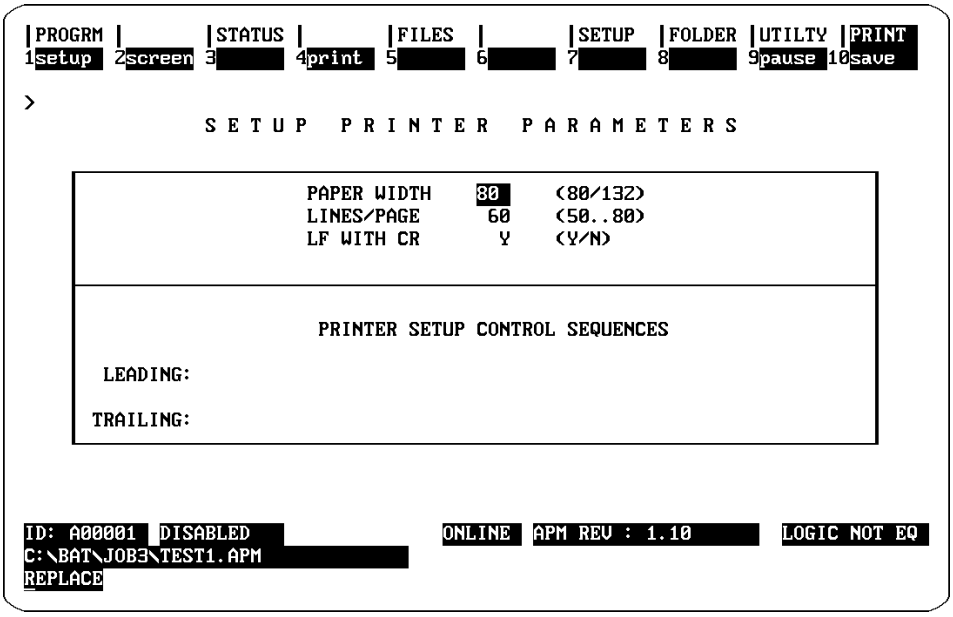


Figure 2-34. Setup Printer Parameters Screen

To setup printer parameters:

1. Type in or step through (**Tab** key) the desired setup parameters. Use the cursor keys to move between fields.

   These settings will remain active until you change them or until the next powerup of the Motion Programmer.

2. Enter any leading or trailing print control sequences that may be desired.

3. If you want to save them permanently, press the **(F10) save** key. This will save these parameters to the PRINT.SET file in the startup directory.

Paper Width. The entry for paper width determines the number of characters that will be printed on a line. If the printer is set up for standard 8 1/2-inch wide paper, select 80 characters for the paper width. If the printer uses 11-inch (or wider) paper you may select either 80 or 132 characters.

Number of Lines per Page. The number of lines that can be printed on a page ranges from 50 to 80, with 60 lines per page being the default value. If an associated group of lines will not fit on one page you can specify more lines per page to achieve the result.

Line Feed with a Carriage Return. The line feed character advances the paper to the next line for printing. This item determines whether the system automatically inserts a line feed each time the printer head should return to the left page margin.

Note

Some printers can be set up to automatically advance to the next line. Refer to the documentation which accompanied you printer to determine how your specific printer works.

Printer Setup Sequences. If the printer uses leading or trailing control characters, enter the characters here. This feature can be used to put the printer into a particular mode (for example, compressed output), returning to the original mode after the listing is finished. From 0 to 60 characters can be specified. To enter non-printing characters, use a backslash followed by the 3-digit decimal equivalent of the ASCII representative. For example, to identify the Escape character <ESC>, you would enter \027. A leading zero is required. If you need to enter the backslash character itself,enter\ \.

Selecting the Screen Print Destination

This option allows you to select the Print Screen destination as either a serial or parallel printer or a disk file.

To access the Select Screen Print Destination screen:

Press the **(F2) screen** key from the Motion Programmer Print Functions screen.

The following screen will appear. You can always return to the Print Functions screen by pressing the **Esc** key.

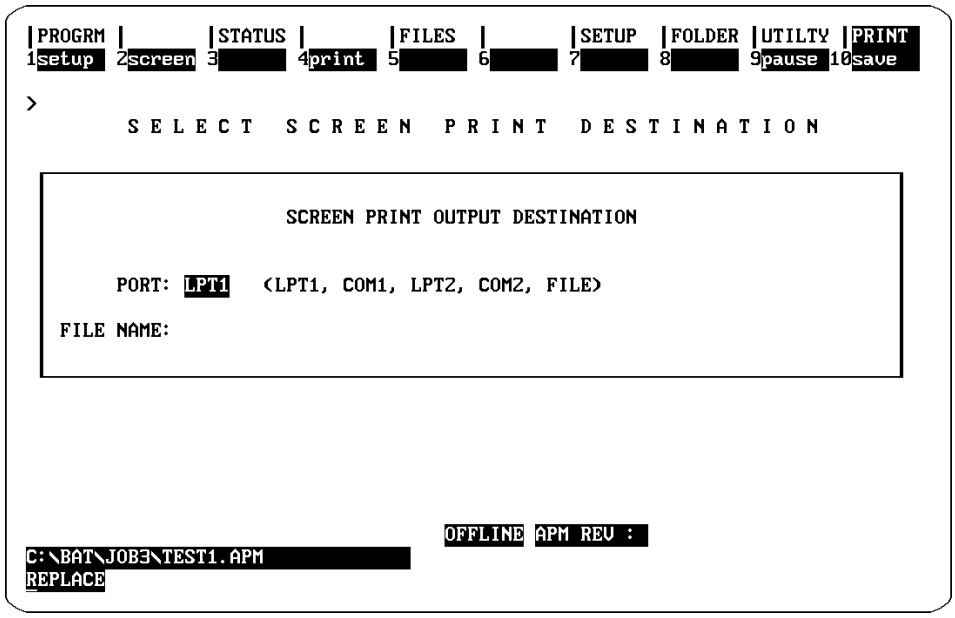


Figure 2-35. Select Screen Print Destination Screen

To print a screen:

1. Select the destination port using the **Tab** key. If you select “FILE”, type in the destination file name for the screen.

   These settings will remain active until you change them or until the next powerup of the Motion Programmer.

2. If you want to save them permanently, press the **(F10) save** key. This will save these parameters to the SCRPRINT.SET file in the startup directory.

3. Go to the Motion Programmer screen you want to print.

4. Print the screen by pressing **Alt-P**.

Printing Programs and Subroutines

This screen allows you to select the Programs and Subroutines to print.

To access the Print Motion Programs screen:

Press the **(F4) print** key from the Motion Programmer Print Functions screen.

The following screen will appear. You can always return to the Print Functions screen by pressing the **Esc** key.

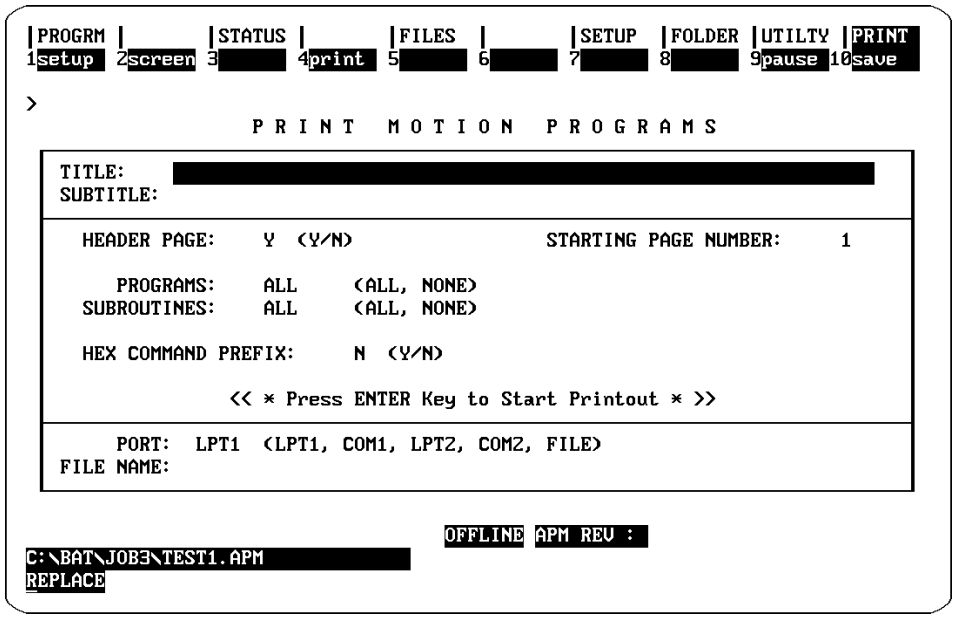


Figure 2-36. Print Motion Program Screen

To print programs and subroutines:

1. Fill in the fields on the screen, see the description of the fields below.

2. Press the **Enter** key to begin the print operation.

Title. The title of the printout, which is printed at the top of every page.

Subtitle. The subtitle of the printout, which is printed underneath the title of every page.

Header Page Option. You can set this to "no" if you do not want the header page to be printed. Default = yes.

Starting Page Number. You can specify the starting page number for the print out. Default = 1.

Programs. This option allows you to specify whether to print all or none of the programs in an APM file. Default = All.

Subroutines. This option allows you to specify whether to print all or none of the subroutines in an APM file. Default = All.

Hex Command Prefix. This option allows you to specify whether to print the 6-byte command (in hex) before its corresponding line in the program. If the line contains comments that extend out to the 80 column length, and the printer is only set up for 80 columns, then the comment will be truncated on the printout (if necessary). This is done since the 6-byte hex command needs to prefix the line being printed. Default = N.

Destination Port. This allows you to select the destination for the printout, a serial or parallel port or a disk file.

File Name. This field is used when you have specified FILE as the destination of the printout. If no path is specified (only the filename), then the print file will be put into the current folder. If a path is specified, the print file will be put in that path.

Pausing the Print Operation

You can pause the printout by pressing the **(F9) pause** key. To resume printing, press the **(F9) pause** key again.

Chapter 3 Motion Program Examples

This chapter includes examples of typical moves and combinations of moves.

Calculating Move Distances

Programmed *Move* commands must allow enough move distance to achieve the programmed velocity at the specified acceleration rate. If this is not the case, an error profile will result . The formulas below show how to calculate the various acceleration distances.

Table 3-1. Formulas for Calculating *Linear* Move Distances

Minimum distance to accelerate when initial velocity is 0 or to decelerate when final velocity is 0.

$$\text{Distance} = \left| \frac{\text{Velocity}^2}{2 \times \text{Acceleration}} \right|$$

Minimum distance to accelerate/decelerate when initial and final velocities are *not* 0.

$$\text{Distance} = \left| \frac{(\text{Velocity}_1)^2 - (\text{Velocity}_2)^2}{2 \times \text{Acceleration}} \right|$$

Distance = Required Move distance in counts.
Velocity = Programmed Velocity in counts / sec.
Acceleration = Programmed Acceleration in counts / sec / sec.

Table 3-2. Formulas for Calculating *S-Curve* Move Distances

Minimum distance to accelerate when initial velocity is 0 or to decelerate when final velocity is 0.

$$\text{Distance} = \left| \frac{\text{Velocity}^2}{\text{Acceleration}} \right|$$

Minimum distance to accelerate/decelerate when initial and final velocities are *not* 0.

$$\text{Distance} = \left| \frac{(\text{Velocity}_1)^2 - (\text{Velocity}_2)^2}{\text{Acceleration}} \right|$$

Distance = Required Move distance in counts.
Velocity = Programmed Velocity in counts / sec.
Acceleration = Programmed Acceleration in counts / sec / sec.

Single Pmove

To completely define a single move you must specify a minimum of three parameters: acceleration, velocity, and final position (or move distance). If you do not specify the acceleration and velocity values in the motion program, the APM30 will use the configured *Jog Acceleration* and *Jog Velocity* values. If you do specify the acceleration and velocity in the motion program, all successive move commands will use these values until new programmed values are encountered in the program.

These parameters relate directly to Motion Commands as shown in the program below. The APM converts these parameters to a velocity versus time profile as illustrated in the figure below.

Example

Starting at axis position 0, accelerate at a rate of 50,000 counts / sec / sec to a velocity of 20,000 counts / sec. Decelerate at a rate of 50,000 counts / sec / sec to a stop at axis position 22,000 counts.

Block No.	Command	Operand 1 (Axis #)	Operand 2	Operand 3	Operand 4
	ACCEL	1,	50000		
	VELOC	1,	20000		
	PMOVE	1,	22000	INC,	LINEAR

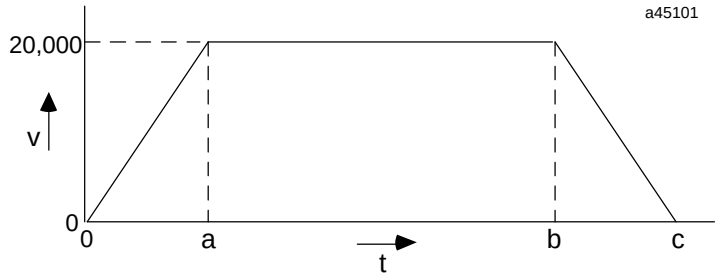


Figure 3-1. Single Pmove Profile

Minimum Move Distance

The minimum Move distance to reach the programmed velocity = (Distance 0→a) + (Distance b→c). This distance is calculated using the formulas in Table 3-1 as follows.

$$\text{Distance } (0 \rightarrow a) = \frac{\text{Velocity}^2}{2 \times \text{Acceleration}} = \frac{20{,}000^2}{2 \times 50{,}000} = 4{,}000 \text{ counts}$$

$$\text{Distance } (b \rightarrow c) = \frac{\text{Velocity}^2}{2 \times \text{Acceleration}} = \frac{20{,}000^2}{2 \times 50{,}000} = 4{,}000 \text{ counts}$$

So, the minimum Move distance = 4,000 counts + 4,000 counts = 8,000 counts. If the total distance programmed is less than 8000 counts, the programmed velocity will not be reached, but the axis will reach the programmed position without generating an error.

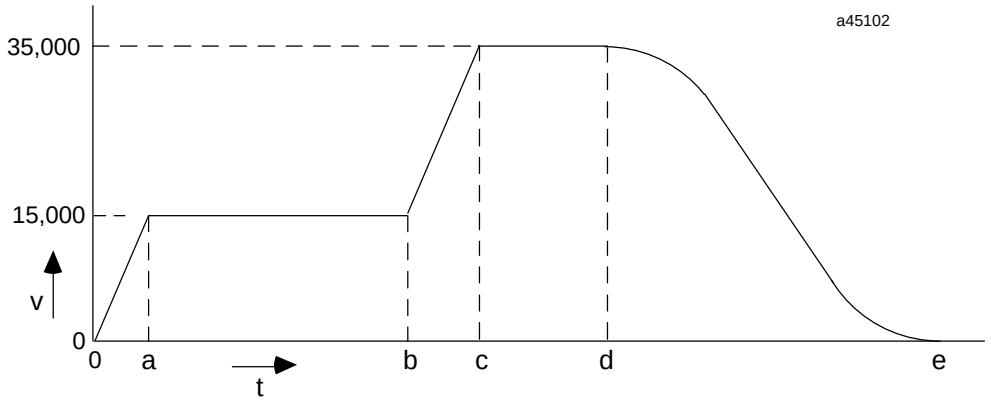
Cmove Followed by Pmove

This example illustrates a *Cmove* followed by a *Pmove.*

Example

Starting at axis position 0, use a *Cmove* (I*ncremental)* to move 23,000 counts an acceleration rate of 60,000 counts / sec / sec to a velocity of 15,000 counts / sec. Move 27,000 more counts using a *Pmove (S-curve)* with the same acceleration rate but a velocity of 35,000 counts / sec.

Block No.	Command	Operand 1 (Axis #)	Operand 2	Operand 3	Operand 4
	ACCEL	1,	60000		
	VELOC	1,	15000		
	CMOVE	1,	23000	INC,	LINEAR
	VELOC	1,	35000		
	PMOVE	1,	22000	INC,	LINEAR



0→c represents the Cmove distance of 23,000 counts
c→e represents a Pmove distance of 27,000 counts

Figure 3-2. Cmove followed by Pmove Profile

Note

In this example, the programmed Velocity for the *Pmove* is attained at the programmed distance for the *Cmove.*

Minimum Move Distance

The minimum Move distance for the programmed Move = (Distance 0→a) + (Distance b→c)+ (Distance d→e). This distance is calculated using the formulas in Table 3-1 as follows.

$$\text{Distance } (0\rightarrow a) = \frac{\text{Velocity}^2}{2 \times \text{Acceleration}} = \frac{15{,}000^2}{2 \times 60{,}000} = 1{,}875 \text{ counts}$$

$$\text{Distance}(b\rightarrow c) = \left| \frac{(Vel_2)^2 - (Vel_1)^2}{2 * \text{Acceleration}} \right| = \left| \frac{(35{,}000)^2 - (15{,}000)^2}{60{,}000} \right| = 8{,}333 \text{ counts}$$

Since the Pmove is S-curve,

$$\text{Distance } (d\rightarrow e) = \frac{(Vel_2)^2}{\text{Acceleration}} = \frac{35{,}000^2}{60{,}000} = 20{,}417 \text{ counts}$$

So, the total distance required = 1,875 counts + 8,333 counts + 20,417 = 30,625 counts. If the total distance programmed is less than 30,625 counts, an error will result.

Multiple Cmoves

This example illustrates that any number of Cmoves can be programmed sequentially to achieve a high velocity in steps.

Example

Starting at axis position 0, program 3 successive Cmoves (absolute, s-curve). Program the Cmoves as follows:

Cmove 1 - 10,000 counts, velocity of 9,000 counts / sec, acceleration 30,000 counts / sec / sec
Cmove 2 - 30,000 counts, velocity of 21,000 counts / sec, acceleration 120,000 counts / sec / sec
Cmove 3 - 100,000 counts, velocity of 45,000 counts / sec, acceleration 120,000 counts / sec / sec

Block No.	Command	Operand 1 (Axis #)	Operand 2	Operand 3	Operand 4
	ACCEL	1,	30000		
	VELOC	1,	9000		
	CMOVE	1,	10000	ABS,	SCURVE
	ACCEL	1,	120000		
	VELOC	1,	21000		
	CMOVE	1,	30000	ABS,	SCURVE
	VELOC	1,	45000		
	CMOVE	1,	100000	ABS,	SCURVE

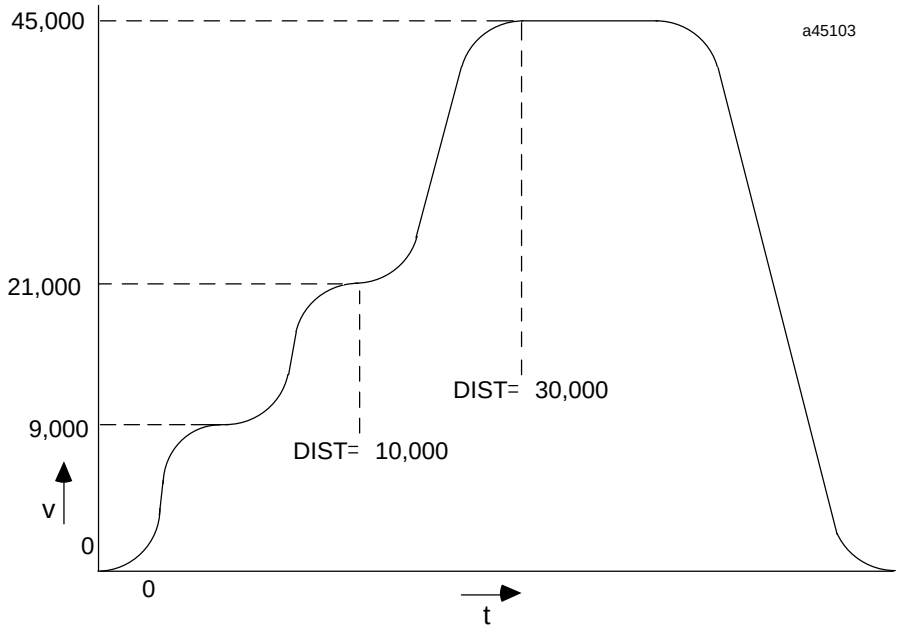


Figure 3-3. Multiple Cmoves Profile

Using the Jump Command

The figure below illustrates how the placement of the Jump command within a program can be used to produce different testing effect. Whenever the specified Jump condition tests "True", the present motion is aborted, and program control is immediately transferred to the new location. When Jump condition testing is active, any programmed Cmoves will be treated as single independent moves.

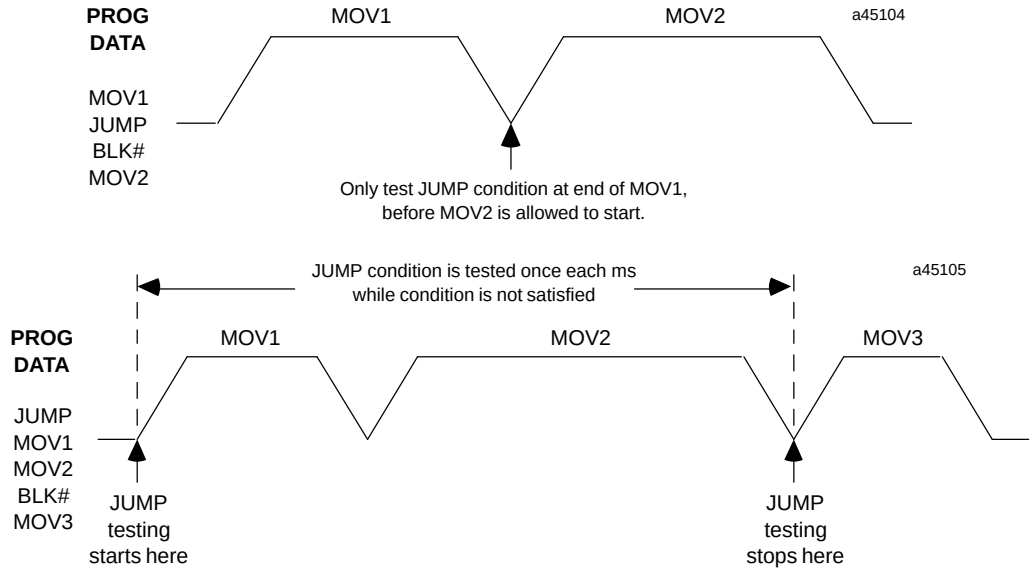


Figure 3-4. Jump Command Condition Testing

Appendix A

APM Motion Commands

This appendix describes each APM Motion Program command in detail. Both the *Motion Programmer* format and the *Program Zero Editor* format are presented for the commands when applicable.

Acceleration

Motion Programmer Format

Command Name	Operand 1	Operand 2
Acceleration (ACCEL)	Axis # (1, 2)	User Units / sec / sec (1 ... 134,217,727) or Parameter Number (P0 ... P0255) Default: 5000 User Units / sec / sec

Program Zero Editor Format

ACCEL	User Units / sec / sec (1 ... 134,217,727)
ACCEL-P	Parameter Number (P0 ... P255)

Comments

This command is used to specify the axis acceleration and deceleration rate for subsequent moves. Once encountered, the specified rate will remain in effect until overridden by a later *Acceleration* command.

Block

Motion Programmer Format

In the Motion Programmer, block numbers are entered in a separate field to the left of the command field. In multi-axis programs, a separate field is used to specify whether the block is a sync-block.

Program Zero Editor Format

BLOCK	Block Number (1 ... 65535)
BLOCK-SYNC	Block Number (1 ... 65535)

Comments:

Block numbers are used as jump destinations and to control jump testing in the APM. They identify the block the APM is currently executing and may be used by a PLC program to monitor motion program progress.

Call Subroutine

Motion Programmer Format

Command Name	Operand 1	Operand 2
CallSubroutine (CALL)	Axis # (1, 2)	SubroutineNumber (1-40) Default: none

Comments

This command is used to activate a specified subroutine as the command data source. The subroutine will remain the active data source until the end of the subroutine is encountered or until it calls another subroutine. Any subroutine may call any other subroutine with the following restriction: a subroutine that is active (started and not yet completed) cannot be called again until it has completed. Starting or ending a subroutine ends jump testing.

Cmove

Motion Programmer Format

Command Name	Operand 1	Operand 2	Operand 3	Operand 4
ContinuousMove (CMOVE)	Axis # (1, 2)	User Units (-8,388,608...+8,388,607) or ParameterNumber (P0 ... P255)	Positioning Mode (ABS, INCR)	Acceleration Mode (LINEAR, SCURVE)

Program Zero Editor Format

CMOVE-AL n	User Units (–8,388,608...+8,388,607)
CMOVE-AL-P n	Parameter Number (P0 ... P255)
CMOVE-AS n	User Units (–8,388,608...+8,388,607)
CMOVE-AS-P n	Parameter Number (P0 ... P255)
CMOVE-IL n	User Units (–8,388,608...+8,388,607)
CMOVE-IL-P n	Parameter Number (P0 ... P255)
CMOVE-IS n	User Units (–8,388,608...+8,388,607)
CMOVE-IS-P n	Parameter Number (P0 ... P255)

I = Incremental L = Linear P = Parameter
A= Absolute S = S-curve

Comments

Continuous Move - this command is used when it is not necessary for the axis to be within the configured *In Position Zone* before proceeding to the next command. If the *Cmove* is followed by another *Move* command, the velocity specified for the *Move* is attained at the programmed distance of the previous *Move.* If no previous acceleration or velocity has been specified in a motion program, the configured *Jog Acceleration* and / or *Jog Velocity* will be used.

Dwell

Motion Programmer Format

Command Name	Operand 1	Operand 2
Dwell (DWELL)	Axis # (1, 2)	Milliseconds (0 ... 65,535) or Parameter Number (P0 ... P255) Default: none

Program Zero Editor Format

DWELL Milliseconds (0 ... 65,535)
DWELL-P n Parameter Number (P0 ... P255)

Comments

This command causes motion to cease for a specified time period (in milliseconds) before proceeding to the next command. The *Program Active* status bit will remain ON during the *Dwell*, the *Moving* status bit will be OFF.

Jump

Motion Programmer Format

Command Name	Operand 1	Operand 2	Operand 3	Operand 4
Conditional Jump (JUMP)	Axis # (not used)	CTL # (CTL01-CTL12) Default: none	Block Number (1-65,535)	n/a

Comments

This command is used to jump to another location in the program or subroutine under certain specified states of the Faceplate control inputs (CTL 1-8) and %Q control outputs (CTL 9-12). The jump will occur when the condition tests "true" (logic 1). An unconditional *Jump* can also be selected using the Motion Programmer. Operand 3 contains the Block Number target for the jump. Operand 2 specifies the CTL Input or Output to test. The jump may be forward or backward in the program. The jump condition will be tested as soon as the move prior to the *Jump* command has completed.

Once the condition testing is allowed to start, the test will occur once each millisecond until the next Block Number is encountered. This will allow continued testing while a move or series of moves takes place, if the *Jump* command is located prior to the moves in the block. If a jump occurs during a move, the remainder of the move is aborted and the command at the target location is immediately effective. The range for searching for the jump target is limited to the bounds of the program or subroutine containing the *Jump* command.

Load Parameter

Motion Programmer Format

Command Name	Operand 1	Operand 2	Operand 3
Load Parameter (LOAD)	n/a	ParameterNumber (0 - 255)	Parameter Value (Depends on Parameter Type)

Program Zero Editor Format

LOAD-P01... P20 Parameter Value

Comments

This command initializes or changes the APM parameter value. The new value becomes effective immediately when encountered in the program. Parameters 246 - 255 are Special Purpose Function Parameters and data may be written into them by the APM.

Null

Motion Programmer Format

Command Name
Null (NULL)

Program Zero Editor Format

NULL

Comments

This command performs no function. It can be used at a jump destination when no command is desired.

Pmove

Motion Programmer Format

Command Name	Operand 1	Operand 2	Operand 3	Operand 4
Positioning Move (CMOVE)	Axis # (1, 2)	User Units (–8,388,608...+8,388,607) or Parameter Number (P0 ... P255)	Positioning Mode (ABS, INCR)	Acceleration Mode (LINEAR, SCURVE)

Program Zero Editor Format

PMOVE-AL n User Units (–8,388,608...+8,388,607)
PMOVE-AL-P n Parameter Number (P0 ... P255)
PMOVE-AS n User Units (–8,388,608...+8,388,607)
PMOVE-AS-P n Parameter Number (P0 ... P255)
PMOVE-IL n User Units (–8,388,608...+8,388,607)
PMOVE-IL-P n Parameter Number (P0 ... P255)
PMOVE-IS n User Units (–8,388,608...+8,388,607)
PMOVE-IS-P n Parameter Number (P0 ... P255)

I = Incremental S = S-curve P = Parameter
A= Absolute L = Linear

Comments

Positioning Move - this command is used when it is necessary for the axis to be within the configured *In Position Zone* before proceeding to the next command. If no previous acceleration or velocity has been specified in a motion program, the configured *Jog Acceleration* and / or *Jog Velocity* will be used.

Velocity

Motion Programmer Format

Command Name	Operand 1	Operand 2
Velocity (VELOC)	Axis # (1, 2)	User Units / sec (1 ... 8,388,607) or Parameter Number (P0 ... P255) Default: 5000

Program Zero Editor Format

VELOC	User Units / sec (1 - 8,388,607)
VELOC-P	Parameter Number (P0 ... P255)

Comments

This command specifies the velocity of axis motion.

Wait

Motion Programmer Format

Command Name	Operand 1	Operand 2
Wait (WAIT)	Axis # (1, 2)	CTL # (CTL01 - CTL12) Default: none

Program Zero Editor

WAIT	CTL # (CTL01 - CTL12)

Comments

This command synchronizes the start of axis motion with an external input or event reported in CTL 1-12. The start of motion is suspended until the bit being monitored is true.

Appendix B APM Discrete Commands (%Q

The following %Q outputs representing Discrete Commands are sent automatically to the APM from the CPU each PLC sweep. A command is executed simply by turning on the output bit of the desired command. This can be done directly through the PLC status tables or using PLC ladder logic. The actual addresses of the outputs depend on the %Q Reference Address you select when configuring the APM.

Table B-1. %Q Discrete Commands

Bit [1]	Description		Bit [1]	Description
00	Abort All Moves	Axis 1	16	CTL09 Output Control
01	Feedhold	Axis 1	17	CTL10 Output Control
02	Enable Drive	Axis 1	18	CTL11 Output Control
03	Find Home	Axis 1	19	CTL12 Output Control
04	Jog Plus	Axis 1	20	Execute Motion Program 0
05	Jog Minus	Axis 1	21	Execute Motion Program 1
06	Reset Strobe Flag	Axis 1	22	Execute Motion Program 2
07	reserved		23	Execute Motion Program 3
08 [2]	Abort All Moves	Axis 2	24	Execute Motion Program 4
09 [2]	Feedhold	Axis 2	25	Execute Motion Program 5
10 [2]	Enable Drive	Axis 2	26	Execute Motion Program 6
11 [2]	Find Home	Axis 2	27	Execute Motion Program 7
12 [2]	Jog Plus	Axis 2	28	Execute Motion Program 8
13 [2]	Jog Minus	Axis 2	29	Execute Motion Program 9
14 [2]	Reset Strobe Flag	Axis 2	30	Execute Motion Program 10
15	reserved		31	Clear Error

1 The bit numbers represent an offset to the starting address configured for %Q references. For example, if %Q0001 is configured as the starting address, then Bit 0 is %Q0001, Bit 01 is %Q0002, etc.

2 Not used for one - axis APMs.

The %Q Discrete Commands are described in detail on the following pages.

Abort All Moves

Command Bit: 0, 8

Command Action: Continuous

Required Preconditions:

- none

Resulting Conditions:

- *In Zone* status bit set
- *Program Active* status bit cleared
- All Move Pointers set to null condition to clear all move buffers
- *Wait* condition canceled

Comments: This command causes *any* motion in progress to halt at the active acceleration rate. Any pending programmed or immediate command is canceled and therefore not allowed to become effective. If a Jog is commanded, motion will not be permitted as long as *Abort All Moves* is set. If motion was in progress when the command was received, the *Moving* status bit will remain set and the *In Zone* status bit will remain cleared until the commanded velocity reaches zero and the *In Zone* condition is achieved.

Clear Error

Command Bit: 31

Command Action: One-shot

Required Preconditions:

- *Error* status bit is set

Resulting Conditions:

- *Error* status bit is cleared
- *Status Code* word is cleared

Comments: When an error condition is reported, this command is used to clear the *Error* status bit and its associated *Status Code* word. Error conditions that are still present (such as an *End of Travel* limit switch error) will not be cleared and must be cleared by some other corrective action .

CTL09- CTL12 Output Controls

Command Bits: 16-19

Command Action: One-shot and Continuous

Comments: These command bits control the CMOS digital outputs on FaceplateI/O Connector B. The bits may also be tested by the APM during execution of *Wait* or *Conditional Jump* commands.

Enable Drive

Command Bit: 2, 10

Command Action: Continuous

Required Preconditions:

- *Error* status bit cleared
- *Drive Enabled* status bit cleared

Resulting Conditions:

- *Drive Enabled* status bit set

Comments: If the *Error* and *Drive Enabled* status bits are cleared, this command will cause the Drive Enable relay contact to close (enabling the drive) and the *Drive Enabled* bit to be set; otherwise, it has no effect. When the *Drive Enabled* bit is set, the path generation and position control functions are enabled and program move commands may be executed. *Enable Drive* must be maintained ON to allow normal servo motion (except when using *Jog* commands).

The *Enable Drive* command will cause the following to occur:

1. The Positive and Negative Overtravel switches are checked and if either is open, an error condition is reported.
2. The *Commanded Position* status value is set equal to the *Actual Position* value.
3. Drive Enable relay is energized, closing the Drive Enable contact.
4. The *Drive Enabled* status bit is set.

Operation of Enable Drive with Jog Commands

Jog commands do not require that Enable Drive be set. A Jog command will initiate the Drive Enable timing sequence at the beginning of the Jog (unless the drive is already enabled). At the end of a Jog, a Disable Drive sequence will be initiated unless Enable Drive is set.

Execute Motion Program 0-10

Command Bit: 20 - 29

Command Action: One-shot

Required Preconditions:

- Error status bit cleared
- Drive Enabled bit set
- *Moving* cleared
- No other program conflicting active

Resulting Conditions:

- Program Motion halted / inhibited
- *In Zone* status bit is set
- *Moving* status bit is cleared

Comments: This command is used to select a stored program for immediate execution. Programs may be temporarily interrupted by a *Feedhold* command, but once the *Feedhold* is released, program execution will continue to the end. The *Execute Motion Program* command bits use one shot action. A command bit must transition from OFF to ON each time a program is to be executed.

Feedhold (On Transition)

Command Bit: 1, 9 ON

Command Action: Continuous

Required Preconditions:

- None

Resulting Conditions:

- Program Motion halted / inhibited
- *In Zone* status bit is set
- *Moving* status bit is cleared

Comments: This command causes any programmed motion in progress to halt at the active acceleration rate. Once the motion is stopped, the *Moving* status bit is cleared and the *In Zone* status bit is set when the In Zone condition is attained. *Jog* commands are allowed when in the Feedhold condition.

Note

This command has no effect on motion resulting from executing a *Find Home, Move At Velocity,* or a *Force D/A Output* command.

Feedhold (Off Transition)

Command Bit: 1, 9 OFF

Command Action: Continuous

Required Preconditions:

- *In Zone* status bit set

Resulting Conditions:

- Programmed motion allowed to start or resume.

Comments: This command causes any programmed motion interrupted by *Feedhold (On Transition)* to resume at the programmed acceleration and velocity rate. Additional program moves will then be processed and normal program execution will continue.

If Jogging occurred while Feedhold was ON, the APM still moves the axis to the programmed position at the programmed velocity. Then normal program execution will continue from there.

Find Home

Command Bit: 3, 11

Command Action: One-shot

Required Preconditions:

- *Drive Enabled* status bit set
- *Error* status bit cleared
- *Moving* status bit cleared
- *Program Active* status bit cleared

Resulting Conditions:

- *Position Valid* status bit is set

Comments: This command is used to establish the *Home Position* for systems with an incremental feedback device that also provides a marker pulse. A Home Limit Switch Input from the I/O connector roughly indicates the reference position for Home and the next marker encountered when traveling in the negative direction indicates the exact location. The configured *Home Offset* defines the location of *Home Position* as the offset distance from the Home Marker.

APM Configured in Home Switch Mode. The response to an OFF to ON transition of the *Find Home* command is as follows: (If initiated from a position on the positive side of the Home switch, starts with step 1, otherwise starts with step 3.)

Note

The *Abort All Moves* command will cause the cycle to abort without finding *Home Position* and *Position Valid* will not be set.

1. If the Home switch is open (indicating a present location on the positive side of the Home switch), the axis is moved in the negative direction at the configured *Find Home Velocity* with the configured *Jog Acceleration* rate until the Home switch closes.
2. The axis is then stopped at the configured *Jog Acceleration* rate.
3. The axis is accelerated in the positive direction at the *Jog Acceleration* rate and moved at the configured *Find Home Velocity* until the Home switch opens.
4. The axis is then stopped at the configured Jog Acceleration rate.
5. The axis is accelerated in the negative direction at the *Jog Acceleration* rate and moved at the configured *Final Home Velocity* until the Home switch closes and a marker pulse is sensed (in that order). The Home reference position is established at this point.
6. The axis is stopped at the configured *Jog Acceleration* rate.
7. The axis is moved to the *Home Position* (which is indicated by the *Home Offset* value from the reference marker position). An internal Set Position is then performed to insert the configured *Home Position* value.
8. The *Position Valid* status bit is set indicating that the *Find Home* cycle is complete.

APM Configured in Move+ or Move- mode. The axis moves at the configured *Jog Acceleration* and *Final Home Velocity* to the next marker pulse. The direction is positive in Move+ mode, negative in Move- mode. Then the above sequence begins at step 6.

Jog Minus

Command Bit: 5, 13

Command Action: Continuous

Required Preconditions:

- *In Zone* status bit set
- *Moving* status bit cleared
- *Jog Plus* command inactive
- *Feedhold* command active (only if executing program)

Resulting Conditions:

- *Drive Enabled* status bit is set
- *Moving* status bit is set
- *In Zone* status bit is cleared

Comments: When this command is ON, it moves the axis in the negative direction at the configured *Jog Acceleration* and *Jog Velocity*. The motion continues as long as the *Jog Minus* command is maintained and the configured *Negative End Of Travel* limit is not encountered or the **Negative End of Travel** limit switch is not activated (if enabled).

If *Jog Minus* and *Jog Plus* are both ON, no motion will occur. If the Negative Overtravel limit switch opens while *Jog Minus* is ON, an error will be reported. To jog away from the open limit switch, *Clear Error* and *Jog Plus* must both be maintained ON.

Jog Minus commands will be ignored when at the **Negative End of Travel** limit if Position Valid is set. If Position Valid is not set, Jogging will continue as long as it is commanded and will roll over at the configured **Low Count Limit**.

Jog Plus

Command Bit: 4, 12

Command Action: Continuous

Required Preconditions:

- *In Zone* status bit set
- *Moving* status bit cleared
- *Jog Minus* command inactive
- *Feedhold* command active (only if executing program)

Resulting Conditions:

- *In Zone* status bit is cleared
- *Moving* status bit is set
- *Drive Enabled* status bit is set

Comments: When this command is ON it moves the axis in the positive direction at the configured *Jog Acceleration* and *Jog Velocity*. The motion continues as long as the *Jog Plus* command is maintained and the configured *Positive End Of Travel* limit is not encountered or the **Positive End of Travel** limit switch is not activated (if enabled).

If *Jog Minus* and *Jog Plus* are both ON, no motion will occur. If the Positive Overtravel limit switch opens while *Jog Plus* is ON, an error will be reported. To jog away from the open limit switch, *Clear Error* and *Jog Minus* must both be maintained ON.

Jog Plus commands will be ignored when at the **Positive End of Travel** limit if Position Valid is set. If Position Valid is not set, Jogging will continue as long as it is commanded and will roll over at the configured **High Count Limit**.

Reset Strobe Flag

Command Bit: 6, 14

Command Action: Continuous

Required Preconditions:

Resulting Conditions:

- Strobe Flag status bit cleared

Comments: The *Strobe Flag* status bit informs the PLC that a Strobe Input has captured an axis position that is now stored in the *Strobe Position* status word. When the PLC acknowledges this data, it may use the *Reset Strobe Flag* command to clear the *Strobe Flag* status bit. Once the *Strobe Flag* bit is set, additional Strobe Inputs will not cause new data to be captured. The flag must be cleared before another Strobe Position will be captured. As long as the *Reset Strobe Flag* command is active, the *Strobe Flag* bit will be held in the cleared state. In this condition, the latest Strobe Input position is reflected in the *Strobe Position* status word, although the flag cannot be used by the PLC to indicate when new data is present.

Appendix C APM Immediate Commands (%AQ)

There are 6 %AQ Words sent automatically from the CPU to the APM each sweep. These words are used to transfer Immediate Commands automatically. Axis 1 uses Words 1-3, and Axis 2 uses Words 4-6.

The Immediate Commands are in a 6-Byte format as defined in the table below. When you wish to download an Immediate Command, enter the appropriate command and data in the 3 words reserved when you configured the system. The actual addresses of the Immediate Command Words depend on the starting address configured for the %AQ references.

Table C-1. Immediate Commands Using the 6-Byte Format

Word 3		Word 2		Word 1		**Immediate Command Definition**
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	
xx	xx	xx	xx	00	00h	**Null**
xx	xx	xx	RO%	00	20h	**Rate Override** RO% = 0 ...120%
xx	xx	xx	Incr.	00	21h	**Position Increment** Incr= 0 ... 255 user units
Velocity				00	22h	**Move At Velocity** Vel. = –8,388,608 ... + 8,388,607 user units / sec
Position				00	23h	**Set Position** Pos. = –8,388,608 ... + 8,388,607 user units
xx	xx	D / A Output		00	24h	**Force D/A Output** D / A Output = –32,000 ... + 32,000
Velocity				00	28h	**Jog Velocity** Vel. = 1 ... + 8,388,607 user units / sec
Acceration				00	29h	**Jog Acceleration** Acc. = 1 ... + 134,217,727 user units / sec
xx	xx	Time Constant		00	2Ah	**Position Loop Time Constant** Time Constant = 0, 5 ... 10000 ms
xx	xx	VFF%		00	2Bh	**Velocity Feedforward** VFF% = 0 ... 100%
xx	xx	Integr. TC		00	2Ch	**Integrator Time Constant** Integr. TC = 0, 10 ... 10,000
Parameter Data				Par #h	50h	**Load Parameter Immediate** Par # = 0 ... 255 Parameter Data = Range depends on parameter usage.

xx = don't care

The %AQ Immediate Commands are described in detail on the following pages.

Force D/A Output

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	D/A Output		00	24h	**Force D/A Output** D/A Output = –32,000 ... + 32,000 Default = none

Comments: This command forces the Velocity Command D/A outputs at the I/O connector to some constant output level. The *Force D/A Output* command sets the output to any value within the +/– 10 volt range.

A command of +32,000 will produce an output of +10 V and –32,000 will produce an output of –10 V.

Integrator Time Constant

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	Integr. TC		00	2Ch	**Integrator Time Constant** Integr. TC = 0, 10 ... 10,000 Default = 0

Comments: This commands sets the Integrator Time Constant for the position error integrator. The time constant is the number of milliseconds in which 63% of the *Position Error* will be removed. A time constant of 0 means the position error integrator is OFF. The time constant should be at least 10 times greater than the *Position Loop Time Constant*.

Jog Acceleration

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
Acceleration				00	29h	**Jog Acceleration** Acc. = 1 ... +134,217,727 Default = 10,000

Comments: This command sets the acceleration used for *Jog, Move at Velocity, Find Home, Abort,* and Jump stops.

Jog Velocity

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
Velocity				00	28h	**Jog Velocity** Vel. = 1 ... +8,388,607 Default = 1,000

Comments: This command sets the velocity used when the *Jog Plus* or *Jog Minus* %Q bit is set.

Load Parameter Immediate

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
Parameter Data				Par #h	50h	**Load Parameter Immediate** Par # = 0 ... 255 Parameter Data = Range depends on parameter usage. Default = None

Comments: This command immediately changes an APM parameter value. A command for each parameter change is required. Parameters 246-255 are special purpose parameters into which the APM itself may place data.

Move At Velocity

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
Velocity				00	22h	**Move At Velocity** Vel. = –8,388,608 ... + 8,388,607 user units / sec Default = none

Comments: This command is executed from the PLC to move the axis at a constant velocity. The configured *Jog Acceleration* rate and *Jog Acceleration Mode* (linear or S-curve) are used for *Move at Velocity* commands. *Commanded Position* and *Actual Position* will roll over at the configured *High* and *Low Count Limits* when reached during these moves.

Position Increment

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	xx	Incr	00	21h	**Position Increment** Incr = –128 ... +127 user units Default = none

Comments: This command offsets *Actual Position* by a small amount. The *Actual Position* is changed by the increment value. The command uses one-shot action; each increment value is inserted only once.

Position Loop Time Constant

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	Time Constant		00	2Ah	**Position Loop Time Constant** Time Constant = 0, 5 ... 10000 ms Default = Set by configuration

Comments: This command allows the servo position loop time constant to be changed from the PLC. If this command is not sent, then the initial position loop time constant configured with the Logicmaster 90 configuration software is used.

Rate Override

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	xx	RO%	00	20h	**Rate Override** RO% = 0 ...120% Default = none

Comments: This command immediately changes the % feedrate override value (frequently referred to as MFO value). This new value will become effective immediately when received by the APM. It is stored and will remain effective until overwritten by a different value.

If sent when the *Moving* status bit is set, the override change will occur when the *Commanded Velocity* is constant.

Set Position

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
Position				00	23h	**Set Position** Pos. = –8,388,608 ... + 8,388,607 user units Default = none

Required Preconditions:

- *In Zone* set
- *Moving* cleared

Comments: This command changes the axis position register value without moving the axis. The *Commanded Position* and *Actual Position* values will both be changed so that no motion command will be generated. The *Actual Position* will be set to the value designated and the *Commanded Position* will be set to the value + *Position Error.*

Velocity Feedforward

Command Format:

Word 3		Word 2		Word 1		
Byte 6	**Byte 5**	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Immediate Command Definition**
xx	xx	VFF%		00	2Bh	**Velocity Feedforward** VFF% = 0 ... 100% Default = none

Required Preconditions:

Resulting Conditions:

Comments: This command sets the *Velocity Feedforward* gain (percent). It is the percentage of Commanded Velocity that is added to the APM velocity command output. Increasing *Velocity Feedforward* causes the servo to operate with faster response and reduced position error. Optimum *Velocity Feedforward* values are 80-90 %. The “Vel at 10 V” value must be set correctly for proper operation of the *Velocity Feedforward* gain factor.

Appendix D

APM Status Bits (%1)

The following %I Status Bits are transferred automatically from the APM to the CPU each sweep. The actual addresses of the Status Bits depend on the %I Reference Address you select when configuring the APM. The Status Bits can be viewed in the Motion Programmer Status Screen.

Table D-1. %I Status Bits

Bit[1]	Description		Bit[1]	Description
00	AxisEnabled	Axis1	16	Front Panel Input CTL01 Switch Status
01	Position Valid	Axis1	17	Front Panel Input CTL02 Switch Status
02	DriveEnabled	Axis1	18	Front Panel Input CTL03 Switch Status
03	Program Active	Axis1	19	Front Panel Input CTL04 Switch Status
04	Moving	Axis1	20	Front Panel Input CTL05 Switch Status
05	In Zone	Axis1	21	Front Panel Input CTL06 Switch Status
06	Position Strobe	Axis1	22	Front Panel Input CTL07 Switch Status
07	In Error Limit	Axis1	23	Front Panel Input CTL08 Switch Status
08[2]	AxisEnabled	Axis2	24	reserved
09[2]	Position Valid	Axis2	25	reserved
10[2]	DriveEnabled	Axis2	26	reserved
11[2]	Program Active	Axis2	27	reserved
12[2]	Moving	Axis2	28	reserved
13[2]	In Zone	Axis2	29	reserved
14[2]	Position Strobe	Axis2	30	PLC Control Active
15[2]	In Error Limit	Axis2	31	Error

[1] The bit numbers represent an offset to the configured starting address for %I references. For example, if %I0001 is configured as the starting address, then Bit 0 is %I0001, Bit 01 is %I0002, etc.

[2] Not used in the one-axis APM.

The %I Status Bits are described in detail on the following pages.

Axis Enabled

Input Number: 0, 8

Default State: Set on power-up

Comments: The *Axis Enabled* status bit is On when the APM is ready to receive commands and control a servo. An error condition which stops the servo will turn *Axis Enabled* Off.

Drive Enabled

Input Number: 2, 10

Default State: Cleared on power-up

Comments: The *Drive Enabled* status bit indicates the state of the *Enable Drive* discrete command and the relay contact supplied by the APM. The ON state of the *Drive Enabled* status bit corresponds to the CLOSED state of the relay contact. *Drive Enabled* is cleared following power-up or an error condition which stops the servo.

While the *Drive Enabled* status bit is cleared, the Velocity Command output from the APM I/O connector will stay at zero, and the value of the *Commanded Position* status word will not change. The *Actual Position* status word, however, will continue to track any axis motion and therefore the value of the *Position Error* status word will change with axis movement.

Error

Input Number: 31

Default State: Set on power-up

Comments: This status bit is set when the APM detects any error. When set, the %AI *Status Code* word identifies the error condition. *Clear Error* is the only command that will clear the *Error* status bit and the associated *Status Code* word. If the condition causing the error is still present, the *Error* status bit will not be cleared.

Faceplate Input Status CTL 01-08

Input Number: 16-23

Default State: None

Comments: These inputs always indicate the state of the external input devices connected to the APM faceplate terminals CTL01-08. These inputs (as well as CTL09-CTL12 from the PLC %Q table) may be tested by the APM during the execution of *Wait* and *Conditional Jump* commands.

Several inputs may serve alternate purposes:

CTL01 - Axis 1 Strobe input
CTL03 - Axis 1 Home switch input
CTL05 - Axis 1 Overtravel (+) switch input
CTL06 - Axis 1 Overtravel (–) switch input
CTL02 - Axis 2 Strobe input
CTL04 - Axis 2 Home switch input
CTL07 - Axis 2 Overtravel (+) switch input
CTL08 - Axis 2 Overtravel (–) switch input

In Error Limit

Input Number: 5, 13

Default State: Cleared on Power-up and following a *Drive Enable* command

Comments: The *In Error Limit* status bit is set when the absolute value of the position error exceeds the configured *Position Error Limit* value. When the *In Error Limit* status bit is set, commanded velocity and commanded position are frozen to allow the axis to "catch up" to the commanded position.

The *In Error Limit* status bit could be set for any of the following reasons:

1. Configured value for *Position Error Limit* is too low for the configured value of *Position Loop Time Constant*.
2. The maximum servo drive velocity (produced by a 10 V Velocity Command) is lower than the programmed velocity.
3. The servo drive is incapable of following the programmed velocity and/or acceleration.
4. Feedback signals have been lost or corrupted.

Some applications may intentionally run in error limit to make use of its command velocity limiting feature, but unless it is expected, once a system has been "tuned" it should not occur for normal operation.

In Zone

Input Number: 5, 13

Default State: None

Comments: The *In Zone* status bit indicates that the position error is equal to or less than the configured *In Position Zone* value. This condition occurs at the end of each *Positioning Move* command or any time the axis commands are halted and the actual position has caught up to the commanded position (e.g. for *Dwell* and *Feedhold*).

Moving

Input Number: 7, 15

Default State: Cleared on power-up.

Comments: The *Moving* status bit is set when commanded velocity is non-zero, otherwise it is cleared. All *Positioning Move*, *Continuous Move*, *Jog*, and *Move at Velocity* commands will cause the *Moving* bit to be set. The *Force D/A Output* command will not set the *Moving* bit. *Moving* remains ON when MFO is 0%.

PLC Control Active

Input Number: 30

Default State: Set on power-up

Comments: Normally the *PLC Control Active* status bit is set, indicating that the %Q discrete commands or %AQ immediate commands from the PLC are controlling the APM.

When the Logicmaster 90 Motion Programmer requests control of the APM via its Control / Status screen, the *PLC Control Active* bit will be cleared. The APM will ignore the PLC %Q discrete and %AQ immediate commands and respond only to the %Q and %AQ commands from the Motion Programmer. If the Motion Programmer loses communications with the APM while the Control / Status screen is controlling the APM, then the APM will enter the Disable state. The *PLC Control Active* bit will remain cleared unless the Motion Programmer re-establishes communications and the user requests that control be switched back to the PLC.

Note

In this release of the Motion Programmer software, the Control / Status screen performs only Status functions. The *PLC Control Active* status bit is always 1.

Position Strobe

Input Number: 6, 14

Default State: Cleared.

Comments: The *Position Strobe* status bit indicates that the Strobe Input at the I/O connector has captured an axis position that is currently indicated by the *Strobe Position* %AI status word. The data will remain in the *Strobe Position* status word until the *Position Strobe* bit is cleared by the *Reset Strobe Flag* %Q bit. Once the *Position Strobe* bit is cleared, new data may be captured by another Strobe Input.

Position Valid

Input Number: 1, 9

Default State: Cleared on power-up.

Comments: The *Position Valid* status bit indicates that the value in the *Actual Position* status word has been initialized by a *Set Position* command or successful completion of the *Find Home* cycle.

Position Valid Cleared

The *Position Valid* status bit is cleared under the following conditions:

1. At power-up.
2. When position feedback synchronization is lost.

Position Valid Set

The *Position Valid* status bit is set by the following:

1. Successful completion of the Find Home cycle.
2. When a *Set Position* immediate command is received.

Program Active

Input Number: 3, 11

Default State: Cleared on power-up and following an *Abort All Moves* command

Comments: The *Program Active* status bit for each axis indicates that a Motion Program (0-10) is executing on that axis. On a two axis APM, executing a multi-axis program will set both *Program Active* bits.

Program Active Cleared

The *Program Active* status bit is cleared under the following conditions:

1. At power-up.
2. When a selected motion program has completed execution.
3. When an *Error* condition causes the axis to stop.
4. When an *Abort All Moves* command is executed.

Program Active Set

The *Program Active* status bit is set by the following:

1. Executing a valid motion program.

Appendix E

APM Status Words (%AI)

The following %AI Status Words are transferred automatically from the APM to the CPU each sweep. The actual addresses of the Status Words depend on the %AI Reference Address you select when configuring the APM. The contents of the Status Words can be viewed in the PLC %AI tables or in the Motion Programmer Status Screen.

Table E-1. %AI Status Words (1 Axis)

Word[1]	Description	
000	Status Code	
001	Command Block Number	Axis 1
002–003	Commanded Position	Axis 1
004–005	Actual Position	Axis 1
006–007	Strobe Position	Axis 1
008–009	Commanded Velocity	Axis 1
010–011	Actual Velocity	Axis 1
012–13	Position Error[2]	Axis 1
14	Analog Input Value[2]	

[2] The APM30 may be configured with Analog Input Value in word 012 with no Position Error reported for backward compatibility with earlier versions of the APM30.

Table E-2. %AI Status Words (2 Axis)

Word[1]	Description	
000	Status Code	
001	Command Block Number	Axis 1
002–003	Commanded Position	Axis 1
004–005	Actual Position	Axis 1
006–007	Strobe Position	Axis 1
008–009	Commanded Velocity	Axis 1
010–011	Actual Velocity	Axis 1
012–13	Position Error	Axis 1
014	Analog Input Value	
015	Command Block Number	Axis 2
016–017	Commanded Position	Axis 2
018–019	Actual Position	Axis 2
020–021	Strobe Position	Axis 2
022–023	Commanded Velocity	Axis 2
024–025	Actual Velocity	Axis 2
026–27	Position Error	Axis 2

[1] The word numbers represent an offset to the configured starting address for %AI references. For example, if %AI0001 is configured as the starting address, then Word 0 is %AI0001, Word 01 is %AI0002, etc.

The %AI Status Words are described in detail on the following pages.

Actual Position

Actual Position (user units) is a value maintained by the APM30 to represent the physical position of the axis. It is set to an initial value by the Set Position command or to Home Position by the Find Home cycle. It is updated by the motion of the feedback device.

If *Actual Position* is counted past either of its limits, it will "rollover" to the other limit and continue counting in the direction of the axis motion.

Actual Velocity

Actual Velocity (user units / sec) is a value maintained by the APM30 that is derived from the input counts of the feedback device. Therefore, it represents the instantaneous velocity of the axis movement.

Analog Input

Analog Input returns the digital value representing the voltage applied to the analog input terminal. +10 V is indicated by +32,000 and –10 V by –32,000.

Command Block Number

Command Block Number indicates the block number of the command that is presently being executed in the active Program or Subroutine. It changes at the start of each new block as the program commands are executed, and thus identifies the present operating location within the program.

Commanded Position

Commanded Position (user units) is where the axis is commanded to be at any instant in time. The difference between Commanded Position and Actual Position is a position error value which produces the Velocity Command to drive the axis. The rate at which the Command Position is incremented (or decremented) determines the velocity of axis motion.

Commanded Velocity

Commanded Velocity (user units) is a value generated by the APM30 that indicates the instantaneous velocity command that is producing axis motion. At the beginning of a move it will increase at the acceleration rate, and once the programmed velocity has been reached, it will stabilize at the programmed velocity value.

Position Error

Position Error (user units) is equal to Commanded Position – Actual Position.

Status Code

Status Code indicates the current operating status of the module. When the Error status bit is set, it contains the offending error code number.

For a list of APM30 error codes refer to Table E–3.

Strobe Position

Strobe Position (user units) contains the axis position when a Strobe Input occurs. When a Strobe Input occurs, the Position Strobe input is set to indicate to the PLC that new Strobe data is available in the Strobe Position status word. When acknowledged, the PLC then sets the Reset Strobe Flag output to clear the Position Strobe input.

Strobe Position will be maintained and will not be overwritten by additional Strobe Inputs until the Position Strobe input has been cleared. If the Reset Strobe Flag output is left in the On state (thus holding Position Strobe input flag in the cleared state), then each Strobe Input that occurs will cause the axis position to be captured in Strobe Position.

The Status Code Word

The *Status Code* word contains a code which describes the error indicated when the *Error* status bit is set. There are three categories of errors reported by the *Status Code.*

- Non–fatal programming errors
- Fatal programming errors that halt the servo
- Hardware errors (encoder out of sync, ”Run” switches off, loss of programmer comm. , etc.)

Error Codes are placed in the Status Code %AI word. The format for the Status Code word is:

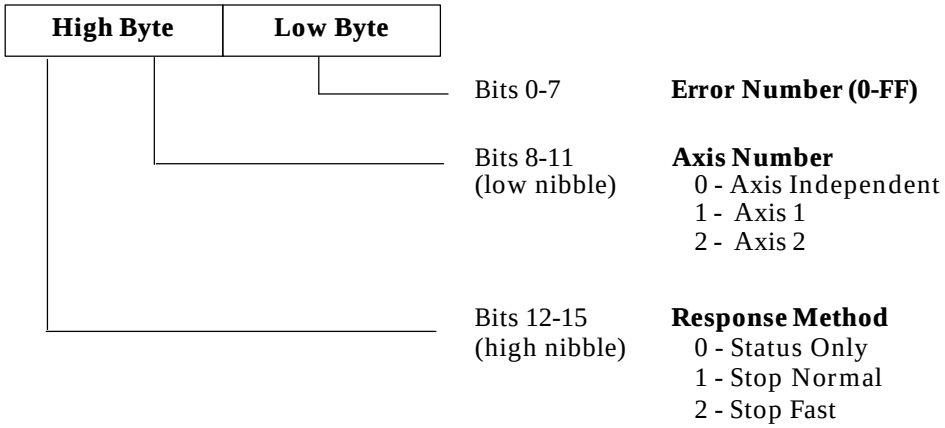


Figure E-1. Status Code Organization

Response Methods

1. **Status Only Errors:** Set the error flag and status code, but do not affect motion.
2. **Stop Normal Errors:** Perform an internal abort of any current motion. The *Drive Enabled* and *Axis Enabled %I* bits are turned OFF after the configured *Drive Disable Delay.*
3. **Stop Fast Errors:** Instantly abort all motion by setting the analog output voltage to zero. The *Drive Enabled* and *Axis Enabled %I* bits are turned off after the configured *Drive Disable Delay.*

Table E-3. Status Word Error Codes

Error Number (Hexadecimal)	Response	Description
0	None	No Error
		Configuration Errors
2	Status Only	Scaled data too big
3	Status Only	Home Position < Positive EOT, Positive EOT used
4	Status Only	Home Position < Negative EOT, Negative EOT used
		Configuration Parameter Errors
10	Status Only	Position Loop Time Constant too large
11	Status Only	Position Loop Time Constant too small
12	Status Only	Position Loop Time Constant computation overflow
1E	Status Only	Immediate command Jog Velocity out of range
1F	Status Only	Immediate command Jog Acceleration out of range
		Configuration Errors
20	Status Only	Program Acceleration overrange
21	Status Only	Program Acceleration is zero
22	Status Only	Scaled Velocity greater than 1 million cts / sec
23	Status Only	Program Velocity is zero, 1 count per second used
24	Stop Normal	Program Position too large
25	Stop Normal	Unconditional Jump Destination not found
26	Stop Normal	Jump Mask Error
27	Stop Normal	Wait Mask Error
28	Stop Normal	Parameter Position too large
29	Status Only	Dwell time greater than 60 seconds, 5 seconds used
		Position Increment Errors
2C	Status Only	Position Increment Overrange Error
		Find Home Errors
30	Status Only	Find Home while Drive Not Enabled error
31	Status Only	Find Home while Program Selected error
32	Status Only	Find Home while D / A forced error
33	Status Only	Find Home while Jog error
34	Status Only	Find Home while Move at Velocity error
36	Status Only	Find Home while Abort bit set error
		Move at Velocity Errors
39	Status Only	Move at Velocity while Drive Not Enabled error
3A	Status Only	Move at Velocity while Program Selected error
3B	Status Only	Move at Velocity while Home Cycle active error
3C	Status Only	Move at Velocity while Jog error
3D	Status Only	Move at Velocity while Abort All Moves bit is set error
3E	Status Only	Move at Velocity Data greater than 8,388,607 uu / sec error
3F	Status Only	Move at Velocity Data greater than 1 million cts / sec error
		Jog Errors
40	Status Only	Jog while Find Home error
41	Status Only	Jog while Move at Velocity error
42	Status Only	Jog while Force D / A error
43	Status Only	Jog while Program Selected and not Feedholding error
		Force D/A Errors
47	Status Only	Force D / A while Jog error
48	Status Only	Force D / A while Move at Velocity error
49	Status Only	Force D / A while Program Selected error

Table E-3. Status Word Error Codes (Continued)

Error Number (Hexadecimal)	Response	Description
		Set Position Errors
50	Status Only	Set Position while Program Selected error
51	Status Only	Set Position Data overrange error
52	Status Only	Set Position while not In Zone error
		End of Travel Limit Errors
56	Status Only	Commanded Position greater than Positive End of Travel
57	Status Only	Commanded Position less than Negative End of Travel
		Drive Disable Errors
5B	Stop Normal	Drive Disabled while Moving
5C	Stop Normal	Drive Disabled while Program Active
		Software Errors
5F	Status Only	Software Error (Call G.E. Field Service)
		Program and Subroutine Errors
61	Stop Normal	Subroutine not in list
62	Stop Normal	Call Error (subroutine already active)
63	Stop Normal	Subroutine End command found in Program
64	Stop Normal	Program End command found in Subroutine
		Program Execution Errors
70	Status Only	Request Program 0 with other programs active
71	Status Only	Too many programs requested in same PLC sweep
72	Status Only	Request Program 1–10 with multi–axis program active
73	Status Only	Request two programs on same sweep with program active
74	Status Only	Request two programs for same axis, lower number program executed
75	Status Only	Empty or Invalid Program requested
		Program Execution Conditions Errors
80	Status Only	Execute Program while Home Cycle active
81	Status Only	Execute Program while Jog
82	Status Only	Execute Program while Move at Velocity
83	Status Only	Execute Program while D / A Forced
84	Status Only	Execute Program while Program Selected
85	Status Only	Execute Program while Abort All Moves bit set
86	Status Only	Execute Program while Position Valid not set
87	Status Only	Execute Program while Drive Enabled not set
		Program Synchronous Block Errors
8C	Status Only	Sync Block Error during CMOVE
8D	Status Only	Sync Block Error during Jump
		EEPROM Errors
90	Status Only	Flash EEPROM memory programming failure
		Hardware Limit Switch Errors
A0	Stop Fast	Limit Switch (+) error
A1	Stop Fast	Limit Switch (–) error
		Hardware Errors
A8	Stop Fast	Out of Sync error
A9	Stop Fast	Encoder Loss of Quadrature error
AA	Stop Normal	Analog Input Failure
		Special Purpose Errors
E0	Status Only	Custom Loop Type Mismatch
EF	Status Only	2 axis firmware in 1 axis hardware

A

Abort All Moves Discrete Command (%Q), B-2
Acceleration Command, A-2
Actual Position Status Word, E-2
Actual Velocity Status Word, E-2
Analog Input Status Word, E-2
APM Immediate Commands (%AQ), C-1
APM Motion Commands, A-1
APM Status Bits (%I), D-1
APM Status Words (%AI), E-1
Axis Enabled Status Bit, D-2

B

Backing–Up All APM Files in a Folder, 2-32
Block Number Command, A-2
Block Numbers, Renumbering, 2-19

C

Calculating Move Distances, 3-1
Call Subroutine Command, A-3
Checking Program or Subroutine , 2-18
Clear Error Discrete Command (%Q), B-2
Clearing All APM Files in a Folder, 2-34
Clearing APM Memory, 2-48
Cmove Command, A-3
Cmove Followed by Pmove Example, 3-5
Command Block Number Status Word, E-2
Command Line, 2-9
Commanded Position Status Word, E-2
Commanded Velocity Status Word, E-2
Comment Lines, Adding, 2-17
Communications, Setting–Up, 2-40
Configuring Programs and Subroutines, 2-26
Connecting the Communications Cable, 2-11
Copying All APM Files in a Folder, 2-36
Copying APM Files, 2-28
Creating a Folder, 2-4
Creating a Folder, 2-30
Creating an APM File, 2-7 , 2-23
Creating and Editing Programs and Subroutines, 2-12
CTL09–CTL12 Output Controls Discrete Command s (%Q), B-3

D

Deleting a Folder, 2-31
Deleting an APM File, 2-25
Deleting Lines, 2-18
Discrete Commands (%Q), B-1
  Abort All Moves, B-2
  Clear Error, B-2
  CTL09–CTL12 Output Controls, B-3
  Enable Drive, B-3
  Feedhold (OFF), B-5
  Feedhold (ON), B-4
  Find Home, B-5
  Jog Minus, B-7
  Jog Plus, B-8
  Reset Strobe Flag, B-9
Display Mode, 2-13
DisplayingaProgram, 2-12
Drive Enabled Status Bit, D-2
Dwell Command, A-4

E

Edit Mode, 2-17
Editing a Program or Subroutine, 2-17
EEPROM Operations, 2-50
Enable Drive Discrete Command (%Q), B-3
Error Codes, E-3
Error Status Bit, D-2
Establishing Communications, 2-11
Examples, Motion Program, 3-1
  Cmove Followed by Pmove, 3-5

Jump Command, 3-8
Multiple Cmoves, 3-7
Single Pmove, 3-3

F

Faceplate Input Status CTL 01–08 Status Bit, D-3

Feedhold (OFF) Discrete Command (%Q), B-5

Feedhold (ON) Discrete Command (%Q), B-4

File Functions, 2-23

Files, 2-22
Copying, 2-28
Creating, 2-23
Deleting, 2-25
Loading and Storing, 2-43
Printing, 2-52
Selecting, 2-23
Verifying, 2-47

Files, Hierarchy, 2-22

Find Home Discrete Command (%Q), B-5

Folder Functions (F8), 2-29

Folders, 2-22
Backing–Up All APM Files, 2-32
Clearing All APM Files, 2-34
Copying All APM Files, 2-36
Creating, 2-30
Deleting, 2-31
Locking and Unlocking, 2-35
Restoring All APM Files, 2-33
Selecting, 2-30

Folders, Hierarchy, 2-22

Force D / A Output Immediate Command, C-2

Function Keys, Help and Special, 2-10

Functions Accessed from , 2-13

G

GoTo Function, 2-19

H

Help and Special Function Keys, 2-10

I

Immediate Commands (%AQ)
Force D / A Output, C-2
Integrator Time Constant, C-2
Jog Acceleration, C-3
Jog Velocity, C-3
Load Parameter, C-4
Move at Velocity, C-4
Position Increment, C-5
Position Loop Time Constant, C-5
Rate Override, C-6
Set Position, C-6
Velocity Feedforward, C-7

In Error Limit Status Bit, D-3

In Zone Status Bit, D-4

Insert Mode, 2-16

Inserting Lines in a Program or Subroutine, 2-16

Installing the Motion Programmer Software, 2-2

Integrator Time Constant Immediate Command, C-2

Introduction to APM Programming, 1-1

J

Jog Acceleration Immediate Command, C-3

Jog Minus Discrete Command (%Q), B-7

Jog Plus Discrete Command (%Q), B-8

Jog Velocity Immediate Command, C-3

Jump Command, A-4

Jump Command Example, 3-8

L

Load Parameter Command, A-5

Load Parameter Immediate Command, C-4

Loading a File from the APM to the Programmer, 2-43

Loading and Storing APM Files, 2-43

Locking Current Program Folder, 2-35

M

Making Changes to a Program or Subroutine, 2-17

Memory, Clearing, 2-48

Message Line, 2-9

Mode, Programmer, 2-39

Motion Commands
  Acceleration, A-2
  Block Number, A-2
  Call Subroutine, A-3
  Cmove, A-3
  Dwell, A-4
  Jump, A-4
  Load Parameter, A-5
  Pmove, A-5 , A-6
  Velocity, A-7
  Wait, A-7

Motion Program Examples, 3-1

Motion Programmer, 1-1 , 1-2 , 2-1
  Functions, 2-8
  Installing the Software, 2-2
  Instruction Format, 1-3
  Main Screen, 2-8
  Startup, 2-2

Move at Velocity Immediate Command, C-4

Move Distances, Calculating, 3-1

Moving Status Bit, D-4

Moving the Cursor, 2-15

Multiple Cmoves Example, 3-7

P

Pausing the Print Operation, 2-56

PLC Control Active Status Bit, D-4

Pmove Command, A-5 , A-6

Ports, 2-37
  Setting–Up, 2-37

Position Increment Immediate Command, C-5

Position Loop Time Constant Immediate Command, C-5

Position Strobe Status Bit, D-5

Position Valid Status Bit, D-5

Printer Parameters, Setting–Up, 2-52

Printer Port, Setup, 2-37

Printing, 2-52
  Pausing, 2-56
  Printer Parameters, 2-52
  Programs and Subroutines, 2-55
  Selecting Screen Print Destination, 2-54

Printing APM Files, 2-52

Program Active Status Bit, D-6

Program Edit Fields, 2-14

Program Zero Editor, 1-1 , 1-4

Programming, Overview, 1-1

Programs, Selecting and Configuring, 2-26

R

Rate Override Immediate Command, C-6

Renumbering Block Numbers, 2-19

Reset Strobe FlagDiscrete Command (%Q), B-9

Restoring All APM Files in a Folder, 2-33

S

Screen Print Destination, Selecting, 2-54

Selecting a Folder, 2-4

Selecting a Folder, 2-30

Selecting an APM File, 2-23

Selecting Programs and Subroutines, 2-26

Selecting the Screen Print Destination, 2-54

Set Position Immediate Command, C-6

Setting the Programmer Mode, 2-39

Setting–Up APM Communications, 2-40

Setting–Up Printer Parameters, 2-52

Setting–Up the Printer Serial Port, 2-37

Setting–Up the Programmer Ports, 2-37

Single Pmove Example, 3-3

Size Limitations, 2-20

Starting–Up the Motion Programmer, 2-2

Starting–Up the Motion Programmer Software, 2-4

Status Bits (%I)
  Axis Enabled, D-2
  Drive Enabled, D-2
  Error, D-2

Faceplate Input Status CTL 01–08, D-3
In Error Limit, D-3
In Zone, D-4
Moving, D-4
PLC Control Active, D-4
Position Strobe, D-5
Position Valid, D-5
Program Active, D-6

Status Code Status Word, E-2

Status Code Word, E-3

Status Fields, Additional, 2-13

Status Functions, 2-21

Status Line Definition, 2-9

Status Words (%AI)
Actual Position, E-2
Actual Velocity, E-2
Analog Input, E-2
Command Block Number, E-2
Commanded Position, E-2
Commanded Velocity, E-2
Status Code, E-2

Storing a File from the Programmer to the APM, 2-45

Subroutines, Selecting and Configuring, 2-26

Syntax, Checking Program or Subroutine, 2-18

U

Unlocking Current Program Folder, 2-35

V

Velocity Command, A-7

Velocity Feedforward Immediate Command, C-7

Verifying the APM with the Programmer, 2-47

W

Wait Command, A-7