



GE Fanuc Automation

PowerMotion™ Products

*Power Mate J
for Series 90™ -30*

User's Manual

GFK1256

August 1996

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 Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation 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.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	GEnet	PowerMotion	Series One
CIMPLICITY	Genius	ProLoop	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	PROMACRO	Series Three
CIMPLICITY 90-ADS	Helpmate	Series Five	VuMaster
CIMSTAR	Logicmaster	Series 90	Workmaster
Field Control	Modelmaster		

This manual describes the Power Mate J, which is a complete Series 90-30 integrated motion control system. The Power Mate J system consists of two Series 90-30 modules; a controller module and an interface module, which communicate over a high speed serial link. The controller provides the position control, motion program, and interface to the I/O. The interface module is the interface to the digital servo system providing digital torque and velocity control.

Content of this Manual

Chapter 1. Product Description: This chapter provides a description of the controller and interface modules, including module features, specifications, and I/O connections.

Chapter 2. Installation and Configuration: This chapter provides information on installing the Power Mate J in a Series 90-30 PLC system and configuring the modules with Logixmaster 90-30 software.

Chapter 3. Power Mate J to PLC Interface: This chapter describes the operation of the Power Mate J interface to the Series 90-30 PLC.

Chapter 4. Motion Control: This chapter provides practical information on Power Mate J motion control and includes several examples.

Appendix A. Error Word Status Codes: This appendix describes the errors reported by the status code word of the %AI Status Words.

Appendix B. Visual Status Error Codes: This appendix describes the status indications provided by the LEDs on the Power Mate J controller module, and the status information available from the 7-segment display on the Digital Servo Interface module.

Appendix C. Data Parameters From the PLC: This appendix describes how to use the COMM_REQ function block to load Data Parameter Memory from the PLC to the Power Mate J.

Appendix D. Serial Encoders: This appendix describes information needed to use serial encoders with the Power Mate J.

Related Publications

For more information, refer to these publications:

- *Series 90-30 Programmable Controller Installation Manual* - GFK-0356
- *Installation Requirements for Conformance to Standards* - GFK-1179
- *Logicmaster 90 Series 90-30/20/Micro Programming Software User's Manual* - GFK-0466
- *Hand-Held Programmer for Series 90-30/20/Micro User's Manual* - GFK-0402
- *Power Mate APM for Series 90-30 Standard Mode Users Manual* GFK-0840
- *Series 90 PLC Power Mate APM (PM-APM) Programmers Manual* - GFK-0664
- *a Series Amplifier Manuals (SVU):* GFZ-65192EN/01
- *Control Motor Amplifier a Series (SVM):* GFZ-65162E/02
- *a Series Servo Motor Manuals:* GFZ-65165E/01,GFZ-65150E/02,GFZ-65142E/02.
- *b Series Servo Product Specification Guide:* GFH-001

We Welcome Your Comments and Suggestions

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

Henry A. Konat
Senior Technical Writer

Chapter 1	Product Description	1-1
	The Power Mate J System	1-1
	System Overview	1-2
	Section 1: Description of the Power Mate J	1-3
	Controller Module (Power Mate APM)	1-3
	Serial COMM Connector	1-4
	Multidrop Connection	1-5
	I/O Connectors	1-6
	I/O Cable and Terminal Block	1-6
	I/O Specifications	1-7
	I/O Cable Connections (1-Axis)	1-10
	I/O Cable Connections (2-Axis)	1-11
	Functional Connection Diagrams	1-12
	Functional Connection Diagram, 1-Axis	1-12
	Functional Connection Diagram, 2-Axis	1-13
	Interface Module (DSI)	1-14
	Ports and Connectors	1-15
	JS1 Connector to Servo, Axis 1	1-15
	JS2 Connector to Servo, Axis 2	1-15
	Power Mate APM to DSI, High Speed Serial Communications	1-16
	High Speed Serial Cable Diagram	1-17
	Power Mate J System Specifications	1-18
	Section 2: Compatibility	1-19
Chapter 2	Installation and Configuration	2-1
	Section 1: Installing the Power Mate J Hardware	2-1
	Section 2: Configuring the Power Mate J Using Logicmaster 90-30	2-2
	Controller Configuration (Power Mate APM)	2-2
	Rack/Slot Configuration	2-2
	Module Configuration	2-2
	Setting the Configuration Parameters	2-3
	Digital Servo Interface Configuration	2-11
	Section 3: Program Zero	2-11
	Program Zero Programmer Instruction Format	2-12
	Program Zero Motion Command Descriptions	2-14
	Essential Configuration Parameters	2-15
	Important Configuration Considerations	2-17

Contents

Chapter 3	Power Mate J to PLC Interface	3-1
	Section 1: Power Mate APM Command and Status Data	3-2
	Controller Module %I Status Bits (PM-APM)	3-2
	Controller Module %AI Status Words (PM-APM)	3-5
	Controller Module %Q Discrete Commands (PM-APM)	3-8
	Controller Module %AQ Immediate Commands (PM-APM)	3-11
	Section 2: Interface Module %I Status Bits (DSI)	3-18
Chapter 4	Motion Control	4-1
	Position Feedback Types	4-1
	Non-Programmed Motion	4-2
	Power Mate J Home Cycle	4-2
	Jogging with the Power Mate J	4-3
	Move at Velocity Command	4-4
	ForceD/A Command	4-4
	Position Increment Commands	4-4
	Other Considerations	4-5
	Programmed Motion	4-6
	Prerequisites for Programmed Motion	4-7
	Conditions Which Stop a Motion Program	4-7
	Parameters for Programmed Moves	4-7
	Types of Programmed Move Commands	4-10
	Programmed Moves	4-11
	Dwell Command	4-15
	Wait Command	4-15
	Subroutines	4-16
	Block Numbers and Jumps	4-16
	Unconditional Jumps	4-16
	Conditional Jumps	4-17
	S-CURVE Jumps	4-23
	Other Programmed Motion Considerations	4-25
	Feedhold with the Power Mate J	4-26
	Feedrate Override	4-27
	Multiaxis Programming	4-29
	Parameters in the Power Mate J	4-30
Appendix A	Error Word Status Codes	A-1
	Error Code Format	A-2
	Response Methods	A-2

Appendix B	Visual Status Error Codes	B-1
	Section 1: Controller Module Status LEDs (PM-APM)	B-1
	LED Indicators	B-1
	Section 2: Interface Module Visual Status Display (DSI)	B-2
	AxisError Indicator	B-2
	MultipleAlarms	B-2
	Power-Up Considerations	B-2
	Clearing Alarms	B-2
	DSI Status Information	B-3
	Servo Unit Error Alarm Table	B-4
	Power Mate J Internal Alarms	B-5
Appendix C	Data Parameters From the PLC	C-1
Appendix D	Serial Encoders	D-1
	Serial Encoder Models	D-1
	Serial Encoder - First Time Use or Use After Loss of Encoder Battery Power	D-1
	Serial Encoder Modes	D-2
	Limitations on Total Travel for EOT Mode	D-2
	Incremental Encoder Mode Considerations	D-3
	Absolute Encoder Mode Considerations	D-3
	Absolute Encoder Mode - Position Initialization	D-3
	Absolute Encoder Mode - Power Mate J Power-Up	D-4
	Absolute Encoder Mode with Rotary Mode	D-4

Contents

Figure 1-1. Power Mate J	1-1
Figure 1-2. Power Mate J System Block Diagram	1-2
Figure 1-3. Controller Module LEDs and Connectors	1-3
Figure 1-4. Connecting Power Mate J Modules in a multidrop Configuration	1-5
Figure 1-5. Multidrop Cable for the Power Mate J	1-5
Figure 1-6. I/O Cable and Terminal Block	1-6
Figure 1-7. Enable Relay Output Circuitry	1-7
Figure 1-8. Analog Output Circuitry	1-7
Figure 1-9. Strobe Input Circuitry	1-8
Figure 1-10. 5 VDC Power Circuitry	1-8
Figure 1-11. General Purpose Input Circuitry	1-9
Figure 1-12. Connection Diagram for 1-Axis Power Mate J (Digital Mode) I/O Connector A	1-12
Figure 1-13. Connection Diagram for 2-Axis Power Mate J (Digital Mode) I/O Connector A	1-13
Figure 1-14. Interface Module Visual Status Display and Connectors	1-14
Figure 1-15. Controller Module to Interface Module Cable Diagram	1-17
Figure 2-1. Example of a Program Zero Editor (One Axis)	2-11
Figure 4-1. Sample Linear Motion	4-8
Figure 4-2. Sample S-Curve Motion	4-9
Figure 4-3. Combining PMOVEs and CMOVEs	4-11
Figure 4-4. Changing the Acceleration Mode During a Profile	4-12
Figure 4-5. Not Enough Distance to Reach Programmed Velocity	4-13
Figure 4-6. Hanging the Power Mate J When the Distance Runs Out	4-14
Figure 4-7. Dwell	4-15
Figure 4-8. Unconditional Jump	4-16
Figure 4-9. Normal Stop Before JUMP	4-20
Figure 4-10. JUMP Without Stopping	4-21
Figure 4-11. Jump Stop	4-22
Figure 4-12. Jump Followed by PMOVE	4-22
Figure 4-13. Jumping After the Midpoint of Acceleration or Deceleration	4-23
Figure 4-14. Jumping before the Midpoint of Acceleration or Deceleration	4-24
Figure 4-15. Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating	4-25
Figure 4-16. Maximum Acceleration Time	4-26
Figure 4-17. Feedhold	4-27
Figure 4-18. Feedrate Override	4-28
Figure 4-19. Multiaxis Programming	4-29
Figure A-1. Status Code Organization	A-2

Table 1-1. Pin Definitions of the Serial COMM Port Connector	1-4
Table 1-2. Cable Connections to Faceplate I/O Connector A 1-Axis Power Mate J	1-10
Table 1-3. Cable Connections to Faceplate I/O Connector A 2-Axis Power Mate J	1-11
Table 1-4. JS1 Connector to Servo Axis	1-15
Table 1-5. JS2 Connector to Servo Axis	1-15
Table 1-6. High Speed Serial Communications, Power Mate APM to DSI JD35 Connector	1-16
Table 1-7. Power Mate J Specifications	1-18
Table 2-1. Module Configuration Data	2-3
Table 2-2. Serial Communications Port Configuration Data	2-6
Table 2-3. Axis Configuration Data	2-7
Table 2-4. Program Zero Motion Program Commands	2-13
Table 3-1. %I Status Bits (1-Axis)	3-2
Table 3-2. %I Status Bits (2-Axis)	3-3
Table 3-3. %AI Status Words (1-Axis)	3-5
Table 3-4. %AI Status Words (2-Axis)	3-5
Table 3-5. %Q Discrete Commands (1-Axis)	3-8
Table 3-6. %Q Discrete Commands (2-Axis)	3-9
Table 3-7. Immediate Commands Using the 6-Byte Format	3-12
Table 3-8. DSI Status Bits	3-18
Table A-1. Status Word Error Codes	A-3
Table A-1. Status Word Error Codes (Continued)	A-4
Table A-1. Status Word Error Codes (Continued)	A-5
Table B-1. Power Mate J Interface Module Internal Status Alarms	B-3
Table B-2. Power Mate J Servo Status Alarms	B-4
Table B-3. Status Alarms	B-5

Chapter 1

Product Description

The information in this chapter is intended to present a description of the Power Mate J motion control system, provide module specifications, and indicate system functionality by way of a block diagram. Additionally, I/O connections and diagrams are provided.

The Power Mate J System

The Power Mate J (PM-J) for Series 90-30 is a complete integrated motion control system that supports *Standard mode* operation. Standard mode is the normal controller module motion control loop, which provides a velocity command proportional to position error, with optional Velocity Feed forward. The Power Mate J consists of two PowerMotion modules: The controller module (Power Mate APM) and the interface module (Digital Servo Interface). The controller provides the position control, motion program and interface to the I/O. The interface module provides the interface to the digital servo system including digital torque and velocity control. The two modules communicating together across a high speed serial link form a complete Power Mate J, digital motion controller system.

a45635

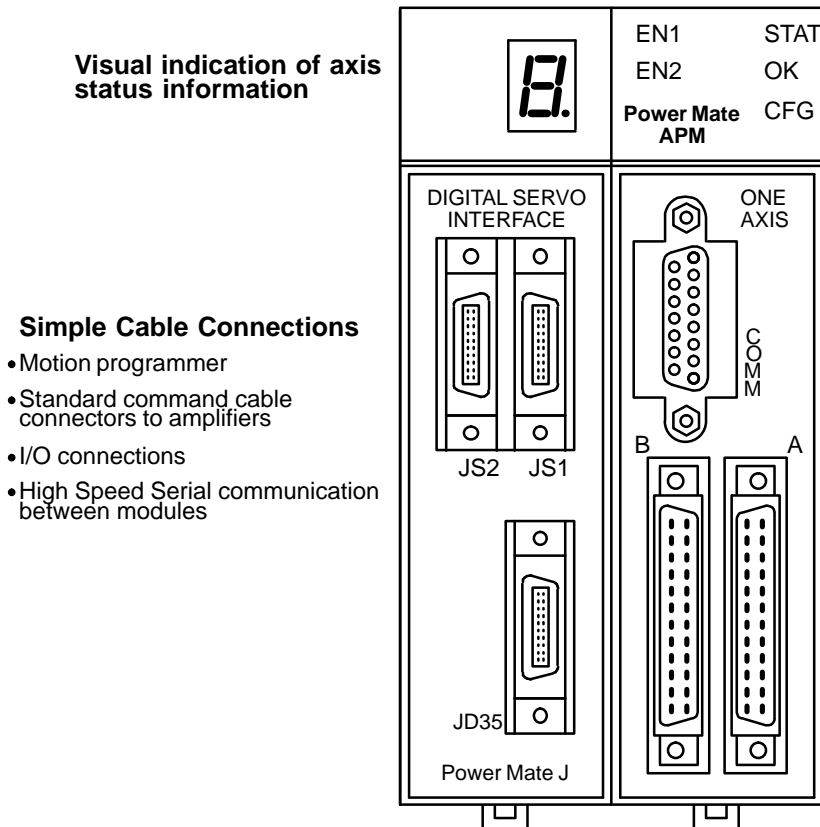


Figure 1-1. Power Mate J

System Overview

The Power Mate J for Series 90-30 allows a Series 90-30 PLC to interface to a Series or b Series Servo Motor Amplifiers and servo motors. a Series and b Series digital servo motors are connected by a simple cable interface to the motor amplifier and to the Power Mate J. The Power Mate J installs easily into the Series 90-30 CPU baseplate or any expansion baseplate and will draw all needed power from the PLC backplane power supply.

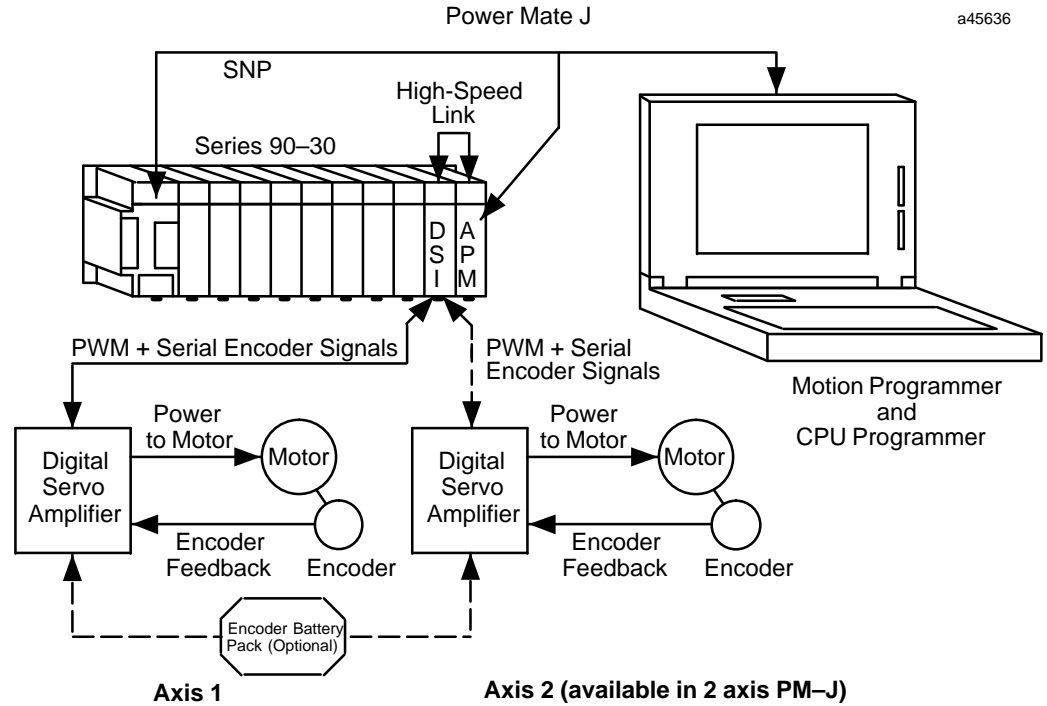


Figure 1-2. Power Mate J System Block Diagram

A Power Mate J system is available as either a one or two axis motion control system as described below.

Description	
Power Mate J	One Axis
<i>Includes:</i>	
Controller Module (one axis module)	
Interface Module	
Eight inch Serial Interface Cable (Used to connect controller to interface module)	
Power Mate J	Two Axis
<i>Includes:</i>	
Controller Module (two axis module)	
Interface Module	
Eight inch Serial Interface Cable (Used to connect controller to interface module)	

Section 1: Description of the Power Mate J

This section describes the Power Mate J, which consists of a controller module and an interface module.

Controller Module (Power Mate APM)

The following pages describe the user interfaces of the controller module in Standard control mode when Digital servo command and Digital feedback are selected.

The release 2.5 (IC693APU301K or IC693APU302K) or later of the controller module can be configured in DIGITAL mode to operate with the interface module. The figure below illustrates module features when used in the DIGITAL configuration.

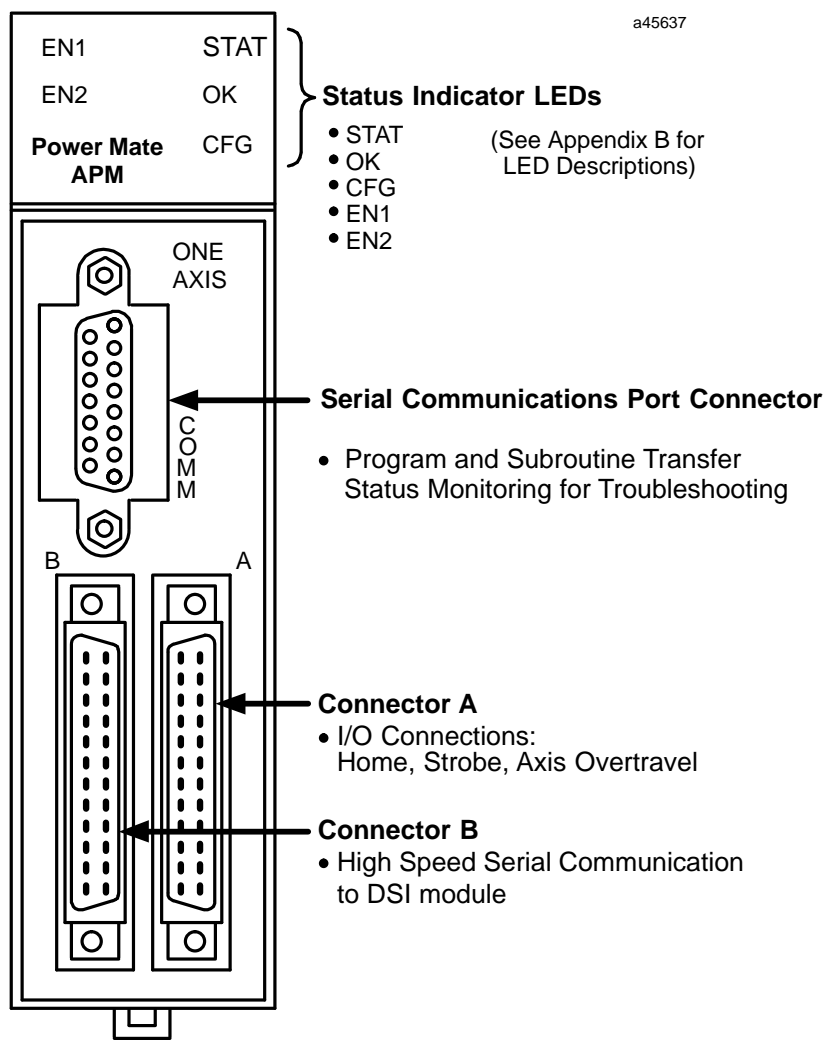


Figure 1-3. Controller Module LEDs and Connectors

Serial COMM Connector

The controller module's front Panel contains a single 15-pin, female, D-connector for serial communications. It is used to connect a computer running the Motion Programmer software to the controller. The serial COMM port uses the GE Fanuc SNP protocol and is RS-485 compatible. The baud rate is selectable from 300 to 19,200 baud.

The connection between the programming computer and the controller is typically made from the RS-232 port of the computer through an RS-232 to RS-485/RS-422 converter to the Serial Communications Connector. A Mini-converter Kit (IC690ACC901) which includes a converter and 10 feet (3 meters) cable is available for this purpose.

The port is configured using the Logicmaster Configuration Package or the Series 90-30 PLC Hand Held Programmer.

The pin definitions for the serial COMM port are listed below.

Table 1-1. Pin Definitions of the Serial COMM Port Connector

Pin	Signal	Description	Pin	Signal	Description
1	Shield	Cable Shield	9	RT	120 Ohm Termination for RXD (A)
2	DCD (A)	Carrier Detect	10	RD (A)	Receive Data
3	DCD (B)	Carrier Detect	11	RD (B)	Receive Data
4	ATCH	HHP Attach Input	12	SD (A)	Transmit Data
5	+5 V	+5 V HHP/RS-442 to RS-232 Converter Power	13	SD (B)	Transmit Data
6	RTS (A)	Ready to Send	14	RTS (B)	Ready to Send
7	0 V	0 V Signal Common	15	CTS (A)	Clear to Send
8	CTS (B)	Clear to Send			

Multidrop Connection

Power Mate J modules can be connected in multidrop fashion. A sample configuration is shown in the figure below. One cable is necessary for each Power Mate J in the system.

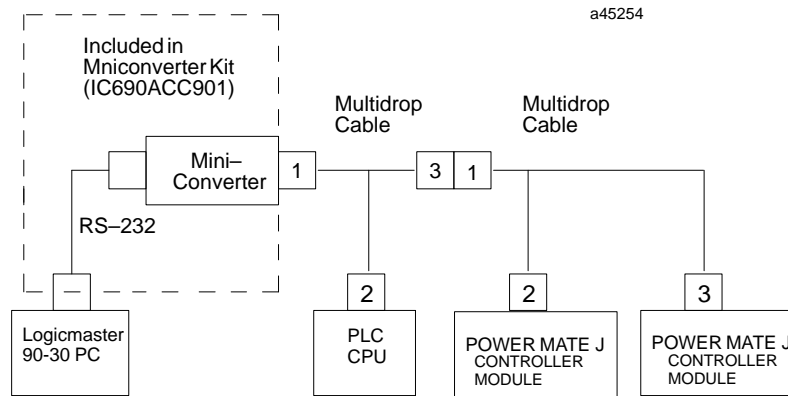


Figure 1-4. Connecting Power Mate J Modules in a multidrop Configuration

The multidrop cable should be made according to the following diagram.

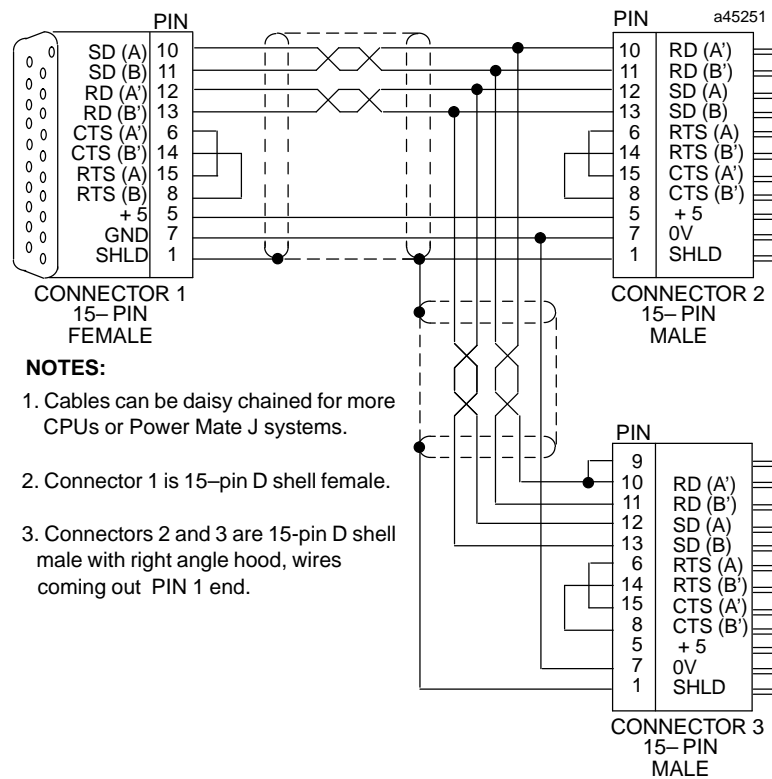


Figure 1-5. Multidrop Cable for the Power Mate J

I/O Connectors

The controller module's front panel contains two 24-pin, male high-density connectors. Connector A contains connections for I/O. Connector B contains connections for the high speed serial interface to the Power Mate J Digital Servo Interface Module.

I/O Cable and Terminal Block

High-density connectors are used on the controller to permit a large number of I/O connections within the physical size limitations of the controller. To facilitate wiring to the drive and machine, the high-density I/O connector is typically connected by a short cable to a terminal block. A three meter I/O cable (IC693CBL311) is available with the Power Mate J system.

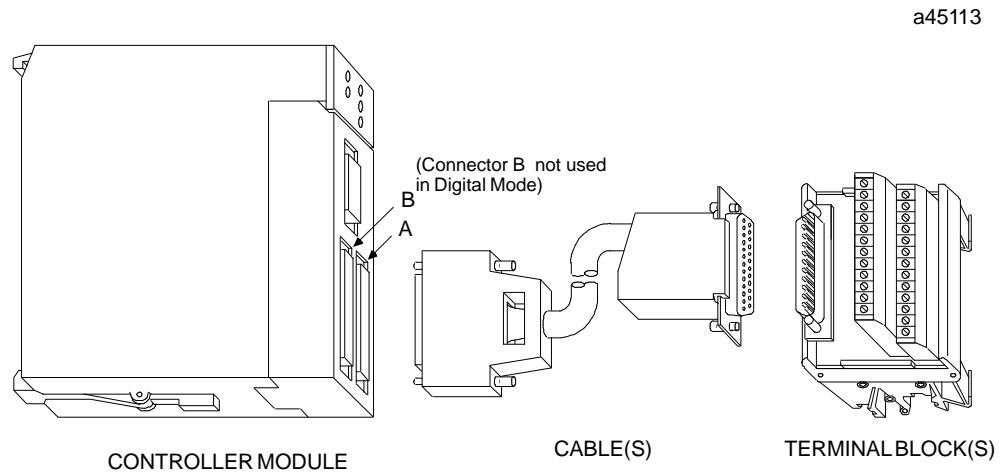


Figure 1-6. I/O Cable and Terminal Block

I/O Specifications

The specifications and circuitry for the controller's I/O connections are provided below.

Enable Relay Output

Normally-open DC solid state relay contact; Contacts rated at 30 volts DC, 100 mA. Resistive load only. The off-state leakage current is 10 μ A maximum.

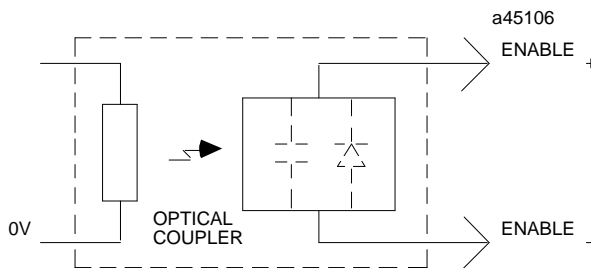


Figure 1-7. Enable Relay Output Circuitry

Analog Output

Output of D/A converter with the following characteristics (1 axis Power Mate J only).

- Resolution: 13 bits including sign
- Linearity: .02% of full scale output
- Offset Voltage: $\pm 500 \mu$ V maximum
- Maximum Output: ± 10.0 V , ± 0.3 V
- Minimum Load Resistance: 2000 Ω
- Maximum allowable Voltage Between Analog Common and Ground: ± 1.0 V

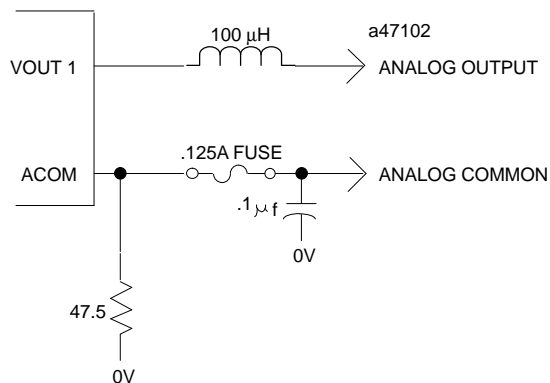


Figure 1-8. Analog Output Circuitry

Strobe Inputs CTL01, CTL02

- Input Type: 5V differential or single ended
- Input Impedance: 4000 Ohms (common mode)
- Input Threshold: Single ended: +1.5 V nominal ($\pm 0.4V$), Differential: +0.5 V nominal ($\pm 0.4V$)
- Input Common Mode Range: ± 15 V
- Single Ended Input Voltage: +15 V maximum
- Minimum 3 microsecond pulse width is required
- A pulse on the strobe input causes the latest value of Actual Position to be reported in the **Strobe Position** portion of %AI data.
- Position Capture Delay: 0 to 1 millisecond (IC693APU301)
0 to 2 millisecond (IC693APU302)

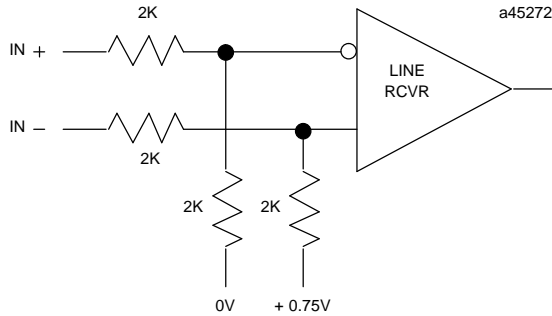


Figure 1-9. Strobe Input Circuitry

5 VDC Power

- 5 VDC Power: Non-isolated current limited +5 V supply is available at the controller module's front panel I/O connector A for general use. Maximum load must be limited to 500 mA @ 40°C (104°F), 300 mA @ 55°C (131°F)

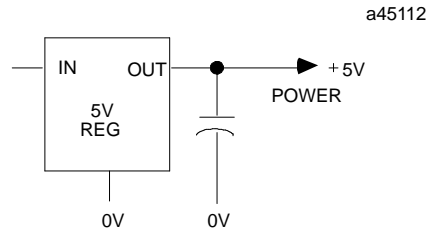


Figure 1-10. 5 VDC Power Circuitry

General Purpose 24 VDC Digital Inputs

I/O Connector A:

- CTL03
- CTL05
- CTL06

Optically Isolated with the following specifications:

- Optically isolated DC source/sink (bi-directional input opto coupler)
- Input ON threshold: 18.0 V to 30.0 V
- Input OFF threshold: 0 V to 4.0 V
- Input resistance: 5000 $\Omega \pm 10\%$
- Input filtering: 5 milliseconds nominal
- Isolation voltage: 1500 V peak transient

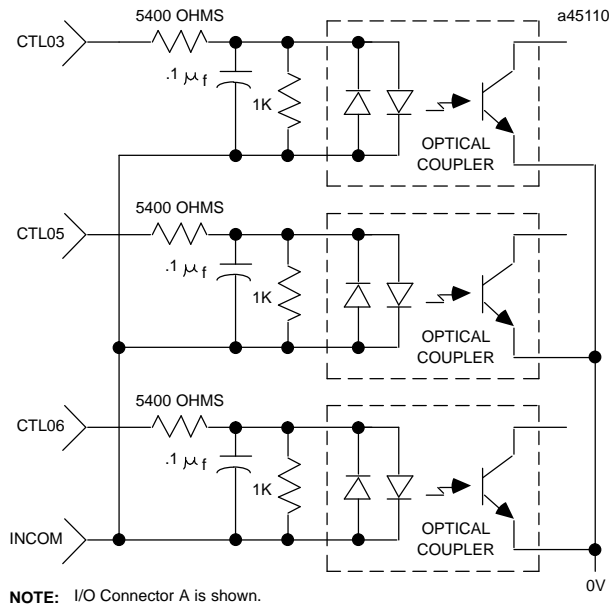


Figure 1-11. General Purpose Input Circuitry

I/O Cable Connections (1-Axis)

The following table lists the I/O cable connections for connector A on the 1-Axis Power Mate J when the controller is configured for *Digital Mode*.

Table 1-2. Cable Connections to Faceplate I/O Connector A 1-Axis Power Mate J

I/O Module Connector A Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Reserved
A2	11	Reserved
B2	23	CTL02 (+) Input
A3	10	CTL02 (-) Input
B3	22	Strobe 1 (+) / CTL01 (+) Input
A4	9	Strobe 1 (-) / CTL01 (-) Input
B4	15	Common for CTL03, 05, 06 Inputs
A5	2	Overtravel 1 (- direction) / CTL06 Input
B5	14	Overtravel 1 (+ direction) / CTL05 Input
A6	1	Home 1 / CTL03 Input
B6	16	Enable Relay (-) Output
A7	3	Enable Relay (+) Output
B7	17	Analog Output Common
A8	4	Analog Output
B8	21	0 V
A9	8	+5 VDC Power
B9	20	Reserved
A10	7	Reserved
B10	19	Reserved
A11	6	Reserved
B11	18	Reserved
A12	5	Reserved
B12	13	Cable Shield
	25	no connection

I/O Cable Connections (2-Axis)

The following table lists the I/O cable connections for connector A on the 2-Axis Power Mate J when the controller is configured for *Digital Mode*.

Table 1-3. Cable Connections to Faceplate I/O Connector A 2-Axis Power Mate J

I/O Module Connector A Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Reserved
A2	11	Reserved
B2	23	Strobe 2 (+) / CTL02 (+) Input
A3	10	Strobe 2 (-) / CTL02 (-) Input
B3	22	Strobe 1 (+) / CTL01 (+) Input
A4	9	Strobe 1 (-) / CTL01 (-) Input
B4	15	Common for CTL03, 05, 06 Inputs
A5	2	Overtravel 1 (- direction) / CTL06 Input
B5	14	Overtravel 1 (+ direction) / Home 2 / CTL05 Input
A6	1	Home 1 / CTL03 Input
B6	16	Enable Relay (-) Output
A7	3	Enable Relay (+) Output
B7	17	Reserved
A8	4	Reserved
B8	21	0 V
A9	8	+5 VDC Power
B9	20	Reserved
A10	7	Reserved
B10	19	Reserved
A11	6	Reserved
B11	18	Reserved
A12	5	Reserved
B12	13	CableShield
	25	no connection

Functional Connection Diagrams

The Figures below illustrate how the 1- and 2-Axis Power Mate J I/O is connected to a drive and a machine in a typical application. Shielded cable should be used as indicated.

Functional Connection Diagram, 1-Axis

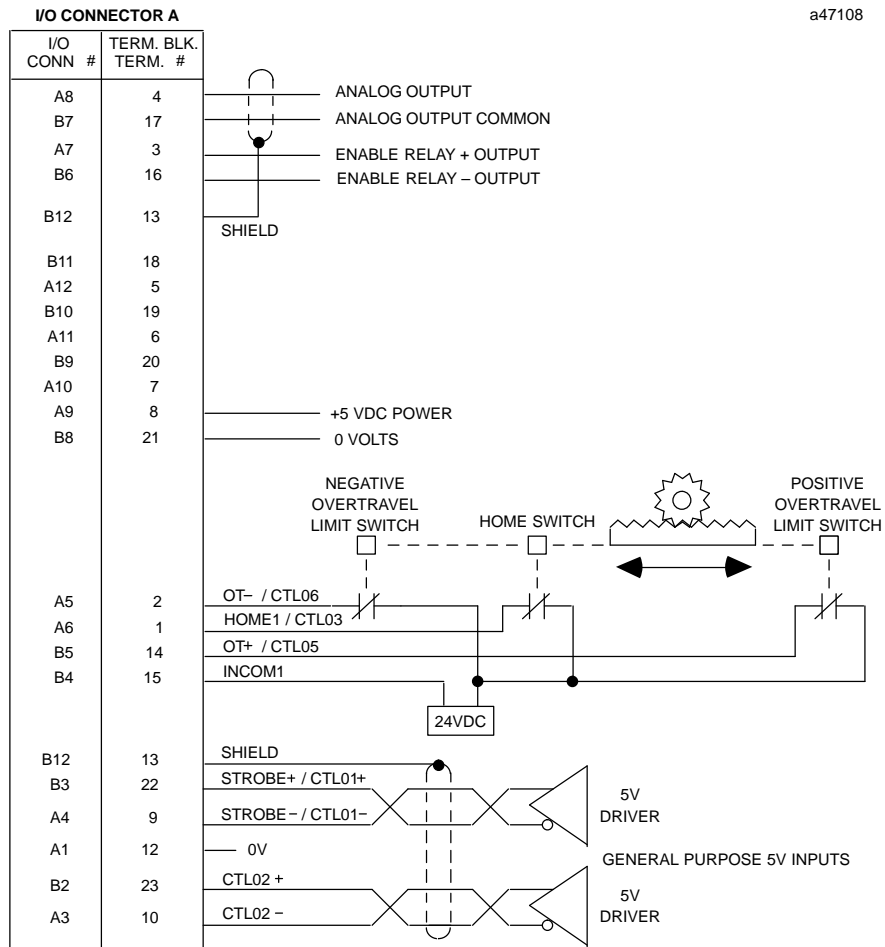


Figure 1-12. Connection Diagram for 1-Axis Power Mate J (Digital Mode) I/O Connector A

Functional Connection Diagram, 2-Axis

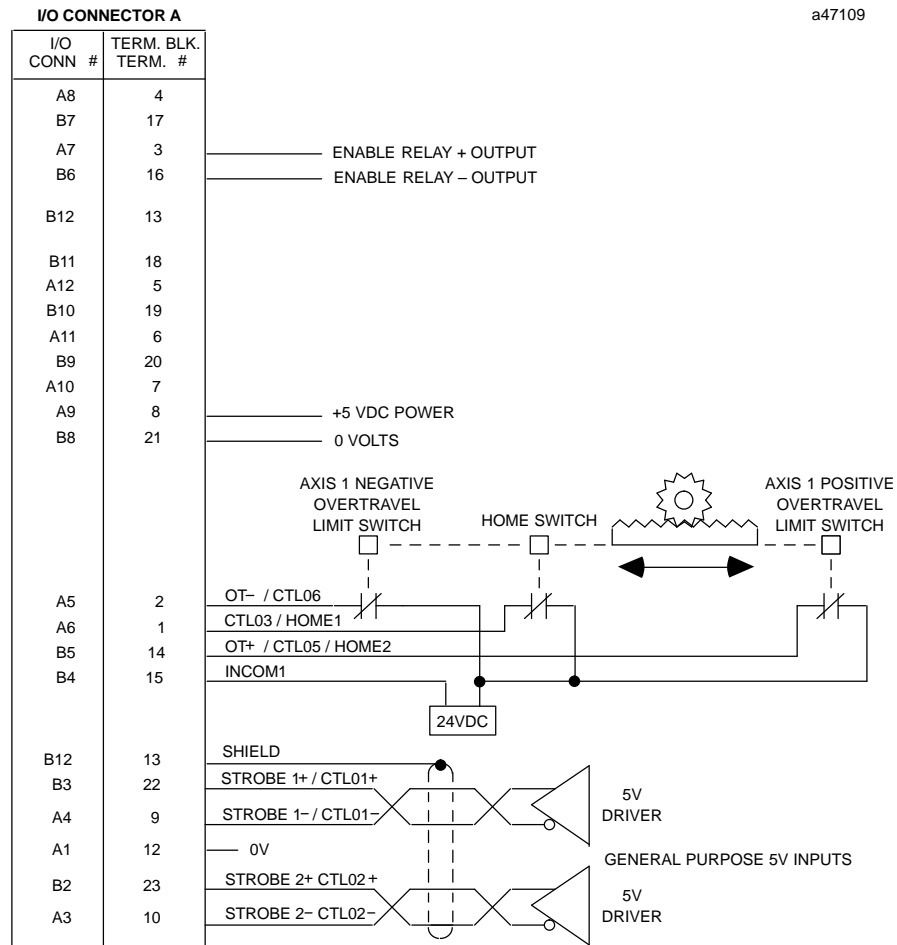


Figure 1-13. Connection Diagram for 2-Axis Power Mate J (Digital Mode) I/O Connector A

Note

If the Axis 1 Overtravel limit Switches are disabled in the Configuration Software, then CTL05 is available for use as the Axis 2 Home Switch. Overtravel Limit Switches are not available for Axis 2 when the controller is used with the interface module.

Interface Module (DSI)

The interface module is a standard PowerMotion single slot module. The interface module allows the motion controller to interface to and directly control *a* Series or *b* Series Servo motors and amplifiers. Absolute position feedback is supported with the addition of the optional battery pack to the digital servo amplifier unit.

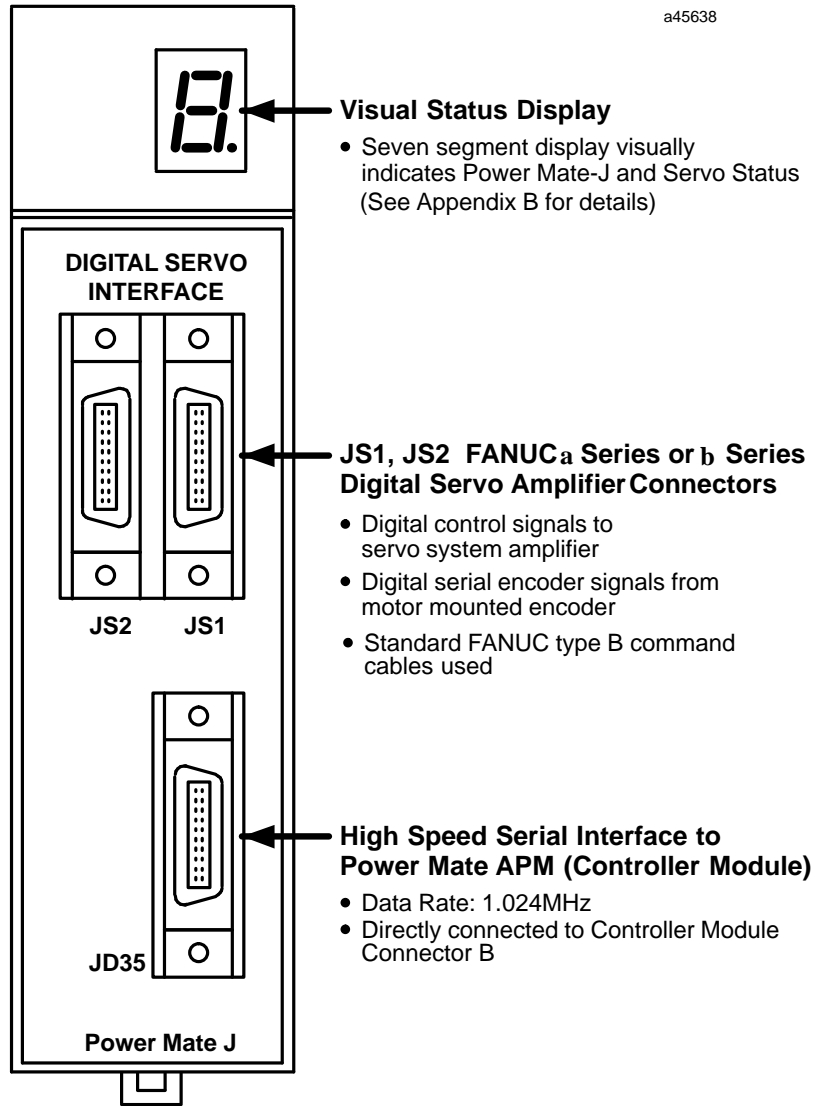


Figure 1-14. Interface Module Visual Status Display and Connectors

Ports and Connectors

The DSI has three connectors: JS1, JS2 and JD35. All connectors are Honda Tsushin PCR20 Male connectors. Premade 2 meter cables (44C742196-004) or connector kits may be ordered with your system.

The Power Mate APM to DSI cable (44A738935-001) is shipped with your system.

JS1 Connector to Servo, Axis 1

Table 1-4. JS1 Connector to Servo Axis

Pin Number	Description	Pin Number	Description
01	IR1	11	IS1
02	GDR1	12	GDS1
03	*PWMA1	13	*ENBL1
04	0V	14	0V
05	*PWMC1	15	PD1
06	0V	16	*PD1
07	*PWME1	17	PRQ1
08	0V	18	*PRQ1
09	*DRDY1	19	0V
10	*MCON1	20	0V

JS2 Connector to Servo, Axis 2

Table 1-5. JS2 Connector to Servo Axis

Pin Number	Description	Pin Number	Description
01	IR2	11	IS2
02	GDR2	12	GDS2
03	*PWMA2	13	*ENBL2
04	0V	14	0V
05	*PWMC2	15	PD2
06	0V	16	*PD2
07	*PWME2	17	PRQ2
08	0V	18	*PRQ2
09	*DRDY2	19	0V
10	*MCON2	20	0V

Power Mate APM to DSI, High Speed Serial Communications

Table 1-6. High Speed Serial Communications, Power Mate APM to DSI JD35 Connector

Pin Number	Description	Pin Number	Description
01		11	
02		12	
03	SDCLK	13	*SDCLK
04	0V	14	0V
05		15	SDIN
06	0V	16	*SDIN
07		17	SDOUT
08	0V	18	*SDOUT
09		19	0V
10		20	0V

Caution

The connector contains three RS-422 signal pairs, SDIN - *SDIN, SDOUT - *SDOUT, and SDCLK - *SDCLK. These must be connected as twisted pairs.

High Speed Serial Cable Diagram

The following figure is a High Speed Serial cable diagram for the cable that connects the Power Mate APM module to the DSI.

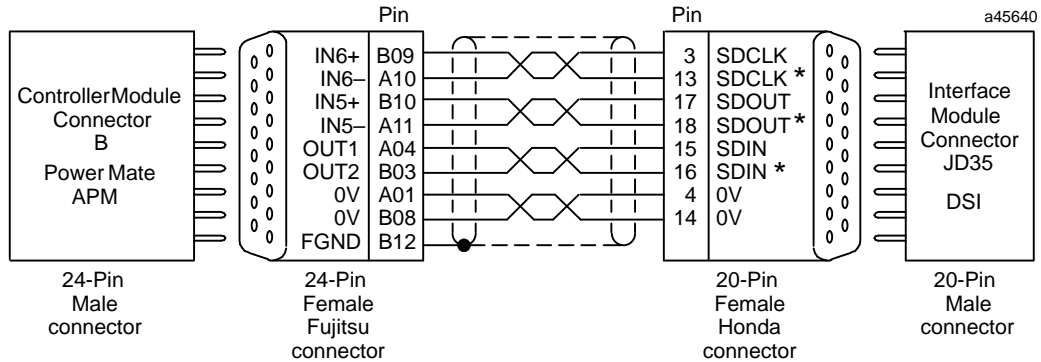


Figure 1-15. Controller Module to Interface Module Cable Diagram

Cable Components

- The cable should be made from high speed communications data quality cable consisting of #28 AWG wire (x 4 twisted pairs), with an overall shield.
- 20 Pin (Female) Honda solder connector kit
- 24 Pin (Female) Fujitsu connector kits:

Note

An eight inch pre-fabricated High Speed Serial, Power Mate J cable is included with the Power Mate J.

Power Mate J System Specifications

Table 1-7. Power Mate J Specifications

Controller Module (Power Mate APM): Power Supply Voltage: Power Supply Current: Available +5V Current/Module:	5 VDC from PLC backplane 800 mA + (1.4 x faceplate connector A 5 VDC current drain) + (95 mA SNP Mini-converter Current, if used) 500 mA @40°C (104°F) derated to 300 mA @ 55°C (131°F)
Interface Module (DSI): Module Type PLC I/O Points Used: Power Supply requirements +5 VDC +24 VDC (Isolated)	Series 90-30 PLC input module 16 inputs (%I) 450mA 70mA
Maximum Number of modules/system: † Model 311, 313, 321, 323 PLCs : (5 or 10-slot CPU baseplates) Model 331, 340, 341 PLC : (5 and 10-slot CPU baseplates, 5 and 10-slot expansion or remote baseplates - 5 total baseplates per system) Model 351 and 352 PLC: ‡: (5 and 10-slot CPU baseplates, 5 and 10-slot expansion or remote baseplates - 8 total baseplates per system)	1 PM-J system in CPU baseplate with PWR321/322 3 PM-J systems in CPU baseplate with PWR330 1 PM-J system in CPU baseplate with PWR321/322 3 PM-J systems in CPU baseplate with PWR330 2 PM-J system in expansion/remote baseplate with PWR321/322 3 PM-J system in expansion/remote baseplate with PWR330 9 total PM-J systems per PLC system with PWR321/322 15 total PM-J systems per PLC system with PWR330 1 PM-J systems in CPU baseplate with PWR321/322 3 PM-J systems in CPU baseplate with PWR330 2 PM-J system in expansion/remote baseplate with PWR321/322 3 PM-J systems in expansion/remote baseplate with PWR330 15 total PM-J systems per PLC system with PWR321/322 24 total PM-J systems per PLC system with PWR330

† The number of Power Mate J systems in a PLC depends on the available power from the installed power supply.

The total power consumption of all modules in a baseplate must be calculated so that the total load capacity of the supply is not exceeded. Refer to Chapter 3 of GFK-0356 for detailed information on load requirements for Series 90-30 modules.

The available power supplies are:

- IC693PWR321 - Standard AC/DC Power Supply (allows 15 watts for +5 VDC)
- IC693PWR322-24/48VDC Power Supply (allows 15 watts for +5 VDC)
- IC693PWR330 - High Capacity AC/DC Power Supply (allows 30 watts for +5 VDC)

‡ The CPU 352 will be available 4th Quarter, 1996.

NOTE Refer to GFK-0867B, (*GE Fanuc Product Agency Approvals, Standards, General Specifications*), or later version for product standards and general specifications.

Note

Installation instructions in this manual are provided for installations that do not require special procedures for noisy or hazardous environments. For installations that must conform to more stringent requirements (such as CE Mark), see GFK-1179, *Installation Requirements for Conformance to Standards*.

Section 2: Compatibility

The Power Mate J is compatible with the following devices:

■ Host CPU

- Series 90-30 PLC's (models 311, 313, 321, 323, 331, 340, 341, 351, and 352): release 3.52 or later.

The Power Mate J modules can be installed in any I/O module slot in the CPU rack, in any expansion rack, or in any remote rack in the Series 90-30 PLC. However for best performance, they should be located in a CPU rack or in a local expansion rack.

■ Programmer

- Series 90-30 Hand Held Programmer:
- Logicmaster 90-30 Programming Software Configuration, release 6.01 or later. *The Power Mate J system is not compatible with earlier versions of the Logicmaster software. When using the Logicmaster 90-30 configuration software, the DSI must be configured as a generic 16 point input module. Beginning with Logicmaster 90-30 Programming Software, release 7.01 or later., the DSI will exist as a configuration selection within the configuration software.*
- CIMPLICITY Control 90-30: With CIMPLICITY Control 90-30 version (future product), the DSI will exist as a configuration selection within the configuration software. No further configuration of the DSI itself will be required. Any additional configuration requirements will be satisfied through enhancements to the configuration of the controller module.

■ Controller Modules

The interface module must be used with controller modules release 2.50 or later (IC693APU301K or IC693APU302K).

When a two axis controller module is configured to work with a digital servo on one axis, it cannot simultaneously be configured to work with an analog servo on the second axis.

Caution

The controller module must be at least release R09 (Label on circuit board). Firmware upgrades of earlier hardware release versions will not function with the interface module.

■ Servo Units

- a Series Servo Amplifier Units (SVU or SVM).
- b Series Servo Amplifiers.
- a Series and b Series Servo Motors.

Chapter 2

Installation and Configuration

This chapter contains information about installing the Power Mate J in the Series 90-30 PLC and configuring the modules with Logicmaster software.

The Power Mate J can operate in *any* Series 90-30 CPU or expansion baseplate (Series 90-30 release 3.52 or later required). For limitations on the maximum number of Power Mate J systems you can install per baseplate and system, refer to *Module Specifications* in Chapter 1. The modules are connected by an eight inch high speed serial interface cable (included with Power Mate J) and are installed adjacent to each other, with the interface module to the left of the controller module.

The configuration files created by Logicmaster 90-30 configuration software must match the physical configuration of the modules.

Section 1: Installing the Power Mate J Hardware

To install the Power Mate J modules on the baseplate, follow these steps:

1. Use the Logicmaster 90-30 software or the Series 90-30 Hand-Held Programmer to stop the PLC. This will prevent the local application program, if any, from initiating any command that may affect the operation of the module.
2. Power-down the Series 90-30 PLC system.
3. Align the controller module (Power Mate APM) with the desired slot in the baseplate and backplane connector. Tilt the module upward so that the top rear hook of the module engages the slot on the baseplate (Detailed module installation information can be found in the *Series 90-30 Programmable Controller Installation Manual*, GFK-0356).
4. Swing the module down until the connectors mate and the lock-lever on the bottom of the module snaps into place engaging the baseplate notch. Repeat steps 2 and 3 for the interface module (Digital Servo Interface). The DSI should be installed adjacent and to the left of the controller module.
5. Connect the included cable from controller module connector B to interface connector JD-35. Complete cable connections to I/O and digital servo systems as indicated in Chapter 1, *Product Description*.
6. Repeat this procedure for each Power Mate J.
7. Power-up the PLC rack. The Status LED of the Power Mate APM will turn ON when the module has passed its power-up diagnostics.
8. When you have completed the hardware installation, configure the controller module(s) as explained in the next section.

For instructions about installation of the Power Mate J when IEC and other standards must be observed, see *Installation Requirements for Conformance to Standards*, GFK-1179. Please pay particular attention to the installation specifics of the controller module and the interface module.

Section 2: Configuring the Power Mate J Using Logicmaster 90-30

Controller Configuration (Power Mate APM)

The controller module (Power Mate APM) is configured using the Logicmaster 90-30 configuration software. Configuration is a two part procedure consisting of:

- Rack Slot Configuration
- Module Configuration

Rack/Slot Configuration

Logicmaster 90-30 is used to define the type and location of all modules present in the PLC racks. This is done by completing setup screens which represent the modules in a baseplate and saving the information to a configuration file, which is then downloaded to the CPU.

Once a baseplate and slot location is defined for the Power Mate APM, you can continue to the second part of the configuration process, Module Configuration.

Module Configuration

This section is divided into three parts:

- Setting the Configuration Parameters
- Essential Configuration Parameters
- Important Configuration Considerations

Setting the Configuration Parameters

As with I/O Rack Configuration, module configuration is done by completing screens in the Logicmaster 90-30 configuration software. The Series 90-30 Hand Held Programmer can only configure the Module and Serial Port Configuration Data.

Controller module configuration data consists of four types:

- Module Configuration Data
- Programmer Port Configuration Data
- Axis Configuration Data
- Program Zero

Module Configuration Data

During each CPU sweep certain data is automatically transferred both ways between the controller module and the CPU. CPU Interface data references the starting locations for the automatic transfers. The configuration parameters in Module Configuration Data are described in Table 3-1.

Table 2-1. Module Configuration Data

Configuration Parameter	Description	Values	Defaults	Units
Ref Adr	Start address for %I ref type (32 bits)	CPU Dependent	%I00001 or next higher reference	N/A
Ref Adr	Start address for %Q ref type (32 bits)	CPU Dependent	%Q00001 or next higher reference	N/A
Ref Adr	Start address for %AI ref type (15 words for 1-axis, 28 words for 2-axis)	CPU Dependent	%AI00001 or next higher reference	N/A
Ref Adr	Start address for %AQ ref type (6 words)	CPU Dependent	%AQ00001 or next higher reference	N/A
%AI Pos Err	On Single Axis Power Mate Js this parameter adds Position Error to %AI Data	DISABLED/ ENABLED	DISABLED	N/A
Fdback Type	Feedback Type ¹	DIGITAL	ENCODER	N/A
Ctl Loop	Control Loop Type	STANDARD	STANDARD	N/A
Servo Cmd	Servo Interface Type	DIGITAL	ANALOG	N/A
Motor1 Type Motor2 Type	GE Fanuc Motor Type	0 . . . 127	0 (no motor)	N/A
Motor1 Dir Motor2 Dir	Motor direction for positive velocity command	POS NEG	POS	N/A

Fdback Type. DIGITAL selects GE Fanuc Digital AC servo encoder input mode. If DIGITAL is selected, the SERVO CMD configuration parameter must also be set to DIGITAL. (Default = ENCODER).

Ctl Loop. STANDARD selects the normal controller module motion control loop. The STANDARD loop provides a velocity command proportional to position error, with optional Velocity Feed forward. CCL1 and CCL2 (Custom Control Loops) are individually designed for special applications and should not be selected. FOLLOWER selects a control loop that allows ratio tracking of a master input with zero following error and is not supported in the Power Mate J. (Default = STANDARD).

Servo Cmd. This parameter defines the type of command output provided to the servo system. DIGITAL selects a special digital output for GE Fanuc Digital servo drives. Digital servos are only supported in the controller module firmware versions 2.50 or higher. (Default = ANALOG).

Motor Type. Selects the type of GE Fanuc Digital AC servo motor to be used with the Power Mate J. The Power Mate J internally stores setup motor parameter tables for each of the servos supported. A particular motor for the indicated axis is selected in the Logicmaster 90-30 configuration fields *Motor1 Type* or *Motor2 Type*. A *Motor Type* of 0 for a particular axis disables servo control by the interface module for the axis. Supported motor types are listed in the tables below.

Motor Model - Motor model information is in the form *series, peak torque in Newton meters / maximum rpm*.

Motor Specification - Digital motor specification part numbers are in the form A06B-xxxx-yyyy, where xxxx represents the motor type field. For example: When reading a Motor Specification number from a motor label of A06B-0032-B078, the significant digits 0032 indicate the motor model of b 2/3000 the Series table references the *Motor Type Code* (36) needed for the configuration field.

a Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
13	a0.5/3000	0113
61	a1/3000	0371
46	a2/2000	0372
62	a2/3000	0373
15	a3/3000	0123
16	a6/2000	0127
17	a6/3000	0128
18	a12/2000	0142
19	a12/3000	0143
27	a22/1500	0146
20	a22/2000	0147
21	a22/3000	0148
28	a30/1200	0151
22	a 30/2000	0152
23	a30/3000	0153
30	a40/2000	0157
29	a40/FAN	0158

a L Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
56	aL3/3000	0561
57	aL6/3000	0562
58	aL9/3000	0564
59	aL25/3000	0571
60	aL50/2000	0572

a C Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
7	aC3/2000	0121
8	aC6/2000	0126
9	aC12/2000	0141
10	aC22/1500	0145

a HV Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
3	a12HV/3000	0176
4	a22HV/3000	0177
5	a30HV/3000	0178

a M Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
24	aM3/3000	0161
25	aM6/3000	0162
26	aM9/3000	0163

b Series Servo Motor

Motor Type Code	Motor Model	Motor Specification
35	b 1/3000	0031
36	b2/3000	0032
33	b3/3000	0033
34	b6/2000	0034

Motor Dir. A configured motor direction of POS (Positive) defines the positive axis direction as counter clockwise (CCW) motor shaft rotation when viewed looking into the motor. A configured motor direction of NEG (Negative) defines the positive axis direction as clockwise (CW) shaft rotation. In practice, the motor direction configuration allows the user to reverse the motion caused by all commands.

Serial Communications Port Configuration Data

The Power Mate J can be programmed using the Motion Programmer software. The computer running the Motion Programmer software connects to the Serial Communications Port (which supports the SNP protocol) on the face plate of the controller module.

The Serial Communications Port must be configured properly to communicate with the Motion Programmer. Make sure the configuration parameters for the Motion Programmer and the Serial Communications Port match. The configuration parameters for the Serial Communications Port Configuration Data are described in Table 3-2.

Table 2-2. Serial Communications Port Configuration Data

Configuration Parameter	Description	Values	Defaults	Units
Baud Rate	Baud rate of SNP Port	300, 600, 1200, 2400, 4800, 9600, 19200	19200	N/A
Parity	Parity	ODD, EVEN, NONE	ODD	N/A
Stop Bits	Number of stop bits	1 or 2	1	N/A
Data Bits	Number of data bits	7 or 8	8	N/A
Modem TT	Modem turnaround time	0..2550, in multiples of 10 milliseconds	0	ms
Idle Time	Maximum link idle time	1..60	10	sec
SNP ID	SNP ID	6 characters consisting of A-F and 0-9. First character must be A-F	A00001	N/A

Axis Configuration Data

The controller module Axis Configuration Data consists of base values for configuration parameters used by one or more motion programs. The values of these configuration parameters do not usually change, therefore they are not included in the motion program. The configuration parameters are defined and briefly described here.

Table 2-3. Axis Configuration Data

Configuration Parameter	Description	Values	Defaults	Units
User Units	User Units Value	1...65,535	1	N/A
Counts	Feedback Counts	1...65,535	1	N/A
OT Limit Sw	Overtravel Limit Sw En/Dis	ENABLED/DISABLED	ENABLED	N/A
Pos EOT	Positive End of Travel	-8,388,608...+8,388,607	+8,388,607	user units
Neg EOT	Negative End of Travel	-8,388,608...+8,388,607	-8,388,608	user units
Pos Err Lim [*]	Position Error Limit	256...60,000	+ 4096	user units
In Pos Zone	In Position Zone	0..2000	10	user units
Pos Loop TC ¹	Position Loop Time Constant	0,5...10,000	1000	ms
Vel at 10 V ¹	Velocity for 10 V Output	400...1,000,000	+4000	user units/sec
Vel FF %	Velocity Feed Forward	0..100	0	%
Intgr TC ²	Integrator Time Constant	0...10,000	0	ms
Intgr Mode ³	Integrator Mode	OFF/CONT/INZONE	OFF	N/A
Rev Comp	Reversal Compensation	0...255	0	user units
DisDly	Drive Disable Delay	0...65,535	100	ms
Jog Vel	Jog Velocity	1...8,388,607	+1000	user units/sec
Jog Acc	Jog Acceleration	1...134,217,727	+10000	user units/sec/sec
Jog Acc Mod	Jog Acceleration Mode	LINEAR/SCURVE	LINEAR	N/A
HiLimit	High Count Limit	-8,388,608...+8,388,607	+8,388,607	user units
LoLimit	Low Count Limit	-8,388,608...+8,388,607	-8,388,608	user units
Home Positn	Home Position	-8,388,608...+8,388,607	0	user units
Home Offset	Home Offset Value	-32,768...+32,767	0	user units
Finl Hm Vel	Final Home Velocity	1...8,388,607	+500	user units/sec
Find Hm Vel	Find Home Velocity	1...8,388,607	+2000	user units/sec
Home Mode	Find Home Mode	HOMESW/MOVE+/ MOVE-	HOMESW	N/A

¹ Scaling dependent configuration parameter, value range depends upon user scaling.

² Intgr TC is not used. This location is used to configure Velocity Loop gain with a range of 0 - 255.

³ Intgr Mode is not used. CONT selects Absolute Encoder. OFF or IN ZONE selects Incremental Encoder.

User Units, Counts. The ratio of User Units to Counts sets the number of programming units for each feedback count. This allows the user to program the controller module in units appropriate for the application. The RANGE for user units and counts is 1 to 65,535. The RATIO of user units to counts must be in the range 8:1 to 1:32. For example, if there is 1.000 inch of travel for 8192 feedback counts, a user units:counts ratio of 1000:8192 sets 1 user unit = 0.001 inch. Default is 1:1.

OT Limit Sw. Selects whether the controller module will monitor axis 1 hardware overtravel input limit switches. If axis 1 OT Limit Switches are disabled, then Axis 2 Home Switch may be used (two axis controller only).

The one axis controller module can use the hardware Over travel Limit Switch inputs. If Limit Switches are DISABLED, the Limit Switch inputs can be used as general purpose

inputs. If ENABLED, then 24 VDC must be applied to both of the inputs in order for the controller module to operate. If not, then whenever the drive is enabled a Limit Switch Input Error will occur. The Jog and Clear Error %Q bits can be used simultaneously to back away from the Limit Switch.

The *OT Limit Switch* setting for the two axis controller module may be set to ENABLED for axis one only. (The *OT Limit Switch* setting for axis 2 should always be *DISABLED*.) When axis one overtravel switches are enabled the Home Switch input for axis two (Home 2) is unavailable. Axis 1 Default: ENABLED.

Pos EOT. Positive Software End of Travel limit (User Units). If the controller module is programmed to go to a position greater than the Positive EOT then an error will result and the controller module will not allow axis motion. Default: +8,388,607

Neg EOT. Negative Software End of Travel limit (User Units). If the controller module is programmed to go to a position less than the Negative EOT then an error will result and the controller module will not allow axis motion. Default: -8,388,608

If Pos / Neg EOT limits are both set to zero, the controller module uses +8,388,607 / -8,388,608 instead.

Pos Err Lim. Position Error Limit (User Units). The maximum Position Error (*Commanded Position - Actual Position*) allowed when the controller module is controlling a servo. Position Error Limit should normally be set to a value 10% to 20% higher than the highest position error encountered under normal servo operation. Default: 4096. The range formula for Position Error Limit is:

$$256 \times (\text{user units/counts}) \leq \text{Position Error Limit} \leq 60,000 \times (\text{user units/counts})$$

If Velocity Feed forward is not used, Position Error Limit can be set to a value approximately 20% higher than the position error required to produce a 4096 rpm command. *The Position Error* (User Units) required to produce a 4096 rpm command with 0% Velocity Feed forward is:

$$\text{Position Error (user units)} = \frac{\text{Position Loop Time Constant (ms)} \times \text{Servo Velocity @ 10v (user units/sec)}}{1000}$$

If Velocity Feed forward is used to reduce the following error, a smaller error limit value can be used, but in general, the error limit value should be 10% - 20% higher than the largest expected following error.

Note

An *Out of Sync* error will occur and cause a fast stop if the *Position Error Limit* Value is exceeded by more than 1000 counts. The controller module attempts to prevent an *Out of Sync* error by temporarily halting the internal command generator whenever position error exceeds the *Position Error Limit*. Halting the command generator allows the position feedback to catch up and reduce position error below the error limit value.

If the feedback does not catch up and the position error continues to grow, the *Out of Sync* condition will occur. Possible causes are:

1. erroneous feedback wiring
2. feedback device coupling slippage
3. servo drive failure.
4. mechanically forcing the motor/encoder shaft past the servo torque capability.

In Pos Zone. In Position Zone (User Units). When the magnitude of servo position error is less than or equal to this value and neither Jog nor Move at Velocity is commanded, the *In Zone* status bit will be set. In Position Zone also determines the position error at which *PMOVEs* are considered to be complete. Default: 10

Pos Loop TC. Position Loop Time Constant (milliseconds). The desired servo position loop time constant. The lower the value, the faster the system response. Values which are too low will cause system instability and oscillation. For accurate tracking of the commanded velocity profile, *Pos Loop TC* should be 1/4 to 1/2 of the MINIMUM system deceleration time. Setting Position Loop Time Constant to 0 will place the controller module in *open loop* mode where only Velocity Feed forward is used to produce the velocity command output to the interface module. **The Position Loop Time Constant will not be accurate unless the Vel at 10 V value is set correctly.** Default: 1000

Velocity at 10 Volts All controller module and servo functions depend on this value being correct for proper operation.

The *Velocity at 10 v*: configuration field should be set to a conversion constant value of 139820 multiplied by the decimal value of the *User Units to Counts* ratio. For example: with a *User Unit* value of 1 and a *Counts* value of 2 the decimal value of the ratio would be 0.5. The conversion constant multiplied by 0.5 yields the value 69910 for the *Velocity at 10 v* ($1/2 * 139820 = 69910$)

If the user sends the Power Mate J a velocity command which exceeds the capability of the servo system, the Power Mate J will *clamp* that command value at the appropriate maximum boundary; no error will be reported back to the controller module

Default: 4000

Vel FF %. Velocity Feed forward gain (percent). The percentage of Commanded Velocity that is added to the controller module velocity command output. Increasing Velocity Feed forward causes the servo to operate with faster response and reduced position error. Optimum feed forward values are 80-90 %. The *Vel at 10 V* value must be set correctly for proper operation of Velocity Feed forward. Default: 0

Intgr TC. Integrator Time Constant. The position error integrator in digital mode is not used and this setting is used to configure Velocity Loop Gain. Default: 0

Note

Configuration software prior to Release 7.0 only allows Intgr TC to be 0 or 0 - 10,000. Configuration software versions 7.0 and later allow a range of 0 – 10,000. Therefore, VLGn values of 1 - 9 can only be configured with Configuration software versions 7.0 and later.

Intgr Mode. Integrator Mode. Operating mode for position error integrator. However, the position error integrator is not used for DIGITAL mode of operation. The Integrator Mode field is used to configure Incremental or Absolute feedback type for the digital serial encoder. OFF or IN ZONE means that feedback counts from the serial encoder will be interpreted as *INCREMENTAL* counts, encoder battery alarms are not reported. CONTINUOUS means the serial encoder feedback will be considered as *ABSOLUTE*, battery alarms will be reported. See Appendix D for more information about using absolute mode encoders. Default: OFF (Selects Incremental Encoder)

Rev Comp. Reversal Compensation (User Units). A compensation factor which allows the servo to reverse direction and still provide accurate positioning in systems exhibiting backlash. Default: 0.

DisDly. Servo Drive Disable Delay (milliseconds). The time delay from zero velocity command to the drive enable output switching off. Disable Delay is effective when the *Enable Drive* %Q bit is turned off or certain error conditions occur. Disable Delay should be longer than the deceleration time of the servo from maximum speed. Default: 100

Jog Vel. Jog Velocity (User Units/second). The velocity at which the servo moves during a *Jog* operation. Default: 1000

Jog Acc. Jog Acceleration Rate (User Units/second/second). The acceleration rate used during *Jog*, *Find Home*, *Move at Velocity*, and *Abort* operations. The *Jog Acceleration* is used when an acceleration has not been programmed. Default: 10000

Jog Acc Mod. Jog Acceleration Mode (LINEAR or S-CURVE). The acceleration mode for *Jog*, *Find Home*, *Move at Velocity*, and *Abort* operations. LINEAR causes commanded velocity to change linearly with time. S-CURVE causes commanded velocity to change more slowly than the linear mode at the beginning and end of acceleration intervals. Default: LINEAR

Hi Limit. High Count Limit (User Units). When moving +, the Actual Position will roll over to the low limit when this value is reached. The Count Limits are used for rotary applications. See Appendix D for considerations when using absolute mode encoder. Default: 8,388,607

Lo Limit. Low Count Limit (User Units). When moving -, the Actual Position will rollover to the high limit when this value is reached. The Count Limits are used for rotary applications. See Appendix D for considerations when using absolute mode encoder. Default: -8,388,608

If Hi / Low Count limits are both set to zero, the controller module uses default limits +8,388,607 / -8,388,608 instead.

Home Positn. Home Position (User Units). The value assigned to *Actual Position* at the end of a *Find Home* cycle. Default: 0

Home Offset. Home Position Offset (User Units). The offset of the servo final stopping point at the completion of a *Find Home* cycle. Home Offset adjusts the final servo stopping point relative to the Encoder marker. Default: 0

Fnl Home Vel. Final Home Velocity (User Units/second). The velocity at which the servo seeks the final Home Switch transition and Encoder Marker pulse at the end of a *Find Home* cycle. Final Home Velocity must be slow enough to allow a 10 millisecond (filter time) delay between the final Home Switch transition and the Encoder Marker pulse. Default: 500

Find Home Vel. Find Home Velocity (User Units/second). The velocity at which the servo seeks the initial Home Switch transitions during the *Find Home* cycle. If desired, Find Home Velocity can be set to a high value to allow the servo to quickly locate the Home Switch. Default: 2000

Home Mode. Find Home Mode. This method is used to find home during a *Find Home* cycle. HOME SWITCH indicates that a Home Switch is to be monitored to Find Home. MOVE+ and MOVE- specify direct positive and negative movement to the next marker at the Final Home Velocity. Default: HOMESW

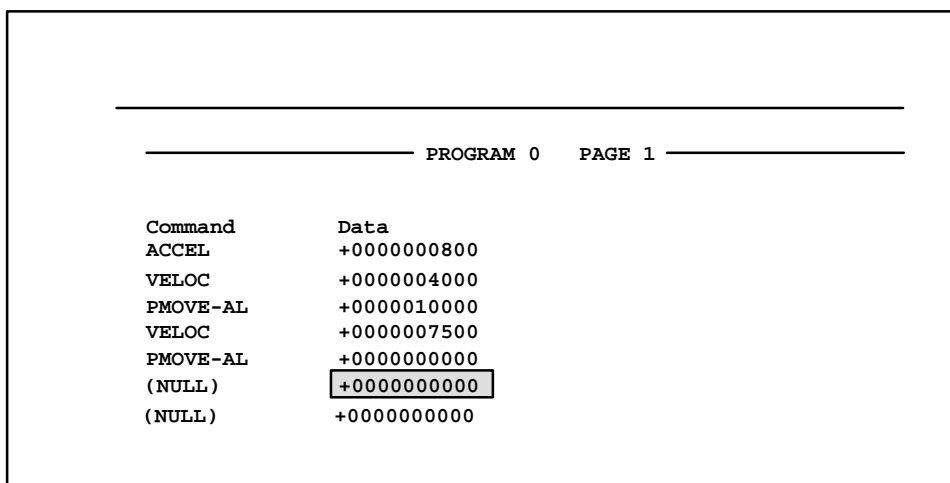
Digital Servo Interface Configuration

The interface module can be completely configured by the Logicmaster 90-30 configuration software, release 6.01 or later. It will be necessary to configure the interface module slot as a generic 16 point input module. The interface module should be installed in an empty slot to the left of the controller module.

Section 3: Program Zero

Program Zero is a short default motion program (20 commands maximum) which is defined in the Logicmaster 90-30 configuration software and downloaded to the Power Mate J whenever the controller module is initialized by the PLC. Program Zero is programmed by entering motion commands in an English-language format similar to that of the Motion Programmer software.

In a 2-axis module, the controller module determines whether Program Zero is an axis 1, axis 2, or multiaxis program according to which axis or axes are used in the program. For example, if all Program Zero commands contain Axis number = 1, Program Zero will be classified as a Single Axis program for Axis 1. Therefore Program Zero will be allowed to execute **concurrently** with another Single Axis Program for Axis 2. A multiaxis program 0 requires that both axes in a two axis module be home referenced prior to execution.



The screenshot shows a text-based editor window titled "PROGRAM 0 PAGE 1". It contains a table with two columns: "Command" and "Data". The data values are positive integers with leading zeros. The value "+0000000000" in the fifth row is highlighted with a rectangular box.

Command	Data
ACCEL	+0000000800
VELOC	+0000004000
PMOVE-AL	+0000010000
VELOC	+0000007500
PMOVE-AL	+0000000000
(NULL)	+0000000000
(NULL)	+0000000000

Figure 2-1. Example of a Program Zero Editor (One Axis)

The Program Zero commands are entered using the Function keys (F1-F9). Paired with each command is a data field for entering either a signed double integer or the number of a controller module parameter, as appropriate for the configured command.

Program Zero Programmer 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.

Command Name Field. The 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-F9). Most of the commands fall into groups such as CMOVE, PMOVE, VELOC, ACCEL, and so forth. The variations of these commands can be displayed (using the Tab key) after pressing the function key for the particular group.

For example, to program a CMOVE-IS-P (Continuous Move, incremental, s-curve, using a parameter), first select a command field then press the *CMOVE* function key. This causes CMOVE-AL to be displayed in the command name field of the screen. The variations of the command are cycled through by pressing the Tab key.

To program a JUMP command press F10 (MORE) and then press F1 (JUMP).

Data Field. Paired with each command is a data field. In the data field enter either a signed double integer or the number of a parameter (0-255), as appropriate for the configured command. Parameters 1-20 can be loaded with data using the LOAD-P command. Block parameters can be entered in the data field when paired with Block or Jump instructions. The data field of a dwell command indicates the number of milliseconds that the program will pause in execution.

Axis# Field.

This field is only used in the configuration for a two axis controller module. It is used to set the axis number for commands such as VELOC or PMOVE

Table 2-4. Program Zero Motion Program Commands

Command Name	Definition	Range	Default
(NULL)	Do nothing	0	0
BLOCK	Define Block Number	1..65,535	1
BLOCK-SYNC	Define Synchronous Block Number	1..65,535	1
CMOVE-AL	Continuous move, Absolute, Linear	-8,388,608.. 8,388,607	1
CMOVE-AL-P	Continuous move, Absolute, Linear, Use data in Parameter	0..255	1
CMOVE-AS	Continuous move, Absolute, S-curve	-8,388,608.. 8,388,607	1
CMOVE-AS-P	Continuous move, Absolute, S-curve, Use data in Parameter	0..255	1
CMOVE-IL	Continuous move, Incremental, Linear	-8,388,608.. 8,388,607	1
CMOVE-IL-P	Continuous move, Incremental, Linear, Use data in Parameter	0..255	1
CMOVE-IS	Continuous move, Incremental, S-curve	-8,388,608.. 8,388,607	1
CMOVE-IS-P	Continuous move, Incremental, S-curve, Use data in Parameter	0..255	1
PMOVE-AL	Positioning move, Absolute, Linear	-8,388,608.. 8,388,607	1
PMOVE-AL-P	Positioning move, Absolute, Linear, Use data in Parameter	0..255	1
PMOVE-AS	Positioning move, Absolute, S-curve	-8,388,608.. 8,388,607	1
PMOVE-AS-P	Positioning move, Absolute, S-curve, Use data in Parameter	0..255	1
PMOVE-IL	Positioning move, Incremental, Linear	-8,388,608.. 8,388,607	1
PMOVE-IL-P	Positioning move, Incremental, Linear, Use data in Parameter	0..255	1
PMOVE-IS	Positioning move, Incremental, S-curve	-8,388,608.. 8,388,607	1
PMOVE-IS-P	Positioning move, Incremental, S-curve, Use data in Parameter	0..255	1
VELOC	Set Velocity	1..8,388,607	2000
VELOC-P	Set Velocity to data in Parameter	0..255	1
ACCEL	Set Acceleration	1..134,217,727	5000
ACCEL-P	Set Acceleration to data in Parameter	0..255	1
WAIT	Wait for CTL XX bit to go high before moving	1..12	1
LOAD-P01 ↓ LOAD-P20	Load controller module Parameter register number	-8,388,608.. 8,388,607	0
DWELL	Wait for specified time in milliseconds	0..60,000	0
DWELL-P	Wait X milliseconds, X is the value in the Parameter	0..255	1
JUMP-UNCOND	Continue program execution at given block #	1..65,535	1
JUMP-CTL01 ↓ JUMP-CTL12	When the given CTL bit goes high during the current block execution, transfer program execution to the given block #	1..65,535	1

Program Zero Motion Command Descriptions

Each command is briefly described below. For a comprehensive explanation of Motion Programming on the controller module, refer to GFK-0664, *The Series 90 Power Mate APM Programmer's Manual*.

Acceleration (ACCEL). This modal 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 (BLOCK). Block numbers are used to monitor and synchronize program execution, terminate jump testing, and as jump destinations. Block number is an actual command in the Program Zero Editor. The default block number is zero. The most recently used block number will be reported to the PLC until a subsequent block instruction is processed, or the controller is reset or power cycled.

Cmove (CMOVE). 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 no previous acceleration or velocity has been specified in a motion program, the configured *Jog Acceleration* and/or *Jog Velocity* will be used.

Dwell (DWELL). This command causes motion to cease for a specified time period (in milliseconds) before proceeding to the next command.

Jump (JUMP). This command is used to Branch program execution to another location in the program 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. 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. A maximum of 220 jumps can exist for all programs and subroutines.

Once the condition testing is allowed to start, the test will occur once each millisecond (every two milliseconds for two axis controller module) until a Block Number or another Jump command is encountered. This will allow continued testing while a move or series of moves takes place, if the *Jump* command is located ahead of the moves in the block. If a jump occurs during a move, the remainder of the move is aborted and the command at the destination location is immediately effective. Jump destinations must be limited to the bounds of the program containing the *Jump* command.

Load Parameter (LOAD). This command initializes or changes a controller module data parameter value. The new value becomes effective immediately when encountered in the program. Program Zero may load parameters 1 to 20.

Pmove (PMOVE). 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. The axis movement will stop for one or two milliseconds in the *In Zone* range, then the next program command will execute.

Velocity (VELOC). This modal command specifies the velocity of axis motion. Once encountered, this command will remain in effect until overridden by a later *Velocity* command.

Wait (WAIT). 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.

Essential Configuration Parameters

To correctly configure the controller module, several configuration parameters must be properly set. This section explains how these parameters affect controller module operation and how the parameters relate to each other. While all parameters are important, some parameters are absolutely essential to correct operation.

Velocity at 10 volts

All controller module and servo functions depend on this value being correct for proper operation. The *Velocity at 10 V*: configuration field should be set to a conversion constant value of 139820 multiplied by the decimal value of the *User Units to Counts* ratio. For example: with a *User Unit* value of 1 and a *Counts* value of 2 the decimal value of the ratio would be 0.5. The conversion constant multiplied by 0.5 yields the value 69910 for the *Velocity at 10 v* ($1/2 * 139820 = 69910$)

If the user sends the Power Mate J a velocity command which exceeds the capability of the servo system, the Power Mate J will *clamp* that command value at the appropriate maximum boundary; no error will be reported back to the controller module

Position Loop Time Constant

The lower the Position Loop Time Constant, the faster the axis will respond. However, if the Time Constant is too low, the system may become unstable or even oscillate. When decelerating, especially at high speeds, the controller module could command a servo to stop at a certain point faster than the servo could respond. This would result in overshoot.

For accurate tracking of the commanded velocity profile, *Pos Loop TC* should be 1/4 to 1/2 of the MINIMUM system deceleration time.

For users familiar with servo bandwidth expressed in rad/sec:

$$\text{Bandwidth (rad/sec)} = 1000 / \text{Position Loop Time Constant (ms)}$$

For users familiar with servo gain expressed in ipm/mil:

$$\text{Gain (ipm/mil)} = 60 / \text{Position Loop Time Constant (ms)}$$

Gain (ipm/mil)	Bandwidth (rad/sec)	Position Loop Time Constant (ms)
0.5	8.5	120
0.75	12.5	80
1.0	16.6	60
1.5	25.1	40
2.0	33.4	30
2.5	41.8	24
3.0	50	20

Open Loop Mode

For applications which do not require feedback control or employ very crude positioning systems, an Open Loop Mode exists. Setting a zero Position Loop Time Constant, which indicates that the positioning loop is disabled, selects this mode. Note that in Open Loop Mode, the only way to generate motion is to program a non-zero Velocity Feed forward. The *Position Error* is no longer used to generate motion because *Position Error* is based on feedback and Open Loop Mode ignores all feedback.

User Units and Counts

The controller module has a very powerful scaling feature. A User Units to Counts ratio can be configured to allow programming in other than default counts. In a simplified example, all digital encoders used with the Power Mate J produces 8,192 counts per revolution and is geared to a machine which produces one inch per revolution. The default units would be one thousandth of an inch (1000:8192). However, you may want to write programs and use the controller module with metric units. A ratio of 2540 User Units to 8192 Counts can be configured to allow this. With this ratio, one user unit would represent .01 millimeters. 2540 user units would represent 25.4 millimeters (one inch) of travel.

The range for both User Units and Counts is 1 to 65,535. However, the ratio of User Units to Counts must be between 8:1 and 1:32. A more detailed example is described below.

Example

A machine has a digital servo motor (8192 counts per revolution) that is connected through a 20:1 gear reduction to a 14.336 inch diameter spur gear. The programmer wishes to program in 0.01 inch resolution.

The following is the given data:

- 14.336 inch diameter spur gear
- 8192 count encoder
- 20:1 reduction gear
- 0.01 inch is the desired programming unit

First, determine the number of Encoder Counts per revolution of the spur gear.

$$8192 * 20 = 163,840 \text{ Counts per rev}$$

The multiply by 20 results from the reduction gear

Next, determine the number of User Units per revolution using 0.01 inch resolution.

$$14.336 \text{ inches} * \pi = 45.0378 \text{ inches per revolution}$$

$$45.0378 \text{ inches per rev} / 0.01 \text{ inches per User Unit} = 4503.78 \text{ User Units per revolution}$$

This User Units to Counts ratio would be $4503 / 163,840$ or 0.02748 which is about 1:27. This ratio is less than 1:32 so the resolution is adjusted.

To assume a new programming unit resolution of 0.001 inch.

Determine the revised User Units per revolution using 0.001 inch resolution.

$$14.336 \text{ inches} * \pi = 45.0378 \text{ inches per revolution}$$

$$45.0378 \text{ inches per rev} / 0.001 \text{ inches per User Unit} = 45,037.8 \text{ User Units per revolution}$$

This User Units to Counts ratio would be $45,037 / 163,840$ or 0.27488 which is about 1:3.6. Thus values of 2749 for User Units and 10,000 for Counts could be used to obtain 0.001 inch resolution.

Acceleration Mode

The controller module supports two types of acceleration, Linear and S-Curve. Linear motion consists of constant linear acceleration and deceleration to specified velocities. A profile of velocity versus time would show straight lines. S-Curve motion uses a variable acceleration. An S-Curve acceleration would begin slowly and increase, up to the specified acceleration, then decrease back to zero when the specified velocity was reached. ***S-Curve motions require twice the time and distance to change velocity compared to linear motions with the same acceleration.***

CMOVE and PMOVE motion commands specify the acceleration mode for all programmed motion, except the jump stop condition which is described in Chapter 4. *Jog, Find Home, Abort, and Move at Velocity* use the configured *Jog Acceleration Mode*.

Important Configuration Considerations

Software End of Travel Limits, positive and negative, are used to bound commanded motion. The controller module will not execute any programmed motion which takes the Commanded Position to or past an EOT Limit. If the Position Valid %I bit is ON, a Jog will immediately be stopped at the EOT Limit. Jogs with Position Valid OFF and Move at Velocity immediate commands ignore the EOT Limits. This enables movement outside the EOTs while setting up a system.

High and Low Count Limits can be used for rotary type motion in which a servo can move forever in either direction. When a Count Limit is reached, the reported position *wraps around* to the opposite limit where it can continue changing. *Thus if the Count Limits are set equal to or within the End of Travel Limits, the End of Travel Limits will never be reached.* See Appendix D for consideration in using absolute encoder mode.

In Rotary mode, Jogs or Incremental Moves can be used to cause continuous motion in either direction and the EOT limits will never be exceeded.

In EOT Mode (Non Rotary Mode) Jogs and Programmed Motions beyond either EOT limit are not allowed.

Rotary Mode selection is based on the relative values of Hi/Lo Count Limits and the Pos/Neg EOT limits:

The controller module will be in Rotary Mode if the Hi Count Limit is \leq the Pos EOT limit **AND** the Lo Count limit is \geq the Neg EOT limit.

The controller module will be in EOT Mode (Non Rotary Mode) if the Hi Count Limit is $>$ the Pos EOT limit **OR** if the Lo Count limit is $<$ the Neg EOT limit.

Rotary Mode is the DEFAULT operating mode for the Standard Control Loop because the default Count limits and EOT limits are +8,388,607 / -8,388,608.

The Over travel Limit Switches are external hardware switches. The switch status is returned in the CTL bits CTL05 -CTL06. The controller module can be configured to use these CTL bits as Over travel Limit Switches. If the switches are enabled, whenever the drive is enabled and an Over travel bit is OFF the controller module commands all motion on that axis to immediately cease. If this happens, the Jog and Clear Error %Q bits can be turned on simultaneously to move the servo away from the Over travel switch.

A Reversal Compensation feature on the controller module allows accurate positioning on systems with backlash. Backlash is exhibited by a servo that must move a small amount before the system begins moving when it switches directions. For example, consider a dead bolt door lock. Imagine the servo controls the key in the lock and the feedback reports movement of the bolt. When the servo turns the key counterclockwise, the bolt moves left. However, as the servo turns the key clockwise, the bolt does not move until the key turns to a certain point. The Reversal Compensation feature adds in the necessary motion to move the servo to where motion will begin on the feedback device. The controller module removes the compensation when a move in the negative direction is commanded, and adds the compensation before a move in the positive direction.

The Servo Drive Disable Delay specifies how long the controller module will wait after a zero velocity is commanded before it turns OFF the Drive Enable output. Because turning OFF the Drive Enable Relay stops the controller module from commanding the servo, there are times when the relay should stay ON. For example, if the servo runs into an End of Travel Limit and the Drive Enable Relay was immediately turned OFF because of the error, the servo may continue moving until it coasted to a stop. Thus to allow the controller module to command and control a fast stop, the Drive Disable Delay should be longer than the deceleration time of the servo from maximum speed.

The disable delay may be used to control when torque is removed from the motor shaft. Applications using an electro-mechanical brake may need time for the brake to engage prior to releasing servo torque.

Chapter 3

Power Mate J to PLC Interface

The Power Mate APM and DSI both transfer command data and status bits to the PLC. This chapter describes data used by each module in the Power Mate J.

Input Status Data (Transferred from PM-APM to PLC)

- Status Bits: 32 bits of (%I) data
- Status Words: 15 words of (%AI) data for a 1 axis Power Mate J
28 words of (%AI) data for a 2 axis Power Mate J

Output Command Data (Transferred from PLC to PM-APM)

- Discrete Commands: 32 bits of (%Q) data
- Immediate Commands: 6 words of (%AQ) data

Input Status Data (Transferred from DSI to PLC)

- Status Bits: 16 bits of (%I) data

Section 1: Power Mate APM Command and Status Data

Controller Module %I Status Bits (PM-APM)

The following %I Status Bits are transferred automatically from the controller to the CPU each sweep. The actual addresses of the Status Bits depend on the starting address configured for the %I references. The bit numbers listed in the following table are offsets to this starting address.

Table 3-1. %I Status Bits (1-Axis)

Bit *	Description	Bit *	Description
00	AxisEnabled Axis1	16	Front Panel Input CTL01 Status (Strobe 1 State)
01	Position Valid Axis1	17	Front Panel Input CTL02 Status
02	DriveEnabled Axis1	18	Front Panel Input CTL03 Status (Home Switch 1)
03	ProgramActive Axis1	19	reserved
04	Moving Axis1	20	Front Panel Input CTL05 Status (+ Overtravel)
05	In Zone Axis1	21	Front Panel Input CTL06 Status (- Overtravel)
06	PositionStrobe Axis1	22	reserved
07	In ErrorLimit Axis1	23	reserved
08	reserved	24	Configuration Complete
09	reserved	25	reserved
10	reserved	26	reserved
11	reserved	27	reserved
12	reserved	28	reserved
13	reserved	29	reserved
14	reserved	30	PLC Control Active
15	reserved	31	Error

* The bit numbers represent an offset to the starting address for %I references.

Table 3-2. %I Status Bits (2-Axis)

Bit *	Description		Bit *	Description
00	AxisEnabled	Axis1	16	Front Panel Input CTL01 Status (Strobe 1 State)
01	PositionValid	Axis1	17	Front Panel Input CTL02 Status (Strobe 2 State)
02	DriveEnabled	Axis1	18	Front Panel Input CTL03 Status (Home Switch 1)
03	ProgramActive	Axis1	19	reserved
04	Moving	Axis1	20	Front Panel Input CTL05 Status (+ Overtravel 2/Home Switch 2)
05	In Zone	Axis1	21	Front Panel Input CTL06 Status (- Overtravel 1)
06	PositionStrobe	Axis1	22	reserved
07	In ErrorLimit	Axis1	23	reserved
08	AxisEnabled	Axis2	24	Configuration Complete
09	PositionValid	Axis2	25	reserved
10	DriveEnabled	Axis2	26	reserved
11	ProgramActive	Axis2	27	reserved
12	Moving	Axis2	28	reserved
13	In Zone	Axis2	29	reserved
14	PositionStrobe	Axis2	30	PLC Control Active
15	In ErrorLimit	Axis2	31	Error

* The bit numbers represent an offset to the starting address for %I references.

Axis Enabled. The *Axis Enabled* status bit is ON when the controller is ready to receive commands and control a servo. An error condition which stops the servo will turn *Axis Enabled* OFF.

Position Valid. The *Position Valid* status bit indicates that the value in the %AI *Actual Position* status word has been initialized by a *Set Position* command or successful completion of the *Find Home* cycle. *Position Valid* must be ON in order to execute a motion program.

If the controller is configured to use an absolute feedback digital encoder, *Position Valid* is automatically set whenever a correct encoder status signal is received. See Appendix D for details of operation when absolute mode digital encoders are used.

Drive Enabled. The *Drive Enabled* status bit indicates the state of the *Enable Drive* discrete command and the relay contact supplied by the controller module. 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.

Program Active. The *Program Active* status bit for each axis indicates that a Motion Program (0-10), or a %AQ Move command (27h) is executing on that axis. On a two axis Power Mate J, executing a multi-axis program will set both *Program Active* bits.

Moving. 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.

In Zone. 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*, *Feedhold*, or % *Feedrate* = 0).

Position Strobe. 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 data will be captured within 1 millisecond from a Strobe Input (2 milliseconds for two axis Power Mate J).

In Error Limit. 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.

Faceplate Input %I Status Bits. These inputs indicate the state of the external input devices connected to the controller module's face plate terminals CTL01-03 and CTL05-06. These inputs (as well as CTL09-CTL12 from the PLC %Q table) may be tested by the controller module during the execution of *Wait* and *Conditional Jump* commands.

CTL01	Axis 1 Strobe Input
CTL02	Axis 2 Strobe Input (Two Axis Power Mate J Only)
CTL03	Axis 1 Home Switch Input
CTL05	Axis 1 Overtravel (+) Switch Input or Axis 2 Home Switch Input
CTL06	Axis 1 Overtravel (-) Switch Input

The configuration software selects whether the Axis 1 Overtravel Switches are enabled or disabled. If they are disabled on a Two Axis Power Mate J, CTLO5 can be used as the AxisTwo Home Switch Input.

Configuration Complete. The *Configuration Complete* status bit is set by %AQ Immediate command 49h. This status bit is cleared whenever the PLC sends a reset command or new configuration to the controller module. Configuration Complete can be set by a PLC program after other %AQ Immediate commands such as *In Position Zone* or *Position Loop Time Constant* have been sent to the controller module. The status bit can then be monitored by the PLC. If the bit is cleared, then the controller module has been reset or reconfigured and the PLC should re-send all necessary %AQ configuration commands before setting the bit again.

PLC Control Active. Normally the *PLC Control Active* status bit is set, indicating that the %Q discrete commands or %AQ immediate commands from the PLC can control the controller module. PLC Control Active is cleared only when the Status screen in the Motion Programmer is used instead of the PLC to control the controller module, a capability not yet implemented.

Error. This status bit is set when the controller module 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.

Controller Module %AI Status Words (PM-APM)

The following %AI Status Words are transferred automatically from the controller module to the CPU each sweep. The actual addresses of the Status Words depend on the starting address configured for the %AI references. See Table 3-1, Module Configuration Data. The word numbers listed in the following table are offsets to this starting address.

Table 3-3. %AI Status Words (1-Axis)

Word*	Description
000	Status Code
001	Command Block Number Axis1
002 - 003	Commanded Position ** Axis1
004 - 005	Actual Position Axis1
006 - 007	Strobe Position Axis1
008 - 009	Commanded Velocity Axis1
010 - 011	Actual Velocity Axis1
012 - 013	Position Error*** Axis1
014	reserved

- * The word numbers represent an offset to the starting address for %AI references.
- ** The %AQ Immediate Command *Select Return Data* (40h) can be used to select *Command Position* or other data such as module firmware revision.
- *** If Module configuration parameter %AI Pos Err is DISABLED, the value zero will be reported instead of Position Error in %AI words 12 - 13.

Table 3-4. %AI Status Words (2-Axis)

Word*	Description
000	Status Code
001	Command Block Number Axis1
002 - 003	Commanded Position ** Axis1
004 - 005	Actual Position Axis1
006 - 007	Strobe Position Axis1
008 - 009	Commanded Velocity Axis1
010 - 011	Actual Velocity Axis1
012 - 013	Position Error Axis1
014	reserved
015	Command Block Number Axis2
016 - 017	Commanded Position ** Axis2
018 - 019	Actual Position Axis2
020 - 021	Strobe Position Axis2
022 - 023	Commanded Velocity Axis2
024 - 025	Actual Velocity Axis2
026 - 027	Position Error Axis2

- * The word numbers represent an offset to the starting address for %AI references.
- ** The %AQ Immediate Command *Select Return Data* (40h) can be used to select *Commanded Position* or other data such as module firmware revision.

Note

All %AI data except Actual Velocity is updated within the controller module once every 10 milliseconds. Actual Velocity is updated once every second.

Status Code. *Status Code* indicates the current operating status of the module. When the *Error %I* flag is set, it contains an error code number which describes the condition causing the error.

For a list of Power Mate J error codes refer to Appendix A.

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. Block numbers are displayed only if the motion program uses them. Additionally, the most recently used block number will be displayed until superceded by a new value.

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 the *Position Error* value which produces the Velocity Command to drive the axis. The rate at which the *Commanded Position* is changed determines the velocity of axis motion.

If *Commanded Position* moves past either of the count limits, it will *roll over* to the other limit and continue in the direction of the axis motion.

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

If *Actual Position* moves past either of the count limits, it will *roll over* to the other limit and continue in the direction of the axis motion.

Strobe Position. *Strobe Position* (user units) contains the axis position when a Strobe Input occurs. When a Strobe Input occurs, the *Position Strobe %I* bit is set to indicate to the PLC that new Strobe data is available in the *Strobe Position* status word. The PLC must set the *Reset Strobe Flag %Q* bit to clear the *Position Strobe %I* bit.

Strobe Position will be maintained and will not be overwritten by additional Strobe Inputs until the *Position Strobe %I* bit 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 axis 1 *Strobe Position* is also placed in data parameter 255 for use within motion programs. In a 2-axis Power Mate J the axis 2 *Strobe Position* is placed in data parameter 254.

This feature allows the strobe input to trigger a Conditional JUMP to a program block using the *Strobe Position* as the destination of a CMOVE or PMOVE command.

Commanded Velocity. *Commanded Velocity* (user units/sec) is a value generated by the controller module 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.

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

Position Error. *Position Error* (user units) is the difference between *Commanded Position* and *Actual Position* at any instant in time.

Controller Module %Q Discrete Commands (PM-APM)

The following %Q Outputs representing Discrete Commands are sent automatically to the controller module from the CPU each PLC sweep. A command is executed simply by turning on the Output Bit of the desired command.

The actual addresses of the Discrete Command bits depend on the starting address configured for the %Q references. See Table 3-1, Module Configuration Data. The bit numbers listed in the following table are offsets to this starting address.

Table 3-5. %Q Discrete Commands (1-Axis)

Bit *	Description	Bit *	Description
00	AbortAll Moves Axis1	16	CTL09
01	Feedhold Axis1	17	CTL10
02	EnableDrive Axis1	18	CTL11
03	Find Home Axis1	19	CTL12
04	Jog Plus Axis1	20	Execute Motion Program0
05	Jog Minus Axis1	21	Execute Motion Program 1
06	Reset Strobe Flag Axis1	22	Execute Motion Program2
07	reserved	23	Execute Motion Program 3
08	reserved	24	Execute Motion Program 4
09	reserved	25	Execute Motion Program 5
10	reserved	26	Execute Motion Program 6
11	reserved	27	Execute Motion Program 7
12	reserved	28	Execute Motion Program 8
13	reserved	29	Execute Motion Program 9
14	reserved	30	Execute Motion Program 10
15	reserved	31	Clear Error

* The bit numbers represent an offset to the starting address for %Q references.

Table 3-6. %Q Discrete Commands (2-Axis)

Bit *	Description	Bit *	Description
00	AbortAllMoves Axis1	16	CTL09
01	Feedhold Axis1	17	CTL10
02	EnableDrive Axis1	18	CTL11
03	FindHome Axis1	19	CTL12
04	JogPlus Axis1	20	Execute Motion Program0
05	JogMinus Axis1	21	Execute Motion Program 1
06	Reset Strobe Flag Axis1	22	Execute Motion Program 2
07	reserved	23	Execute Motion Program 3
08	AbortAllMoves Axis2	24	Execute Motion Program 4
09	Feedhold Axis2	25	Execute Motion Program 5
10	EnableDrive Axis2	26	Execute Motion Program 6
11	FindHome Axis2	27	Execute Motion Program 7
12	JogPlus Axis2	28	Execute Motion Program 8
13	JogMinus Axis2	29	Execute Motion Program 9
14	Reset Strobe Flag Axis2	30	Execute Motion Program 10
15	reserved	31	Clear Error

* The bit numbers represent an offset to the starting address for %Q references.

Abort All Moves. This command causes *any* motion in progress to halt at the current *Jog Acceleration* rate. Any pending programmed or immediate command is canceled and therefore not allowed to become effective. The abort condition is in effect as long as this command is on. 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.

Feedhold (On Transition). This command causes any 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. After an ON transition, program motion will stop, even if the command bit transitions back OFF before motion stops.

Feedhold (Off Transition). This command causes any programmed motion interrupted by *Feedhold* 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 interrupted *Move* command will resume from where the axis was left after the *Jog*. The *Move* finishes at the correct programmed velocity and continues to the original programmed position as if no *jog* displacement occurred.

Enable Drive. 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 servo motion can be commanded. *Enable Drive* must be maintained ON to allow normal servo motion (except when using *Jog* commands).

Find Home. This command causes the controller module 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 indicates the exact position. A

configuration option allows the Home Limit Switch to be ignored. The configured *Home Offset* defines the location of *Home Position* as the offset distance from the Home Marker. Position Valid indication is set at the conclusion of the Home cycle. See Appendix D for considerations when using absolute encoder.

Jog Plus. When this command bit is ON, the axis moves in the positive direction at the configured *Jog Acceleration* and *Jog Velocity* as long as the *Jog Plus* command is maintained and the configured *Positive End Of Travel* limit is not encountered. *Jog Plus* may be used to jog off of the Negative End of Travel Limit Switch if the %Q Clear Error bit is also maintained on.

Jog Minus. When this command bit is ON, the axis moves in the negative direction at the configured *Jog Acceleration* and *Jog Velocity* as long as the *Jog Minus* command is maintained and the configured *Negative End Of Travel* limit is not encountered. *Jog Minus* may be used to jog off of the Positive End of Travel Limit Switch if the %Q Clear Error bit is also maintained on.

Reset Strobe Flag. The *Position Strobe %I* status bit flag 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 %Q* command bit to clear the *Position Strobe %I* status bit flag. Once the *Position Strobe %I* 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 %Q* command bit is set, the *Position Strobe* bit flag 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.

CTL09-CTL12 Output Controls. These command bits may be tested by the controller module during execution of *Wait* or *Conditional Jump* commands.

Execute Motion Program 0 - 10. These commands are used to select stored programs for immediate execution. Each command uses a one shot action; thus a command bit must transition from OFF to ON each time a program is to be executed. Programs may be temporarily interrupted by a Feedhold command.

When a program begins execution, Rate Override is always set to 100%. A *Rate Override %AQ* immediate command can be sent on the same sweep as the *Execute Program %Q* bit and will be effective as the program starts.

Only one Motion Program can be executed at a time per axis; the *Program Active %I* status bit must be OFF or Motion Program execution will not be allowed to start. A multi-axis Motion Program uses both axis 1 and axis 2, so both *Program Active* bits must be OFF to start a multi-axis Motion Program.

Clear Error. 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.

Controller Module %AQ Immediate Commands (PM-APM)

Each PLC sweep, six words of data are automatically transferred from the CPU %AQ data to the controller module. These six words are used to send Immediate Commands from the PLC to the controller module. The first three words, offsets 0 through 2, are dedicated to axis 1 of the Power Mate J. The second three words, offsets 3 through 5, are dedicated to axis 2 of the Power Mate J. Thus one command may be sent to each axis of the Power Mate J every sweep.

The only exception is the Load Parameter Immediate command which is axis independent. This command may be sent using either or both sets of three words. Thus two Load Parameter Immediate commands can be sent on the same sweep (one in the first three %AQ words and the other in the second three %AQ words).

In a 1-axis Power Mate J, the only effective commands for the second set of %AQ words are Force D/A Output (Analog Output) and Load Parameter Immediate.

Even though the commands are sent each sweep, the controller module will act on a command ONLY if it changed since the last sweep. When any of the 6-byte data changes, the controller module will accept the data as a new command and respond accordingly.

The 6-byte format for the Immediate Commands is defined in Table 2-7. The actual addresses of the Immediate Command Words depend on the starting address configured for the %AQ references. **The word numbers listed in the following table are offsets to this starting address.**

The word offsets are shown in reverse order and in hexadecimal to simplify the data entry. The following example sends the Set Position command to axis 1. The first word, word 0, contains the actual command number. For the Set Position command, the command number is 0023h. The second and third words contain the data for the Set Position command which is a position. The second word, word 1, is the least significant word of the position and the third word, word 2, is the most significant word. To set a position of 3,400,250, first convert the value to hexadecimal. 3,400,250 decimal equals 0033E23A hexadecimal. For this value, 0033 is the most significant word and E23A is the least significant word. The data to be sent to the controller module would be:

Word 2	Word 1	Word 0	Command
0033	E23A	0023	Set Position 3,400,250

Setting up word 0 as a hexadecimal word and words 1 and 2 as a double integer in the PLC will simplify immediate command entry.

The following table applies to both the 1-Axis and the 2-Axis Power Mate J

Table 3-7. Immediate Commands Using the 6-Byte Format

Word 2		Word 1		Word 0 *		Immediate Command Definition
Word 5		Word 4		Word 3 *		
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	
xx	xx	xx	xx	00	00h	Null
xx	xx	xx	RO%	00	20h	Rate Override RO% = 0 ... 120%
xx	xx	**	Incr	00	21h	Position Increment Without Position Update Incr. = -128 ... +127 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 = -4,095 ... +4,095 (RPM Digital) D/A Output = -32,000 ... + 32,000 (Analog)
xx	xx	**	Incr.	00	25h	Position Increment With Position Update Incr. = -128 ... +127 User Units
xx	xx	In Position Zone		00	26h	In Position Zone Range = 0 ... 2000
Data				Move Type	27h	Move Command
Velocity				00	28h	Jog Velocity Vel. = +1 ... + 8,388,607 User Units/sec
Acceleration				00	29h	Jog Acceleration Acc. = +1 ... + 8,388,607 User Units/sec/sec
xx	xx	Time Constant		00	2Ah	Position Loop Time Constant Time Constant = 0,5 ... 10000
xx	xx	xx	VFF%	00	2Bh	Velocity Feedforward VFF% = 0 ... 100%
xx	xx	xx	VLGN	00	2Eh	Velocity Loop Gain VLGN = 0 ... 255
xx	xx	xx	Torque Limit %	00	2Fh	Torque Limit Range = 1-100%
xx	xx	Offset		Mode	40h	Select Return Data
xx	MCON CTRL	xx	xx	00	45h	Configuration Modes
Feedback Offset (counts)				xx	46h	Absolute Feedback Offset
xx	xx	xx	xx	00	49h	Set Configuration Complete
xx	xx	xx	xx	00	4Ah	Update Flash Memory
Parameter Data				Par #h	50h	Load Parameter Immediate Par # = 0 ... 255 Parameter Data = Range depends on parameter usage.

* = The word numbers represent an offset to the starting address for the %AQ references.

** = Only 00 or FFh are acceptable.

xx = don't care

Null. This is the default %AQ Immediate command. Since the %AQ words are automatically transferred each PLC sweep, the *Null* command should always be used to avoid inadvertent execution of another Immediate command.

Rate Override. 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 controller module. It is stored and will remain effective until overwritten by a different value. A rate override has no effect on non-programmed motion. Feedrate is set to 100% whenever a program is initiated.

Position Increment Without Position Update. (user units) This command offsets the axis motion from -128 to +127 user units without updating the Actual or Commanded Position. The controller module will immediately move the axis by the increment commanded if the servo is enabled.

Move At Velocity. (user units/sec) This command is executed from the PLC to move the axis at a constant velocity. The configured *Jog Acceleration* rate is used for *Move at Velocity* commands. Axis position data will roll over at the configured Hi or Lo Limit when reached during these moves.

Set Position. (user units) This command changes the axis position register values 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*. *Set Position* cannot be performed when the %I *Moving* bit or the %I *Program Active* bit is ON. The position value must be within the End of Travel Limits and Count Limits or a status error will be reported. *Position Valid* %I flag is set after a successful *Set Position* command. See Appendix D for considerations when using absolute mode encoders.

Note

When a servo system is first powered up after removal or replacement of the encoder battery, the digital encoder must be rotated past its internal reference point. If this is not done the *Set Position* command will be ignored and Error Code 53h (Set Position before encoder passes reference point) will be reported.

Force D/A Output. This command forces a Velocity Command to the DSI. Acceleration control is not used and changes in velocity take effect immediately. A *D/A Output* value of +4095 will produce a motor output of + 4,095 RPM and -4095 will produce a motor output of -4,095 RPM. The DSI may limit actual motor speed to a lower value. Care should be taken not to operate a servo motor past its rated duty cycle.

The *Enable Drive* %Q bit must be active with no other motion commanded for the Force D/A Output command to operate. Force D/A output is the only continuous %AQ immediate command. It must remain continuously in the %AQ data for proper operation. Thus any other %AQ immediate command will remove the Force D/A command.

For the single axis Power Mate J only, a *Force D/A Output* immediate command and a *D/A Output* value in the %AQ Words 3-5, will operate the analog output on I/O connector A. A *D/A Output* value of +32000 will produce +10.00 Vdc and a *D/A Output* value of -32000 will produce -10.00 Vdc.

Position Increment With Position Update. (user units) This command is similar to the Position Increment without position update command (#21h) except that Actual and Commanded Position (returned in %AI data) are both updated by the increment value. If the servo is enabled, the controller module will immediately move the axis by the increment value.

In Position Zone. This command can be used to set the active *In Position Zone* to a value different than the configured value.

The *In Position Zone* is used by the controller module to determine when a PMOVE is complete and also when the axis motion (feedback position) is close enough to the commanded position to allow position critical operations (such as *Set Position*) to take place. The *In Zone* %I bit is set to indicate this.

If the controller module is power cycled or the PLC CPU is reset for any reason, the value set by this command will be lost and the *In Position* zone value set by configuration software will be reinstated.

Move Command. This command will produce a single move to the commanded position each time it is sent. The current Jog acceleration and velocity (which can also be changed by %AQ commands) will be used for the move.

The data field for this command may contain the move position or distance in bytes 2-5 with the command type (in hexadecimal format) as defined below:

Move Type (byte 1):

00h = Abs, Pmove, Linear
 01h = Abs, Cmove, Linear
 10h = Abs, Pmove, S
 11h = Abs, Cmove, S
 40h = Inc, Pmove, Linear
 41h = Inc, Cmove, Linear
 50h = Inc, Pmove, S
 51h = Inc, Cmove, S

The data field for this command may contain a parameter number in byte 2 (bytes 3-5 unused) with the command type as defined below:

Move Type (byte 1):

80h = Abs, Pmove, Linear
 81h = Abs, Cmove, Linear
 90h = Abs, Pmove, S
 91h = Abs, Cmove, S
 C0h = Inc, Pmove, Linear
 C1h = Inc, Cmove, Linear
 D0h = Inc, Pmove, S
 D1h = Inc, Cmove, S

The *Move Command* is executed as a single move motion program, therefore, all the restrictions that apply to motion program execution will also apply to it. For example, if a program is already active for axis 1, then an attempt to send this command for axis 1 will result in an error condition reported.

Jog Velocity. (user units/sec) This command sets the velocity used when a *Jog* %Q bit is used to jog in the positive or negative direction. A PLC reset or power cycle returns this value to the configured data.

Jog Acceleration. (user units/sec/sec) This command sets the acceleration value used by *Jog*, *Move at Velocity*, a *Home Cycle*, and *Abort All Moves*. A PLC reset or power cycle returns this value to the configured data.

Position Loop Time Constant. (milliseconds) This command allows the servo position loop time constant to be changed from the configured value. The lower the value, the faster the system response. Values which are too low will cause system instability and oscillation. For accurate tracking of the commanded velocity profile, the *Position Loop Time Constant* should be 1/4 to 1/2 of the MINIMUM system deceleration time. The *Vel at 10 V* configuration value must be set correctly for proper operation of the *Position Loop Time Constant* gain factor. A PLC reset or power cycle returns this value to the configured data.

Velocity Feedforward. This command sets the *Velocity Feedforward* gain (percent). It is the percentage of Commanded Velocity that is added to the controller module 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” configuration value must be set correctly for proper operation of the *Velocity Feedforward* gain factor. A PLC reset or power cycle returns this value to the configured data.

Velocity Loop Gain. (VLGN) The velocity control loop gain for an axis may be set with the *Velocity Loop Gain* command. The VLGN value is used to match the load inertia (J_L) to the motor inertia (J_M). VLGN is defined with a value of 16 representing an inertia ratio of 1 to 1. The VLGN value is calculated assuming that the load is rigidly applied to the motor. Therefore, in actual machine adjustment the set value may significantly differ from the calculated value due to rigidity, friction, backlash, and other factors. A PLC reset or power cycle returns this value to the configured data. The user must set the velocity loop gain (VLGN) such that it satisfies the following equation:

$$VLGN = \frac{J_L}{J_M} * 16$$

Where:

JL = Load Inertia

JM = Motor Inertia

VLGN = Velocity Control Loop Gain (0 - 255)

For example: The rotor inertia (J_M) of a particular servo is 0.10 lb-in-s². The load inertial (J_L) in this application is 0.05 lb-in-s². VLGN = (0.05 / 0.10) * 16 = 8

The default *Velocity Loop Gain* is set using the *Integrator Time Constant* setting in the Configuration Software. Since the Integrator is unused by the controller module in Digital mode, the Integrator Time Constant data is used to set the initial *Velocity Loop Gain*.

Caution

Changing the VLGN value may cause an axis to be unstable and care should be used when making any change to the VLGN value.

Select Return Data. This command allows alternate data to be reported in the %AI Commanded Position location for each axis. The alternate data includes information such as Parameter Register contents, controller module's Firmware Revision and interface module's Firmware Revision.

The *Select Return Data* command uses a mode selection and an offset selection. The mode selection (byte offset +1 of the six byte command) determines the Return Data type. The offset selection (byte offsets +2, +3 of six byte command) selects an individual data item for some modes. Setting the mode to 00h causes the default Commanded Position to be reported. The following Return Data selections are allowed:

<u>Selected Return Data</u>	<u>Data Mode</u>	<u>Data Offset</u>
Commanded Position	00h	not used
Absolute Feedback Offset(cts)	07h	not used
Parameter Register	08h	Parameter Number (0-255)
APM Firmware Revision	10h	not used
DSI Firmware Revision	11h	not used

Absolute Feedback Offset is the position offset that is used to initialize Actual Position when Absolute Encoder is used. Actual Position = Absolute Encoder Data + Absolute Feedback Offset.

Controller and interface module Firmware Revisions should be interpreted as two separate words for major-minor revision codes. At least three PLC sweeps or 20 milliseconds (whichever represents more time) must elapse before the new Selected Return Data is available in the PLC.

Torque Limit (TRLMT). The *Torque Limit Command* provides a method of limiting the torque produced by a digital servo motor on each axis. The controller module will set the torque limit (TRLMT) at the default 100% whenever a power cycle or reset occurs. The PLC application logic must set any other value for desired TRLMT. The valid range for TRLMT is 1 to 100 % in units of 1%. TRLMT can be changed during axis motion. Refer to the appropriate servo motor manual for the actual motor torque curve to determine the actual value of torque output at a given velocity.

Configuration Modes. This configuration command permits various configuration changes to the controller module by modifying the byte in the command string associated with the desired feature. The values established by the *Configuration Mode* command are not saved through a power cycle or reset and must be re-established on system startup. Use of the *Set Configuration Complete* command is recommended, with appropriate application logic to enable the PLC verification of desired modes.

MCON CTRL has the following possible values:

- MCON CTRL = 00h = Digital Servo Magnetic Contactor (MCON) NOT controlled by CTL Output (Default).
- MCON CTRL = 01h = Digital Servo Magnetic Contactor (MCON) is controlled by CTL Output.

With the default mode of operation the %Q *Enable Drive* command On will energize the digital servo power relay (MCON) and enable servo power to the amplifier. The *Enable Drive* %Q Off will cause the MCON and digital servo power to be removed after the configured *Servo Drive Disable Delay (DisDly)*.

With MCON CTRL (byte 4) set to 01h, the %Q *Enable Drive ON/OFF* command Enables and Disables servo power to the motor. The motor shaft will have no torque when the

servo power is Off. The MCON relay will remain energized. This function is useful when the torque must be removed and reapplied repeatedly from the motor and will help avoid wear on the MCON relay.

With MCON CTL set to 01h, a %Q CTL output will control the ON/OFF state of the magnetic contactor (MCON).

CTL11 is the Axis 1 MCON control output. CTL12 is the Axis 2 MCON control output for the two axis Power Mate J.

Absolute Feedback Offset. This command updates the Absolute Feedback Offset (counts) used to initialize Actual Position from an AbsoluteDigital Encoder. To permanently save the encoder offset entered with this command, the *Update Flash Memory* %AQ command must also be used. Normally the Absolute Feedback Offset is calculated automatically by the *Set Position* or *Find Home* commands. Refer to Appendix D, *Serial Encoders*, for additional information.

Set Configuration Complete. This command sets the *Configuration Complete* %I bit. Once set, the *Configuration Complete* bit is only cleared when the PLC resets or reconfigures the controller module. The PLC can monitor the bit to determine if it must resend other %AQ commands, such as *In Position Zone* or *Jog Acceleration*. This would only be necessary if the %AQ commands were used to override controller module configuration data programmed with the PLC configuration software.

Update Flash Memory. This command copies important RAM data including Motion Programs and the latest Absolute Feedback Offset into Flash memory. Normally the *Update Flash Memory* command is sent after a *Set Position* or *Find Home* operation when an Absolute Digital Encoder is used. Updating the Flash memory permanently saves the Absolute Feedback Offset so that Actual Position can be restored correctly after a system loss of power. Refer to Appendix D, *Serial Encoders*, for additional information.

Load Parameter Immediate. This command is executed from the PLC to immediately change a controller module data parameter value. Data parameters are only used by motion programs. A command for each parameter change is required.

Section 2: Interface Module %I Status Bits (DSI)

The DSI will return 16 bits of status information to configured PLC %I references via the normal input scan of the Series 90-30 CPU. The meaning of these status bits is shown in the table below.

Table 3-8. DSI Status Bits

PLC Status Bits (%I)			
Bit Number	Description	Bit Number	Description
00	Reserved	08	Reserved
01	Axis 1 Servo Enabled	09	Axis 2 Servo Enabled
02	Axis 1 MCON Enabled	10	Axis 2 MCON Enabled
03	Reserved	11	Reserved
04	Axis 1 Error	12	Axis 2 Error
05	Reserved	13	Reserved
06	Reserved	14	Reserved
07	Controller Comm OK	15	DSI Board OK or Interface Module OK

Servo Enabled. This bit is set whenever the servo amplifier power inverter is on. The motor shaft will be uncontrolled and may freely rotate whenever this bit is off. Consult appropriate amplifier manual for further details.

MCON Enabled. This bit is set whenever the magnetic contactor (MCON) is commanded on. When this bit is off hysteresis braking is applied to the motor. Consult the appropriate amplifier manual for further details.

Axis Error. This bit is set whenever an amplifier or serial encoder problem is detected. The specific error will be displayed on the DSI visual indicator and will also be reported to the PLC in the controller module %AI Status Word.

Controller Comm OK. Bit 7, *Controller Comm OK* is always set to a 1 *except* when a communications failure within the controller module has been detected. During the power up sequence, while the DSI is waiting for the controller module to communicate for the first time, this bit will be initialized to a 1 until the power-up wait time has been exhausted. This will avoid false reporting of a communications error while the DSI waits on its initial communication from the controller module.

DSI Board OK or Interface Module OK. Bit 15, *DSI Board OK or Interface Module OK*, will always be set to a 1 *except* when a fatal system alarm has been detected.

Chapter 4

Motion Control

This chapter provides practical information on Power Mate J Motion Control with a number of examples.

The main topics discussed are:

- Position Feedback Types
- Non-Programmed Motion
- Programmed Motion

Position Feedback Types

The initial release of the Power Mate J supports only digital serial encoders used with the a Series or b Series digital servo. Encoder resolution is fixed at 8192 encoder counts per motor shaft revolution. Incremental or Absolute (battery backed) encoders may be configured. Refer to Appendix D for additional information about absolute encoder usage.

Non-Programmed Motion

The Power Mate J can generate motion in an axis in one of five ways without the use of any motion programs.

- *Find Home* and *Jog* use the %Q bits to command motion.
- *Move at Velocity*, *Force D/A Output*, and *Position Increment* use %AQ immediate commands.

During *Jog*, *Find Home*, *Move at Velocity*, and *Force D/A Output*, any other commanded motion, programmed or non-programmed, will generate an error. The only exception is *Position Increment* which can be commanded any time. See the description of *Position Increment* motion for more details.

Non-programmed motions (Abort, Jog, Move at Velocity) use the JOG acceleration and acceleration mode. *FEEDHOLD* uses the programmed acceleration and acceleration mode.

Power Mate J Home Cycle

A Home Cycle establishes the Home Position for systems with a serial pulse encoder used in incremental mode. The configured Home Offset defines the location of Home Position as the offset distance from the Home Marker.

The *Enable Drive %Q bit and Drive Enabled %I* bit must be ON during an entire Home Cycle. However, the *Find Home %Q* bit does not need to be held ON during the cycle; it may be turned on momentarily with a one-shot. Note that turning ON the *Find Home* bit immediately turns OFF the *Position Valid %I* bit until the end of the Home Cycle. The *Abort All Moves %Q* bit halts a Home Cycle, but the *Position Valid* bit does not turn back ON. No motion programs can be executed unless the *Position Valid* bit is ON.

Home Switch Mode

If the Find Home Mode is configured as Home Switch, the **Home Switch** input from the I/O connector is used to roughly indicate the reference position for Home. The next marker encountered when traveling in the negative direction indicates the exact location. An open **Home Switch** indicates the servo is on the positive side of the **Home Switch**. An OFF to ON transition of the *Find Home* command yields the following Home Cycle. Unless otherwise specified, acceleration is at the current *Jog Acceleration* and configured *Jog Acceleration Mode*. (If initiated from a position on the positive side of the **Home Switch**, the cycle starts with step 1, otherwise the cycle starts with step 3)

1. The axis is moved in the negative direction at the configured *Find Home Velocity* until the **Home Switch** closes.
2. The axis decelerates and is stopped.
3. The axis is accelerated in the positive direction and moved at the configured *Find Home Velocity* until the **Home Switch** opens.
4. The axis decelerates and is stopped.
5. The axis is accelerated in the negative direction and moved at the configured *Find Home Velocity* until the **Home Switch** closes.
6. The Power Mate J continues negative motion at the configured *Final Home Velocity* until a marker pulse is sensed. The marker establishes the Home reference position.

7. The axis decelerates and is stopped.
8. The axis is moved, at the current *Jog Velocity*, the number of user units specified by the *Home Offset* value from the Home reference position.
9. The axis decelerates and is stopped.
10. An internal *Set Position* sets the *Commanded* and *Actual* Positions to the configured *Home Position* value. Finally, the Power Mate J sets the *Position Valid %I* bit to indicate the Home Cycle is complete.

Move+ and Move- Modes

If *Find Home Mode* is configured as Move+ or Move-, the first marker pulse encountered when moving in the appropriate direction (positive for Move+, negative for Move-) is used to establish the exact location. An OFF to ON transition of the Find Home %Q bit will perform the following operation.

1. The axis is accelerated at the *Jog Acceleration rate* and moved at the configured *Final Home Velocity* (positive direction for Move+, negative direction for Move-) until a marker pulse is sensed. This marker pulse establishes the Home reference position.
2. The axis is stopped at the configured *Jog Acceleration rate* and with the configured *Jog Acceleration Mode*.
3. The axis is moved, at the configured *Jog Velocity* and with the configured *Jog Acceleration rate* and *Jog Acceleration Mode*, the number of user units specified by the *Home Offset* value from the Home reference position.
4. The axis is stopped at the configured *Jog Acceleration rate* and with the configured *Jog Acceleration Mode*.
5. An internal *Set Position* sets the *Commanded* and *Actual* Positions to the configured *Home Position* value; the Power Mate J sets the *Position Valid %I* bit to indicate the Home Cycle is complete.

Jogging with the Power Mate J

The *Jog Velocity*, *Jog Acceleration*, and *Jog Acceleration Mode* are configurable in the Power Mate J. These values are used whenever a *Jog Plus* or *Jog Minus %Q* bit is turned ON. Note that both bits ON generates no motion. The *Jog Acceleration* and *Jog Acceleration Mode* are also used during a *Find Home Cycle* and when a *Move at Velocity* immediate command is performed. Programmed motions use the *Jog Velocity* and *Jog Acceleration* as defaults.

A *Jog* can be performed when no other motion is commanded, or while programmed motion is temporarily halted due to a Feedhold command. The *Enable Drive %Q* bit does not need to be ON to Jog, but it can be ON. Turning on a *Jog %Q* bit will automatically close the Enable Relay, and turn on the *Drive Enabled %I* bit. When an **Overtravel Limit Switch** is OFF, *Jog* and *Clear Error* may be turned on simultaneously to move away from the open Limit Switch. Thus a *Jog Plus* will not work while the **Positive End of Travel Switch** is open and *Jog Minus* will not work while the **Negative End of Travel Switch** is open.

Move at Velocity Command

A *Move at Velocity* command is generated by placing the value 22h in the first word of %AQ data assigned to an axis. The second and third words together represent a signed 32 bit velocity. Note that the third word is the most significant word of the velocity. Once the command is given, the %AQ data can be cleared by sending a NULL command, or changed as desired. *Move at Velocity* will not function unless the servo drive is enabled (%Q *Enable Drive* and %I *Drive Enabled* are set).

The listing of %AQ immediate commands shows the words in reverse order to make understanding easier. For example, to command a velocity of 512 user units per second in an Power Mate J configured with %AQ data starting at %AQ1, the following values should be used: 0022h (34 decimal) in %AQ1, 0200h (512 decimal) in %AQ2, and 0 in %AQ3. When the Power Mate J receives these values, if *Drive Enabled %I* is ON, *Abort All Moves %Q* is OFF, and no other motion is commanded it will begin moving the axis at 512 user units per second in the positive direction using the current Jog Acceleration and Acceleration Mode.

The *Drive Enabled %I* bit must be ON before the Power Mate J receives the immediate command or an error will occur. Also, if a *Move at Velocity* command is already in the %AQ data, the velocity value must change while the *Drive Enabled* bit is ON for the Power Mate J to accept it. The Power Mate J detects a *Move at Velocity* command when the %AQ values change.

When the Power Mate J is performing a *Move at Velocity*, it ignores the *Software End of Travel Limits*. Hardware Overtravel Limits must be ON if they are enabled.

A *Move at Velocity* can be stopped without causing an error in two ways: a *Move at Velocity* command with a velocity of zero, or turning the *Abort All Moves %Q* bit ON for at least one PLC sweep.

Force D/A Command

The *Force D/A* command is the only command that must be maintained in the %AQ data for proper operation. If any other immediate command is sent to the Power Mate J, the *Force D/A* operation will end. A one-shot *Force D/A* command will operate only during the sweep in which it appears. Refer to Chapter 2 for details on use of this command.

Position Increment Commands

To generate small corrections between the axis position and the Power Mate J tracking, the *Position Increment* commands can be used to offset Actual Position by a specific number of user units. If the *Drive Enabled %I* bit ON, the axis will immediately move the increment amount. If the position increment without position update is used (%AQ command 21h), the *Actual Position* reported by the Power Mate J will remain unchanged. If the position increment with position update is used (%AQ command 25h), the *Actual Position* and *Commanded Position* reported by the Power Mate J will be changed by the increment. *Position Increment* can be used at any time, though simultaneous use with the *Force D/A* command is impossible because the *Force D/A* command must continuously appear in the %AQ data.

Other Considerations

Other considerations when using non-programmed motion are as follows:

- The *Abort All Moves* bit when ON will prevent any non-programmed motion from starting.
- Turning ON the *Abort All Moves* bit will immediately stop any current non-programmed motion at the current *Jog Acceleration*.
- A *Set Position* command during non-programmed motion will cause a status error.
- Turning OFF the *Enable Drive* bit while performing a Home Cycle or *Move at Velocity* will cause a stop error.
- The *Feedhold* bit has no effect on non-programmed motion.
- The *Rate Override* command has no effect on non-programmed motion.
- Changing the *Jog Velocity* or *Jog Acceleration* will not affect moves in progress.

Programmed Motion

The Power Mate J executes program motion commands sequentially in a block-by-block fashion once a program is selected. The program commands can be categorized as follows:

Type 1 Commands

- Call Subroutine
- Jump

Type 2 Commands

- Block #
- Null
- Acceleration
- Velocity

Type 3 Commands

- Positioning Move
- Continuous Move
- Dwell
- Wait
- End of Program

Type 1 commands can redirect the program path execution, but do not directly affect positioning. Call Subroutine executes a subroutine before returning execution to the next command. Jump either continues execution at another location, or it tests CTL bits and, based on the bit condition, may or may not alter the program path.

Type 2 commands also do not affect position. The Block # command provides an identification or label for the following Type 3 command. If no Block # is found in the current program block, the previous Block # is used. The Velocity and Acceleration commands specify velocity and acceleration rates for motion of the following Type 3 command.

Type 3 commands start or stop motion and thus affect positioning control. Positioning and Continuous moves command motion; Dwell, Wait, and End of Program stop motion.

A program block consists of one (and only one) Type 3 command with any number and combination of preceding Type 1 and 2 commands. Type 2 commands are optional; a program block could contain a single Type 3 command. Type 2 commands, and Conditional Jumps, do not take effect until the following Type 3 command is executed.

While the Power Mate J is executing one program block, the following program block is processed into a buffer command area to minimize the transition time from one block to another. Thus parameters used in a move must be loaded before the move two blocks previous completes execution.

When a 2-axis Power Mate J is executing a 2-Axis program, the program commands are scanned independently by each axis and only the data designated for that axis is executed. Note that some commands do not specify an axis (Block #, Jump, Call, and End) and therefore will apply to both axes.

A 2-Axis program can contain Sync Block # commands to synchronize the two motions at designated points. When the first axis reaches the Sync Block, it will not execute the next block until the other axis has also reached the Sync Block. Refer to Example 18 for an illustration of this type of operation. Several aspects of programmed motion are discussed below.

Prerequisites for Programmed Motion

The following conditions must be satisfied before a motion program can be initiated (for a multi-axis program, the conditions must be met for BOTH axis 1 and axis 2):

- The Enable Drive %Q bit must be ON
- The *Drive Enabled* %I bit must be ON
- The *Position Valid* %I bit must be ON
- The *Moving* %I bit must be OFF
- The *Program Active* %I bit must be OFF
- The *Abort All Moves* %Q bit must be OFF
- The axis position must be within the configured End of Travel Limits
- The Overtravel Limit Switches must be asserted if enabled
- A Force D/A command must not be active
- The program to be executed must be a valid program stored in the Power Mate J

Conditions Which Stop a Motion Program

A motion program will immediately cease when one of the following conditions occurs:

- The *Abort All Moves* %Q bit turns ON
- The *Enable Drive* %Q bit turns OFF
- An **OvertravelLimitSwitch** turns OFF when OT Limits are configured to be enabled.
- The next programmed move, either PMOVE or CMOVE, will pass a Software End of Travel Limit
- A Stop Error occurs. See Appendix A (Error Codes)

Parameters for Programmed Moves

Programmed moves have three parameters:

1. The *distance* to move or *position* to move to,
2. The *type of positioning reference* to use for the move, and
3. The *type of acceleration* to use while performing the move.

Types of Positioning Reference

The choices for the type of reference to use for the move are ABSOLUTE and INCREMENTAL. This reference determines how the first parameter, the distance to move or position to move to, is interpreted.

Absolute Positioning

In an absolute positioning move, the first parameter is the position to move to. The following is an example of an absolute positioning move.

```
PMOVE 5000, ABS, LINEAR
```

This move will move the axis from its current position, whatever it may be, to the position 5000. Thus the actual distance moved depends upon where the axis is when the move is encountered. If the initial position was 0, the axis would move 5000 user units in the positive direction. If the initial position was 10000, the axis would move 5000 user units in the negative direction. And, if the initial position was 5000, no motion would be generated.

Incremental Positioning

In an incremental move, the first parameter is interpreted as the distance to move from the position where the move begins. The Power Mate J translates incremental move distances into absolute move positions so no error accumulates. The following is an example of an incremental positioning move.

```
PMOVE 5000, INC, LINEAR
```

This incremental move will move the axis from its current position to a position 5000 user units greater. With an incremental move, the first parameter specifies the actual number of user units the axis moves.

Types of Acceleration

Choices for the last parameter, which specifies the type of acceleration to use while performing the move, are LINEAR and S-CURVE.

Linear Acceleration

A sample profile of a linear move plotting velocity versus time is shown in Figure 4-1. The straight lines on the graph show that a linear move uses constant acceleration. The area under the graph is the distance moved.

```
ACCEL      1000  
VELOC      2000  
PMOVE     6000, INC, LINEAR
```

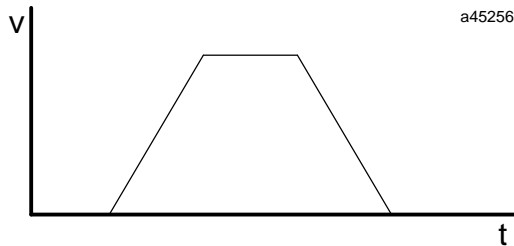


Figure 4-1. Sample Linear Motion

S-Curve Acceleration

An s-curve motion sample, again plotting velocity versus time, is shown below. The curved lines on the graph indicate that the acceleration was not constant. When the move begins, the acceleration starts slowly and builds until it reaches the programmed acceleration. This should be the midpoint of the acceleration. Then, the acceleration begins decreasing until it is zero, at which time the programmed velocity has been reached. An s-curve move requires twice the time and distance to accelerate and decelerate that a linear move needs if the acceleration is the same. The area under the velocity vs time graph is also the distance moved.

ACCEL 2000
VELOC 2000
PMOVE 8000, INC, SCURVE

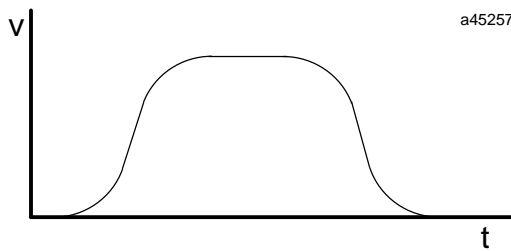


Figure 4-2. Sample S-Curve Motion

Types of Programmed Move Commands

Positioning Move (PMOVE)

A PMOVE uses the most recently programmed velocity and acceleration. If a VELOC command has not been encountered in the motion program, the *Jog Velocity* is used as a default. If an ACCEL command has not been encountered in the motion program, the *Jog Acceleration* is used as a default.

A PMOVE will always stop when it is completed to allow the *IN ZONE %I* bit to turn ON.

Continuous Move (CMOVE)

A CMOVE uses the most recently programmed velocity and acceleration. If a VELOC command has not been encountered in the motion program, the *Jog Velocity* is used as a default. If an ACCEL command has not been encountered in the motion program, the *Jog Acceleration* is used as a default.

A CMOVE does not stop when completed unless it is followed by a DWELL or a WAIT, the next programmed velocity is zero, or it is the last program command. It does not wait for the position to be *IN ZONE* before going to the next move. A CMOVE reaches its programmed position at the same time it reaches the velocity of the following Move command.

A special form of the CMOVE command can be used to force the Power Mate J to reach the programmed CMOVE position *before* starting the velocity change associated with the next move command (that is, execute the entire CMOVE command at a constant velocity). Programming an incremental CMOVE command with an operand of 0 (CMOVE INC 0) will force a delay in the servo velocity change for the next move command in sequence. The following sequence of commands illustrates this effect:

Command	Data	Comments
VELOC	10000	Set velocity of first move = 10000
CMOVE	15000, ABS, LINEAR	Reach velocity of second move at position = 15000
VELOC	20000	Set velocity of second move = 20000
CMOVE	0, INC, LINEAR	Force next velocity change to wait for next move command
CMOVE	30000, ABS, LINEAR	Stay at velocity = 20000 until position = 30000, then change to velocity = 5000
VELOC	5000	Set velocity of third move = 5000
PMOVE	45000, ABS, LINEAR	Final stop position = 45000

Programmed Moves

By combining CMOVES and PMOVES, absolute and incremental moves, and linear and s-curve motion, virtually any motion profile can be generated. The following examples show some simple motion profiles, as well as some cases of incorrect motion programming.

Example 1: Combining PMOVES and CMOVES

This example shows how simple PMOVES and CMOVES combine to form motion profiles.

```

ACCEL      1000
VELOC     2000
PMOVE     5000, ABS, LINEAR
VELOC     1200
PMOVE     10000, ABS, SCURVE
ACCEL     1500
VELOC     2800
CMOVE     6000, INC, LINEAR
VELOC     1200
CMOVE     23000, ABS, SCURVE
ACCEL     1000
VELOC     2800
PMOVE     5000, INC, LINEAR
    
```

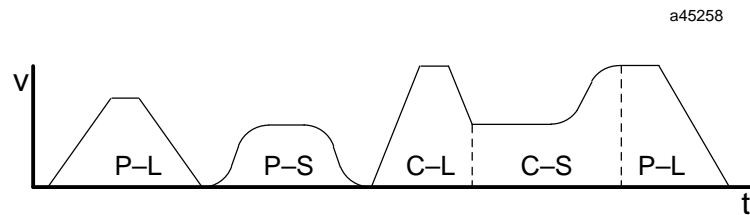


Figure 4-3. Combining PMOVES and CMOVES

The move types are indicated under the corresponding move; for example, P-L indicates linear PMOVE.

The first PMOVE accelerates to programmed velocity, moves for a distance, and decelerates to a stop. This is because motion stops after all PMOVES. When the first move stops, it is at the programmed distance.

The second move is an s-curve PMOVE. It, like the first, accelerates to the programmed velocity, moves for a time, and decelerates to zero velocity because it is a PMOVE.

The next move is a linear CMOVE. It accelerates to programmed velocity, moves for a time, and then decelerates to a lower velocity using linear acceleration. When a CMOVE ends, it will be at the programmed position of the move just completed, and at the velocity of the next move. Thus when the fourth move begins, it is already at its programmed velocity.

The fourth move is a CMOVE, so as it approaches its final position, it accelerates to be at the velocity of the fifth move when it completes. The graph shows the acceleration of the fourth move is s-curve.

Finally, the fifth move begins and moves at its programmed velocity for a time until it decelerates to zero. Any subsequent moves after the fifth would begin at zero velocity because the fifth move is a PMOVE.

Example 2: Changing the Acceleration Mode During a Profile

The following example shows how a different acceleration, and an even acceleration mode, can be used during a profile using CMOVEs. The first CMOVE accelerates linearly to the programmed velocity. Because the second CMOVE's velocity is identical to the first, the first CMOVE finishes its move without changing velocity. The acceleration of the second move is s-curve as it decelerates to zero velocity.

```

ACCEL      2000
VELOC      6000
CMOVE      13000, ABS, LINEAR
ACCEL      4000
CMOVE      15000, INC, SCURVE
    
```

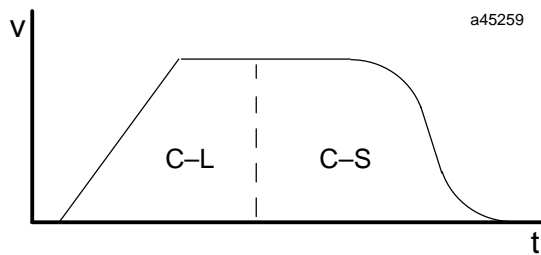


Figure 4-4. Changing the Acceleration Mode During a Profile

Example 3: Not Enough Distance to Reach Programmed Velocity

CMOVES and PMOVES can be programmed that do not have enough distance to reach the programmed velocity. The following graph shows a CMOVE which could not reach the programmed velocity. The Power Mate J accelerates to the point where it must start decelerating to reach the programmed position of C1 at the velocity of the second CMOVE.

```

ACCEL      2000
VELOC      8000
CMOVE      7000, INC, LINEAR
ACCEL      10000
VELOC      2000
CMOVE      4400, INC, LINEAR
    
```

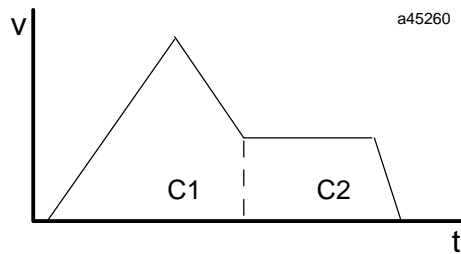


Figure 4-5. Not Enough Distance to Reach Programmed Velocity

Example 4: *Hanging* the Power Mate J When the Distance Runs Out

A serious programming error involves hanging the Power Mate J at a high velocity when the distance runs out. In the following example, the first CMOVE accelerates to a high velocity. The second CMOVE has an identical velocity. However, the distance specified for the second CMOVE is very short. Thus the axis is running at a very high velocity and must stop in a short distance. If the programmed acceleration is not large enough, the following profile could occur. In order to not pass the final position, the Power Mate J instantly commands a zero velocity. This rapid velocity change is undesirable and could cause damage to the controlled machine.

```

ACCEL      500
VELOC     3000
CMOVE     9000, ABS, LINEAR
ACCEL     600
CMOVE     4800, INC, LINEAR
    
```

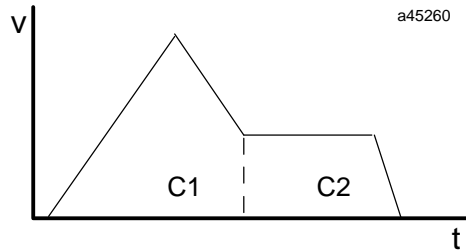


Figure 4-6. *Hanging* the Power Mate J When the Distance Runs Out

Dwell Command

A DWELL command is used to generate no motion for a specified number of milliseconds. A DWELL after a CMOVE will make the CMOVE perform similar to a PMOVE, even if the specified dwell duration is zero.

A DWELL command using a parameter to set the dwell time will be treated as a NULL command if the parameter value is 65000. This feature allows a DWELL-P command between a CMOVE and another Move to be skipped if the DWELL-P value is 65000 (The CMOVE continues to the Move following the DWELL without stopping).

Example 5: Dwell

A simple motion profile, which moved to a specific point, waited, and returned to the original point could use the following program and would have the following velocity profile.

```
ACCEL      30000
VELOC      15000
PMOVE      120000, ABS, LINEAR
DWELL      4000
PMOVE      0, ABS, LINEAR
```

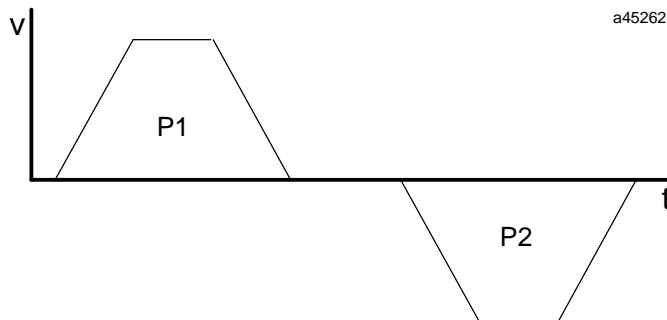


Figure 4-7. Dwell

Wait Command

The WAIT command is similar to the DWELL command; instead of generating no motion for a specified period of time, a WAIT stops program motion and monitors a CTL bit until it is ON. Thus motion will stop any time a WAIT is encountered, even if the CTL bit is on before the WAIT is reached in the program. The trigger to continue the program can be any of the twelve CTL bits.

If, in the previous example WAIT was substituted for DWELL, the motion profile would be the same except the second PMOVE would not start until the CTL bit turned ON. If the CTL bit was ON when the program reached the WAIT, the second PMOVE would begin immediately when the first PMOVE finished.

Also, if WAIT was used instead of DWELL in the previous example, CMOVEs and PMOVEs would generate similar velocity profiles. The WAIT will stop motion whether the previous move is a CMOVE or PMOVE.

Subroutines

The Power Mate J can store up to ten separate programs and forty subroutines. Subroutines can be defined as two types: *single axis* and *multiaxis*. Commands within single axis subroutines do not contain an axis number; this allows single axis subroutines to be called from any single axis program written for either axis 1 or axis 2. Commands within multiaxis subroutines contain axis numbers just like commands within multiaxis programs. multiaxis subroutines can only be called from multiaxis programs or subroutines. Single axis subroutines can only be called from single axis programs or subroutines. On a two axis Power Mate J, a single axis program for axis 1 and a single axis program for axis 2 can call the same single axis subroutine simultaneously.

The CALL command is used to execute a subroutine, with the subroutine number specified as an argument. Program execution continues at the start of the subroutine and resumes at the command after the CALL command when the subroutine finishes. Subroutines can be called from another subroutine, but once a subroutine has been called, it must be completed before it can be called again for the same axis. Thus recursion is not allowed.

Block Numbers and Jumps

Block numbers are used as reference points within a motion program and to control jump testing. A %AI data word displays the current block number which can be monitored to ensure correct program execution or to determine when events should occur. A block number can also be the destination of a JUMP command.

Unconditional Jumps

Jumps are divided into unconditional and conditional. An unconditional jump command simply tells the Power Mate J to continue program execution at the destination block number. An example of an unconditional jump follows.

Example 6: Unconditional Jump

The program executes a PMOVE, dwells for 2 seconds, and then unconditionally jumps back to the beginning of the program at block 1. Thus the PMOVE will be repeated until an *End of Travel Limit* or **Overtravel Limit Switch** is reached.

```

ACCEL      10000
VELOC      30000
BLOCK      1
PMOVE      200000, INC, LINEAR
DWELL      2000
JUMP       UNCOND, 1
    
```

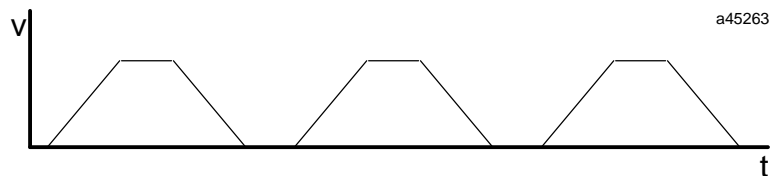


Figure 4-8. Unconditional Jump

Conditional Jumps

A conditional jump is a JUMP command with a CTL bit specified in the command. Conditional jumps are Type 1 commands in that they affect program path execution, but they are also similar to Type 2 commands because they do not take effect until a Type 3 command following the JUMP command is executed. When a conditional JUMP command is executed, the Power Mate J examines the specified CTL bit. If the bit is ON, program execution jumps to the destination BLOCK #; if the bit is OFF, the program continues executing the command after the JUMP. Note that the Type 3 command after the conditional jump and at the jump destination will affect jump behavior.

Conditional JUMP commands should not be used with multiaxis programs containing sync blocks unless the Jump is triggered while both axes are testing the same JUMP command.

Conditional Jump testing **starts** when the next PMOVE, CMOVE, DWELL, WAIT or END Program command following a Conditional JUMP becomes active.

When Conditional Jump testing is active, the designated CTL bit is tested once every millisecond for the 1 axis Power Mate J and once every 2 milliseconds for the 2 axis Power Mate J.

Conditional Jump testing **ends** when the designated CTL bit turns ON (Jump Trigger occurs) or when a new Block Number becomes active.

If more than one Conditional JUMP is programmed without an intervening PMOVE, CMOVE, DWELL, WAIT or END Program command, only the last Conditional JUMP will be recognized.

Motion Program Example:

```

Begin Program
1.  JUMP  CTL01, 2           This JUMP command will be ignored
    JUMP  CTL02, 3           This JUMP command will be recognized
    CMOVE 1, +40000, INC, LINEAR
2.  CMOVE 1, +20000, INC, LINEAR
3.  PMOVE 1, +100000, ABS, LINEAR
4.  DWELL 1, 100
End Program

```

When a new Block Number becomes active AFTER a Conditional JUMP command, Jump testing occurs one more time.

Motion Program Example:

```

Begin Program
1.  CMOVE 1, +20000, ABS, LINEAR
    JUMP  CTL01, 3
2.  PMOVE 1, +40000, ABS, LINEAR           CTL01 tested only once
3.  DWELL 1, 100
End Program

```

In the example above, The CTL01 bit will only be tested once because the PMOVE following the JUMP contains a new Block Number (2).

Changing the location of Block Number 2 will cause the CTL bit to be tested throughout the PMOVE following the JUMP:

Begin Program

1. CMOVE 1, +20000, ABS, LINEAR
2. JUMP CTL01, 3
PMOVE 1, +40000, ABS, LINEAR CTL01 tested throughout PMOVE
3. DWELL 1, 100

End Program

The Power Mate J can perform a Conditional JUMP from an active CMOVE to a program block containing a CMOVE or PMOVE without stopping. **For the axis to jump without stopping, the distance represented by the CMOVE or PMOVE in the Jump block must be greater than the servo stopping distance.** The servo stopping distance is computed using the present commanded velocity and the acceleration parameters that would be in effect when the jump block became active.

The axis will STOP before jumping if a Conditional Jump trigger occurs under any of the following conditions:

- When a PMOVE is active
- When a CMOVE is active and the Jump destination block contains a CMOVE or PMOVE representing motion in the opposite direction.
- When a CMOVE is active and the Jump destination block contains a CMOVE or PMOVE representing motion in the same direction with insufficient distance for the axis to stop.
- When a CMOVE is active and the Jump destination block contains a DWELL, WAIT or END (program) command.

If the axis does STOP before a Conditional Jump, the JOG acceleration and acceleration mode will be used.

Unconditional Jumps do not force the axis to stop before jumping to a new program block. For example, a CMOVE followed by a JUMP Unconditional to another CMOVE will behave just as if the two CMOVES occurred without an intervening Unconditional JUMP.

If Conditional Jump testing is active when the Power Mate J command processor encounters a CALL SUBROUTINE command, the axis will **stop** and terminate jump testing before the CALL is executed.

If Conditional Jump testing is active when the Power Mate J command processor encounters an END SUBROUTINE command, the axis will **stop** and terminate jump testing before the END SUBROUTINE is executed.

Jump Testing

Conditional jumps perform jump testing. If the CTL bit is ON, the jump is immediately performed. If the CTL bit is OFF, the Power Mate J watches the CTL bit and keeps track of the JUMP destination. This monitoring of the CTL bit is called jump testing. If during jump testing the CTL bit turns ON before a BLOCK command, another JUMP command, or a CALL command is encountered, the jump is performed. These commands will end jump testing.

Example 7: Jump Testing

Consider the following two program sections. In the program on the left, the move to position 2000 is completed before jump testing begins. The BLOCK command immediately after the JUMP command ends jump testing. Thus the duration for which the CTL bit is monitored is very short. In the program on the right, however, the JUMP command is encountered before the move command. This starts jump testing before motion begins and jump testing continues as long as the move lasts. If the CTL bit turns ON while the move is being performed, the jump will be performed. After the move completes, the BLOCK command ends jump testing and program execution continues normally. Jump testing would continue during subsequent moves encountered before the BLOCK command.

ACCEL	5000	ACCEL	5000
VELOC	1000	VELOC	1000
BLOCK	1	BLOCK	1
CMOVE	2000, ABS, LINEAR	JUMP	CTL01, 3
JUMP	CTL01, 3	CMOVE	2000, ABS, LINEAR
BLOCK	2	BLOCK	2

Normal Stop Before JUMP

A conditional jump command is similar to Type 2 commands in that jump testing does not start until the Type 3 command immediately after the JUMP is executed. If this Type 3 command would normally stop motion, then motion will stop before jump testing begins. Type 3 commands that will stop motion are: DWELL, WAIT, End of Program, and moves in the opposite direction.

Thus even though the CTL bit may be ON before the block with the conditional JUMP and Type 3 command is executed, axis motion will stop before program execution continues at the jump destination. This stopping is NOT a Jump Stop, which is described in Example 10.

Example 8: Normal Stop Before JUMP

The following example contains a jump followed by a DWELL command. The Power Mate J, because it processes ahead, knows it must stop after the CMOVE command. Thus it comes to a stop before the DWELL is executed. Since jump testing does not begin until the DWELL is executed, testing begins after motion stops. Jump testing ends when the following CMOVE begins because of the BLOCK command associated with it. The dotted lines in the velocity profile indicate when jump testing is taking place. The CTL03 bit does not turn ON during the program.

```

BLOCK      1
ACCEL     5000
VELOC    10000
CMOVE    60000, INC, LINEAR
BLOCK     2
JUMP     CTL03, 4
DWELL    4000
BLOCK     3
ACCEL    10000
VELOC    5000
CMOVE    15000, INC, LINEAR
BLOCK     4
NULL
  
```

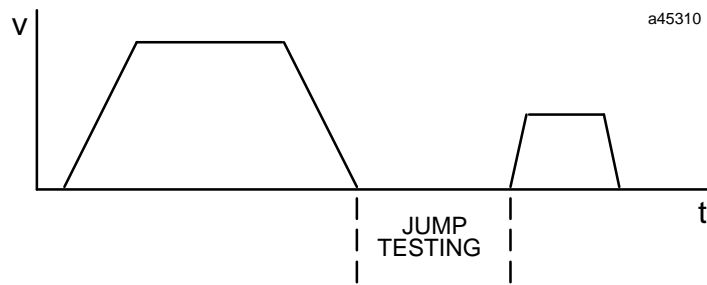


Figure 4-9. Normal Stop Before JUMP

Jumping Without Stopping

If the Type 3 command following a conditional jump is a CMOVE and the Type 3 command at the destination is a move command with sufficient distance to fully decelerate to zero when completed, the jump will be executed without stopping. This is the only way to sustain motion when a jump is performed.

Example 9: JUMP Without Stopping

This is a simple example of a conditional jump from one CMOVE to another. While jump testing the CTL03 bit, the first CMOVE accelerates to the programmed velocity. Before the dotted line, the CTL03 bit is OFF, but at the dotted line the CTL03 bit turns ON. Program execution is immediately transferred to block 3 and the CMOVE there begins. Because the velocity at the jump destination is different, the velocity changes at the acceleration programmed of the jump destination block. Finally, as the second CMOVE completes, velocity is reduced to zero and the program ends.

```

BLOCK      1
ACCEL     2000
VELOC    10000
JUMP     CTL03, 3
CMOVE    120000, INC, LINEAR
BLOCK     3
ACCEL    20000
VELOC    5000
CMOVE    15000, INC, LINEAR
  
```

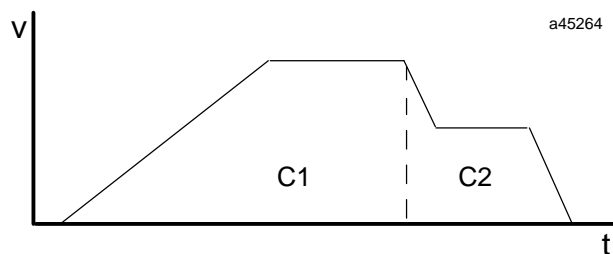


Figure 4-10. JUMP Without Stopping

Jump Stop

A jump stop is a stop that is caused by a jump. When a jump stop occurs, the *Jog Acceleration* and *Jog Acceleration Mode* are used instead of any programmed acceleration. Note that s-curve motion will achieve constant velocity before using the *Jog Acceleration* and beginning to decelerate. See the s-curve jump examples for more details. The *Jog Acceleration* is used because a jump stop may indicate something is wrong. The current *Jog Acceleration*, which can be changed with an immediate command, provides more versatility than the programmed acceleration. There are two ways of generating a jump stop, each described below.

A conditional JUMP triggered during a PMOVE will always generate a jump stop. Because a PMOVE always stops before continuing to a subsequent motion, a jump stop always occurs when a jump takes place during a PMOVE.

When a conditional jump trigger occurs during a CMOVE, however, a jump stop will not occur if the motion programmed at the jump destination is a PMOVE or CMOVE representing sufficient distance in the same direction. A jump stop will occur if the PMOVE or CMOVE at the jump destination does not represent sufficient distance or represents motion in the opposite direction.

In an s-curve move, a jump stop will do one of two things. If the jump takes place after the midpoint of the acceleration or deceleration, the acceleration or deceleration is completed before the jump stop is initiated. If the jump occurs before the midpoint of the acceleration or deceleration, the profile will immediately begin leveling off. Once acceleration or deceleration is zero, the jump stop begins. See the s-curve jump examples.

Example 10: Jump Stop

The following is an example conditional jump with a jump stop. An enhancement on Example 5, DWELL, would be to watch an external CTL bit which would indicate a problem with the positive motion. If the CTL bit never turns on, the profile for the following program will be identical to the profile shown in the DWELL example. If the CTL bit turned on during the first PMOVE or the DWELL, the reverse movement would immediately commence.

The following profile would appear if the CTL bit turned on during the first PMOVE, at the dotted line, and the *Jog Acceleration* was 75000. Because the first move completed early due to the CTL bit and a faster acceleration (*Jog Acceleration* versus programmed acceleration) the second move would not have to move as far to get back to 0 position as it did in the DWELL example. Note that because the motion programmed at the jump destination is in the opposite direction as the initial motion, the profile would be identical if the moves were CMOVEs instead of PMOVEs.

```

ACCEL      30000
VELOC     15000
BLOCK     1
JUMP      CTL09, 2
PMOVE    120000, ABS, LINEAR
DWELL    4000
BLOCK     2
PMOVE    0, ABS, LINEAR
    
```

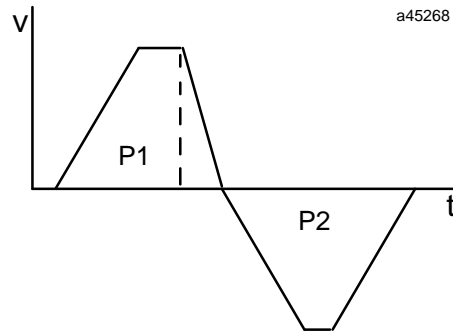


Figure 4-11. Jump Stop

Example 11: Jump Followed by PMOVE

In this JUMP example, the command after the JUMP is a PMOVE in the same direction. The velocity profile below shows the acceleration and movement for the first CMOVE and the deceleration to the PMOVE's velocity. The CTL01 bit, OFF when the PMOVE begins, turns ON at the dotted line. Motion stops after a PMOVE, even if a conditional jump goes to another block. Thus the CTL01 bit triggers a deceleration to zero before the final CMOVE begins.

```

BLOCK     1
ACCEL     2000
VELOC     8000
CMOVE    76000, INC, LINEAR
BLOCK     2
ACCEL     1000
VELOC     4000
JUMP      CTL01, 3
PMOVE    50000, INC, LINEAR
BLOCK     3
ACCEL     6000
VELOC     6000
CMOVE    36000, INC, LINEAR
    
```

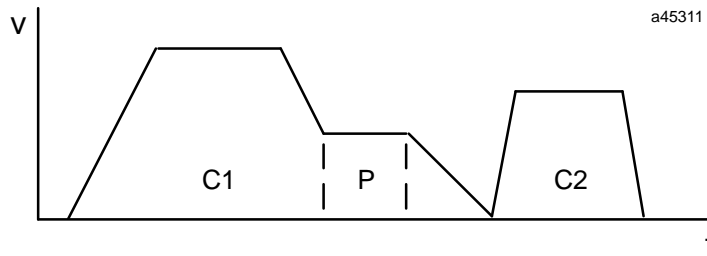


Figure 4-12. Jump Followed by PMOVE

S-CURVE Jumps

Jumps during linear motion and jumps during s-curve motion at constant velocities immediately begin accelerating or decelerating to a new velocity. Jumps during an s-curve acceleration or deceleration, however, require different rules in order to maintain an s-curve profile. What happens when a jump occurs during an s-curve move while changing velocity depends on whether the jump occurs before or after the midpoint (the point where the acceleration magnitude is greatest) and whether the velocity at the jump destination is higher or lower than the current velocity.

If the jump occurs after the midpoint of the change in velocity, the change will continue normally until constant velocity is reached; then the velocity will be changed to the new velocity using the acceleration mode of the move at the jump destination.

Example 12: S-CURVE - Jumping After the Midpoint of Acceleration or Deceleration

In the following example, a jump occurs during the final phase of deceleration, at the dotted line. The deceleration continues until constant velocity is reached and then the acceleration to the higher velocity begins.

```

ACCEL      50000
VELOC      100000
BLOCK      1
JUMP       CTL01, 3
CMOVE      500000, ABS, SCURVE
BLOCK      2
VELOC      60000
CMOVE      -500000, INC, SCURVE
BLOCK      3
VELOC      85000
ACCEL      100000
CMOVE      250000, INC, SCURVE
    
```

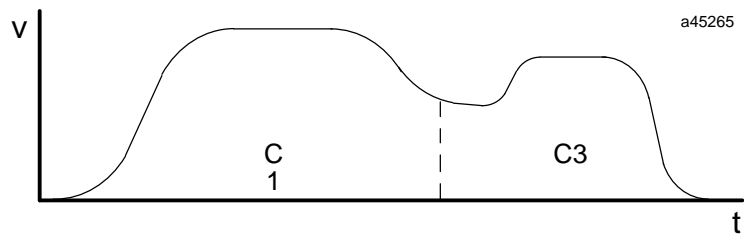


Figure 4-13. Jumping After the Midpoint of Acceleration or Deceleration

If a jump takes place before the midpoint of acceleration or deceleration, the result depends on whether the velocity at the jump destination is higher or lower than the velocity before the jump took place. In the first case, when accelerating but the new velocity is lower, or decelerating and the new velocity is greater, the Power Mate J will immediately begin reducing the acceleration or deceleration to zero; once at zero, the Power Mate J will use the jump destination acceleration and velocity and change to the new velocity.

Example 13: S-CURVE - Jumping Before the Midpoint of Acceleration or Deceleration

In the following example, during the acceleration of the first CMOVE, a jump takes place at the first dotted line. Because the velocity at the jump destination is lower than the velocity of the first CMOVE the Power Mate J slows the acceleration to zero. Constant velocity, zero acceleration, occurs at the second dotted line. There, the Power Mate J begins decelerating to the new velocity using the acceleration at the jump destination. Finally, the second CMOVE finishes.

```

ACCEL      1000
VELOC     50000
BLOCK     1
JUMP      CTL01, 3
CMOVE     50000, INC, SCURVE
BLOCK     3
VELOC     5000
ACCEL     10000
CMOVE     15000, INC, SCURVE
    
```

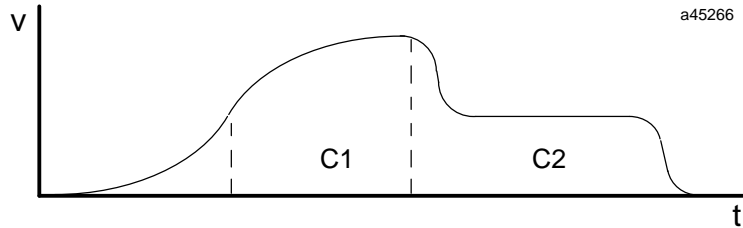


Figure 4-14. Jumping before the Midpoint of Acceleration or Deceleration

The second case involves jumping to a higher velocity while accelerating or a lower velocity while decelerating. When this occurs, the Power Mate J continues to the first move's acceleration or deceleration. This acceleration or deceleration is maintained, similar to be a linear acceleration, until the axis approaches the new velocity. Then the normal s-curve is used to reduce acceleration or deceleration to zero.

Example 14: S-CURVE - Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating

In this example, a JUMP command is triggered during the initial phase of acceleration (before the first dotted line) and the velocity at the jump destination is higher than that of the current move. The first dotted line indicates the maximum acceleration of the first CMOVE. This value is held as the axis continues to accelerate until it s-curves back to constant velocity. Constant velocity, the second dotted line, indicates the beginning of the second CMOVE. This move continues until it decelerates to zero at the end of the program.

```

ACCEL      50000
VELOC     30000
BLOCK     1
JUMP      CTL02, 2
CMOVE     150000, INC, SCURVE
BLOCK     2
VELOC     90000
ACCEL     25000
CMOVE     500000, INC, SCURVE
    
```

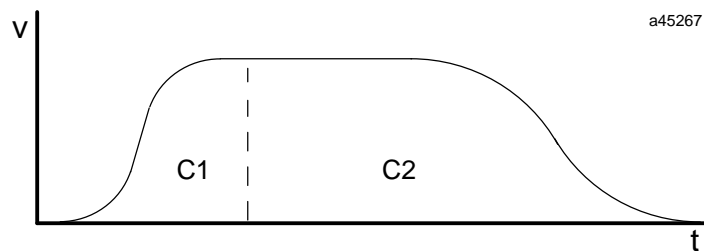


Figure 4-15. Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating

Other Programmed Motion Considerations

Maximum Acceleration Time

The maximum time for a programmed acceleration or deceleration is 64 seconds for the 1 axis Power Mate J and 128 seconds for the 2 axis Power Mate J. If the time to accelerate or decelerate is computed to be longer than this time, the Power Mate J will compute an acceleration to be used based on 64 or 128 seconds. To obtain longer acceleration times, multiple CMOVEs with increasing or decreasing velocities must be used.

Example 15: Maximum Acceleration Time

The following 1 axis program shows a problem with a very long acceleration time and a solution. In the first program and profile, 120 seconds, two minutes, is required to reach the programmed acceleration. Since this is greater than 64 seconds, the Power Mate J calculates that an acceleration of 188 would allow a velocity of 12000 to be reached in 64 seconds. The left-hand velocity profile below shows the slightly higher 188 acceleration used. Also shown is a dotted line indicating the programmed acceleration to constant velocity.

One solution for obtaining a low acceleration for a long period of time breaks the move up into separate moves with individual acceleration times less than 64 seconds. This method requires some calculation. Each acceleration and deceleration must be broken into moves with acceleration times less than 64 seconds. Thus to allow an acceleration of 100 during acceleration and deceleration, three moves will be required.

The second program and profile, those on the right, show how the first program can be broken into three parts. The distance at the midpoint of each acceleration, when velocity is 6000, is calculated to be 180,000, one fourth the distance required to accelerate to 12000. An initial CMOVE uses this distance. The next CMOVE will then accelerate to its velocity at the same acceleration rate. The final PMOVE is the midpoint distance, 180,000 user units, from the final position. The second CMOVE will automatically decelerate to the PMOVE's velocity as it approaches its final position. The dotted lines indicate when the second CMOVE begins and ends.

ACCEL	100	ACCEL	100
VELOC	12000	VELOC	6000
PMOVE	1500000, INC, LINEAR	CMOVE	180000, INC, LINEAR
		VELOC	12000
		CMOVE	1,140000, INC, LINEAR
		VELOC	6000
		PMOVE	180000, INC, LINEAR

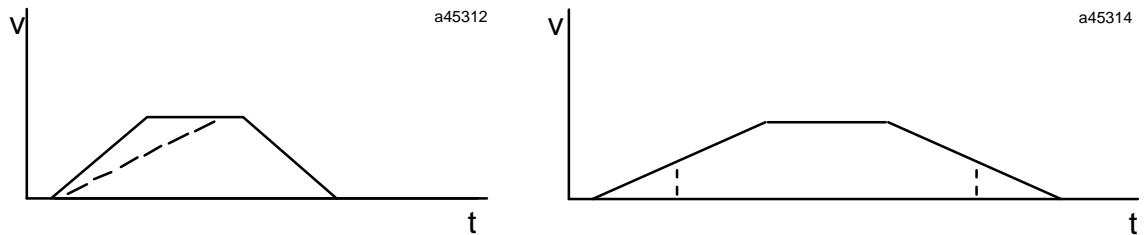


Figure 4-16. Maximum Acceleration Time

Feedhold with the Power Mate J

Feedhold is used to pause program execution without ending the program, often to examine some aspect of a system. It causes all axis motion to end at the programmed acceleration. When feedhold is ended, program execution resumes. Interrupted motion will resume at the programmed acceleration and velocity.

Feedhold is asserted by turning ON the *Feedhold %Q* bit and lasts until the *%Q* bit is turned OFF. The *Abort All Moves %Q* bit turning ON or an error which would normally cause a stop error will end feedhold as well as terminate the program. During feedhold, jogging positive and negative is allowed, but no other motion. When feedhold is terminated and program execution resumes, the Power Mate J remembers and will move to its previous destination.

Example 16: Feedhold

The following example illustrates a motion profile when feedhold is applied. The linear move accelerates to the programmed velocity at the programmed rate. Feedhold is applied at the dotted line, so velocity decreases at the programmed acceleration to zero. Then, a Jog is performed using the *Jog Minus %Q* bit. This is evident because the *Jog Velocity* is negative. Note the acceleration used during the Jog is the current *Jog Acceleration* which is different than the programmed acceleration. Note also, the feedhold must be applied during the entire duration of the Jog. After the jog motion has ceased, the feedhold is ended and the program continues to completion.

```

ACCEL      1000
VELOC      2000
PMOVE      12000, INC, LINEAR
    
```

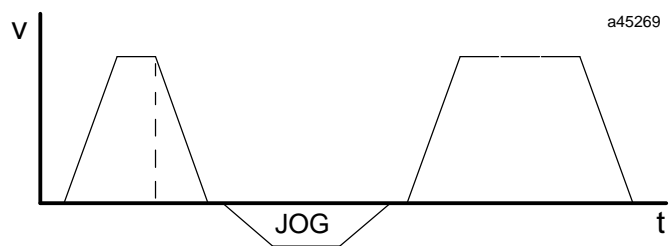


Figure 4-17. Feedhold

Feedrate Override

Some applications require small modifications to a programmed velocity to handle outside changes. A *Feedrate Override* immediate command allows changes to a programmed velocity during program execution. Whenever a program begins, the override rate is initially set to 100%. Thus changes to feedrate before the execute program bit is turned ON will be ignored, but a feedrate commanded on the same sweep as an execute program bit will be effective.

A percentage can be assigned to the feedrate from 0% to 120%. When a *Feedrate Override* is commanded, the Power Mate J internally multiplies the feedrate percentage by programmed velocity to obtain a new velocity. If the axis is moving, the current move's *Jog Acceleration Mode* is used to change velocity to the new velocity. All future move velocities will be affected by the feedrate change. Note that when a feedrate of 0% is applied, no motion will be generated until a new feedrate is commanded. Also note the *Moving %I* bit stays ON when the feedrate is 0%.

Feedrate Override has no effect on non-programmed motion such as *Jog*, *Find Home*, or *Move at Velocity*.

Example 17: Feedrate Override

During execution of this program, feedrate changes of + or -10% are commanded. Dotted lines indicate -10%, dashed lines indicate +10%.

```
ACCEL      1000  
VELOC      6000  
PMOVE     110000, INC, LINEAR
```

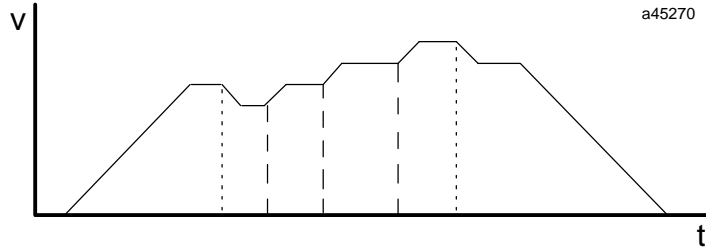


Figure 4-18. Feedrate Override

Multiaxis Programming

Sync Blocks can be used in a multiaxis program to synchronize the axis motion commands at positions where timing is critical.

Example 18: Multiaxis Programming

This example assumes that axis 1 controls vertical motion and axis 2 controls horizontal motion. The objective is to move a piece of material from point A to point C as quickly as possible while avoiding the obstacle which prevents moving directly from A to C.

A simple way would be to move from point A to point B, and then from point B to point C. This sequence, however, wastes time. A better way would begin the horizontal movement before reaching point B. It has been determined that after axis 1 has moved to a position of 30,000 user units, axis 2 could then start and still clear the obstacle. The program segment could be programmed as follows:

```
BLOCK 10          CMOVE, 30000, INC, AXIS 1
BLOCK 20 [SYNC]  PMOVE, 50000, INC, AXIS 1
                  PMOVE, 150000, INC, AXIS 2
```

When this program is executed, axis 1 immediately begins its 30,000 unit move. Axis 2 would ignore the first command, because it applies only to axis 1, and see the Sync Block. Axis 2 waits for axis 1 to reach the Sync Block before it continues executing the program. When axis 1 reaches the 30,000 unit mark, it begins the 50,000 unit PMOVE at the Sync Block without stopping (the first move was a CMOVE). Now that axis 1 has reached the Sync Block, axis 2 begins its 150,000 unit move. Looking at the position profile below, axis 1 completes its move first and stops at the end of the PMOVE. When axis 2 reaches point C, it also stops.

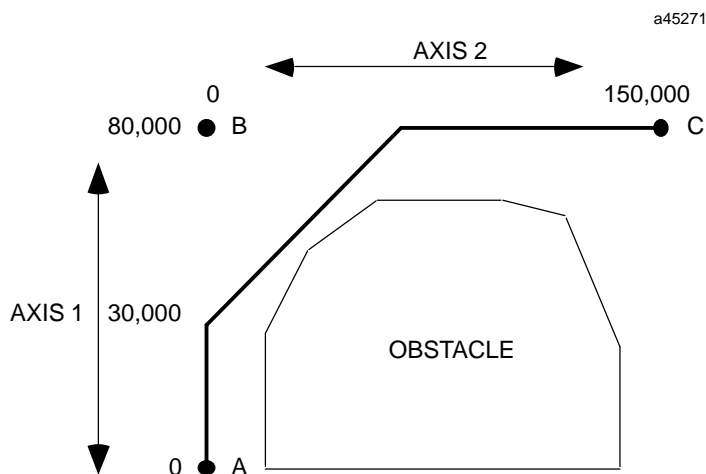


Figure 4-19. Multiaxis Programming

If this program segment is not at the beginning of a program and for some reason axis 2 has not yet reached Block 20 when axis 1 has moved 30,000 counts, an error would occur. Axis 1 would continue to 80,000 counts, and the Power Mate J would report a Block Sync Error during a CMOVE in the Status Code.

If it is imperative that the axes synchronize at Block 20, then changing Block 10 to a PMOVE would guarantee synchronization, but then axis 1 would always stop momentarily at 30,000 counts

Parameters in the Power Mate J

The Power Mate J maintains 256 double word parameters (0 through 255) in memory. These values can be used as a parameter in ACCEL, VELOC, DWELL, PMOVE, and CMOVE motion commands. Be aware that range limits still apply and errors may occur if a parameter contains a value out of range. The last ten parameters are special purpose parameters. The Power Mate J can load data into these parameters which might overwrite user data. The following table describes the function of the special purpose parameters.

Parameter Number	Special Purpose Function
246 - 253	Reserved for future use
254	Stores Axis 2 <i>Strobe Position</i> value (user units), 2 Axis Power Mate J only
255	Stores Axis 1 <i>Strobe Position</i> value (user units)

Parameters are all reset to zero after a power cycle or after a Power Mate J configuration is stored by the PLC. Parameters can be assigned in three ways: the motion program command LOAD, the immediate command Load Parameter Immediate, and the COMM_REQ function block in the PLC. The COMM_REQ function block is described in Appendix B. Assigning a value to a parameter overwrites any previous value. Parameter values can be changed during program execution, but the change must occur before the Power Mate J begins executing the block previous to the block that uses the parameter.

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

- Programming errors which generate a Status Only (warning) message.
- Programming errors that halt the servo
- Hardware errors (encoder out of sync, PLC Run switches off, loss of programmer comm., and others)

Note

The Status LED on the face plate of the module flashes slow (four times/second) for Status Only errors and fast (eight times/second) for errors which cause the servo to stop.

Error Code Format

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

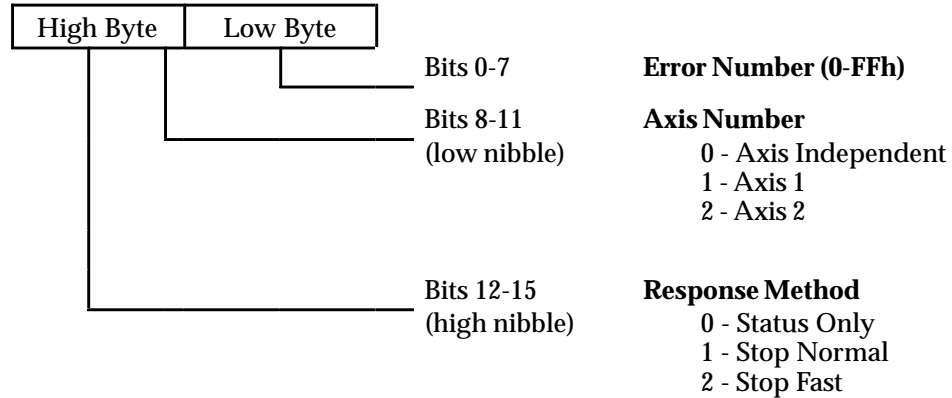


Figure A-1. Status Code Organization

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 using current Jog acceleration and acceleration mode (LINEAR or S-CURVE). 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 servo velocity command to zero. The *Drive Enabled* and *Axis Enabled %I* bits are turned off after the configured *Drive Disable Delay*.

Table A-1. Status Word Error Codes

ErrorNumber (Hexadecimal)	Response	Description
0	None	No Error
ConfigurationErrors		
2	Status Only	Scaled data too big, maximum value in range used
3	Status Only	Home Position > Positive EOT, Positive EOT used
4	Status Only	Home Position < Negative EOT, Negative EOT used
ConfigurationParameter Errors		
10	Status Only	Position Loop Time Constant too large, Immediate command ignored
11	Status Only	Position Loop Time Constant too small, Immediate command ignored
12	Status Only	Position Loop Time Constant computation overflow, reduced to non-overflow value
1E	Status Only	Immediate command Jog Velocity out of range, command ignored
1F	Status Only	Immediate command Jog Acceleration out of range, command ignored
Program Errors		
20	Status Only	Program Acceleration overrange, defaulted to 16.7 million cts/sec/sec
21	Status Only	Program Acceleration too small, defaulted to 32 cts/sec/sec
22	Status Only	Scaled Velocity greater than 1 million cts/sec, 1 million cts/sec is used
23	Status Only	Program Velocity is zero, defaulted to 1 count/sec 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, increment ignored
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 user units/sec *
3F	Status Only	Move at Velocity Data greater than 1 million cts/sec error *

* Status error is reported, command is not executed.

Table A-1. Status Word Error Codes (Continued)

ErrorNumber (Hexadecimal)	Response	Description
Jog Errors		
40	Status Only	Jog while Find Home error *
41	Status Only	Jog while Move at Velocity error *
42	Status Only	Jog while ForceD/Aerror *
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	ForceD/A while Program Selected error *
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 *
53	Status Only	Attempt to initialize position before digital encoder passes reference point.
54	Status Only	Digital encoder position invalid, must use Find Home or Set Position.
End of Travel and Count Limit Errors		
56	Status Only	Commanded Position > Positive End of Travel or High Count Limit *
57	Status Only	Commanded Position < Negative End of Travel or Low Count Limit *
58	Status Only	(AbsolutePosition + Position offset) > Positive End of Travel or High Count Limit
59	Status Only	(AbsolutePosition + Position offset) < Negative End of travel or Low Count Limit
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. Fanuc 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

* Status error is reported, command is not executed.

Table A-1. Status Word Error Codes (Continued)

ErrorNumber (Hexadecimal)	Response	Description
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 or Linear Feedback Loss of Signal error
AA	Stop Normal	Analog Input Failure
Digital Servo Configuration Errors		
B1	Status Only	CRC , SCRDY or SAL1-SAL0 error during configuration
B2	Status Only	Wait > 10 seconds for initial PM-J communications
B3	Status Only	PCGACK not = 1 error
B4	Status Only	PCGACK not = 0 error
B5	Status Only	DSI not ready for control mode
Digital Servo Control Errors		
B8	Stop Fast	DSI interrupt missing
B9	Stop Fast	CRC error Communication to DSI
BE	Stop Fast	Communications alarm from DSI
BF	Stop Fast	Hardware alarm from DSI
C0	Stop Fast	Servo not ready when MCON command is on - may be caused by E-STOP input to amplifier.
C1	Status Only	Serial Encoder Battery Low
C2	Stop Normal	Serial Encoder Battery Failed
C3	Stop Normal	Servo Motor Over Temperature
C4	Stop Fast	Servo Motor Over Current
C5	Stop Fast	Serial Encoder Communications Alarm
C6	Stop Fast	Serial Encoder Alarm
C7	Stop Fast	Servo Unit Alarm
Special Purpose Errors		
E0	Status Only	Custom Loop Type Mismatch
E1	Status Only	Digital mode not supported by hardware
EF	Status Only	Firmware/Hardware Axis Number Mismatch (1-axis firmware in 2-axis module, or 2-axis firmware in 1-axis module)

* Status error is reported, command is not executed.

Appendix B

Visual Status Error Codes

This appendix describes the visual error codes provided by the Power Mate J. These visual displays consist of LEDs on the controller module, and by the 7-segment plus decimal display on the interface module.

Section 1: Controller Module Status LEDs (PM-APM)

LED Indicators

There are five LEDs, on the PM-APM, which provide status indication for the Power Mate J controller module. These LEDs are described below.

STAT Normally ON. FLASHES to provide an indication of operational errors. Flashes *slow* (four times/second) for Status-Only errors. Flashes *fast* (eight times/second) for errors which cause the servo to stop.

OK The OK LED indicates the current status of the Power Mate APM board.

ON: When the LED is steady ON, the Power Mate J is functioning properly. Normally, this LED should always be ON.

OFF: When the LED is OFF, the Power Mate APM is not functioning. This is the result of a hardware or software malfunction.

CFG This LED is ON when a valid module configuration has been received from the PLC. Flashes *slow* (four times/second) during the Motion Program Store function. Flashes *fast* (eight times/second) during the Write User RAM to EEPROM operation.

EN1 When this LED is ON, the servo drive for Axis 1 is enabled.

EN2 When this LED is ON, the servo drive for Axis 2 is enabled.

On a 1-axis Power Mate J, this LED will not be on ON during normal operation unless a Force D/A Analog Output is commanded.

Section 2: Interface Module Visual Status Display (DSI)

Axis Error Indicator

The Power Mate J Digital Servo Interface provides a 7-segment plus decimal point display for visually indicating the current alarm status of the module. For those alarm types which apply to a particular axis, as opposed to the Power Mate J system as a whole, the decimal point will distinguish to which axis the error applies. When the alarm applies to axis 1, the decimal point will not be displayed; if the alarm applies to axis 2, then the decimal point will be displayed.

Multiple Alarms

When more than one error is detected by the Power Mate J prior to the alarm condition being cleared, each alarm will be displayed consecutively at an interval of approximately 1 second.

Power-Up Considerations

When reset, the Power Mate J controller and interface modules will power-up independently. The interface module will usually be ready to communicate with the controller module before the controller module is ready to communicate with the interface module. The interface module will begin to output the communications clock when it completes initialization and wait for the controller to establish communication. After five seconds of no communications from the controller, the interface module will display on its alarm display the error code indicating that no communications is occurring between the controller module and the interface. Once communications with the controller is initially established, this alarm will automatically be removed from the interface module.

Clearing Alarms

The controller's %Q bit *Clear Error* will function to clear Power Mate J alarms or error status indications. Asserting *Clear Error* will cause the interface module to clear all alarms currently identified. The interface module will reset the visual display and will begin to display any error caused by conditions that remain present.

DSI Status Information

The table below summarizes the status information displayed on the 7-segment display on the Power Mate J Interface Module.

Table B-1. Power Mate J Interface Module Internal Status Alarms

Display on 7 Segment LED *	Description	Per Axis	Alarm Level	Corrective Action
0	Ready (No alarm; Servo ON)	No	Information	Nonrequired
-	NotReady (No alarm; Servo OFF)	No	Information	Nonrequired
≡	Not Ready (Waiting >5 seconds for communications)	No	Warning	1. Correct Communications problem (typically cable). 2. Check controller module configuration.
1	Battery Low	Yes	Warning	2. Install a new battery. 2. Issue Clear Error.
2	Battery Failed (Zero)	Yes	Normal Servo Stop	1. Install a new battery. 2. Power cycle servo amp. 3. Issue Clear Error.
3	Motor Over Temperature	Yes	Normal Servo Stop	1. Eliminate cause of alarm. 2. Issue Clear Error.
4	Over Current	Yes	Fast Servo Stop	1. Eliminate cause of alarm. 2. Issue Clear Error.
5	Pulse Coder Communications Error	Yes	Fast Servo Stop	1. Eliminate cause of alarm. 2. Power cycle servo amp. 3. Issue Clear Error.
6	Pulse Coder Error	Yes	Fast Servo Stop	1. Eliminate cause of alarm. 2. Power cycle servo amp. 3. Issue Clear Error.
7 <small>Details in following table</small>	Servo Unit Error	Yes	Fast Servo Stop	1. Eliminate cause of alarm. 2. Power cycle servo amp. 3. Issue Clear Error.

* For alarms which apply to a particular axis, a decimal point will distinguish the axis. When the alarm applies to axis-1, the decimal point will not be displayed. When the alarm applies to axis-2, the decimal point will be displayed.

When multiple alarms are detected, each alarm will be displayed consecutively every second.

Servo Unit Error Alarm Table

The table below summarizes the Power Mate J servo status alarms.

Table B-2. Power Mate J Servo Status Alarms

Display on 7-Segment LED *	Servo Unit Error Alarm Type	Amplifier Type		
		a SVM	a SVU	b Series
c	Inverter Over Current	✓	✓	✓
C	Inverter Fan Alarm or Converter Fan Alarm	✓		
e	Inverter Low Voltage for DC Link	✓		
H	Inverter IPM Alarm	✓	✓	
h	Inverter Low Voltage for Control Power	✓		
J	Converter Excessive Regeneration	✓	✓	✓
L	Converter Over Current or Converter IPM	✓	✓	
o	Converter Overload Alarm	✓		
P	Converter Low Voltage for DC Link	✓	✓	✓
u	Converter Insufficient Charge for DC Link	✓	✓	
U **	Converter Input Power Missing Phase	✓		
U **	Converter Control Power Supply Low Voltage	✓	✓	
y	Converter Over Voltage for DC Link	✓	✓	✓

* For alarms which apply to a particular axis, a decimal point will distinguish the axis. When the alarm applies to axis-1, the decimal point will not be displayed. When the alarm applies to axis-2, the decimal point will be displayed.
When multiple alarms are detected, each alarm will be displayed consecutively every second.

** The same display value is reported for both converter errors.

Power Mate J Internal Alarms

The table below summarizes Power Mate J communication status alarms.

Table B-3. Status Alarms

Display on 7 Segment LED *	Description	Per Axis	Alarm Level	Corrective Action
B	Bad CRC from Controller	No	Fast Servo Stop	<ol style="list-style-type: none"> 1. Make sure controller to interface module cable connected; if not, correct and issue Clear Error; otherwise... 2. Power cycle PLC rack. 3. If alarm immediately reoccurs, replace interface module.
9	Handshake Error from Controller	No	Fast Servo Stop	<ol style="list-style-type: none"> 1. Make sure controller to interface cable connected; if not, correct and issue APM Clear Error; otherwise... 2. Power cycle PLC rack. 3. If alarm immediately reoccurs, replace DSI module.
A	Illegal Parameter from Controller	Yes		<ol style="list-style-type: none"> 1. Review controller module configuration parameters set with Logicmaster 90-30 and correct.
b	Reserved	No		
d	Servo Watchdog Time-out	No	Fast Servo Stop	<ol style="list-style-type: none"> 1. Power cycle servo amp. 2. Power cycle PLC rack. 3. If alarm immediately reoccurs, replace interface module.
E	Parity Error	No	Fast Servo Stop	<ol style="list-style-type: none"> 1. Power cycle PLC rack. 2. If alarm immediately reoccurs, replace interface module.
F	Clock Error	No	Fast Servo Stop	<ol style="list-style-type: none"> 1. Power cycle PLC rack. 2. If alarm immediately reoccurs, replace interface module.

* For alarms which apply to a particular axis, a decimal point will distinguish the axis. When the alarm applies to axis-1, the decimal point will not be displayed. When the alarm applies to axis-2, the decimal point will be displayed.
 When multiple alarms are detected, each alarm will be displayed consecutively every second.

Appendix C

Data Parameters From the PLC

This appendix describes an alternate method of loading the controller module's Data Parameter Memory from the PLC using a COMM_REQ function block with command code E501h. This function block can send up to 16 controller module parameter values at once. The total data length of the COMM_REQ must be set to 68 bytes (34 words) organized as shown below.

Word Offset	Byte Offset	Data
0	0 - 1	Starting parameter number (0 - 255)
1	2 - 3	Number of parameters to load
2 - 3	4 - 7	1st parameter data (4 bytes)
4 - 5	8 - 11	2nd parameter data (4 bytes)
...
32 - 33	64 - 67	16th parameter data (4 bytes)

Only the number of parameters specified in word offset 1 will be loaded into parameter memory. However, the 68 byte data block must always be initialized in the PLC. If the last parameter to be loaded is greater than 255, the COMM_REQ will be rejected. A parameter block download is illustrated in the following PLC program segment.

```

REFERENCE NICKNAME      REFERENCE DESCRIPTION
-----
%R0195 CMREQST          COMM_REQ STATUS WORD (Updated by COM_REQ)
%R0196 HDR_WDS          COMM_REQ HEADER LENGTH IN WORDS (ALWAYS 4)
%R0197 NO_WAIT          NO WAIT (ALWAYS 0)
%R0198 STMEMTP          STATUS MEMORY TYPE (8=REG)
%R0199 STLOCM1          STATUS WORD LOCATION MINUS 1 (194 = %R0195)
%R0200 NO_USE1          NOT USED (ZEROED BY BLKMV)
%R0201 NO_USE2          NOT USED (ZEROED BY BLKMV)

%R0202 CMDTYP           COMMAND TYPE (E501 FOR APM)
%R0203 BYTECNT          BYTE COUNT OF DATA
%R0204 MEMTYP           MEMORY TYPE OF DATA (8=REG)
%R0205 DATAST          START OF DATA BLOCK -1 (205 = %R0206)
%R0206 PAR_NO           STARTING PARAM NUMBER
%R0207 NO_VALS          NUMBER OF PARAMETERS TO SEND

%R0208-%R0239 PAR_DAT   Data for 16 Parameters (32 words)

```



```

(*****
* ADD LOGIC HERE TO MOVE THE PROPER CONSTANTS INTO THE REGISTERS
* (%R208 - %R239) SO THEY CAN BE SENT TO THE APM PARAMETERS.
*)
(*****

(*****
* NOW ACTIVATE THE COMM_REQ TO SEND THE PARAMETER DATA TO THE PM-APM
*)
(*****

SEND
%T0001 +-----+
+--] [-----+COMM_+
      |      |REQ|
HDR_WDS |      |
%R0196  -+IN FT+-
      |      |
CONST  -+SYSID
      |      |
      0007
      |      |
CONST  -+TASK
      |      |
00000000 +-----+

```

NOTE: SYSID HIGH BYTE = COMM_REQ RACK DESTINATION
 SYSID LOW BYTE = COMM_REQ SLOT DESTINATION
 TASK ALWAYS = 0 FOR PM-APM COMM_REQ

Appendix D

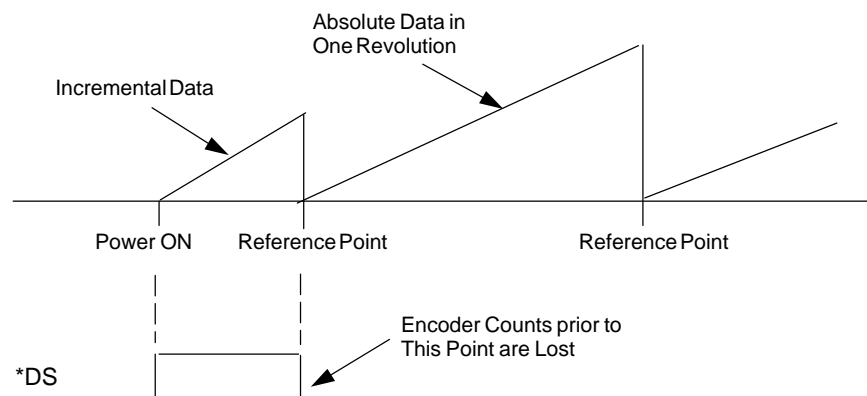
Serial Encoders

Serial Encoder Models

There are three pulse encoder models that will function with the Power Mate J, 8K, 32K or 64K counts per revolution. All encoders return position information in the form of a 32 bit value. To maintain equal number of counts (scaling) per motor revolution, the 8K encoder position increments its absolute position word data by eight counts, whereas the 32K encoder increments its data by two counts, and the 64K encoder increments its data by one count. However, the Power Mate J overrides these additional count inputs; thus, 8K, 32K and 64K encoders are identical for a Power Mate J. **The Power Mate J will always sense 8192 counts per motor shaft revolution from a serial encoder.** The one million count (1M) encoder is not supported.

Serial Encoder - First Time Use or Use After Loss of Encoder Battery Power

The encoder temporarily provides incremental data during the first use or after restoring encoder battery power. The incremental data is lost when motor shaft rotation causes the encoder to pass a reference point (similar to a marker signal) within one revolution of the motor shaft. The diagram below describes this situation, as well as the state of the *DS internal signal. The signal *DS is not directly available to the user but serves to illustrate the sequence of events. The *DS signal indicates that absolute position data is available. If no battery pack is used, *DS will be reset on each power cycle.



Note

The a Series digital encoder must be rotated up to one full revolution after the absolute mode battery has been reattached. Within one revolution the encoder will reference itself and report a referenced status (*DS = 0) to the controller. The encoder will set it's internal counts accumulator to zero at the point where *DS switches to a logic zero. Counts accumulated prior to the *DS transition are lost.

Serial Encoder Modes

The a Series pulse encoders can be operated in either Incremental mode or Absolute mode. The Incremental mode is selected by setting the controller module's Logicmaster 90-30 configuration field *Intgr Mode* (Integrator Mode) to *OFF* or to *IN ZONE*. Absolute mode is selected by setting the controller module's Logicmaster 90-30 configuration field *Intgr Mode* (Integrator Mode) to *CONTINU*. Additionally, proper operation of the Absolute mode requires an external battery pack that must be connected to the appropriate digital servo amplifier connector. Refer to the appropriate amplifier manual for selection and installation of the battery pack.

Note

The Position Error Integrator is not used when the controller module is used to control digital servos. Therefore the *Intgr Mode* (Integrator Mode) configuration is used for the alternate purpose of selecting Incremental or Absolute operating mode of the a Series pulse encoders.

Limitations on Total Travel for EOT Mode

The maximum value that can be used for the *+EOT* is +8,388,607 user units. User units are configured as a ratio to the encoder count value and are bound to the range of 8:1 to 1:32. Additionally, the feedback from a digital servo is fixed at 8192 counts per revolution. Using a simple formula will determine the maximum number of motor shaft revolutions possible for axis travel when EOT (non rotary) mode is used.

$$\text{Maximum Revolutions} = 1024 * \frac{\text{Counts}}{\text{User Units}}$$

Using the above calculation, the maximum possible move is realized with the User Unit to Count ratio at the maximum 1:32 resulting in a maximum travel of +/- 32,768 motor shaft revolutions. (The maximum travel originates at the *Actual Position* zero and can move the number of maximum travel revolutions plus or minus of the zero position.)

There is no restriction on maximum travel for Rotary mode. refer to *Important Configuration Considerations* in Chapter 2 for additional information on EOT Mode and Rotary Mode.

Incremental Encoder Mode Considerations

The digital serial encoder can be used as an incremental encoder returning 8192 counts per shaft revolution, with no revolution counts retained through a power cycle. The equivalent of a *marker* pulse will occur once each motor shaft revolution. All Home modes (Home Switch, Move+, Move- and Set Position) reference the axis, and set the *Position Valid* bit upon successful completion. High count limits and Low Count limits are valid and the *Actual Position* as reported by the controller module will wrap from high to low count or from low to high count values. This is an excellent mode for rotary applications that will always operate via incremental moves, in the same direction. *Home Offset* and *Home Position* configuration items allow simple referencing to the desired location. Incremental mode is selected by setting the controller module's Logicmaster 90-30 configuration field *Intgr Mode* (Integrator Mode) to *OFF* or to *IN ZONE*.

Absolute Encoder Mode Considerations

The a Series serial pulse encoder can be used as an absolute type encoder by adding a battery pack to retain servo position while system power is off. A Find Home cycle or Set position must be performed initially or whenever encoder battery power is lost with the servo amplifier also in a powered down state. Absolute Encoder mode must be configured in the controller module to function properly. Absolute mode is selected by setting the controller module's Logicmaster 90-30 configuration field *Intgr Mode* (Integrator Mode) to *CONTINU*.

Absolute Encoder Mode - Position Initialization

When a system is first powered up in Absolute Encoder mode, a position offset for the encoder must be established. This can be done by using the Find Home cycle or the Set Position command.

Find Home Cycle - Absolute Encoder Mode

The Find Home cycle mode can be configured for Move +, Move - or Home Switch operation. Refer to Chapter 4 for additional details of Home Cycle operation. The Home Offset and Home Position configuration items function the same as in Incremental Encoder mode. At the completion of the Home Cycle, Actual Position is set to the configured Home Position value. The controller internally calculates the encoder Absolute Feedback Offset needed to produce the configured Home Position at the completion of the Home Cycle. This Absolute Feedback Offset must be permanently saved in the controller by sending %AQ command 4Ah *Update Flash Memory* after completion of the Home Cycle.

Once the Absolute Feedback Offset is saved in Flash memory, the APM will automatically initialize Actual Position after a power cycle and set the Position Valid %I bit.

Set Position Command - Absolute Encoder Mode

The %AQ Set Position command functions the same as in Incremental Encoder mode. At the completion of the Set Position operation, Actual Position is set to the Set Position value. The controller internally calculates the encoder Absolute Feedback Offset needed to produce the commanded Set Position value. This Absolute Feedback Offset must be permanently saved in the controller by sending %AQ command 4Ah *Update Flash Memory* after the Set Position command.

Once the Absolute Feedback Offset is saved in Flash memory, the APM will automatically initialize Actual Position after a power cycle and set the Position Valid %I bit.

Absolute Encoder Mode - Power Mate J Power-Up

The battery pack attached to the servo subsystem will maintain power to the encoder counter logic. Once the encoder has referenced through first time start up, the actual position is automatically maintained by the encoder, even if the axis is moved during servo power loss. The encoder will monitor the status of the battery pack, and report loss of battery power or low battery power to the Power Mate J.

The Power Mate J will complete a power-on diagnostic, and when configured for absolute encoder mode, interrogate the referenced status of the serial encoder. A valid referenced status from the encoder will signal the Power Mate J to read the encoder absolute position. The controller will report the *Actual Position* as the sum of the encoder position and the *Absolute Feedback Offset* established by the initial Find Home cycle or Set Position command.

Absolute Encoder Mode with Rotary Mode

Some restrictions are necessary when Absolute Encoder mode is selected along with Rotary mode.

Absolute Encoder mode causes the controller to automatically initialize Actual Position from the battery backed absolute encoder after a power cycle. Absolute Encoder mode is selected by setting the Intgr Mode to CONTINU in the Configuration software.

Rotary mode allows the controller to create continuous motion in one direction by the use of multiple CMOVE or PMOVE incremental commands. As the axis moves, Actual Position will reach a Hi or Low Count Limit then roll over to the other limit. The distance between Actual Position rollovers is the Rotary Count Modulus. Rotary Mode is selected by configuring the +EOT Limit \geq Hi Count Limit and -EOT Limit \leq Low Count Limit.

The battery backed absolute encoder has a total absolute counting range of (8192 cts/rev) x (32767 revs) = 268,427,264 counts. This is the maximum number of counts the encoder can move after a position reference operation (Find Home or Set Position) and still retain absolute position. If the encoder rotates more than this number of counts after a position reference operation, the absolute position is lost and the encoder starts a new counting cycle.

Restrictions when Absolute Encoder Mode is used with Rotary Mode

1. If the Rotary Count Modulus (in counts) is a power of 2, then no restrictions exist on rotary travel. This means the distance defined as $((\text{Hi Limit} - \text{Low Limit}) + 1)$ in counts must be a number which is a power of 2 such as 128, 256, 512 ... 8192, 16384, and so forth. Under this condition the controller will always initialize Actual Position to the correct rotary position after a power cycle.
2. If the power of 2 condition for Rotary Count Modulus is not met, the rotary motion must be limited to a range of $\pm 268,427,264$ counts after a Set Position or Find Home position reference operation. This restriction can be handled in some systems by periodically performing a Set Position when the axis is stopped and holding a known rotary position.

Note

Performing a Set Position or Find Home cycle in Absolute Encoder mode causes the controller to recalculate the encoder Absolute Feedback Offset. An Update Flash Memory command must always be sent to permanently save the new Absolute Feedback Offset. The flash memory in the controller has a limited number of update cycles (20,000 – 100,000 cycles). This limitation should be considered if a system design required the Absolute Feedback Offset to be saved more than 10 times per day.

Symbols

- %AI Position Error, 2-3
- %AI Status Words, 3-5
 - Actual Velocity, 3-7
 - Commanded Velocity, 3-7
 - Actual Position, 3-6
 - Command Block Number, 3-6
 - Commanded Position, 3-6
 - Status Code, 3-6
 - Strobe Position, 3-6
 - for the 1-Axis, 3-5
 - for the 2-Axis, 3-5
- %AQ Immediate Commands, 3-11
 - Load Parameter Immediate, 3-17
 - Absolute Feedback Offset, 3-17
 - Configuration Modes, 3-16
 - Force D/A Output, 3-13
 - In Position Zone, 3-14
 - Jog Acceleration, 3-14
 - Jog Velocity, 3-14
 - Move, 3-14
 - Move At Velocity, 3-13
 - Null, 3-13
 - Position Increment, 3-13
 - Position Increment With Position Update, 3-13
 - Position Loop Time Constant, 3-15
 - Rate Override, 3-13
 - Select Return Data, 3-16
 - Set Configuration Complete, 3-17
 - Set Position, 3-13
 - Torque Limit, 3-16
 - Update Flash Memory, 3-17
 - Velocity Feedforward, 3-15
 - Velocity Loop Gain, 3-15
- %I Status Bit
 - Axis Enabled, 3-3
 - Configuration Complete, 3-4
 - Drive Enabled, 3-3
 - Moving, 3-3
 - PLC Control Active, 3-4
 - Position Strobe, 3-4
 - Position Valid, 3-3
 - Program Active, 3-3
 - Error, 3-4
 - In Error Limit, 3-4
 - In Zone, 3-3
- %Q Discrete Commands, 3-8
 - Feedhold (Off Transition), 3-9
 - Jog Minus, 3-10

- Abort All Moves, 3-9
- Clear Error, 3-10
- Enable Drive, 3-9
- Feedhold (On Transition), 3-9
- Find Home, 3-9
- Jog Plus, 3-10
- Reset Strobe Flag, 3-10
- CTL09 - CTL12 Output Controls, 3-10
- Execute Motion Program 0 - 10, 3-10
 - for the 1-Axis, 3-8
 - for the 2-Axis, 3-9

Numbers

- 24 VDC Digital Inputs, General Purpose, 1-9
- 5 VDC Power, 1-8

A

- Abort All Moves %Q Discrete Command, 3-9
- Absolute Encoder
 - First Time Use, D-1
 - Use After Loss of Encoder Battery Power, D-1
- Absolute Encoder Mode
 - Hi/LoLimits, D-2
 - Limitations on Total Travel, D-2
 - Power Mate J Power-Up, D-4
 - Restrictions when used with Rotary Mode, D-5
 - with Rotary Mode, D-4
- Absolute Encoder Mode, Considerations, D-3
 - Find Home Cycle, D-3
 - Position Initialization, D-3
 - Set Position Command, D-4
- Absolute Feedback Offset %AQ Immediate Command, 3-17
- Absolute Position Offset, 2-7
- Absolute Positioning, 4-8
- ACCEL Command, Program Zero, 2-14
- Acceleration, 4-6
 - Linear, 4-8
 - Mode, 2-17
 - S-Curve, 4-9
 - Time, Maximum, 4-25
 - Types of, 4-8

Actual Position %AI Status Word, 3-6
Actual Velocity %AI Status Word, 3-7
Alarms, Clearing, B-2
Alarms, Multiple, B-2
Analog Output Circuitry, 1-7
Appendix A, Error Word Status Codes, A-1
Appendix B, Visual Status Error Codes, B-1
Appendix C, Data Parameters From the PLC, C-1
Appendix D, Serial Encoders, D-1
Axis Configuration Data, 2-7
 Absolute Position Offset, 2-7
 Counts, 2-7
 Disable Delay, 2-7
 DisDly, 2-10
 Final Home Velocity, 2-7
 Find Home Vel, 2-10
 Find Home Velocity, 2-7
 Fnl Home Vel, 2-10
 Gradient, 2-7
 Hi Limit, 2-10
 High Limit, 2-7
 Home Mode, 2-7, 2-10
 Home Offset, 2-7, 2-10
 Home Position, 2-7
 Home Positn, 2-10
 In Pos Zone, 2-7, 2-9
 Integrator Mode, 2-7
 Integrator Time Constant, 2-7
 Intgr Mode, 2-9
 Intgr TC, 2-9
 Jog Acc, 2-10
 Jog Acc Mod, 2-10
 Jog Acceleration, 2-7
 Jog Acceleration Mode, 2-7
 Jog Vel, 2-10
 Jog Velocity, 2-7
 Lo Limit, 2-10
 Low Limit, 2-7
 Neg EOT, 2-8
 Negative End Of Travel, 2-7
 OT Limit SW, 2-7
 Overtravel Limit Switch, 2-7
 Pos EOT, 2-8
 Pos Err Lim, 2-8
 Pos Loop TC, 2-9
 Position Error Limit, 2-7
 Position Loop Time Constant, 2-7

Positive End Of Travel, 2-7
Recircs, 2-7
Rev Comp, 2-9
Reversal Compensation, 2-7
User Units, 2-7
User Units, Counts, 2-7
Velocity at 10 V, 2-7
Velocity at 10 Volts, 2-9
Velocity FF%, 2-7, 2-9

Axis Enabled %I Status Bit, 3-3
AxisError Indicator, B-2
AxisField, Program Zero, 2-12

B

Baud Rate, 2-6
Block #, 4-6
BLOCK Command, Program Zero, 2-14
Block Diagram of PM-J System, 1-2
Block Numbers and Jumps, 4-16

C

Cable, Controller to Interface Module, 1-17
Call Subroutine, 4-6
Catalog Numbers
 IC690ACC901, Mini-Converter Kit (RS-232 to RS-485/422), 1-4
 IC693APU301, 1-Axis Controller, 1-3
 IC693APU302, 2-Axis Controller, 1-3
CFG LED, B-1
Clear Error %Q Discrete Command, 3-10
Clearing Alarms, B-2
CMOVE, 4-10
CMOVE Command, Program Zero, 2-14
COMM_REQ, C-1
COMM_REQ, example of, C-1
Command Block Number %AI Status Word, 3-6
Command Name Field, Program Zero, 2-12
Command, Dwell, 4-15
Command, ForceD/A, 4-4

Command, Velocity, 4-4
Command, Wait, 4-15
Commanded Position %AI Status Word, 3-6
Commanded Velocity %AI Status Word, 3-7
Commands, Position Increment, 4-4
Commands, Program Zero Motion Program, 2-13
Communications, High Speed Serial, PM-APM to DSI, 1-16
Compatibility
 Controller Modules, 1-19
 Host CPU, 1-19
 Programming, 1-19
 Servo Units, 1-19
Conditional Jumps, 4-17
Conditions Which Stop a Motion Program, 4-7
Configuration, Controller (Power Mate APM), 2-2
Configuration Considerations, Important, 2-17
Configuration Modes %AQ Immediate Command, 3-16
Configuration Parameters, 2-3
Configuration Parameters, Essential, 2-15
Configuring the Interface Module, 2-11
Configuring the Power Mate J, 2-2
Configuring the Rack/Slot, 2-2
Connection, Multidrop, 1-5
Connector, serial COMM, 1-4
Connectors and Ports, 1-15
Considerations for Power-Up, B-2
Considerations. Absolute Encoder Mode, D-3
 Find Home Cycle, D-3
 Position Initialization, D-3
 Set Position Command, D-4
Considerations, Incremental Encoder Mode, D-3
Continuous Move, 4-6
Continuous Move (CMOVE), 4-10
Control Loop, 2-3

Controller Module
 Configuration, 2-2
 description of, 1-3
 Module Compatibility, 1-19
 serial COMM connector, 1-4
Count, 2-7
CTL09 - CTL12 Output Controls %Q Discrete Command, 3-10

D

D/A Command, Force, 4-4
Data Bits, 2-6
Data Field, Program Zero, 2-12
Description
 Controller Module, 1-3
 Interface Module, 1-14
 Power Mate J, 1-1
Digital Servo Interface, 1-14
Digital Servo Motors, list of, 2-4
Disable Delay, 2-7
DisDly (Servo Drive Disable Delay), 2-10
Display, Visual Status, DSI, 1-14
Drive Enabled %I Status Bit, 3-3
Dwell, 4-6
Dwell Command, 4-15
DWELL Command, Program Zero, 2-14

E

EN1 LED, B-1
EN2 LED, B-1
Enable Drive %Q Discrete Command, 3-9
Enable Relay Output Circuitry, 1-7
End of Program, 4-6
Error %I Status Bit, 3-4
Error Code Format, A-2
Error Word Status Codes, A-1
Essential Configuration Parameters
 Acceleration Mode, 2-17
 Open Loop Mode, 2-15
 Position Loop Time Constant, 2-15
 User Units and Counts, 2-16
 Velocity at 10 Volts, 2-15

Example 1, Combining PMOVES and CMOVES, 4-11

Example 10, Jump Stop, 4-21

Example 11, Jump Followed by PMOVE, 4-22

Example 12, S-Curve, Jumping After the Midpoint of Acceleration or Deceleration, 4-23

Example 13, S-Curve, Jumping Before the Midpoint of Acceleration or Deceleration, 4-24

Example 14, S-Curve, Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating, 4-25

Example 15, Maximum Acceleration Time, 4-25

Example 16, Feedhold, 4-27

Example 17, Feedrate Override, 4-28

Example 18, Multiaxis Programming, 4-29

Example 2, Changing the Acceleration Mode During a Profile, 4-12

Example 3, Not Enough Distance to Reach Programmed Velocity, 4-13

Example 4, Hanging the PM-J When the Distance Runs Out, 4-14

Example 5, Dwell, 4-15

Example 6, Unconditional Jump, 4-16

Example 7, Jump Testing, 4-19

Example 8, Normal Stop before JUMP, 4-19

Example 9, JUMP Without Stopping, 4-20

Execute Motion Program 0 - 10 %Q Discrete Command, 3-10

F

Feedback Type, 2-3

Feedback Types, Position, 4-1

Feedhold (Off Transition) %Q Discrete Command, 3-9

Feedhold (On Transition) %Q Discrete Command, 3-9

Feedhold with the PM-J, 4-26

Feedrate Override, 4-27

Final Home Velocity, 2-7

Find Home %Q Discrete Command, 3-9

Find Home Cycle, D-3

Find Home Vel (Find Home Velocity), 2-10

Find Home Velocity, 2-7

First Time Use, Absolute Encoder, D-1

Final Home Vel (Final Home Velocity), 2-10

Force D/A Command, 4-4

Force D/A Output %AQ Immediate Command, 3-13

Functional Connection Diagrams
for 1-Axis, 1-12
for 2 Axis, 1-13

G

General Purpose Input Circuitry, 1-9

H

Hi Limit (High Count Limit), 2-10

Hi/LoLimits, D-2

High Limit, 2-7

High Speed Serial Cable Diagram, 1-17

Home Cycle, 4-2

Home Mode, 2-7

Home Mode (Find Home Mode), 2-10

Home Offset, 2-7

Home Position, 2-7

Home Positn (Home Position Offset), 2-10

Home Positn (Home Position), 2-10

Home Switch Mode, 4-2

Host CPU, Compatibility, 1-19

I

I/O Cable and Terminal Block, 1-6

I/O Cable Connections
for 1-Axis, 1-10
for 2-Axis, 1-11

I/O Connectors, 1-6

- I/O Specifications, 1-7
 - Idle Time, 2-6
 - Immediate Commands Using the 6Byte Format, 3-12
 - Important Configuration Considerations, 2-17
 - In Error Limit %I Status Bit, 3-4
 - In Pos Zone, 2-7
 - In Pos Zone (In Position Zone), 2-9
 - In Position Zone %AQ Immediate Command, 3-14
 - In Zone %I Status Bit, 3-3
 - Incremental Encoder Mode Considerations, D-3
 - Incremental Positioning, 4-8
 - Indicators, LEDs, B-1
 - Input Circuitry, General Purpose, 1-9
 - Installation
 - Interface Module, 2-11
 - Power Mate J, 2-1
 - Step-by-Step Procedure, 2-1
 - Installing the Power Mate J, 2-1
 - Integrator Mode, 2-7
 - Integrator Time Constant, 2-7
 - Interface Module, 1-14
 - See also* Digital Servo Interface Configuration, 2-11
 - Description of Module, 1-14
 - Interface Module Status Bits, 3-18
 - Internal Alarm table, B-3
 - Internal Alarm Type, B-5
 - Intgr Mode (Integrator Mode), 2-9
 - Intgr TC (Integrator Time Constant), 2-9
- J**
- Jog ACC (Jog Acceleration Rate), 2-10
 - Jog ACC Mod (Jog Acceleration Mode), 2-10
 - Jog Acceleration, 2-7
 - Jog Acceleration %AQ Immediate Command, 3-14
 - Jog Acceleration Mode, 2-7
 - Jog Minus %Q Discrete Command, 3-10
 - Jog Plus %Q Discrete Command, 3-10
 - Jog Vel (Jog Velocity), 2-10
 - Jog Velocity, 2-7
 - Jog Velocity %AQ Immediate Command, 3-14
 - Jogging with the PM-J, 4-3
 - JS1 Connector to Servo Axis 1, 1-15
 - JS2 Connector to Servo Axis 2, 1-15
 - Jump, 4-6
 - JUMP Command, Program Zero, 2-14
 - Jump Stop, 4-21
 - Jump Testing, 4-19
 - Jumping Without Stopping, 4-20
 - Jumps and Block Numbers, 4-16
 - Jumps, Conditional, 4-17
 - Jumps, S-CURVES, 4-23
 - Jumps, Unconditional, 4-16
- L**
- LED Indicators, Controller Module, B-1
 - LED Indicators on Power Mate APM, STAT, OK, CFG, EN1, EN2, 1-3
 - Limitations on Total Travel, D-2
 - Linear Acceleration, 4-8
 - Lo Limit (Low Count Limit), 2-10
 - LOAD Command, Program Zero, 2-14
 - Load Parameter Immediate %AQ Immediate Command, 3-17
 - Logicmaster 90-30 Configuration of Controller Module, 2-2
 - of Interface Module, 2-11
 - Loss of Encoder Battery Power, D-1
 - Low Limit, 2-7
- M**
- Maximum Acceleration Time, 4-25
 - Mode, Home Switch, 4-2

- Models of Serial Encoders, D-1
- Modem Turnaround Time, 2-6
- Modes, Move + and Move -, 4-3
- Modes, Serial Encoders, D-2
- Module Configuration
 - Controller (Power Mate APM), 2-2
 - Interface Module, 2-11
- Module Configuration Data
 - %AI Position Error, 2-3
 - Control Loop, 2-3
 - Feedback Type, 2-3
 - Power Mate J, 2-3
 - Ref Adr, 2-3
- Motion Control, 4-1
- Motion Program Commands, Program Zero, 2-13
- Motion Program, Conditions Which Stop, 4-7
- Motion, Non-Programmed, 4-2
- Motion, Programmed, 4-6
- Move %AQ Immediate Command, 3-14
- Move At Velocity %AQ Immediate Command, 3-13
- Move at Velocity Command, 4-4
- Move Commands, Programmed, 4-10
- Move, Continuous (CMOVE), 4-10
- Move, Positioning (PMOVE), 4-10
- Move+ and Move - Modes, 4-3
- Moves, Programmed, 4-11
- Moving %I Status Bit, 3-3
- MultiAxisProgramming, 4-29
- Multidrop Connection, 1-5
- MultipleAlarms, B-2

N

- Neg EOT (Negative Software End of Travel), 2-8
- Negative End Of Travel, 2-7
- Non-Programmed Motion, 4-2
- Normal Stop Before JUMP, 4-19
- Null, 4-6

- Null %AQ Immediate Command, 3-13

O

- OK LED, B-1
- Open Loop Mode, 2-15
- OT Limit SW, 2-7
- Other Considerations, 4-5
- Other Programmed Motion Considerations, 4-25
- Out of Sync, 2-8
- Override, Feedrate, 4-27
- Overtravel Limit Switch, 2-7
- Overview of PM-J System, 1-2

P

- Parameters for Programmed Moves
 - distance to move, 4-7
 - position to move to, 4-7
 - type of acceleration, 4-7
 - type of positioning reference, 4-7
- Parameters in the PM-J, 4-30
- Parity, 2-6
- Pin Definitions of the Serial COMM Connector, 1-4
- PLC Configuration Complete %I Status Bit, 3-4
- PLC Control Active %I Status Bit, 3-4
- PM-APM, 1-18
- PM-J, 1-1
- PMOVE, 4-10
- PMOVE Command, Program Zero, 2-14
- Ports and Connectors, 1-15
- Pos EOT (Positive Software End of Travel), 2-8
- Pos Err Lim (Position Error Limit), 2-8
- Pos Loop TC (Position Loop Time Constant), 2-9
- Position Error Limit, 2-7
- Position Feedback Types, 4-1
- Position Increment %AQ Immediate Command, 3-13

- Position Increment Command, 4-4
 - Position Initialization, D-3
 - Position Loop Time Constant, 2-7, 2-15
 - Position Loop Time Constant %AQ Immediate Command, 3-15
 - Position Strobe %I Status Bit, 3-4
 - Position Valid %I Status Bit, 3-3
 - Positioning Move, 4-6
 - Positioning Move (PMOVE), 4-10
 - Positioning Reference, Types of, 4-7
 - Positioning, Absolute, 4-8
 - Positioning, Incremental, 4-8
 - Positive End Of Travel, 2-7
 - Power Mate APM, to DSI, JD35 Connector, 1-16
 - Power Mate J
 - Description of, 1-1, 1-3
 - Installing Hardware, 2-1
 - Power-Up, D-4
 - Specifications, 1-18
 - Power-Up Considerations, B-2
 - Prerequisites for Programmed Motion, 4-7
 - Program Active %I Status Bit, 3-3
 - Program Motion Commands
 - Type 1
 - Call Subroutine, 4-6
 - Jump, 4-6
 - Type 2
 - Acceleration, 4-6
 - Block #, 4-6
 - Null, 4-6
 - Velocity, 4-6
 - Type 3
 - Continuous Move, 4-6
 - Dwell, 4-6
 - End of program, 4-6
 - Positioning Move, 4-6
 - Wait, 4-6
 - Program Zero, 2-11
 - Program Zero Commands, 2-14
 - ACCEL, 2-14
 - BLOCK, 2-14
 - CMOVE, 2-14
 - DWELL, 2-14
 - JUMP, 2-14
 - LOAD, 2-14
 - PMOVE, 2-14
 - VELOC, 2-14
 - WAIT, 2-14
 - Program Zero Motion Program Commands, 2-13
 - Program Zero Programmer Instruction Format, 2-12
 - AxisField, 2-12
 - Command Name Field, 2-12
 - Data Field, 2-12
 - Programmed Motion, 4-6
 - Programmed Motion Considerations, Other, 4-25
 - Programmed Move Commands, 4-10
 - Programmed Moves, 4-11
 - Programmed Moves, Parameters, 4-7
 - Programmer, Compatibility, 1-19
 - Programming, multiaxis, 4-29
- ## R
- Rack/Slot Configuration, 2-2
 - Rate Override %AQ Immediate Command, 3-13
 - Recircs, 2-7
 - Ref Adr, 2-3
 - Reset Strobe Flag %Q Discrete Command, 3-10
 - Response Methods, A-2
 - Restrictions, Absolute Encoder Mode with Rotary Mode, D-5
 - Rev Comp (Reversal Compensation), 2-9
 - Reversal Compensation, 2-7
 - Rotary Mode, D-4
 - Restrictions, D-5
- ## S
- S-Curve Acceleration, 4-9
 - S-CURVE Jumps, 4-23
 - Select Return Data %AQ Immediate Command, 3-16
 - Serial COMM Connector, 1-4
 - Serial COMM connector, pin definitions, 1-4

- Serial Communications Port Configuration Data, 2-6
 - Baud Rate, 2-6
 - Data Bits, 2-6
 - Idle Time, 2-6
 - Modem Turnaround Time, 2-6
 - Parity, 2-6
 - SNP ID, 2-6
 - Stop Bits, 2-6
- Serial Communications, High Speed, PM-APM to DSI, 1-16
- Serial Encoder Modes, D-2
- Serial Encoders, Models of, D-1
- Servo Motors, Digital, 2-4
- Servo Unit Error Alarm table, B-4
- Servo Units, Compatibility, 1-19
- Set Configuration Complete %AQ Immediate Command, 3-17
- Set Position %AQ Immediate Command, 3-13
- Set Position Command, D-4
- Setting the Configuration Parameters, 2-3
- SNP ID, 2-6
- Specifications, Power Mate J, 1-18
- Status Bits, Interface Module, 3-18
- Status Code %AI Status Word, 3-6
- Status Display, Visual, on DSI, 1-14
- Status LED, B-1
- Stop Before JUMP, Normal, 4-19
- Stop Bits, 2-6
- Strobe Input Circuitry, 1-8
- Strobe Position %AI Status Word, 3-6
- Subroutines, 4-16
- Sync, Out of, 2-8
- System Overview, of PM-J, 1-2

T

- Table, Visual Display
 - Internal Alarm, B-3
 - Internal Alarm Type, B-5
 - Servo Unit Error Alarm, B-4
- Terminal Block and I/O Cable, 1-6

- Testing, Jump, 4-19
- Torque Limit %AQ Immediate Command, 3-16
- Turnaround Time, 2-6
- Types of Acceleration, 4-8
- Types of Positioning Reference, 4-7

U

- Unconditional Jumps, 4-16
- Update Flash Memory %AQ Immediate Command, 3-17
- User Units, 2-7
- User Units and Counts, 2-16
- User Units, Counts, 2-7

V

- VELOC Command, Program Zero, 2-14
- Velocity, 4-6
 - Velocity at 10 V, 2-7
 - Velocity at 10 Volts, 2-9
 - Velocity at 10 volts, 2-15
 - Velocity Command, Move at, 4-4
 - Velocity Feedforward %AQ Immediate Command, 3-15
 - Velocity FF%, 2-7
 - Velocity FF% (Velocity Feed Forward Gain), 2-9
 - Velocity Loop Gain %AQ Immediate Command, 3-15
- Visual Status Display, DSI, 1-14
- Visual Status Error COdes, B-1

W

- Wait, 4-6
- Wait Command, 4-15
- WAIT Command, Program Zero, 2-14
- Wiring, Controller to Interface Module Cable, 1-17