GE Fanuc Automation

PowerMotion ™ Products

Power Mate APM
for Series 90 ™ -30 PLC
Follower Mode

User's Manual

GFK-0781C

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.

Preface

Note

This manual describes the Power Mate APM module follower mode operation. This module was previously referred to as the Axis Positioning Module.

Content of This Manual

Chapter 1. Introduction: This chapter provides a brief overview of the software and hardware used to setup and operate a servo system.

Chapter 2. Installing the Power Mate APM: This chapter describes all the user interfaces of the Power Mate APM and how to install the module on the Series 90-30 baseplate.

Chapter 3. Configuring the Power Mate APM: This chapter explains how to configure the APM using the Logicmaster configuration package.

Chapter 4. Automatic Data Transfers: This chapter describes the %I, %AI, %Q, and %AQ data that is transferred between the PLC CPU and the Power Mate APM each sweep.

Chapter 5. Follower System Startup Procedures: This chapter explains the procedures for properly starting up a follower system.

Chapter 6. Power Mate APM Motion Control: This chapter provides practical information on Power Mate APM control with a number of examples.

Appendix A. Error Codes

Appendix B. Parameter Download Using COMM_REQ

Appendix C. Specifications

Appendix D. Wiring to SS-90 Drives

Appendix E. Ordering Information

Related Publications

GFK-0664 *Series 90 PLC Axis Positioning Module Programmer's Manual*

GFK-0840 *Power Mate APM for Series 90-30 - Standard Mode User's Manual*

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.

Contents

Contents

Chapter 1 Introduction

The Follower option of the 1 and 2-axis Power Mate APM motion control modules provides a high performance, easy-to-use, master-slave controller which is highly integrated with the logic solving and communications functions of the Series 90-30 PLC.

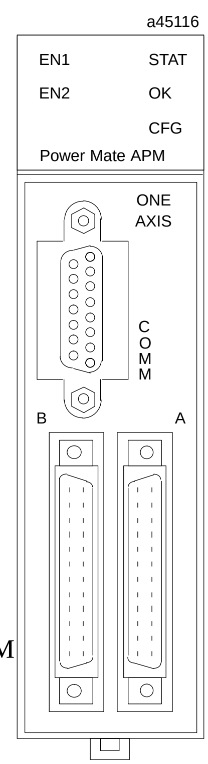
The Power Mate APM operates in two primary control loop configurations:

- Standard Mode
- Follower Mode

This manual describes the Power Mate APM in the *Follower* control loop mode. For information on the operation of the Power Mate APM in *Standard* control loop mode, refer to GFK-0840, *Power Mate APM for Series 90-30 - Standard Mode User's Manual*.

The Power Mate APM Follower option is selected by setting the Module Configuration Parameter, *Control Loop Type*, to FOLLOWER. The *Control Loop Type* parameter is accessed when performing Module Configuration using the Logicmaster 90-30 Configuration package.

Features of the Power Mate APM



High Performance

- Fast Servo Loop Update (1 millisecond for APU301; 2 milliseconds for APU302)
- Velocity Feedforward and Position Error Integrator to enhance tracking accuracy
- Accurate electronic gearing
- Follower motion with superimposed motion program

Easy to Use

- Simple and powerful Motion Program instruction set
- Configured with Logicmaster 90 software
- Automatic Data Transfer between PLC tables and Power Mate APM without user programming
- Ease of I/O connection with factory cables and standard terminal blocks

Versatile I/O

- 10 volt Velocity Command analog output.
- 12-bit plus sign analog input.
- Home and Drive OK switch inputs.
- User-defined control inputs and outputs.
- Encoder Feedback (up to 175 kHz per channel).

Firmware Compatibility Tables

1 Axis Power Mate APM						
Released Firmware Version	**Catalog Number**	**Control Loop Capability**	**Power Mate APM Manual**	**PLC Programmer Version**	**Motion Programmer Version**	**Motion Programmer Manual**
1.00, 1.10	IC693APU301A-D	Standard	GFK-0707A	3.50 or later	1.01 or later	GFK-0664
2.02	IC693APU301E-H	Standard Follower	GFK-0840 GFK-0781	4.01 or later *	1.50 or later	GFK-0664A
2.11	IC693APU301J	Standard Follower	GFK-0840 GFK-0781	6.01 or later	1.50 or later	GFK-0664A
2.50	IC693APU301K	Standard Follower	GFK-0840 GFK-0781	6.01 or later	1.50 or later	GFK-0664A

* *The 1 axis Power Mate APM firmware release 2.02 requires PLC Programmer version 4.50 or later to allow configuration in Follower mode.*

2 Axis Power Mate APM						
Released Firmware Version	**Catalog Number**	**Control Loop Capability**	**Power Mate APM Manual**	**PLC Programmer Version**	**Motion Programmer Version**	**Motion Programmer Manual**
1.50	IC693APU302A-B	Follower	GFK-0781	3.50 or later	1.50 or later	GFK-0664A
2.01, 2.02	IC693APU302GG	Standard Follower	GFK-0840 GFK-0781	4.01 or later	1.50 or later	GFK-0664A
2.10, 2.11	IC693APU302HJ	Standard Follower	GFK-0840 GFK-0781	6.01 or later	1.50 or later	GFK-0664A
2.50	IC693APU302K	Standard Follower	GFK-0840 GFK-0781	6.01 or later	1.50 or later	GFK-0664A

Overview of Power Mate APM Follower Operations

The Power Mate APM Follower is an intelligent, fully configurable, motion control option module for the Series 90-30 Programmable Logic Controller (PLC). The Power Mate APM allows a PLC user to combine high performance follower control with PLC logic solving functions in one integrated system. The figure below illustrates the hardware and software needed to set up and operate a follower system. This section will discuss briefly each piece of the system to provide an overall picture of system operation.

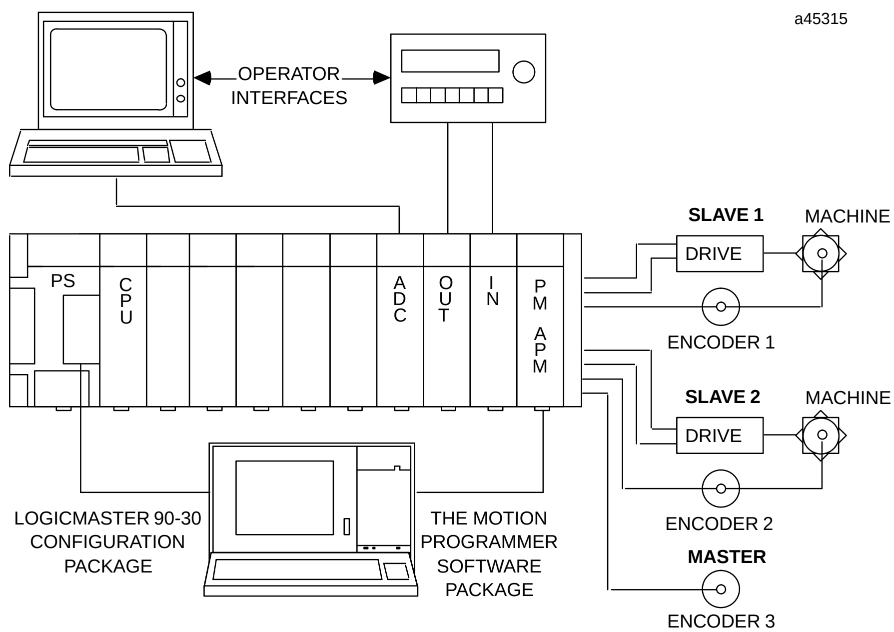


Figure 1-1. Hardware and Software Used to Configure and Operate a Power Mate APM Follower System

The system shown above can be divided into 4 categories:

1. The Series 90-30 PLC and Power Mate APM
2. Configuration Software Package
3. Operator Interfaces
4. Servo Drive and Machine Interfaces

The Series 90-30 PLC and Power Mate APM

The Series 90-30 PLC and Power Mate APM operate together as one integrated motion control package. The Power Mate APM controls axis motion and handles all direct connections to the drive and machine while the PLC automatically transfers data between PLC tables and the Power Mate APM. The PLC also provides a means for connecting Operator Interfaces such as CIMPLICITY 90 - ADS which can control and monitor system operation.

Logicmaster 90 -30 Configuration software is used to configure the Power Mate APM.

The PLC and Power Mate APM are tightly integrated and transfer Power Mate APM Commands and Status Data over the back plane through the PLC tables. Some data is transferred only when directed by an operator. Other data is transferred automatically each PLC sweep.

Configuration Software Package

The Power Mate APM is easily configured using the Logicmaster 90-30 Configuration software. The Power Mate APM must be assigned to a particular slot and rack like any other PLC module.

In addition, there are other types of configuration data which must be entered such as:

- I/O addresses where the CPU to Power Mate APM transfers take place
- Serial Port Setup for connecting a monitor
- Power Mate APM Setup Data

Operator Interfaces

Operator interfaces provide a way for the operator to control and monitor the servo system through a control panel or CRT display. These interfaces communicate with the PLC through discrete I/O modules or an intelligent Series 90 PLC module such as CIMPLICITY 90 - Alphanumeric Display Coprocessor module or a Programmable Coprocessor Module (PCM).

Operator data is automatically transferred between the PLC and the Power Mate APM through %I, %AI, %Q, and %AQ references which are specified when the module is configured. This automatic transfer of data provides a flexible and simple interface to a variety of operator interfaces in addition to the Motion Programmer.

Servo Drive and Machine Interfaces

The servo drive and machine interface is made through two 24-pin male connectors. This interface carries the signals that control axis position such as the **Velocity Command** and **Encoder Feedback** signals. Also provided are **Home Switch** and **Drive OK Switch** inputs as well as other control inputs and outputs.

Standard cables which connect directly to Weidmuller style terminal blocks simplify user wiring.

Chapter

2

Installing the Power Mate APM

This chapter describes the 1 and 2-axis models of the Power Mate APM (Power Mate APM) and how to install the modules on the Series 90-30 PLC baseplate. The chapter is divided into the following section

- Section 1: Description of the 1 and 2 Axis Power Mate APM
- Section 2: Installing the Power Mate APM

Section 1: Description of the Power Mate APM

This section describes the user interfaces of the 1 and 2 axis Power Mate APM in follower mode.

LED Indicators

There are five LEDs which provide status indication for the Power Mate APM. These LEDs are described below.

Status. 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 Power Mate APM OK LED indicates the current status of the Power Mate APM.

ON: When the LED is steady ON, the Power Mate APM 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.

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 APM, this LED is ON only when a *Force D/A Immediate* command is used for the analog output on Connector B.

Serial COMM Connector

The Power Mate APM Front Panel contains a single 15-pin, female, D-connector for serial communications. This port can be used for status information and monitoring. The Power Mate APM port uses the GE Fanuc proprietary 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 Power Mate APM 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 Miniconverter Kit (IC690ACC901) which includes a converter and 6 foot cable is available for this purpose.

The port is configured using the Logicmaster 90-30 / 20 / Micro Configuration Software or the Series 90-30 / 20 / Micro Hand-Held Programmer. The pin definitions for the serial port are listed in Table 2-1.

Table 2-1. Pin Definitions of the Serial COMM 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 / 422-232 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 APM modules can be connected in multidrop fashion. A sample setup is shown in the figure below. One cable is necessary for each Power Mate APM in the system.

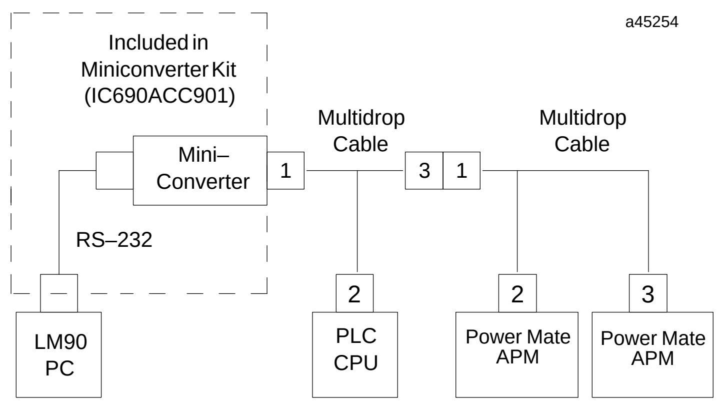


Figure 2-1. Connecting Power Mate APM Modules in a Multidrop Configuration

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

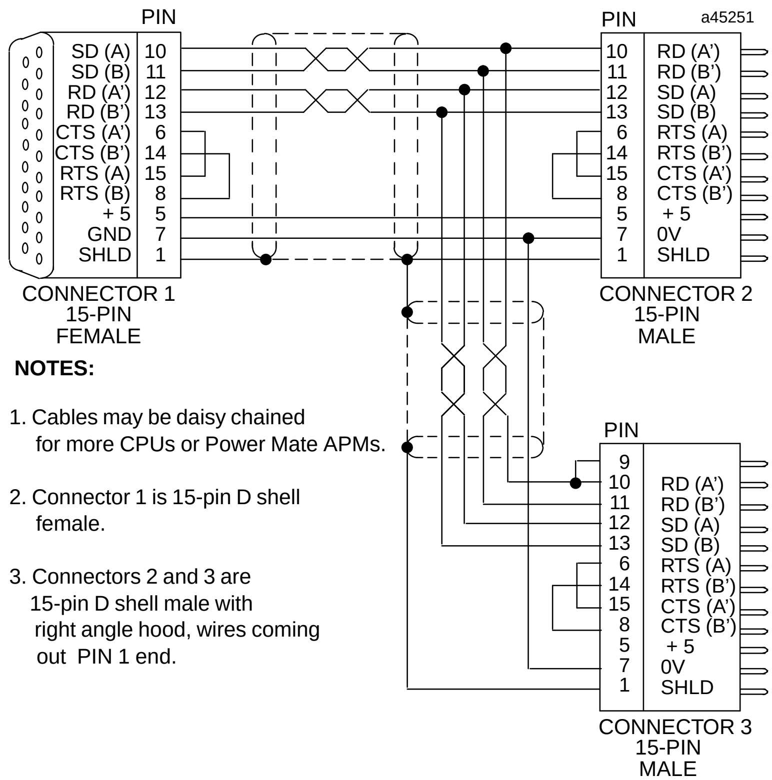


Figure 2-2. Multidrop Cable for the Power Mate APM

I/O Connectors

The Power Mate APM Front Panel contains two 24-pin, male high-density connectors for servo connections. Connector A contains connections for Drive 1 and the Master Encoder. Connector B contains connections for Drive 2 and the Analog Input.

I/O Cable and Terminal Block

High-density connectors are used on the Power Mate APM to permit a large number of I/O connections within the physical size limitations of the Power Mate APM. To facilitate wiring to the drive and machine, each high-density connector is typically connected by a short cable (IC693CBL311A) to a terminal block. Refer to Figure 2-3 below and Tables 2-2 through 2-5.

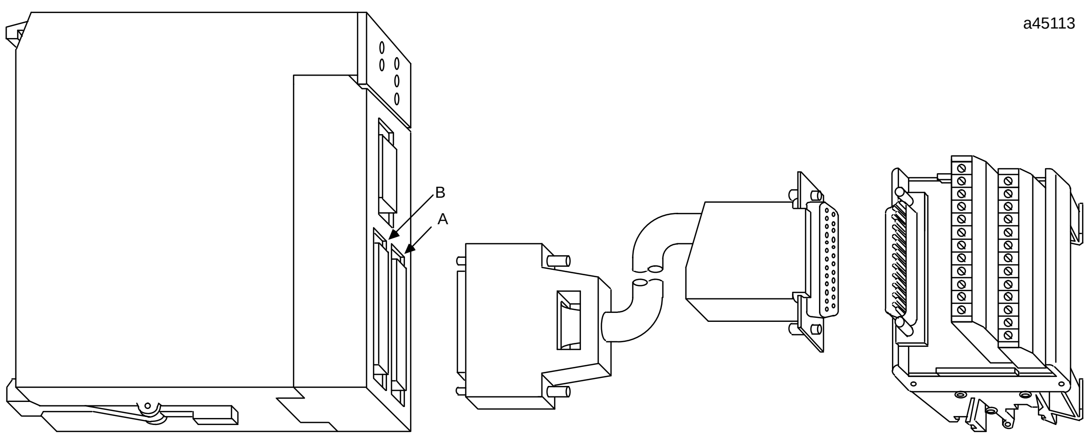


Figure 2-3. I/O Cable and Terminal Block

I/O Cable Connections for the 1-Axis Power Mate APM (IC693APU301)

Tables 2-2 and 2-3 define the cable connections for the 1-Axis Power Mate APM.

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

Power Mate APM Module Connector A Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Encoder Channel Z3+
A2	11	Encoder Channel Z3–
B2	23	Encoder Channel B3+
A3	10	Encoder Channel B3–
B3	22	Encoder Channel A3+
A4	9	Encoder Channel A3–
B4	15	Common for CTL03, 05, 06 Inputs
A5	2	Home 3 / CTL06 Input
B5	14	Drive 1 OK / CTL05 Input
A6	1	Home 1 / CTL03 Input
B6	16	Enable Relay 1 (–) Output
A7	3	Enable Relay 1 (+) Output
B7	17	Analog Output 1 (Velocity Command) Common *
A8	4	Analog Output 1 (Velocity Command)
B8	21	0 V
A9	8	+5 V Encoder Power
B9	20	Encoder Channel Z1+
A10	7	Encoder Channel Z1–
B10	19	Encoder Channel B1+
A11	6	Encoder Channel B1–
B11	18	Encoder Channel A1+
A12	5	Encoder Channel A1–
B12	13	Cable Shield
	25	no connection

* Fused Output – an external source applied to this output could open the fuse.

Table 2-3. Cable Connections to Faceplate I/O Connector B (1-Axis Power Mate APM)

Power Mate APM Module Connector B Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Analog (+) Input
A2	11	Analog (–) Input
B2	23	Reserved
A3	10	Reserved
B3	22	Reserved
A4	9	Reserved
B4	15	Common for CTL04, 07, 08 Inputs
A5	2	CTL08 Input
B5	14	CTL07 Input
A6	1	CTL04 Input
B6	16	Enable Relay 2 (–) Output
A7	3	Enable Relay 2 (+) Output
B7	17	Analog Output 2 Common *
A8	4	Analog Output 2
B8	21	0 V
A9	8	+5 V Output
B9	20	Reserved
A10	7	Reserved
B10	19	CTL02 (+) Input
A11	6	CTL02 (–) Input
B11	18	CTL01 (+) Input
A12	5	CTL01 (–) Input
B12	13	Cable Shield
	25	no connection

* Fused Output – an external source applied to this output could open the fuse.

I/O Cable Connections for the 2-Axis Power Mate APM (IC693APU302)

Tables 2-4 and 2-5 define the cable connections for the 2-Axis Power Mate APM.

Table 2-4. Cable Connections to Faceplate I/O Connector A (2-Axis Power Mate APM)

Power Mate APM Module Connector A Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Encoder Channel Z3+ / CTL02 (+) Input
A2	11	Encoder Channel Z3– / CTL02 (–) Input
B2	23	Encoder Channel B3+
A3	10	Encoder Channel B3–
B3	22	Encoder Channel A3+
A4	9	Encoder Channel A3–
B4	15	Common for CTL03, 05, 06 Inputs
A5	2	Home 3 / CTL06 Input
B5	14	Drive 1 OK / CTL05 Input
A6	1	Home 1 / CTL03 Input
B6	16	Enable Relay 1 (–) Output
A7	3	Enable Relay 1 (+) Output
B7	17	Analog Output 1 (Velocity Command) Common *
A8	4	Analog Output 1 (Velocity Command)
B8	21	0 V
A9	8	+5 V Encoder Power
B9	20	Encoder Channel Z1+
A10	7	Encoder Channel Z1–
B10	19	Encoder Channel B1+
A11	6	Encoder Channel B1–
B11	18	Encoder Channel A1+
A12	5	Encoder Channel A1–
B12	13	Cable Shield
	25	no connection

* Fused Output – an external source applied to this output could open the fuse.

Table 2-5. Cable Connections to Faceplate I/O Connector B (2-Axis Power Mate APM)

Power Mate APM Module Connector B Pin Number	Terminal Block Terminal Number	Description
A1	12	0 V
B1	24	Analog (+) Input
A2	11	Analog (–) Input
B2	23	Reserved
A3	10	Reserved
B3	22	Reserved
A4	9	Reserved
B4	15	Common for CTL04, 07, 08 Inputs
A5	2	CTL08 Input
B5	14	Drive 2 OK / CTL07 Input
A6	1	Home 2 / CTL04 Input
B6	16	Enable Relay 2 (–) Output
A7	3	Enable Relay 2 (+) Output
B7	17	Analog Output 2 (Velocity Command) Common *
A8	4	Analog Output 2 (Velocity Command)
B8	21	0 V
A9	8	+5 V Output
B9	20	Encoder Channel Z2+ / CTL01 (+) Input
A10	7	Encoder Channel Z2– / CTL01 (–) Input
B10	19	Encoder Channel B2+
A11	6	Encoder Channel B2–
B11	18	Encoder Channel A2+
A12	5	Encoder Channel A2–
B12	13	Cable Shield
	25	no connection

* Fused Output – an external source applied to this output could open the fuse.

Functional Connection Diagrams

The figures below illustrate how the Power Mate APM I/O is connected to a drive and a machine in a typical follower application. Shielded cable should be used as indicated.

Functional Connection Diagrams for the 1-Axis Power Mate APM

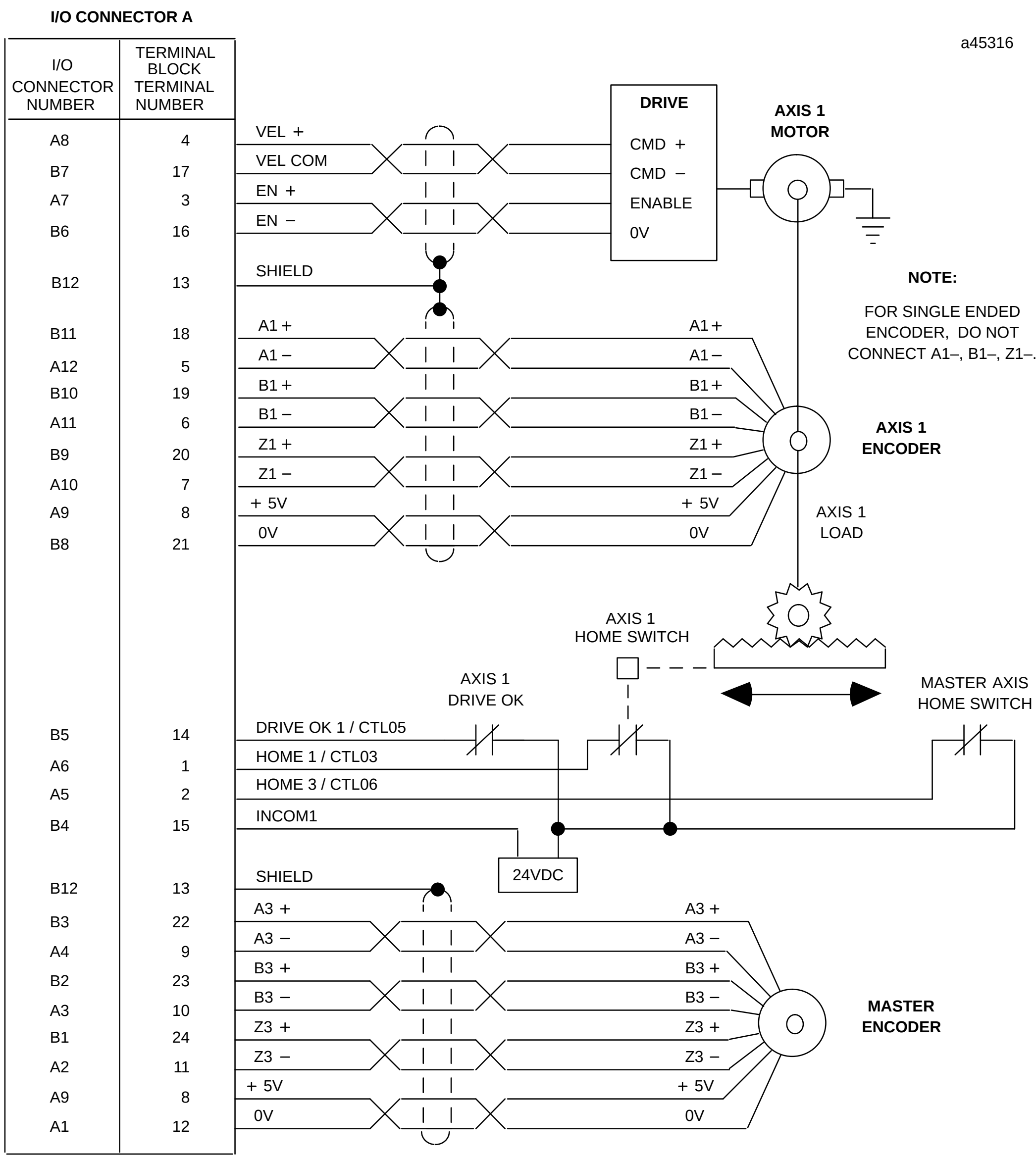


Figure 2-4. Functional Diagram for 1-Axis Power Mate APM I/O Connector A

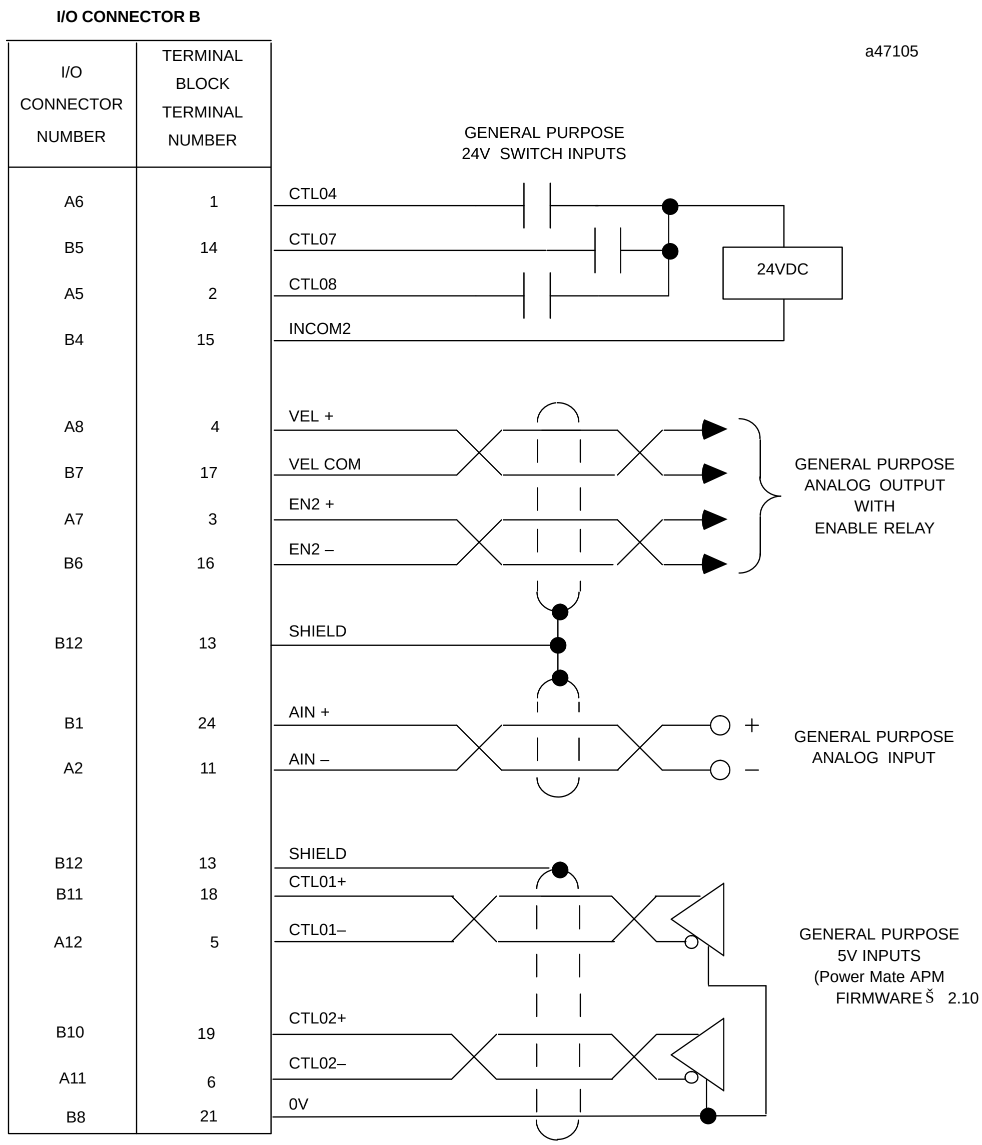


Figure 2-5. Functional Diagram for 1-Axis Power Mate APM I/O Connector B

Functional Connection Diagrams for the 2-Axis Power Mate APM

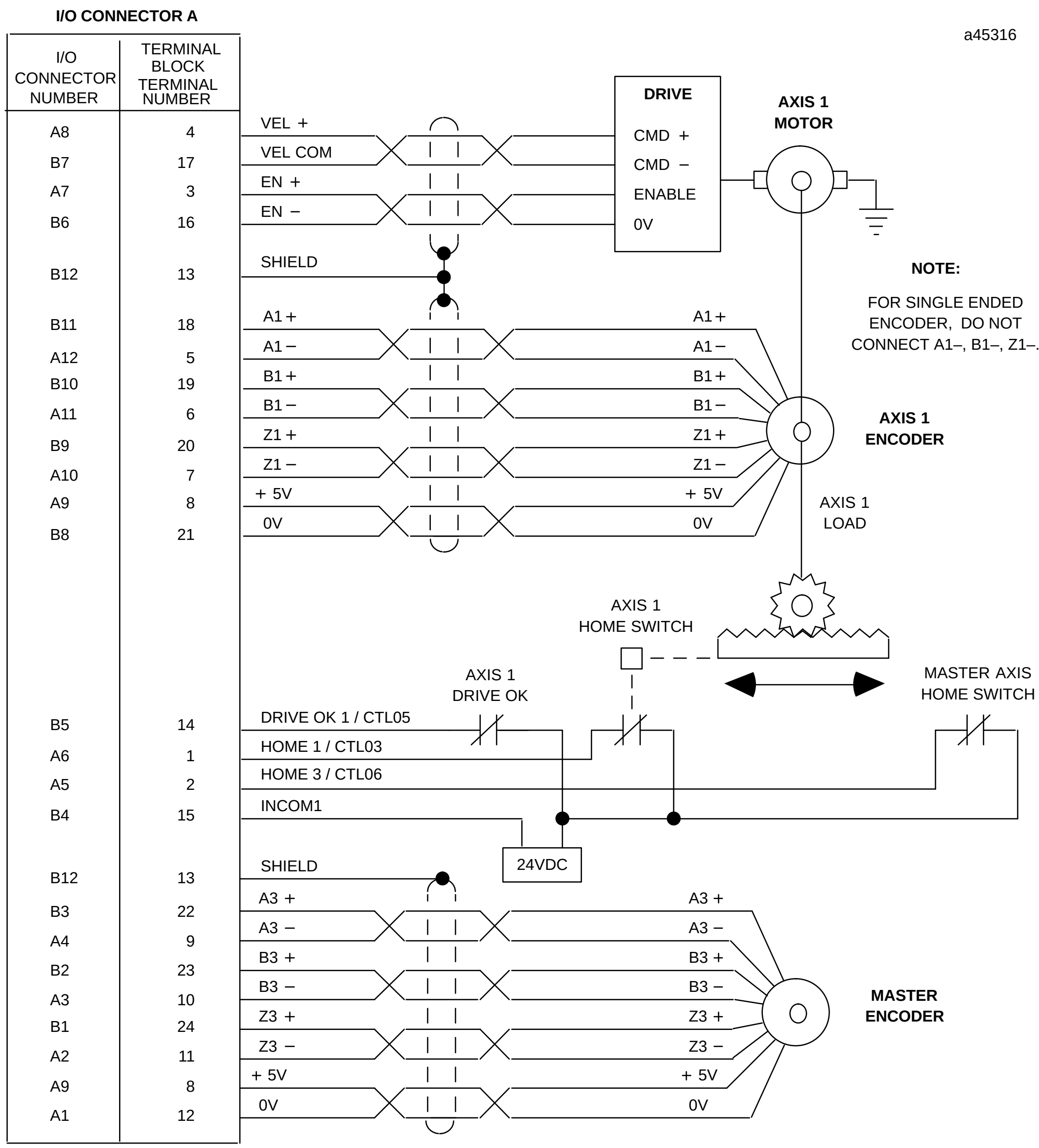


Figure 2-6. Functional Diagram for 2-Axis Power Mate APM I/O Connector A

Note

For Power Mate APM firmware release 2.10 or later, the Z3 input channel is also the CTL02 input.

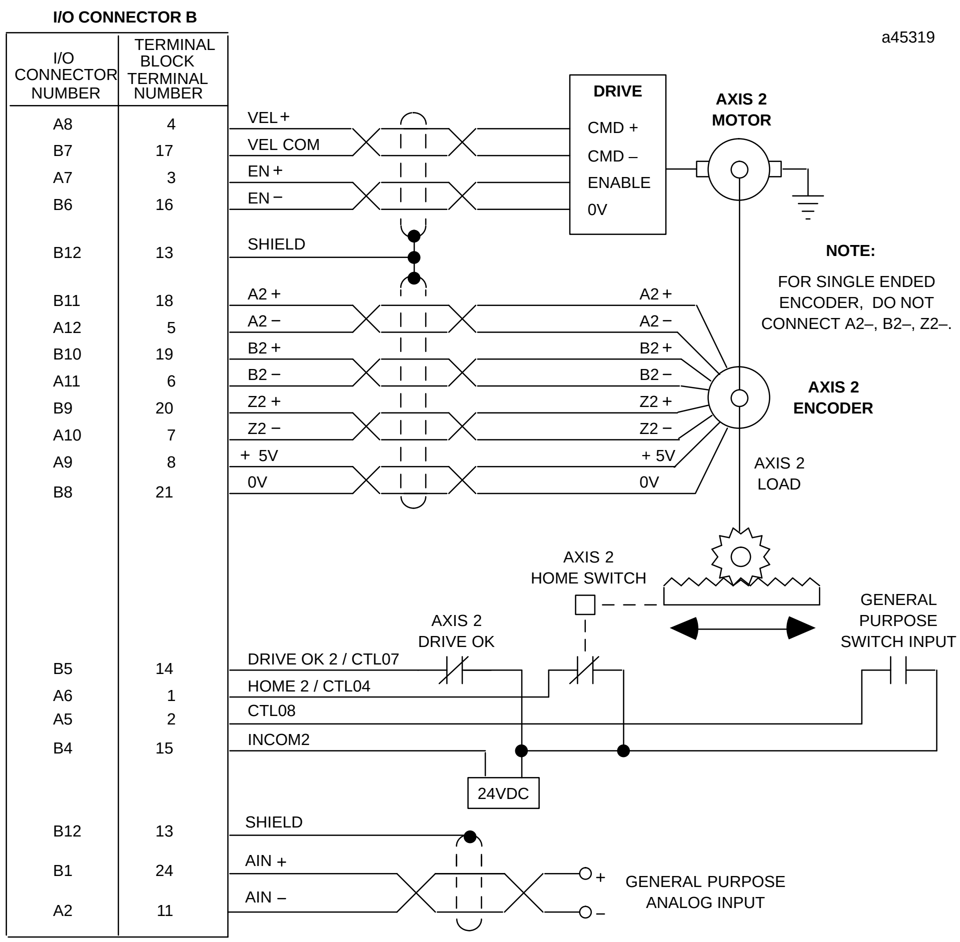


Figure 2-7. Functional Diagram for 2-Axis Power Mate APM I/O Connector B

Note

For Power Mate APM firmware release 2.10 or later, the Z2 input channel is also the CTL01 input.

Functional Block Diagrams for the 2-Axis Power Mate APM

The diagram below illustrates the two axes of the Power Mate APM connected in cascade with encoder 3 or the internal master as the master source for axis 2, and encoder 2 the master source for axis 1.

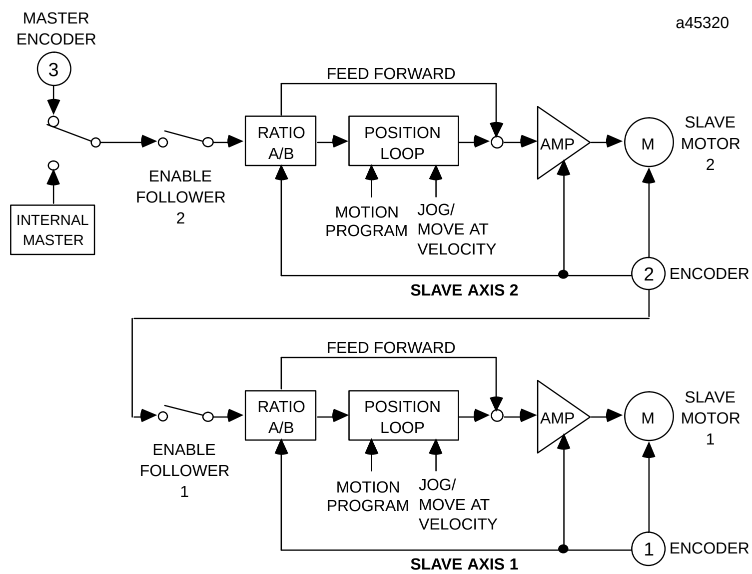


Figure 2-8. 2-Axis Cascade, Master Source Encoder 3/Internal Master

The diagram below illustrates the two axes of the Power Mate APM connected in parallel with encoder 3 or the internal master as the master source for both axis 1 and axis 2.

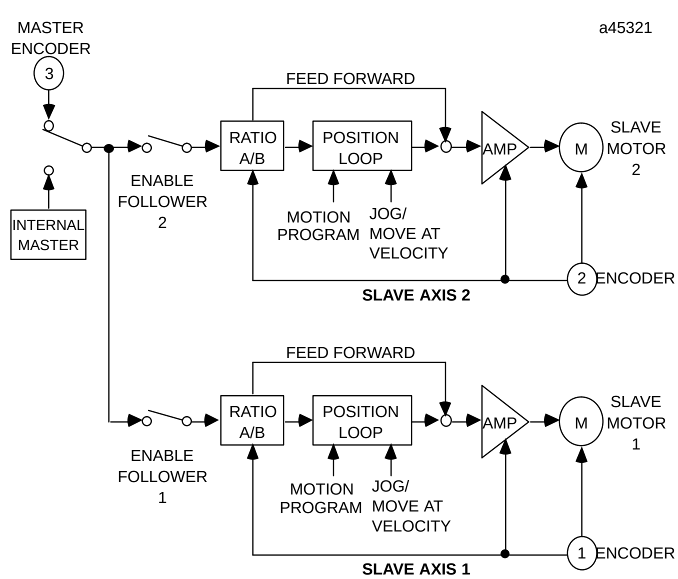


Figure 2-9. 2-Axis Parallel, Master Source Encoder 3/Internal Master

The following diagram illustrates the two axes of the Power Mate APM connected in cascade with the analog input as the master source for axis 2, and encoder 2 the master source for axis 1.

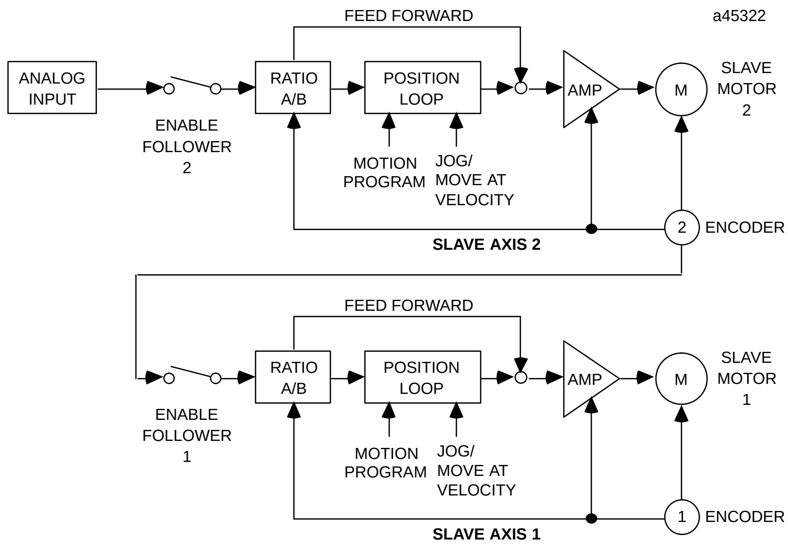


Figure 2-10. 2-Axis Cascade, Master Source Analog Input

The following diagram illustrates the two axes of the Power Mate APM connected in parallel with the analog input as the master source for both axis 1 and axis 2.

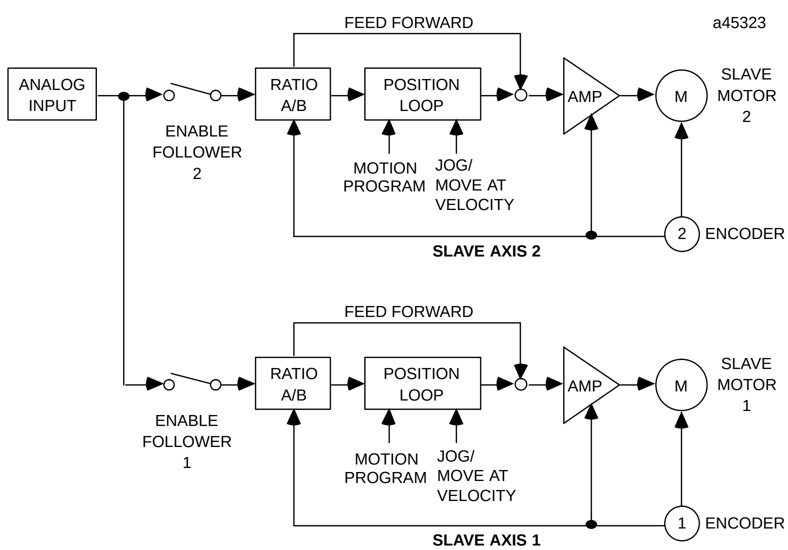


Figure 2-11. 2-Axis Parallel, Master Source Analog Input

The diagram below illustrates the two axes of the Power Mate APM connected with encoder 3 or the internal master as the master source for axis 2, and the analog input the master source for axis 1.

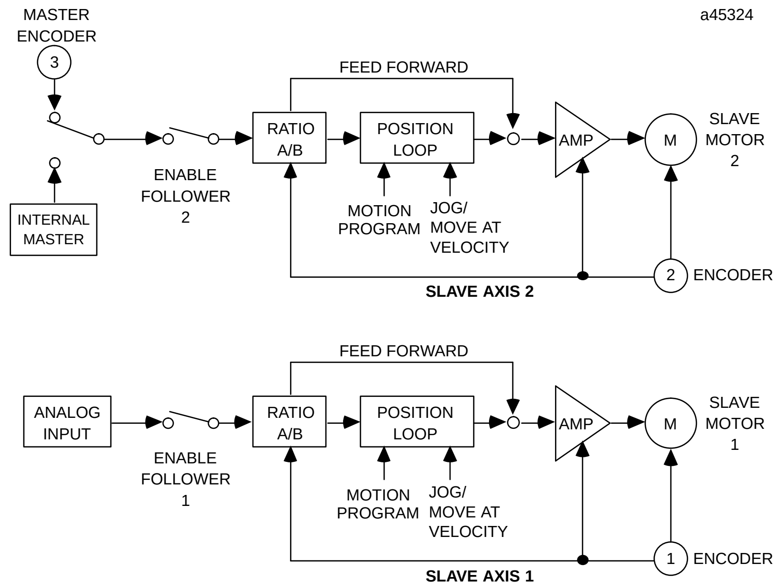


Figure 2-12. 2-Axis Power Mate APM: Axis 2 Master Source Encoder 3/Internal Master, Axis 1 Master Source Analog Input

The diagram below illustrates the two axes of the Power Mate APM connected with the analog input the master source for axis 2 and encoder 3 or the internal master as the master source for axis 1.

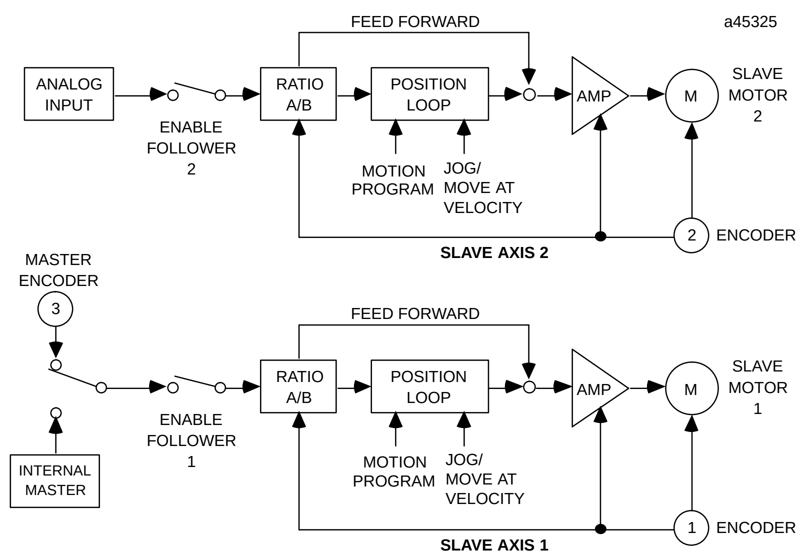


Figure 2-13. 2-Axis Power Mate APM: Axis 2 Master Source Analog Input, Axis 1 Master Source Encoder 3/Internal Master

Section 2. Installing the Power Mate APM

The Power Mate APM can operate in *any* Series 90-30 CPU or expansion baseplate (Series 90-30 release 3 or later). For limitations on the maximum number of Power Mate APM modules you can install per baseplate and system, refer to the section, *Module Specifications.*

The configuration files created by Logicmaster 90-30 / 20 / Micro configuration software must match the physical configuration of the modules.

To install the Power Mate APM in the rack, follow these steps:

1. Use the Logicmaster 90-30 / 20 / Micro software or the 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 module with the desired base slot and connector. Tilt the module upward so that the top rear hook of the module engages the slot on the baseplate.
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.
5. Refer to Tables 2-2 through 2-5 and Figures 2-4 through 2-7 for wiring requirements.
6. Power up the PLC rack. The Status LED of the Power Mate APM 30 will turn ON when the Power Mate APM has passed its power-up diagnostics.
7. Repeat this procedure for each Power Mate APM.
8. Configure the Power Mate APM module(s) as explained in Chapter 3.

Chapter 3 — Configuring the Power Mate APM

The Power Mate APM is configured using the Logicmaster 90-30/20/Micro Configuration software. Configuring the Power Mate APM is a two-part procedure consisting of:

- I/O Rack (Baseplate) Configuration
- Module Configuration

I/O Rack (Baseplate) Configuration

The Power Mate APM is configured using Logicmaster 90-30/20/Micro software in the same way as other Series 90-30 modules. The software is used to define the type and location of all modules present in the PLC baseplates. 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 for the Power Mate APM are defined, you can continue to the second part of Power Mate APM configuration, *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 90-30/20/Micro Configuration software. The Hand-Held Programmer can only configure the Module and Serial Port Configuration Data.

Power Mate APM module configuration data consists of 4 types:

- Module Configuration Data
- Serial Communications Port Configuration Data
- Master Source Configuration Data
- Axis Configuration Data

Module Configuration Data

During each CPU sweep certain data is automatically transferred both ways between the Power Mate APM and the CPU. Power Mate APM to CPU Interface data references the starting locations for the automatic transfers. The configuration parameters in Module Configuration Data are described in the following table.

Table 3-1. 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 APMs this parameter adds Position Error to %AI Data	DISABLED/ ENABLED	DISABLED	n/a
Fdback Type	Feedback Type[1]	ENCODER/ LINEAR RESOLVR [1] CUSTOM 1 [2] CUSTOM 2 [2] DIGITAL [1]	ENCODER	n/a
Ctl Loop	Control Loop Type	STANDARD[3] FOLLOWER CCL1 [2] CCL2 [2]	STANDARD	n/a
Servo Cmd	Servo Interface Type	ANALOG DIGITAL [1]	ANALOG	n/a
Motor1 Type Motor2 Type [4]	GE Fanuc Motor Type [1]	0 . . . 127	0 (no motor)	n/a
Motor1 Dir Motor2 Dir [4]	Motor direction for positive velocity command	POS NEG	POS	n/a

1 Reserved for future use; not implemented at this time.

2 For special-purpose applications only.

3 For information about the Standard Control Loop Type, refer to GFK-0840 *Series 90-30 Power Mate APM (PM-Power Mate APM) - Standard Mode User's Manual*.

4 Two axis Power Mate APM, IC693APU302H, or later.

Fdback Type. ENCODER selects A quad B (x4) incremental encoder input mode. LINEAR selects linear transducer (absolute feedback) input mode. 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. CUSTOM1 and CUSTOM2 configure the Power Mate APM inputs for special applications. DIGITAL feedback is only supported in Power Mate APM firmware revisions greater than 2.10. (Default = ENCODER).

Ctl Loop. STANDARD selects the normal Power Mate APM motion control loop. The STANDARD loop provides a velocity command proportional to position error, with optional Velocity Feedforward and Integrator gain terms. CCL1 and CCL2 (Customer Control Loops) are individually designed for special applications. FOLLOWER selects a control loop that allows ratio tracking of a master input with zero following error. (Default = STANDARD).

Servo Cmd. This parameter defines the type of command output provided to the servo system. ANALOG selects a Ç 10 volt velocity command for standard servo drives. DIGITAL selects a special digital output for GE Fanuc Digital AC servo drives. Digital AC servos are only supported in Power Mate APM firmware versions greater than 2.10. (Default = ANALOG).

Motor Type. Selects the type of GE Fanuc Digital AC servo motor to be used with the Power Mate APM. Refer to the FANUC AC servo drive and Powermate J documentation for additional information. Digital AC servos are only supported in Power Mate APM firmware versions greater than 2.10. (Default = 0)

Motor Dir. A configured motor direction of POS (Positive) defines the positive axis direction as encoder channel A leading channel B (encoder feedback) or motion away from the feedback transducer head (linear feedback). A configured motor direction of NEG (Negative) defines the positive axis direction as encoder channel B leading channel A (encoder feedback) or motion towards the feedback transducer head (linear feedback). In practice, the motor direction configuration allows the user to reverse the motion caused by all commands without reversing motor or encoder wires.

Serial Communications Port Configuration Data

The Power Mate APM can communicate through its Serial Communications Port (which supports the SNP protocol) on the faceplate of the Power Mate APM.

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 Port are described in the table below.

Table 3-2. 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 turn-around 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 char. consisting of A-F and 0-9. 1st char must be A-F	A00001	n/a

Master Source Configuration Data

These Master Source Configuration items are available when the Follower Control Loop is selected:

Table 3-3. Master Source Configuration Data

Configuration Parameter	Description	Values	Defaults	Units
Enc3 Hi Lim	Encoder 3 Position High Count Limit	–8,388,608...+8,388,607	+8,388,607	Counts
Enc3 Lo Lim	Encoder 3 Position Low Count Limit	–8,388,608...+8,388,607	–8,388,608	Counts
Home Positn	Home Position	–8,388,608...+8,388,607	0	Counts
Home Switch	Home Switch Enable	ENABLED / DISABLED	ENABLED	n / a
Anlg Max Vel	AnalogMaximum Velocity	0...+1,000,000	0	Counts / second

Enc3 Hi Lim. Encoder 3 Position High Count Limits (Counts). The maximum position which encoder 3 will reach before rolling over to the *Low Limit*.

Enc3 Lo Lim. Encoder 3 Position Low Count Limits (Counts). The minimum position which encoder 3 will reach before rolling over to the *High Limit*.

Home Position. Encoder 3 Home Position (Counts). The position to which *Encoder 3 Actual Position* is set at the completion of an encoder 3 Find Home Cycle.

Home Switch. Encoder 3 Home Switch Enable. Determines whether CTL06, the Encoder 3 Home Switch, is monitored during an encoder 3 Find Home Cycle.

Anlg Max Vel. Analog Input Maximum Velocity (Counts / second). The velocity which will be produced by 10V on the Analog Input. 0V always produces a velocity of 0 Counts per second.

Axis Configuration Data

Power Mate APM Axis Configuration Data consists of base values for configuration parameters. The configuration parameters are described only briefly here with the intent of providing enough information for entering them on the configuration screen. The table below describes the configuration data for the Power Mate APM when the FOLLOWER Control Loop is selected.

Table 3-4. Axis Configuration Data

Configuration Parameter	Description	Values	Defaults	Units[1]
User Units	User Units Value	1	1	n / a
Counts	Feedback Counts	1	1	n / a
Ratio A Val	Follower A / B Ratio A	–32,768...+32,767	1	n / a
Ratio B Val	Follower A / B Ratio B	+1...+32,767	1	n / a
Mstr Source[2]	Master Axis Source	ENC3 / INT, ENC2, ANALOG	ENC3 / INT	n / a
+Vlim	Master Velocity + Limit	0...+1000	1000	user units / ms
–Vlim	Master Velocity – Limit	–1000...0	–1000	user units / ms
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	Position Loop Time Constant	0,5...10,000	1000	ms
Vel at 10 V	Velocity for 10 V Output	400...1,000,000	4000	user units / sec
Vel FF %	Velocity Feed Forward	0...100	0	%
Intgr TC	Integrator Time Constant	0,10...10,000	0	ms
Intgr Mode	Integrator Mode	OFF / CONT / IN ZONE	OFF	n / a
Rev Comp	Reversal Compensation	0	0	n / a
DisDly	Drive Disable Delay	0...65,535	100	ms
Drive OK Sw	Drive OK Switch Enable	ENABLED / DISABLED	ENABLED	n / a
Jog Vel	Jog Velocity	1...1,000,000	+1000	user units / sec
Jog Acc	Jog Acceleration	1...8,388,607	+10000	user units / sec / sec
Jog Acc Mod	Jog Acceleration Mode	LINEAR	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
Fnl 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	Home Mode	HOMESW / MOVE+ / MOVE–	HOMESW	n / a

[1] Unit measure *user units* is the ratio of the parameters User Units to Counts. In Follower mode, both User Units and Counts are fixed at 1 and cannot be changed.

[2] ENC2 cannot be selected for Axis 2.

User Units and Counts. The ratio of *User Units* to *Counts* is 1:1 and cannot be changed. Thus 1 user unit always equals 1 count.

Ratio A and B Values. The ratio of A over B sets the master / slave gear ratio.

Axismotion (user units) = A / B x Master Reference (user units).

The range for A is –32,768 to +32,767 and B is 1 to +32,767. The ratio must be in the range of 32:1 to 1:32. When A is negative, the slave axis will move in the opposite direction from the master.

Mstr Source. Master Axis Source. The master reference for Axis 1 can be selected as encoder3 / internal master, encoder 2, or the Analog Input. A 1-axis Power Mate APM does not have an encoder 2 selection for master. The master reference for Axis 2, only applicable for 2-axis Power Mate APMs, may be encoder 3 / internal master or the Analog Input. Default: ENC3 / INT

+Vlim cts/ms. Master Velocity + Limit. The positive velocity for each axis is limited to this value. This allows velocity clamping or unidirectional operation.

-Vlim cts/ms. Master Velocity – Limit. The negative velocity for each axis is limited to this value. This allows velocity clamping or unidirectional operation.

Pos Err Lim. Position Error Limit (User Units). The maximum Position Error *(Commanded Position - Actual Position*) allowed when the Power Mate APM 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:

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

If Velocity Feedforward is not used, Position Error Limit can be set to a value approximately 20% higher than the position error required to produce a 10 volt command. *The Position Error* (User Units) required to produce a 10V command with 0% Velocity Feedforward is:

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

If Velocity Feedforward 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 this Error Limit Value is exceeded by more than 1000 counts. The Power Mate APM 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.

In Pos Zone. In Position Zone (User Units). When the magnitude of servo position error is less than or equal to this value and no motion is commanded, the *In Zone* status bit will be set. Note that *In Zone* can be set even when the follower slave axis is actively following a master input. Default: 10

Pos Loop TC. Position Loop Time Constant (milliseconds). 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. Setting Position Loop Time Constant to 0 will place the follower loop in *open loop* mode where only velocity Feedforward is used to produce the analog velocity command output. **Position Loop Time Constant will not be accurate unless the *Vel at 10v* value is set correctly.** Default: 1000

Vel at 10v. Actual Servo Velocity (User Units/second) for an Power Mate APM velocity command output of 10v. **This value must be configured correctly in order for the *Pos Loop TC* and *Vel FF %* factors to be accurate.** The Power Mate APM's *Force D/A Output* %AQ immediate command and the *Actual Velocity* %AI status word can be used to determine the proper configuration value. Default: 4000

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

Intgr TC. Integrator Time Constant (ms). This is the time constant for the position error integrator in milliseconds. This value indicates the length of time in which 63% of the position error will be removed. For example, if the Integrator Time Constant is 1000, or 1 second, the *Position Error* would be reduced to 37% of its initial value after 1 second. A value of zero turns off the integrator. The Integrator Time Constant should be 5 to 10 times greater than the Position Loop Time Constant to prevent instability and oscillation. Default: 0

Intgr Mode. Operating Mode for position error integrator. OFF means the integrator is not used. CONTINUOUS means the integrator runs continuously, even during servo motion. IN ZONE means the integrator only runs when the *In Zone* status bit is set. Default: OFF

Rev Comp. Reversal Compensation is set to 0 and cannot be changed.

DisDly (ms). 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. Default: 10000

Hi Limit. High Count Limit (User Units). When moving +, the *Actual Position* value will roll over to the low limit when this value is reached. Default: +8,388,607

Lo Limit. Low Count Limit (User Units). When moving –, the *Actual Position* value will roll over to the high limit when this value is reached. Default: –8,388,608

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 Vel can be set to a high value to allow the servo to quickly locate the Home Switch. Default: 2000

Home Mode. Find Home Mode. The method 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 or negative movement to the next marker at the Final Home Velocity. Default: HOMESW

Essential Configuration Parameters

To correctly configure the Power Mate APM, many configuration parameters must be properly set. This section explains how these parameters affect Power Mate APM 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 Power Mate APM and servo functions depend on this value being correct for proper operation. This should be the velocity of the axis when the Power Mate APM outputs 10 volts to the servo. See Chapter 5, *Servo System Startup Procedures* to determine the correct value. The allowed range for Velocity at 10 Volts is:

400 user units / sec v Velocity at 10 Volts v 1,000,000 user units / sec

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 Power Mate APM 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:

Bandwidth (rad / sec) = 1000 / Position Loop Time Constant (ms)

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

Gain (ipm / mil) = 60 / 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 or employ very crude positioning systems, an Open Loop Mode exists. This mode is selected by setting Position Loop Time Constant to zero. Note that in Open Loop Mode, the only way to generate motion is to program a non-zero Velocity Feedforward. The *Position Error* is no longer used to generate motion because *Position Error* is based on feedback and Open Loop Mode ignores all feedback.

Ratio A and B Values

The ratio A:B determines the relationship between the axis motion and the master source motion. The generated axis motion is the result of the master source motion multiplied by the ratio. Ratio A can be negative, which would result in the slave moving in the opposite direction as the master axis. The range for A:B ratios is 1:32 to 32:1.

Example of Ratio A and B Values

To make an axis follow the master encoder source, but at only forty percent of the master's velocity (and thus forty percent of the acceleration and distance), set Ratio A to 2 and Ratio B to 5. This ratio is 0.4 or forty percent. If the master source moves at a velocity of 500 counts per second a distance of 2000 counts, the axis following the master with ratio 2:5 would move at a velocity of 200 counts per second and only go a distance of 800 counts.

Important Configuration Considerations

The Integrator will attempt to remove all Position Error and can be used in two modes: IN ZONE and CONTINUOUS.

IN ZONE mode allows the integrator to run when the *In Zone* status bit is set. For Follower systems, *In Zone* is set when Jog or Move at Velocity is not commanded ***and*** when *Position Error* is $\leq$ the configured In Position Zone. Therefore IN ZONE mode can allow the integrator to operate while a slave axis is actively following a master input, provided that *Position Error* is $\leq$ In Position Zone.

CONTINUOUS mode causes the integrator to always operate and is useful for Follower systems which do not require rapid velocity changes. CONTINUOUS mode is not recommended for point to point positioning applications.

The Integrator Time Constant is the amount of time in which 63% of the *Position Error* will be removed. To prevent instability, the Integrator Time Constant must be five to ten times greater than the Position Loop Time Constant.

High and Low Count Limits determine where position rollover occurs. When a Count Limit is reached, the reported position *wraps around* to the opposite limit where it continues changing. The distance between High and Low Count Limits can be set to the distance used in a cyclic operation. At the end of each cycle, the High Count Limit would be reached and the next cycle would begin at the Low Count Limit. High and Low Count Limits are needed for most rotary applications.

Velocity Limits specify the maximum velocity at which an axis will follow the master input. If the master velocity exceeds the limit, the axis will continue at the limit, the *In Velocity Limit* %I bit will be set, and an error will be reported. This error indicates that the follower axis is no longer locked to the master at the specified A/B ratio. An exception is when a Velocity Limit is set to zero. Instead, no motion in that direction is generated and no error is generated. The Power Mate APM will not accept a configuration with both Velocity Limits set to zero.

The Drive OK Switch is an external hardware switch from the system. The switch state is returned in status bits CTL05 (axis 1) and CTL07 (axis 2). Drive OK turning OFF usually indicates a system problem requiring corrective action. If the configuration for the Drive OK switch is enabled, the Power Mate APM will not generate motion while the Drive OK switch is OFF. If the Drive OK switch turns OFF while the axis is moving, the Power Mate APM will command a Fast Stop and report an error.

The Servo Drive Disable Delay specifies how long the Power Mate APM will wait after zero velocity is commanded before the Drive Enable Output is deactivated. Error conditions which instantly zero the velocity command (such as loss of encoder quadrature) should allow the servo to stay activated long enough to decelerate the axis to a complete stop. Thus to allow the Power Mate APM to command a fast stop without coasting of the axis, the Disable Delay should be longer than the deceleration time of the servo from maximum speed.

Chapter 4

Automatic Data Transfers

This chapter defines the data that is transferred automatically each sweep, without user programming, between the CPU and the Power Mate APM. This data is categorized as follows.

Input Status Data (Transferred from Power Mate APM to CPU)

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

Output Command Data (Transferred from CPU to Power Mate APM)

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

%I Status Bits

The following %I Status Bits are transferred automatically from the Power Mate APM to the CPU each sweep. The actual addresses of the Status Bits depend on the starting address configured for the %I references. See Table 3-1, Module Configuration Data. The bit numbers listed in the following table are offsets to this starting address.

Table 4-1. %I Status Bits for the 1-Axis Power Mate APM (IC693APU301)

Bit * Offset	Description		Bit * Offset	Description	
00	Axis Enabled	Axis 1	16	Front Panel Input CTL01 Status	
01	Position Valid	Axis 1	17	Front Panel Input CTL02 Status	
02	Drive Enabled	Axis 1	18	Front Panel Input CTL03 Status (Home 1)	
03	Program Active	Axis 1	19	Front Panel Input CTL04 Status	
04	Command Move	Axis 1	20	Front Panel Input CTL05 Status (Drive 1 OK)	
05	In Zone	Axis 1	21	Front Panel Input CTL06 Status (Home 3)	
06	In Velocity Limit	Axis 1	22	Front Panel Input CTL07 Status	
07	In Error Limit	Axis 1	23	Front Panel Input CTL08 Status	
08	Reserved		24	Configuration Complete	
09	Reserved		25	Position Valid Encoder 3	
10	Reserved		26	Master Velocity Mismatch	
11	Reserved		27	Master Velocity Mismatch Sign	
12	Reserved		28	Follower Enabled	Axis 1
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 4-2. %I Status Bits for the 2-Axis Power Mate APM (IC693APU302)

Bit * Offset	Description		Bit * Offset	Description	
00	AxisEnabled	Axis1	16	Front Panel Input CTL01 Status	
01	Position Valid	Axis1	17	Front Panel Input CTL02 Status	
02	DriveEnabled	Axis1	18	Front Panel Input CTL03 Status (Home 1)	
03	Program Active	Axis1	19	Front Panel Input CTL04 Status (Home 2)	
04	Command Move	Axis1	20	Front Panel Input CTL05 Status (Drive 1 OK)	
05	In Zone	Axis1	21	Front Panel Input CTL06 Status (Home 3)	
06	In VelocityLimit	Axis1	22	Front Panel Input CTL07 Status (Drive 2 OK)	
07	In ErrorLimit	Axis1	23	Front Panel Input CTL08 Status	
08	AxisEnabled	Axis2	24	ConfigurationComplete	
09	Position Valid	Axis2	25	Position Valid Encoder 3	
10	DriveEnabled	Axis2	26	Master VelocityMismatch	
11	Program Active	Axis2	27	Master Velocity Mismatch Sign	
12	Command Move	Axis2	28	FollowerEnabled	Axis1
13	In Zone	Axis2	29	FollowerEnabled	Axis2
14	In VelocityLimit	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 Power Mate APM 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. The *Position Valid* bit can be cleared by PLC commands. To do this, the PLC should send the %Q commands for *Find Home* and *Abort* on the same PLC sweep.

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

Program Active. The *Program Active* status bit for each axis indicates that a Motion Program (1–6) or a %AQ Move Command is executing on that axis. On a two axis Power Mate APM, executing a multi-axis program will set both *Program Active* bits.

Command Move. The *Command Move* status bit represents only the state of the internal motion generator for each follower axis. Therefore, a *Jog*, *Move at Velocity*, *Find Home* , or *Execute Program* Command will set *Command Move*. *Command Move* is also set when the follower acceleration ramp control is accelerating to the master velocity command or decelerating from the master command. Axis Follower motion caused by the external or internal master input will not set *Command Move*.

In Zone. The *In Zone* status bit is set if *Command Move* is OFF and the follower axis position error is less than the configured *In Zone* value. Therefore, *In Zone* may always be set in a follower system which uses velocity feedforward to achieve zero following error.

In Power Mate APM firmware release 2.10, *In Zone* is unconditionally set whenever the Follower is enabled and Command Move is off.

In Velocity Limit. The *In Velocity Limit* status bit is set if the velocity requested by the external or internal master input exceeds the configured *Velocity Limits*. Therefore, *In Velocity Limit* is an indication that the follower axis is no longer locked to its master input. A Status Error code is reported when *In Velocity Limit* is set.

An exception exists when one of the Velocity Limits is set to 0. A zero Velocity Limit indicates unidirectional following motion so no error is generated. For example, if the + Velocity Limit = 0 and + Counts are input, the *In Velocity Limit* Status bit is set, but no Status Error code is reported.

If one of the Follower Control Loop Velocity Limits is set to zero, the position error integrator is inhibited from generating motion in the direction of the Velocity Limit. This feature allows unidirectional systems to operate properly with the position error integrator enabled.

In Error Limit. The *In Error Limit* status bit is set if the position error required to run the follower servo is greater than the configured *Error Limit* value. Whenever *In Error Limit* is set, all internally or externally generated motion commands will be ignored until *Position Error* becomes less than *Error Limit*. Therefore *In Error Limit* is an indication that the follower axis is no longer locked to its master input. A Status Error code will be reported when *In Error Limit* is set and the *Enable Follower* %Q bit is set.

Faceplate Input Status CTL01-08 These inputs always indicate the state of the external input devices connected to the Power Mate APM faceplate terminals CTL01-08. These inputs (as well as CTL09-CTL12 from the PLC %Q table) may be tested by the Power Mate APM during execution of *Wait* and *Conditional Jump* commands. CTL01-08 can also be selected as a follower enable trigger.

Several inputs can serve alternate purposes as shown in the following table.

CTL01	Axis2 Encoder Marker Channel (2 axis Power Mate APM only)
CTL02	Encoder 3 Marker Channel (2 axis Power Mate APM only)
CTL03	Axis 1 Home Switch input
CTL04	Axis 2 Home Switch Input
CTL05	Axis 1 Drive OK Input
CTL06	Encoder 3 Home Switch Input
CTL07	Axis 2 Drive OK Input
CTL08	–

CTL01 and CTL02 are assigned to different faceplate inputs on the 1 axis Power Mate APM and 2 axis Power Mate APM. Refer to the faceplate connection and wiring diagrams in Chapter 2.

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 Power Mate APM. 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 Power Mate APM. The status bit can then be monitored by the PLC. If the bit is ever cleared, then the Power Mate APM has been reset or reconfigured and the PLC should resend all necessary %AQ configuration commands before setting the bit again.

Position Valid Encoder 3. Position valid indication for Encoder 3 Input.

Master Velocity Mismatch. The *Master Velocity Mismatch* indication is set when the difference between the *Encoder 3 Velocity* and the *Internal Master Velocity* is more than 128 cts / sec. This %I bit allows the PLC to detect synchronization of the internal and external master velocities before switching the *Select Internal Master* %Q discrete command bit.

Master Velocity Mismatch Sign. The *Master Velocity Mismatch Sign* bit indicates the sign of the velocity change needed to make *Internal Master Velocity = Encoder 3 Velocity.* "0" indicates that *Internal Master Velocity* must change in the positive direction. "1" indicates that *Internal Master Velocity* must change in the negative direction.

Follower Enabled. This status bit indicates when the Follower is enabled for the axis. The Follower is enabled by the %Q *Enable Follower* bit and an optional CTL01-CTL08 faceplate trigger input. If follower ramp acceleration control is active, when Follower Enabled turns on the axis accelerates to the master velocity command, and when it turns off, the axis decelerates to zero master velocity command.

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 Power Mate APM. *PLC Control Active* is cleared only when the Status screen in the Motion Programmer is used instead of the PLC to control the Power Mate APM, a capability not yet implemented.

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

%AI Status Words

The following %AI Status Words are transferred automatically from the Power Mate APM 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 4-3. %AI Status Words for the 1-Axis Power Mate APM (IC693APU301)

Word* Offset	Description	
000	Status Code	
001	Reserved (Command Block Number)	Axis1
002-003	Commanded Position**	Axis1
004-005	Actual Position	Axis1
006-007	Actual Position	Encoder 3
008-009	Commanded Velocity	Axis1
010-011	Actual Velocity	Axis1
012-13	Position Error***	Axis1
014	Analog Input Value	

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

** This may be configured to display Commanded Position, Program Command Position, or other data types. See %AQ command *ConfigurationModes* and *Select Return Data*.

*** If Module Configuration parameter *%AI Pos Err* is DISABLED, Actual Velocity for Encoder 3 will be reported instead of Axis 1 Position Error.

Table 4-4. %AI Status Words for the 2-Axis Power Mate APM (IC693APU302))

Word* Offset	Description	
000	Status Code	
001	Reserved (Command Block Number)	Axis1
002-003	Commanded Position**	Axis1
004-005	Actual Position	Axis1
006-007	Actual Position	Encoder 3
008-009	Commanded Velocity	Axis1
010-011	Actual Velocity	Axis1
012-13	Position Error	Axis1
014	Analog Input Value	
015	Reserved (Command Block Number)	Axis2
016-017	Commanded Position	Axis2
018-019	Actual Position	Axis2
020-021	Actual Velocity	Encoder 3
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.

** This may be configured to display Commanded Position, Program Command Position, or other data types. See %AQ command *ConfigurationModes* and *Select Return Data*.

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

For a list of Power Mate APM error codes refer to Appendix A, Error Codes.

Commanded Position. *Commanded Position* represents the combination of all position commands from the follower *Master Axis* inputs, the axis *Move at Velocity* or *Jog* commands, or *Stored Program* commands.

Note

Alternate data may be returned in this location. Refer to %AQ Immediate Commands *Select Return Data* and *Configuration Mode* for details. Program Command Position defined below is one selection.

Program Command Position. *Program Command Position* is the reference position for stored motion programs. It is updated only by internally generated MOVE commands and does NOT include commands from the Master input. The %AQ *Configuration Modes* command (45h) or *Select Return Data* (40h) may be used to configure the Power Mate APM to return *Program Command Position* instead of *Commanded Position*.

Actual Position. *Actual Position* (user units) is a value maintained by the Power Mate APM to represent the physical position of the axis. It is set to an initial value by the *Set Position* command or to *Home Position* by the *Find Home* cycle. It is updated by the motion of the feedback device. For Encoder 3, units are counts.

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

Commanded Velocity. *Commanded Velocity* (user units / sec) is a value generated by the Power Mate APM that indicates the instantaneous velocity commanded by a *Jog, Move at Velocity, Find Home*, or *Execute Program* command. Axis motion caused by the Master Axis input is not reflected in *Commanded Velocity.*

Actual Velocity. *Actual Velocity* (user units / sec) is a value maintained by the Power Mate APM that is derived from the feedback device. Therefore, it represents the velocity of the axis movement. For Encoder 3, units are counts / sec.

Position Error. *Position Error* (user units) is equal to *Commanded Position* - *Actual Position* for each follower axis.

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

%Q Discrete Commands

The following %Q Outputs representing Discrete Commands are sent automatically to the Power Mate APM 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 4-5. %Q Discrete Commands for the 1-Axis Power Mate APM (IC693APU301)

Bit * Offset	Description		Bit * Offset	Description
00	Abort All Moves	Axis1	16	CTL09 Bit Control
01	Reserved	Axis1	17	CTL10 Bit Control
02	Enable Drive	Axis1	18	CTL11 Bit Control
03	Find Home	Axis1	19	CTL12 Bit Control
04	Jog Plus	Axis1	20	Reserved
05	Jog Minus	Axis1	21	Execute Motion Program 1
06	Reserved	Axis1	22	Execute Motion Program 2
07	Enable Follower	Axis1	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	Find Home Encoder 3
12	Reserved		28	Select Internal Master
13	Reserved		29	Reserved
14	Reserved		30	Reserved
15	Reserved		31	Clear Error

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

Table 4-6. %Q Discrete Commands for the 2-Axis Power Mate APM (IC693APU302)

Bit * Offset	Description		Bit * Offset	Description
00	Abort All Moves	Axis1	16	CTL09 Bit Control
01	Reserved	Axis1	17	CTL10 Bit Control
02	Enable Drive	Axis1	18	CTL11 Bit Control
03	Find Home	Axis1	19	CTL12 Bit Control
04	Jog Plus	Axis1	20	Reserved
05	Jog Minus	Axis1	21	Execute Motion Program 1
06	Reserved	Axis1	22	Execute Motion Program 2
07	Enable Follower	Axis1	23	Execute Motion Program 3
08	Abort All Moves	Axis2	24	Execute Motion Program 4
09	Reserved	Axis2	25	Execute Motion Program 5
10	Enable Drive	Axis2	26	Execute Motion Program 6
11	Find Home	Axis2	27	Find Home Encoder 3
12	Jog Plus	Axis2	28	Select Internal Master
13	Jog Minus	Axis2	29	Reserved
14	Reserved	Axis2	30	Reserved
15	Enable Follower	Axis2	31	Clear Error

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

Abort All Moves. This command is used to abort a *Jog*, *Move at Velocity*, or *Find Home* command. *Abort All Moves* has no effect on motion generated by the external or internal Master source. *Abort All Moves* uses the current *Jog Acceleration* and *Jog Acceleration Mode*.

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).

Enable Drive must be turned off whenever power is removed from the servo drive. If this is not done, any servo drift could cause a rapid jump of the servo when power is reapplied to the drive.

Find Home (Axis 1, 2). This command causes the Power Mate APM 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 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 end of the Home cycle.

Refer to Home Cycle in Chapter 6, *Power Mate APM Motion Control* for a description of the Find Home cycle.

Jog Plus. When this command bit is ON, the axis moves in the positive direction at the current *Jog Acceleration* and *Jog Velocity* as long as the *Jog Plus* command is maintained.

Jog Minus. When this command bit is ON, the axis moves in the negative direction at the current *Jog Acceleration* and *Jog Velocity* as long as the *Jog Minus* command is maintained.

Enable Follower. When this bit is set and the %I *Follower Enabled* status bit indicates the Follower is enabled, motion commanded by the external or internal master will act as an input to the follower loop. Clearing *Enable Follower* disconnects the follower loop from the master source. *Jog, Move at Velocity,* and *Execute Program* commands will be allowed regardless of the state of *Enable Follower.* When the Follower is enabled, *Jog, Move at Velocity,* or *Execute program* commands will be superimposed on the master command. *Find Home* is not allowed unless *Enable Follower* is cleared.

Execute Motion Program 1-6. 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.

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.

Find Home (Encoder 3). Setting this bit does not cause any motion; instead it initializes the Encoder 3 hardware to wait for a marker pulse transition when the Home Switch 3 input is ON. When the Home Switch / Marker condition is satisfied, the *Encoder 3 Actual Position* will be set to the configured *Home Position* and the *Position Valid Encoder 3* status bit will be set. The Home Switch input may be disabled by a configuration selection, allowing the Encoder 3 Home Cycle to only wait for a marker transition. *Find Home* must be maintained ON until Home is located. Clearing *Find Home* (Encoder 3 only) will cause the Encoder 3 Home cycle to abort without setting *Position Valid Encoder 3.*

Select Internal Master. This bit switches the master axis source from Encoder 3 to the Internal Master Velocity generator. The *Set Master Velocity* %AQ Immediate command can be used to change the velocity of the internal master.

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 a *Drive OK* switch error) will not be cleared and must be cleared by some other corrective action .

%AQ Immediate Commands

Each PLC sweep, 6 words of data are automatically transferred from the CPU %AQ data to the Power Mate APM. These six words are used to send Immediate Commands from the PLC to the Power Mate APM. The first three words, offsets 0 through 2, are dedicated to axis 1 of the Power Mate APM. The second three words, offsets 3 through 5, are dedicated to axis 2 of the 2 axis Power Mate APM. Thus one command may be sent to each axis of the Power Mate APM every sweep. Table 4-7 defines the axis dependent commands, those that will affect only the axis on which they are commanded.

Table 4-8 defines the axis independent commands. Axis independent commands may be sent using either or both sets of three words.

In a 1-axis Power Mate APM, the only effective commands for the second set of %AQ words are Force D / A Output (Analog Output 2) and the axis independent commands.

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

The 6-byte format for the Immediate Commands is defined in Tables 4-7 and 4-8. The actual addresses of the Immediate Command Words depend on the starting address configured for the %AQ references. See Table 3-1, Module Configuration Data. **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 a Set Position 3,400,250 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 Power Mate APM would be:

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

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

Table 4-7. Axis Dependent Immediate Commands Using the 6-Byte Format

Word 2		Word 1		Word 0 *		† Axis 1 Command
Word 5		**Word 4**		**Word 3 ***		**† Axis 2 Command**
Byte 5	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Byte 0**	**Immediate Command Definition**
xx	xx	xx	xx	00	00h	**Null**
xx	xx	**	Incr	00	21h	**Position Increment Without Position Update** Incr = –128 ... +127 User Units
Velocity				00	22h	**Move At Velocity** Vel. = –1,000,000 ... + 1,000,000 User Units / sec
Position				00	23h	**Set Position** Pos. = –8,388,608 ... + 8,388,607 User Units
xx	xx	D / A Output		00	24h	**Force D/A Output** D / A Output = –32,000 ... + 32,000
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 ... + 1,000,000 User Units / sec
Acceleration				00	29h	**Jog Acceleration** Acc. = 32 ... + 8,388,607 User Units / sec / sec
xx	xx	Time Constant		00	2Ah	**Position Loop Time Constant** Time Constant = 0, 5 ... 10,000 ms
xx	xx	xx	VFF%	00	2Bh	**Velocity Feedforward** VFF% = 0 ... 100%
xx	xx	Integr. TC		00	2Ch	**Integrator Time Constant** Time Constant = 0, 10 ... 10,000 ms
Ratio B		Ratio A		00	2Dh	**Follower A/B Ratio** Ratio A = –32,768 ... +32,767 Ratio B = +1 ... +32,767

* The word numbers represent an offset to the starting address for % AQ references.
** Only 00 of FFh are acceptable.
xx = don’t care

Table 4-8. Axis Independent Immediate Commands Using the 6-Byte Format

Word 2		Word 1		Word 0		
Word 5		Word 4		Word 3		
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Immediate Command Definition
Velocity				00	30h	**Internal Master Velocity** Vel = –1,000,000 ... + 1,000,000 counts / sec
Position				00	31h	**Set Encoder 3 Position** Pos = –8,388,608 ... +8,388,607 counts
xx	xx	Offset		Mode	40h	**Select Return Data**
xx	xx	xx	Input#	00	41h	**External Input Enable Follower** Input #: 1 - 8 = CTL01 - CTL08; 00 = none (disabled)
xx	xx	Make-Up Time		00	42h	**Follower Ramp Distance Make-Up Time** Active Range = 0, 10 ... 32000 ms
00	00	Op Modes	Pos Data	00	45h	**Configuration Modes** (see definition below)
xx	xx	xx	xx	00	49h	**Set Configuration Complete**
Parameter Data				Par #h	50h	**Load Parameter Immediate** Par # = 0 ... 255 Parameter Data = Range depends on parameter usage.

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.

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 Power Mate APM 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 current *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 value without moving the axis. The *Commanded Position* and *Actual Position* values will both be changed so that no motion command will be generated. The *Actual Position* will be set to the value designated and the *Commanded Position* will be set to the value + *Position Error. Set Position* cannot be performed when the %I *Command Move* bit is ON. The position value must be within the *End of Travel Limits* and the *Count Limits* or a status error will be generated. *Position Valid* indication is set after a successful Set Position command.

Force D/A Output. This command forces the Velocity Command D / A outputs at the I / O connector to a constant output level. The *Force D/A Output* command sets the output to any value within the 10 volt range. This command will not be allowed if *Enable Follower* is set or if any other motion is in progress.

A Force D / A Output command is the only continuous %AQ command. It must remain continuously in the %AQ data for proper operation. Thus any other %AQ command will remove the Force D / A command. A command of +32,000 will produce an output of +10 V, and –32,000 will produce –10 V.

In a 1-axis follower Power Mate APM, placing this command into axis two %AQ data will automatically enable axis 2. Axis 2 enabled is indicated by the EN2 LED turning on.

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 Power Mate APM 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 Power Mate APM 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 Power Mate APM is power cycled or the PLC CPU is reset for any reason, the configured *In Position* zone value 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 as defined below:

Move Type (byte 1):

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

80 = Abs, Pmove, Linear
81 = Abs, Cmove, Linear
90 = Abs, Pmove, S
91 = Abs, Cmove, S
C0 = Inc, Pmove, Linear
C1 = Inc, Cmove, Linear
D0 = Inc, Pmove, S
D1 = Inc, Cmove, S

All of the moves defined above can be sent and executed when the Power Mate APM is configured for Standard mode. In the Follower mode **only** the moves marked **Linear** should be executed.

This move is executed as a single move motion program, and the Program Active %I indication will be on while the move is in process. 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.

Note that for the Follower mode configuration, an absolute move type will act just like an incremental move, if the Follower is Enabled. It will operate as an absolute move **only** if the Follower is **not** enabled. This is because an absolute move reference position of 0 is used when the move is sent with the Follower enabled and if the Follower is not enabled, the actual position is used for the reference.

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.

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*.

Position Loop Time Constant. (milliseconds) This command allows the servo position loop time constant to be changed from the PLC. If this command is not sent, then the initial value configured with the Logicmaster 90 configuration software is used.

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

Integrator Time Constant. (milliseconds) This command sets the *Integrator Time Constant* for the position error integrator. The value specifies the amount of time in which 63% of the *Position Error* will be removed. *The Integrator Time Constant* should be 5 to 10 times greater than the *Position Loop Time Constant* to prevent instability and oscillation.

Follower A/B Ratio. This command allows the PLC to update the A / B ratio used in each follower loop. A is a 16-bit signed integer with a minimum value of –32,768 and a maximum of +32,767. B is a 16-bit integer with a minimum value of 1 and a maximum value of +32767. The ratio of A / B must be in the range 32:1 to 1:32 or a status error will be generated.

Internal Master Velocity. (counts / second) This axis independent command loads the internal velocity generator which may be used as an alternate source for the master input. The maximum allowed velocity will be 1,000,000 counts / second.

Set Encoder 3 Position. (counts) This axis independent command allows the position of Encoder 3 to be set without using a home cycle.

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 Program Command Position, Parameter Register contents, and the Power Mate APM 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:

Selected Return Data	Data Mode	Data Offset
Commanded Position	00h	not used
Program Command Position	01h	not used
Parameter Register	08h	Parameter Number (0–255)
APM Firmware Revision	10h	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.

Power Mate APM 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.

External Input Enable Follower. This command selects the input that will trigger the follower axis to start following the master command.

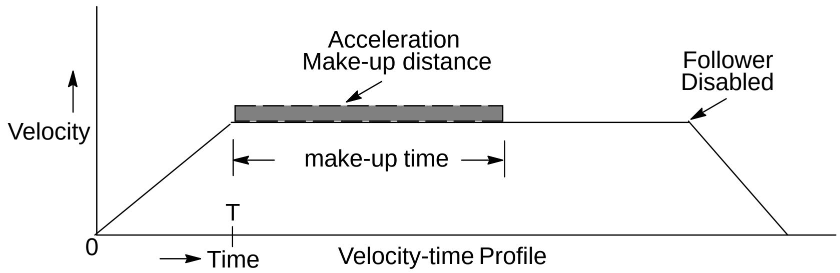
When byte 2 = 00, the external input feature is disabled and the follower is Enabled / Disabled in the normal manner by the ON / OFF state of the Enable Follower %Q command.

Setting Byte 2 = 01 - 08 selects the input (CTL01 - CTL08) that is used to trigger the axis to start following when the Enable Follower %Q bit is ON. When the Enable Follower %Q bit is turned OFF, the follower is disabled and will not be enabled until the Enable Follower %Q bit is turned ON and the selected trigger input turns on.

Follower Ramp Distance Make-Up Time. When the Follower Ramp feature has been selected and the follower is enabled, the following axis is ramped up to the Master velocity at the current jog acceleration rate without losing any of the master counts during ramp-up if the make-up time is set for a non-zero value. The master counts that cannot be sent during ramp-up are stored and when the follower axis reaches the master velocity, they will be inserted at a rate determined by this configured distance make– up time.

The velocity-time profile below indicates the make-up time period immediately after the axis ramp-up. The total time to achieve following velocity can be calculated as:

Time to exact following velocity = [Velocity/Acceleration] + distance make-up time.



Velocity-time Profile
(Follower Enabled at t = 0)

The following table indicates the amount of velocity overshoot for different values of distance make-up time (where *T = acceleration time in seconds = Velocity/Acceleration*).

For most applications the distance make-up time (in seconds) should be set so that it is >= 5T so the velocity overshoot is <= 10%. The resulting velocity including the overshoot must not exceed the servo drive maximum velocity to ensure that no master command counts are lost.

For applications where lost counts do not matter, set the distance make-up time = 0 to prevent Velocity overshoot.

Distance Make-Up Time =	Velocity Overshoot =
1T	50 %
2T	25 %
3T	16.7 %
4T	12.5 %
5T	10 %
6T	8.3 %

Configuration Modes. This command is used to configure the Follower Ramp features and the axis command position data returned in the %AI data Commanded Position location. **This command must be sent only when the Follower is not enabled.**

Configuration definitions are as follows.

Pos Data (Byte 2): specifies data returned in %AI Commanded Position location

00h = normal (Commanded Position)

01h = Program Command Position

Op Modes (Byte 3): specifies Follower operation modes

00h = normal

01h = Ramp

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 Power Mate APM. 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 Power Mate APM configuration data programmed with the PLC configuration software.

Load Parameter Immediate. This command is executed from the PLC to immediately change a Power Mate APM parameter value. Parameters are only used by motion programs. A command for each parameter change is required.

Chapter 5 Follower System Startup Procedures

Startup Procedures

1. Connect the motor to the servo amplifier according to the manufacturer's recommendations.
2. Connect the Power Mate APM **Drive Enable** Relay and **Velocity Command** outputs to the servo amplifier. Connect the position feedback device (A Quad B Encoder) to the Power Mate APM encoder inputs.

Note

If these connections are incorrect or there is slippage in the coupling to the Feedback Device, an ***Out of Sync*** error condition can occur when motion is commanded.

3. If a *Drive OK* relay contact is used (24 Vdc), wire it to the correct Power Mate APM input. Otherwise the *Drive OK* input must be disabled using Configuration Software. If a **Home** switch is used (24 Vdc), wire it to the correct Power Mate APM input. The **Home** switch must be wired so that it is ALWAYS ON when the axis is on the negative side of home and ALWAYS OFF when the axis is on the positive side of home.
4. Use the Configuration Software to set the desired configurable parameters. Store the configuration to the PLC.
5. Clear the program from the PLC, turn off all Power Mate APM %Q bits and place the PLC in RUN mode. Monitor the %I bits for **CTL03 (Home 1), CTL04 (Home 2), CTL06 (Home 3)**, **CTL05 (Drive OK 1)** and **CTL07 (Drive OK 2)** and confirm that each bit responds to the correct switch.
6. Turn on the %Q *Enable Drive* bit and place the command code for *Force D/A Output* 0 in the %AQ table. Confirm that the servo amplifier is enabled. If the motor moves, adjust the amplifier zero offset until the motor stops moving. Note: The %Q *Enable Drive* bit must be maintained on in order for the *Force D/A Output* command to function.
7. Send the command code for *Force D/A Output* +3200 (+1.0v). Confirm that the motor moves in the desired POSITIVE direction and the *Actual Velocity* reported in the Power Mate APM %AI table is POSITIVE. If the motor moves in the wrong direction, consult the manufacturers instructions for corrective action. The Motor Dir parameter in the Configuration Software can also be used to swap the positive and negative axis directions (Power Mate APM Firmware 2.10 or later). If the motor moves in the POSITIVE direction but the Power Mate APM reports that *Actual*

Velocity is NEGATIVE, then the encoder channel A and channel B inputs must be swapped.

8. Record the actual motor velocity reported by the Power Mate APM with a 1.0 volt velocity command. Multiply this velocity by 10 and update the *Vel @ 10V* entry in the Power Mate APM configuration. Initially set the *Pos Loop TC* configuration entry to a high value (typically 100 to 1000 ms).

9. Turn on the %Q *Jog Plus* bit. Confirm that the servo moves in the proper direction and that the *Actual Velocity* reported by the Power Mate APM in the %AI table matches the configured jog velocity.

10. With the *Drive Enabled* %Q bit on and no servo motion commanded, adjust the servo drive zero offset for zero *Position Error.* The integrator should be OFF during this process.

11. Check for proper operation of the *Find Home* cycle by momentarily turning on the %Q *Find Home* bit (the *Drive Enabled* %Q bit must also be maintained on). The axis should move towards the Home Switch at the configured *Find Home* velocity, then seek the Encoder Marker at the configured *Fnl Home* velocity. If necessary, adjust the configured velocities and the location of the **Home Switch** for consistent operation. The final **Home Switch** transition MUST occur at least 10 ms before the Encoder Marker Pulse is encountered. The physical location of *Home Position* can be adjusted by changing the *Home Offset* value with the Configuration Software.

12. Monitor servo performance and use the %Q *Jog Plus* and *Jog Minus* bits to move the servo motor in each direction. The *Pos Loop Time Constant* can be temporarily modified by placing the correct command code in the %AQ table. For most systems the *Position Loop Time Constant* can be reduced until some servo instability is noted, then increased to a value approximately 50% higher. Once the correct time constant is determined, the Power Mate APM configuration should be updated using the Configuration Software. *Velocity Feedforward* can also be set to a non zero value (typically 80 - 90 %) for optimum servo response.

Note

For proper servo operation, the Configuration entry for *Vel at 10v* MUST be set to the actual servo velocity (in User Units / sec) caused by a 10 V command.

System Troubleshooting Hints

1. The Power Mate APM Follower requires PLC firmware release 3.52 or greater and Logicmaster 90-30 / 20 / Micro software release 4.0 or greater.

2. The default Power Mate APM configuration for the Drive OK inputs is ENABLED. Therefore 24 vdc must be applied to the Drive OK inputs or the Power Mate APM will not operate. If Drive OK inputs are not used, the Power Mate APM configuration should be set to Drive OK inputs DISABLED.

3. The ENABLE DRIVE %Q control bit must be set continuously to **1** or no motion other than Jogs will be allowed. If no STOP errors have occurred, the DRIVE ENABLED %I status bit will mirror the state of the ENABLE DRIVE %Q bit. A STOP error will turn off DRIVE ENABLED even though ENABLE DRIVE is still a **1**. The error condition must be corrected and the CLEAR ERROR %Q control bit turned on for one PLC sweep to re-enable the drive.

4. If the ERROR %I status bit is **1** and the AXIS ENABLED and DRIVE ENABLED %I status bits are **0**, then a STOP error has occurred (Status LED flashing fast). In this state, the Power Mate APM will not respond to any commands other than the CLEAR ERROR %Q control bit.

   The CLEAR ERROR %Q control bit uses one-shot action. Each time an error is generated, the bit must be set to **0** then set to **1** for at least one PLC sweep to clear the error.

5. The *CFG OK* LED must be ON or the Power Mate APM will not respond to PLC commands. If the LED is OFF then a valid Power Mate APM configuration has not been received from the PLC.

Follower Control Loop Block Diagram

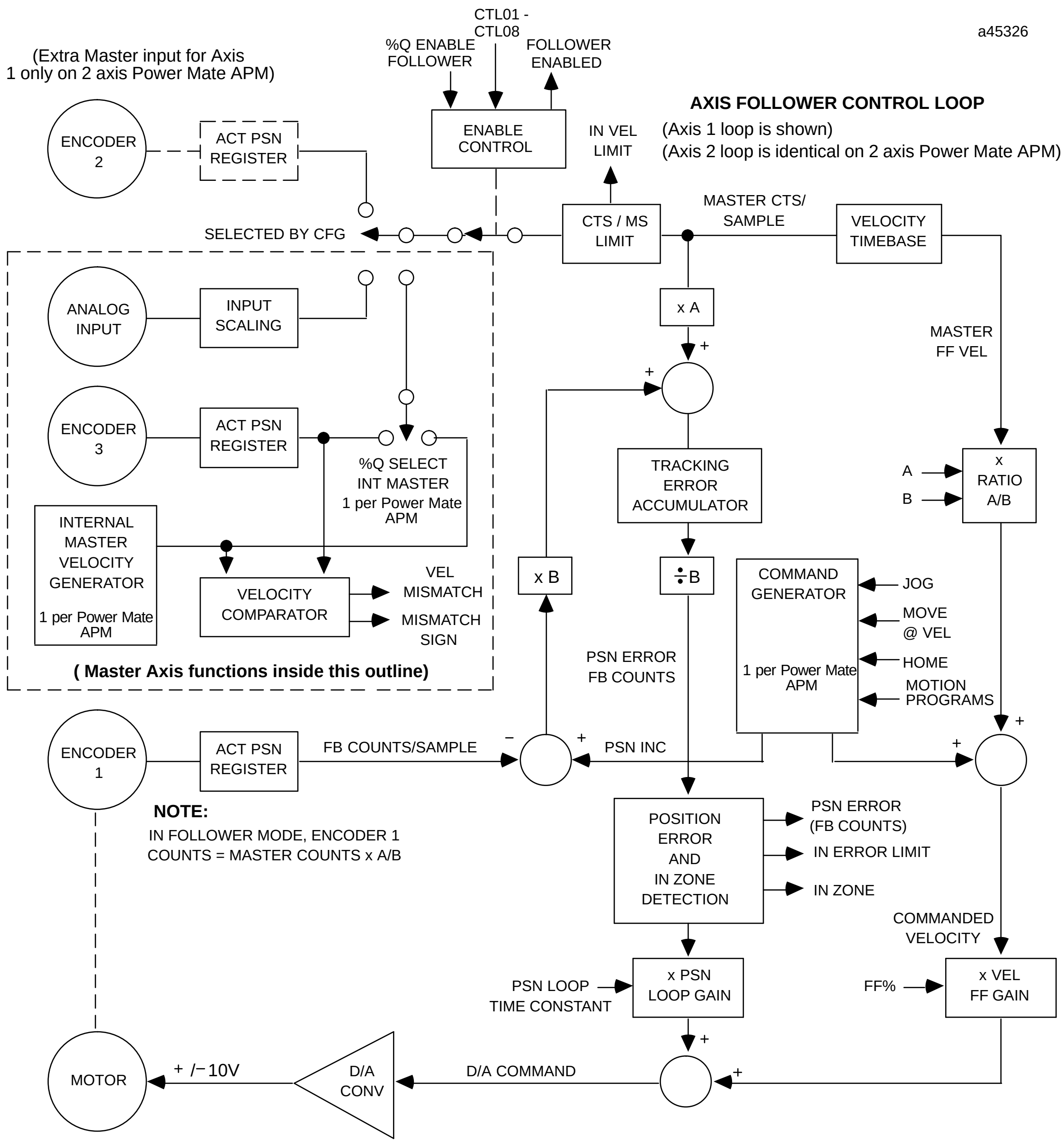


Figure 5-1. Axis Follower Control Loop-Block Diagram

Chapter 6 Power Mate APM Motion Control

This chapter provides practical information on Power Mate APM Motion Control with a number of examples. The main topics discussed are:

- Non-Programmed Motion
- Programmed Motion
- Follower Motion
- Combined Motion

Non-Programmed Motion

The Power Mate APM 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 Outpu*t, any other commanded motion, programmed or non-programmed, will generate an error. The only exception is *Position Increment* which may 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.

Power Mate APM Home Cycle

A Home Cycle establishes the Home Position for systems with an incremental feedback device that also provides a marker pulse. 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 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 APM 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 APM 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 APM sets the *Position Valid* %I bit to indicate the Home Cycle is complete.

Jogging with the Power Mate APM

The *Jog Velocity, Jog Acceleration*, and *Jog Acceleration Mode* are configurable in the Power Mate APM. 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* may be performed when no other motion is commanded. The *Enable Drive* %Q bit does not need to be ON to Jog, but it may be. Turning on a *Jog* %Q bit will automatically close the Enable Relay, and turn on the *Drive Enabled* %I bit.

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 may be cleared by sending a NULL command, or changed as desired. *Move at Velocit*y will not function unless the servo drive is enabled (%Q *Enable Drive* and %I *Drive Enabled* are set). Also, the *Drive OK* input must be ON if it is enabled.

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 a Power Mate APM 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 APM 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 APM 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 APM to accept it. The Power Mate APM detects a *Move at Velocity* command when the %AQ values change.

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

To obtain a specific output voltage from the Power Mate APM to the servo, use the *Force D/A* immediate command. Command values between –32,000 and +32,000 are scaled to the range –10V to +10V. Thus +3,200 is +1V. The *Drive Enabled* %I bit must be on for the *Force D/A* command to work. An exception is the second axis of a one axis Power Mate APM. Since there is no axis 2 *Enable Drive* bit on a one axis Power Mate APM, placing the Force D / A command code in the second axis %AQ data automatically enables the servo drive.

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 APM, the *Force D/A* operation will end. A one-shot Force D / A command will operate only during the sweep in which it appears.

Position Increment Command

To generate small corrections between the axis position and the Power Mate APM 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 APM 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 APM will be changed by the increment. *Position Increment* may 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:

- An ON *Abort All Moves* bit 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.
- Changing the *Jog Velocity* or *Jog Acceleration* will not affect non-programmed moves in progress.

Programmed Motion

Programmed Motion refers to motion caused by the execution of motion programs. The %AQ Move Command (27h) is treated as a single line motion program within the Power Mate APM and is included in the programmed motion category. The Power Mate APM 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 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.

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 APM 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 APM 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 2 motions at designated points. When the first axis reaches the Sync Block, it will not execute the 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 *Drive OK* input must be ON 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 APM

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
- The *Drive OK* input turns OFF when *Drive OK* is configured to be enabled.
- 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

For the type of reference to use for the move, the choices 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 APM 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 SCURVE.

Linear Acceleration

A sample profile of a linear move plotting velocity versus time is shown in Figure 6-1. 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
```

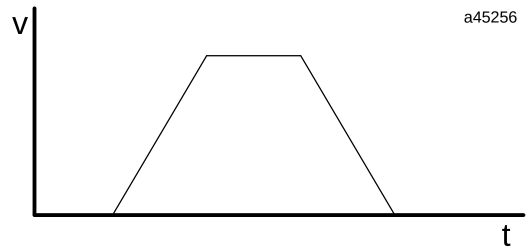


Figure 6-1. Sample Linear Motion

S-Curve Acceleration

Note

> Power Mate APM firmware release 2.10 *does not* allow SCURVE acceleration when the Follower control loop is selected. Any SCURVE accelerations encountered in a program will be converted to linear accelerations. For additional information refer to the *Important Product Information* documentation included with the Power Mate APM module.

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
```

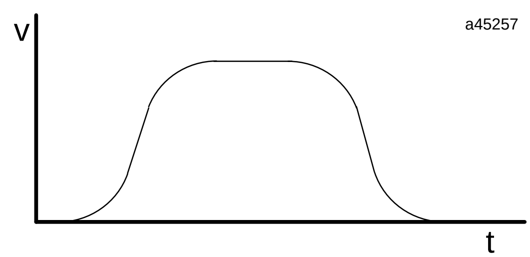


Figure 6-2. Sample S-curve Motion

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 APM 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 scurve 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
```

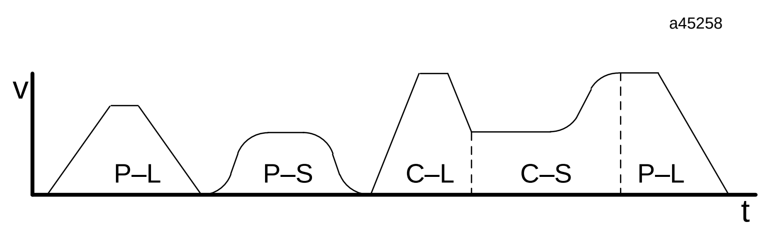


Figure 6-3. Combining PMOVEs and CMOVEs

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 scurve 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 scurve.

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 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 scurve as it decelerates to zero velocity.

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

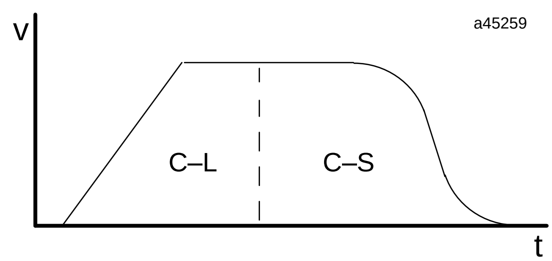


Figure 6-4. Changing the Acceleration Mode During a Profile

Example 3: Not Enough Distance to Reach Programmed Velocity

CMOVES and PMOVES can be programmed which 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 APM 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
```

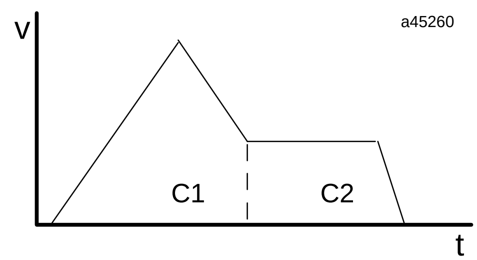


Figure 6-5. Not Enough Distance to Reach Programmed Velocity

Example 4: "Hanging" the Power Mate APM When the Distance Runs Out

A serious programming error involves "hanging" the Power Mate APM 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 APM 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
```

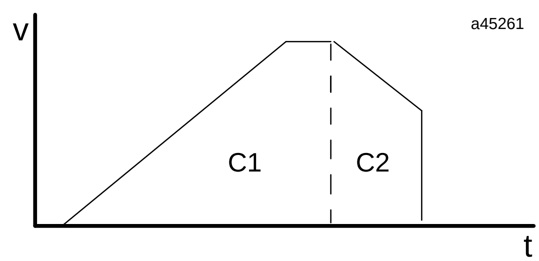


Figure 6-6. "Hanging" the Power Mate APM When the Distance Runs Out

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
```

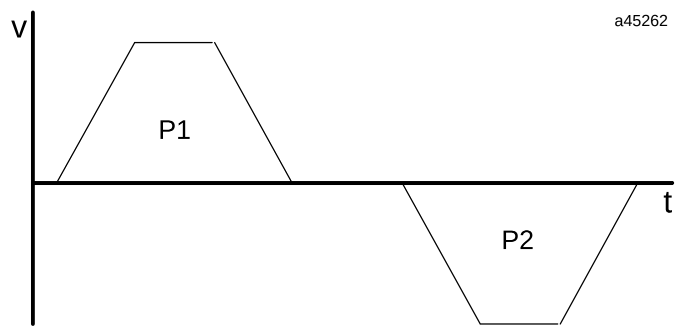


Figure 6-7. Dwell

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 may 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 APM can store up to ten separate programs and forty subroutines. Subroutines can be defined as two types: *single axis* and *multi axis*. 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 multi axis subroutines contain axis numbers just like commands within multi axis programs. Multi axis subroutines can only be called from multi axis programs or subroutines. Single axis subroutines can only be called from single axis programs or subroutines. On a two axis Power Mate APM, a single axis program for axis 1 and a single axis program for axis 2 may 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 insure correct program execution or to determine when events should occur. A block number may 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 APM 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
```

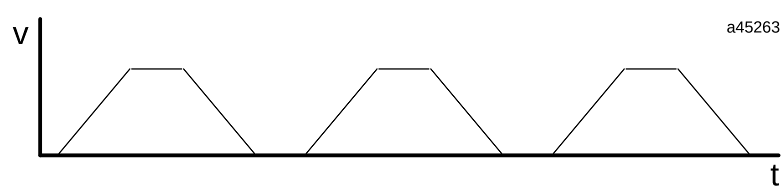


Figure 6-8. Unconditional Jump

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 APM 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 APM and once every 2 milliseconds for the 2 axis Power Mate APM.

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 APM 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 APM 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 APM 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 APM 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 APM, 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
```

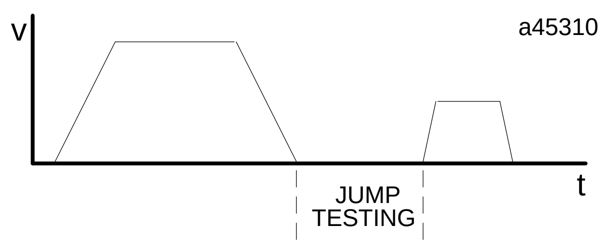


Figure 6-9. Normal Stop before JUMP

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
```

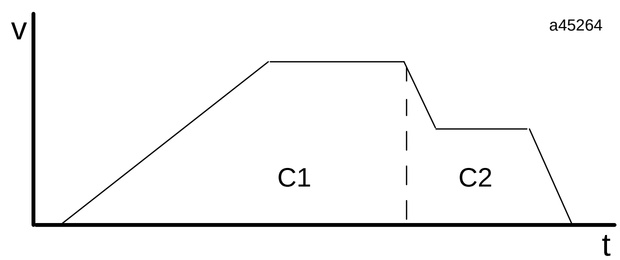


Figure 6-10. JUMP Without Stopping

Jump Stop

A jump stop is a stop 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 scurve motion will achieve constant velocity before using the *Jog Acceleration* and beginning to decelerate. See the scurve 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.

In an scurve 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 scurve 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
```

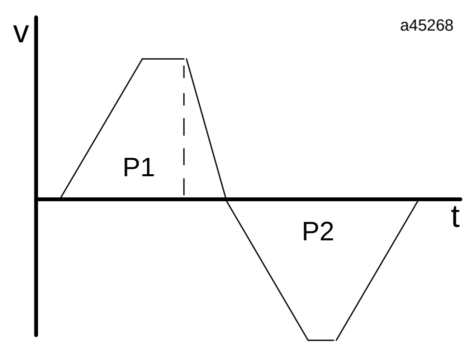


Figure 6-11. Jump Stop

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
```

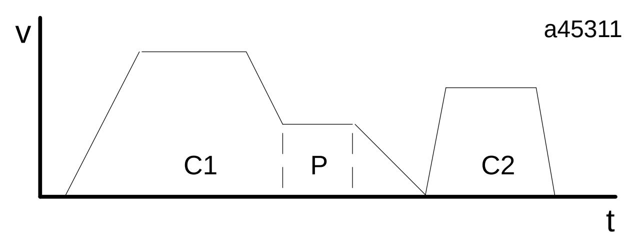


Figure 6-12. Jump Followed by PMOVE

SCURVE Jumps

Jumps during linear motion and jumps during scurve motion at constant velocities immediately begin accelerating or decelerating to a new velocity. Jumps during an scurve acceleration or deceleration, however, require different rules in order to maintain an s-curve profile. What happens when a jump occurs during an scurve 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: SCURVE - 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
```

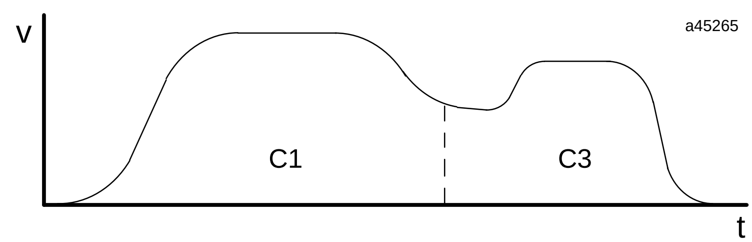


Figure 6-13. Jumping After the Midpoint of Acceleration or Deceleration

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 APM will immediately begins reducing the acceleration or deceleration to zero; once zero, the Power Mate APM will use the jump destination acceleration and velocity and change to the new velocity.

Example 13: SCURVE - 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 APM slows the acceleration to zero. Constant velocity, zero acceleration, occurs at the second dotted line. There, the Power Mate APM 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
```

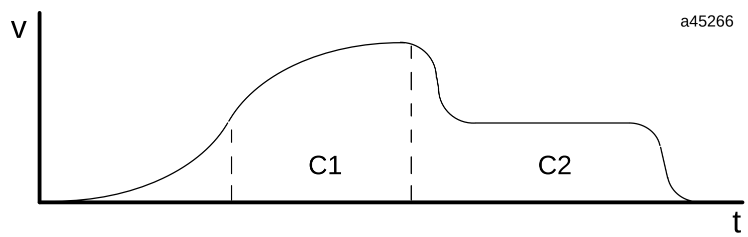


Figure 6-14. Jumping before the Midpoint of Acceleration or Deceleration

The second case involves jumping to a higher velocity while accelerating or a lower velocity while decelerating. When this occurs, the Power Mate APM 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 scurve is used to reduce acceleration or deceleration to zero.

Example 14: SCURVE - 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 scurves 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
```

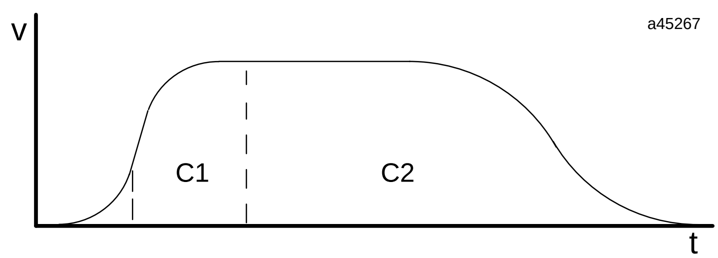


Figure 6-15. Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating

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 APM and 128 seconds for the 2 axis Power Mate APM. If the time to accelerate or decelerate is computed to be longer than this time, the Power Mate APM 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 APM calculates that an acceleration of 188 would allow a velocity of 12000 to be reached in 64 seconds. The lefthand 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
```

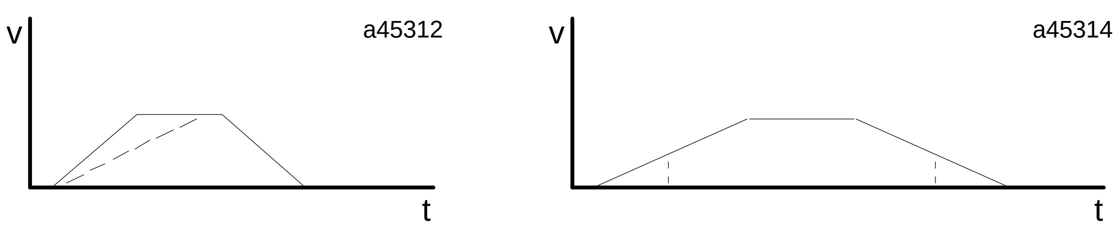


Figure 6-16. Maximum Acceleration Time

Multi-Axis Programming

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

Example 16: Multi-Axis 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.

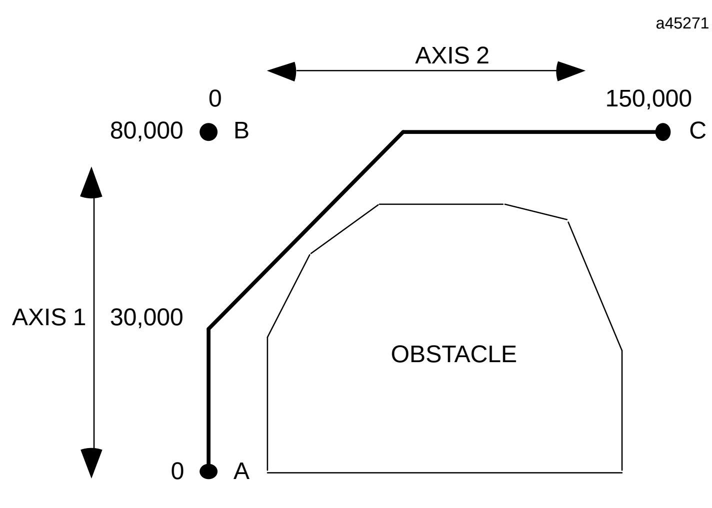


Figure 6-17. Multi-Axis Programming

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 APM 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 APM

The Power Mate APM maintains 256 double word parameters (0 through 255) in memory. These values may be used as a parameter in ACCEL, VELOC, DWELL, PMOVE, and CMOVE motion commands. Note 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 APM 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 - 251	Reserved for future use
252	Stores Axis 2 *Commanded Position* (user units) when Follower Enable Trigger occurs (2 Axis Power Mate APM only)
253	Stores Axis 1 *Commanded Position* (user units) when Follower Enable Trigger occurs
254	Stores Axis 2 *Strobe Position* value (user units), 2 Axis Power Mate APM only
255	Stores Axis 1 *Strobe Position* value (user units)

Parameters are all reset to zero after a power cycle or after a Power Mate APM 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 may be changed during program execution, but the change must occur before the Power Mate APM begins executing the block previous to the block that uses the parameter.

Follower Motion

When the *Enable Follower* %Q bit is turned ON, an axis will immediately begin following the selected *Master Source* unless an external *Enable Trigger* input has been selected. If an external *Enable Trigger* input has been selected then the *Enable Follower* %Q bit must be ON and an OFF to ON transition of the trigger input must occur. The external trigger input CTL01 - CTL08 can be selected using %AQ command 41h (refer to *%AQ Immediate Commands* in Chapter 4).

There are two ways to provide acceleration control when following a master source.

1. External means, such as ramping the velocity of the master input, can be used to limit acceleration.
2. Select the Follower Ramp mode with %AQ Command (45h) to make the Follower axis ramp up at the Jog acceleration rate.

Master Sources

A Power Mate APM axis can follow one of several master input sources:

- The external encoder 3 input
- The internal master velocity generator
- The analog input
- In a 2-axis Power Mate APM, axis 1 can select the axis 2 encoder to be its master input source

If **ENC3/INT** is configured for an axis, the *Select Internal Master* %Q bit is used to choose between the external encoder 3 input and the internal master generator. In a two axis Power Mate APM, if both axes are configured with **ENC3/INT** as the *Master Source,* then both axes will always use the same master source because the *Select Internal Master* %Q bit applies to any axis configured with **ENC3/INT** as master.

Note that follower motion is additive with *Jog*, *Move at Velocity*, or Motion Programs. If an axis is following the internal master at velocity A, and a *Jog* is commanded at velocity B, the axis will move at velocity A+B.

Encoder 3 Master Input

When **ENC3/INT** is configured as the *Master Source* and the *Select Internal Master* %Q bit is OFF, encoder 3 will be the master input source.

The Power Mate APM always tracks and reports the position of encoder 3 in *Encoder 3 Actual Position* %AI data. A 2-axis Power Mate APM also computes and reports the encoder 3 velocity in *Encoder 3 Actual Velocity* %AI data. A 1-axis Power Mate APM will report encoder 3 velocity only if the module is configured with "%AI Pos Err" DISABLED. In this case, the *Encoder 3 Actual Velocity* is reported in place of *Position Error.*

Encoder 3 has configured *High and Low Count Limits.* When a limit is reached, the position rolls over to the other limit and continues changing.

Encoder 3 can be initialized by performing a Home Cycle for encoder 3. **An encoder 3 Home Cycle does not generate any motion**. Instead, it sets the *Encoder 3 Actual Position* %AI data to the configured master source *Home Position* at the next marker pulse when the Home Switch 3 input is ON. It also sets the *Encoder 3 Position Valid* %I bit when it sets the position. See Chapter 4, %Q Discrete Commands, Find Home (Encoder 3) for details about the encoder 3 Home Cycle.

Example 1: Following Encoder 3 Master Input

In this example, a graph of velocity (v) versus time (t) shows the velocities of the master input (encoder 3), and the axis which is following the master. The Power Mate APM is configured with *Master Source* **ENC3/INT** and the *Select Internal Master* %Q bit is OFF. The A:B ratio is 1:1. The velocity profile of the following (slave) axis is identical to the master input.

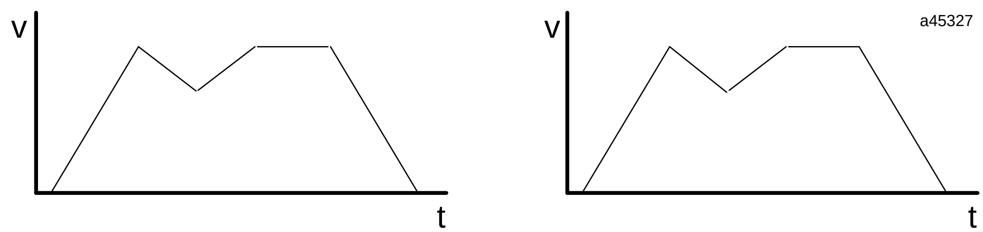


Figure 6-18. Following Encoder 3 Master Input

Internal Master Velocity Generator

The Power Mate APM can internally generate a velocity up to one million counts per second. When **ENC3/INT** is configured as the Master Source and the *Select Internal Master* %Q bit is ON, the internal master will be used as the master input source. The velocity generated is initially zero. The immediate command *Internal Master Velocity* can be used to set the generated velocity.

There is no acceleration control of the internal velocity generator. One method of preventing the instantaneous acceleration attempt when *Enable Follower* is turned ON is to incrementally step up the internal velocity using the *Internal Master Velocity* %AQ command.

Example 2: Following the Internal Master

When following the internal master, the following axis simply moves at the current internal velocity. In this example, dotted lines indicate the times when a master velocity change takes place using the *Internal Master Velocity* %AQ command. The Power Mate APM is configured with *Master Source* **ENC3/INT** and the *Select Internal Master* %Q bit is ON. The A:B ratio is 1:1.

The velocities commanded are the following: initially 0, then 11 thousand, 15 thousand, 4 thousand, 7 thousand, and finally 0 again. The velocity of the following axis is identical to the internal master velocity.

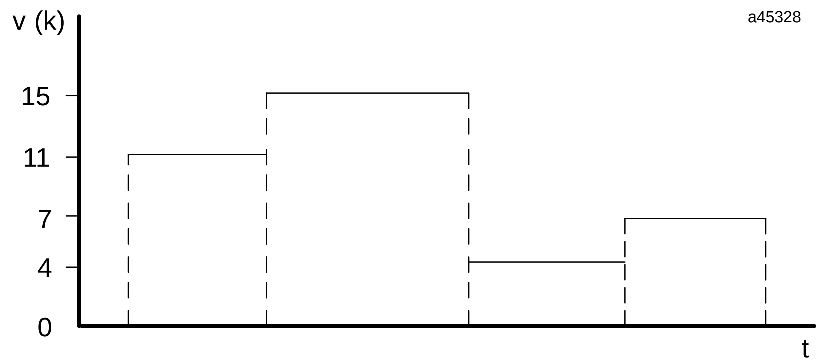


Figure 6-19. Following the Internal Master

Analog Input Master

When **ANALOG** is configured as the Master Source, the faceplate *Analog Input* is the master input source. The Power Mate APM converts the *Analog Input* into a velocity command. The *Analog Maximum Velocity* configuration parameter sets the velocity generated when the *Analog Input* is +10V. Zero velocity is generated when the *Analog Input* is 0V, and the negative of the maximum velocity is generated at –10V.

Example 3: Following the Analog Input

In the graph below, the Power Mate APM is configured with *Master Source* **ANALOG**. The A:B ratio is 1:1. The dotted lines indicate when the *Enable Follower* %Q bit was turned ON and OFF.

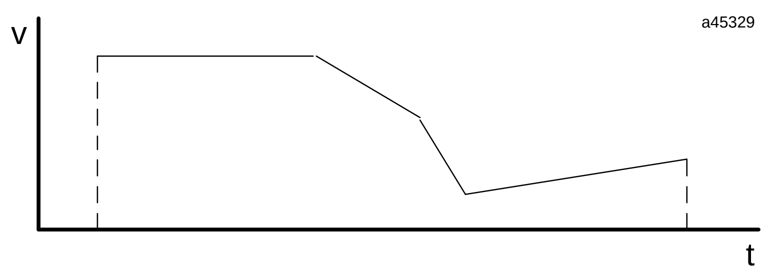


Figure 6-20. Following the Analog Input

Encoder (Axis) 2 Master

A 2-axis Power Mate APM can be configured so that axis 1 will follow axis 2. With “ENC2” as the *Master Source* for axis 1, axis 1 will follow axis 2 at the axis 2 velocity multiplied by the A:B ratio configured for axis 1.

Example 4: Following Encoder (Axis) 2

Axis 1 of a 2-axis Power Mate APM is configured with *Master Source* “ENC2”. With the A:B ratio 1:2, axis 2 is commanded to *Move at Velocity* 12000 and then 0. Axis 1 follows axis 2 at half of the axis 2 velocity and acceleration, and moves only half the distance that axis 2 moves.

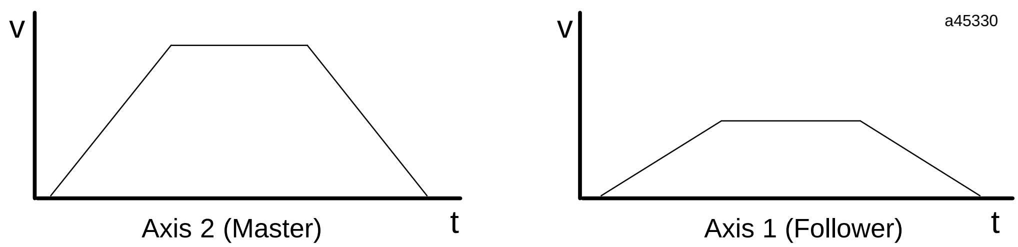


Figure 6-21. Following Encoder (Axis) 2

A:B Ratio

An Power Mate APM axis following a master input can do so at a wide range of master:slave (A:B) ratios. The A value can be any number from –32768 to 32767. The B value can be anywhere between 1 and 32767. The ratio A:B can be from 1:32 to 32:1. Thus very precise ratios such as 12,356:12,354 or 32,000:1024 can be used.

The *Follower A/B Ratio* %AQ command can be used to change the A:B ratio at any time, even while following. However, an invalid ratio will generate a status error and be ignored. An invalid ratio is a ratio with B equal to or less than zero or A:B greater than 32:1 or less than 1:32.

Ratios from 1:1024 to 1024:1 can be obtained with a 2-axis Power Mate APM if the slave axes are cascaded. When axis 2 follows encoder 3 and axis 1 follows axis 2, if both ratios are 32:1 the ratio of axis 1 to encoder 3 is 1024:1.

When following with a non 1:1 ratio, the velocity profile of the master and follower will look somewhat different. Horizontal lines, indicating constant velocity, and slanted lines, indicating acceleration and deceleration, will be different. If the A:B ratio is less than 1, the follower velocity and acceleration will be less than the master. Likewise, if the A:B ratio is greater than 1, the follower velocity and acceleration will be greater than the master. The duration of motion, and time that the slave axis will accelerate, decelerate, or stay at constant velocity are the same for the master and follower.

The distance moved, which in a velocity profile is the area between the graph and the time axis, will be that of the master multiplied by the A:B ratio. If A is zero, no following motion will be generated. If A is negative, the following axis will move with the direction of motion reversed.

Example 5: Sample A:B Ratios

All of the following samples are following the master source input at various A:B ratios.

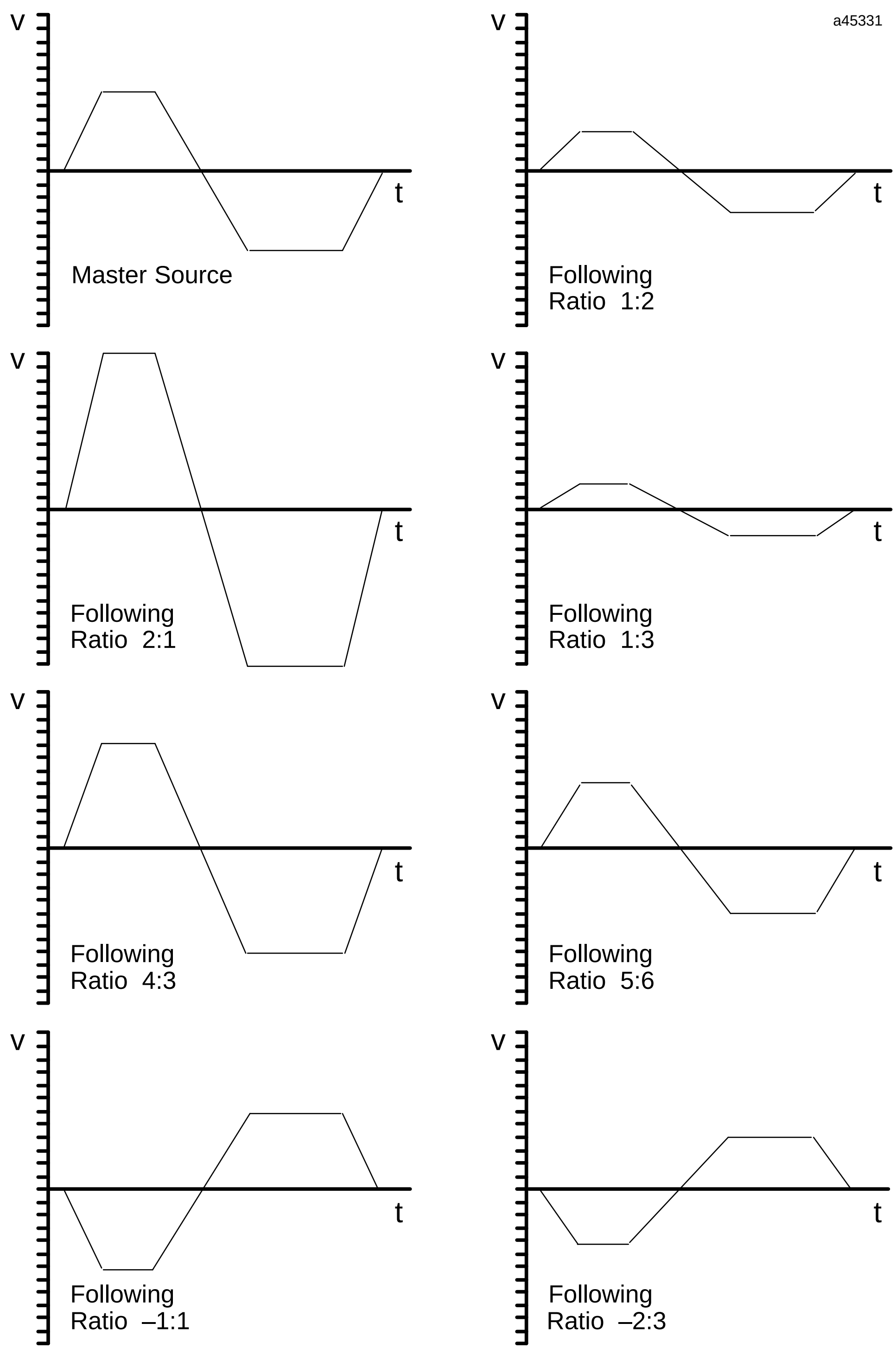


Figure 6-22. Sample A:B Ratios

Example 6: Changing the A:B Ratio

One example of variable A:B ratios is to use one ratio while moving positive, and another when moving negative. Note that determination of positive and negative velocity and transmission of the A:B ratio must be done in the PLC. In profile below, the following axis uses a 2:1 ratio when moving positive, and a 1:2 ratio when moving negative.

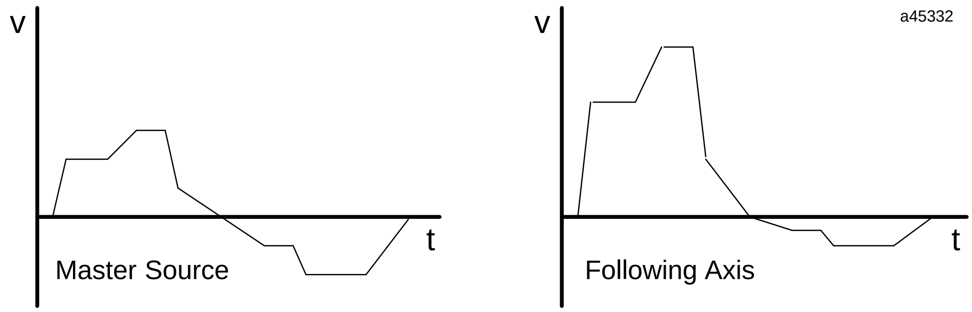


Figure 6-23. Changing the A:B Ratio

Example 7: Ratio Stepping

Another example of variable A:B ratios is step up the ratio as a type of acceleration control. Initially setting a ratio 0:32, and incrementing the numerator after a delay would step the following axis from 0:32 to 1:32 to 2:32 and so forth all the way to 32:32 which is 1:1. In this example, the PLC increments the A:B ratio by 1/16th every second until it reaches 1:1. The dotted lines represent ratio changes, the dashed line indicates when the *Follower Enable* %Q bit is turned OFF.

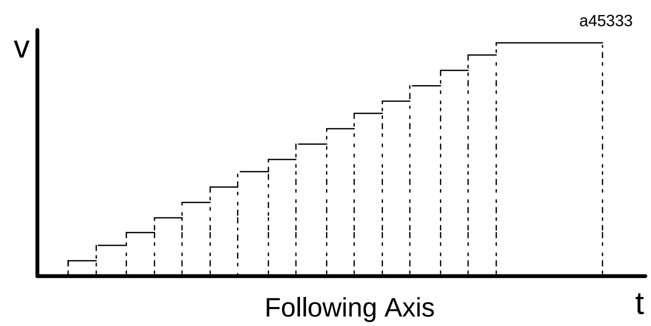


Figure 6-24. Ratio Stepping

Velocity Clamping

Velocity clamping is available using the *Velocity Limits* set in the Configuration software. When the master velocity exceeds the configured limit, the following axis will continue to move at the limit velocity multiplied by the A:B ratio. The *In Velocity Limit* %I bit is set and a status error is generated to indicate that the slave axis is no longer following the master input positioning. The slave axis has essentially *fallen behind* the master input.

The Velocity Limit units are counts per millisecond. Thus a limit of 400 cts / ms will limit velocity to 400,000 cts / sec.

Example 8: Velocity Clamping

The *Velocity Limits* are set to 100 and –100 in this example. Thus the master input velocity is clamped at 100,000 cts / sec in either direction. When the master axis peaks greater than the limits, the following axis stays at the limit. After the master slows to under the limit, the following axis continues tracking the master axis velocity. Counts generated in excess of the Velocity Limits are lost to the follower. The horizontal dashed lines indicate the velocity limits. The shaded area indicates the times when the *In Velocity Limit* bit is ON and the following axis is falling behind the master.

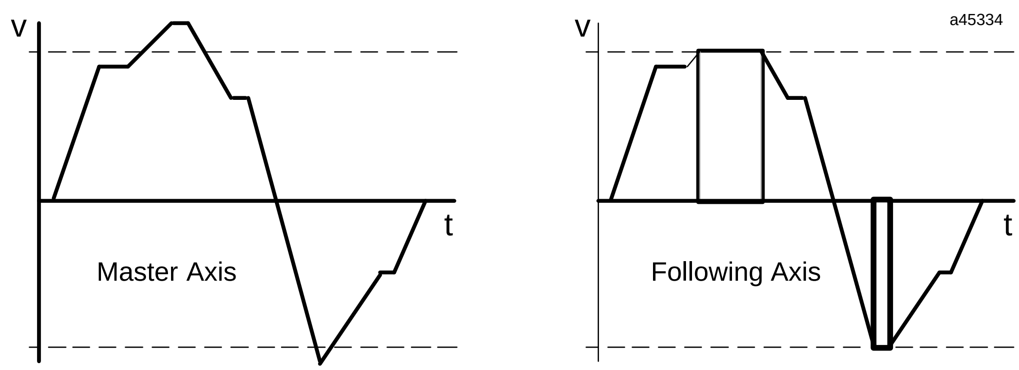


Figure 6-25. Velocity Clamping

Unidirectional Operation

Setting one of the *Velocity Limits* to zero results in unidirectional follower motion. Any encoder pulses in the zero limited direction are ignored. When a *Velocity Limit* is zero, no error is generated by pulses in the zero limited direction. The *In Velocity Limit* %I bit, however, does reflect the presence of pulses in the zero limited direction.

Example 9: Unidirectional Operation

In this example, the negative *Velocity Limit* is set to zero. As shown in the velocity profile below, the following axis follows the positive pulses, but ignores the negative pulses. Note that when the master is moving negative, the *In Velocity Limit* %I bit is ON, but no status error is generated.

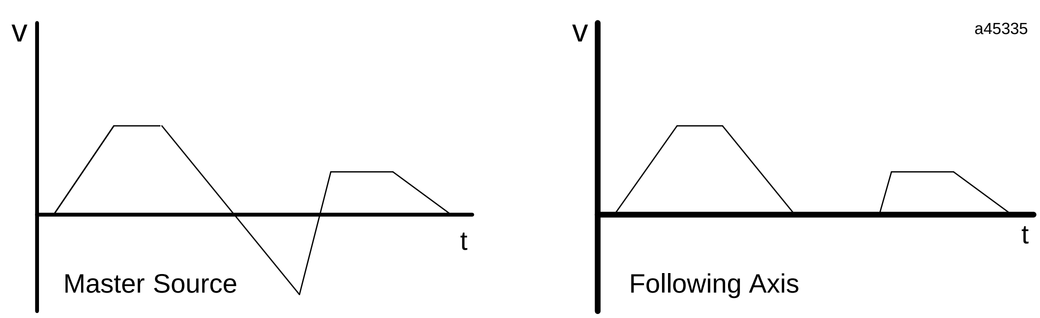


Figure 6-26. Unidirectional Operation

Synchronizing Encoder 3 and the Internal Master

Two %I bits are provided to help synchronize the *Encoder 3* and *Internal Master Velocities*. The *Master Velocity Mismatch* %I bit is ON if the *Encoder 3* and *Internal Master Velocities* differ by more than 128 cts / sec. Another %I bit, the *Master Velocity Mismatch Sign* bit, signals the direction in which the *Internal Master Velocity* must change to equal the *Encoder 3 Velocity*. ON indicates the *Internal Master Velocity* must change in the positive direction, OFF indicates the negative direction.

Synchronization of encoder 3 and the internal master prevents the Power Mate APM from attempting instantaneous accelerations when the *Select Internal Master* %Q bit is toggled. Note, however, that if encoder 3 is undergoing fast acceleration or deceleration, the *Master Velocity Mismatch* bit may not turn OFF because the *Encoder 3 Velocity* could change more than the 256 counts/second in one PLC sweep.

Example 10: Encoder 3 and Internal Master Synchronization

To synchronize the internal master and encoder 3, the *Internal Master Velocity* is stepped up to an expected encoder 3 velocity. When encoder 3 is within 128 cts/sec of this velocity, the PLC will turn the *Select Internal Master* %Q bit OFF to switch the master source to encoder 3. The shaded area indicates the time when the *Select Internal Master* %Q bit is ON.

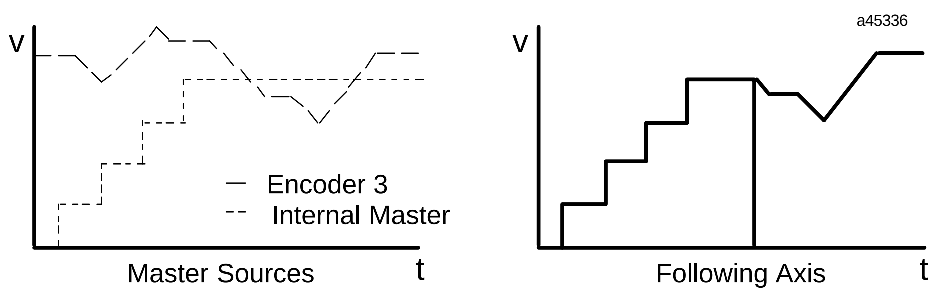


Figure 6-27. Encoder 3 and Internal Master Synchronization

Enabling the Follower with External Input

An external trigger signal applied to one of the inputs CTL01 - CTL08 can be used to enable the follower axis. The input selected for the Enable trigger is defined with %AQ Command 41h. When no input is selected, the follower is enabled and disabled directly by the *Enable Follower %Q* bit command. When an input is selected for the Enable trigger and %Q *Enable Follower* is set, the next positive transition of the defined input will instantly enable the follower. The follower will remain enabled until the %Q *Enable Follower* bit is cleared. Fast Inputs (CTL01 or CTL02) are recommended for this use if a high speed response is required.

When the *Enable Follower* trigger occurs, the *Commanded Position* at that point is captured in a parameter register so that it can be used in a *Programmed Move* command. The position is captured in parameter 253 (for axis 1) or parameter 252 (for axis 2).

The Follower Enable status is returned in %I bit offset 28 (axis 1) or %I bit offset 29 (axis 2).

Follower Axis Acceleration Ramp Control

For applications where the Follower is enabled when the Master command is already up to speed, the Follower Ramp feature can be used to apply a controlled acceleration rate to bring the follower axis up to speed. This may be done without losing any Master command counts from the point at which the Follower was enabled.

To prevent any master counts from being lost, configure the appropriate time (milliseconds) for making up the counts in the Ramp Distance Make-up Time parameter. If lost counts during ramp up are of no concern, set the Distance Make-up Time = 0 for simply ramping up to the Master Commanded Velocity.

To select Ramp Control, use %AQ Command 45h. Set the Follower Ramp distance make-up time to the desired time with %AQ Command 42h. Refer to the description of this %AQ command for more information about this parameter / feature.

The Follower Axis Velocity will be ramped up at the Jog acceleration rate. When the *Enable Follower* %Q bit is turned off, the axis velocity will ramp down at the same rate. The *Command Move* %I bit indication is turned on while the ramp control is in effect for both the ramp up / make-up and ramp down.

The %I *Follower Enabled* and %I *Command Move* bits can be monitored by the PLC to determine which part of the follower ramp up / ramp down cycle is active. The following figure shows the state of *Follower Enabled* and *Command Move* during a follower cycle.

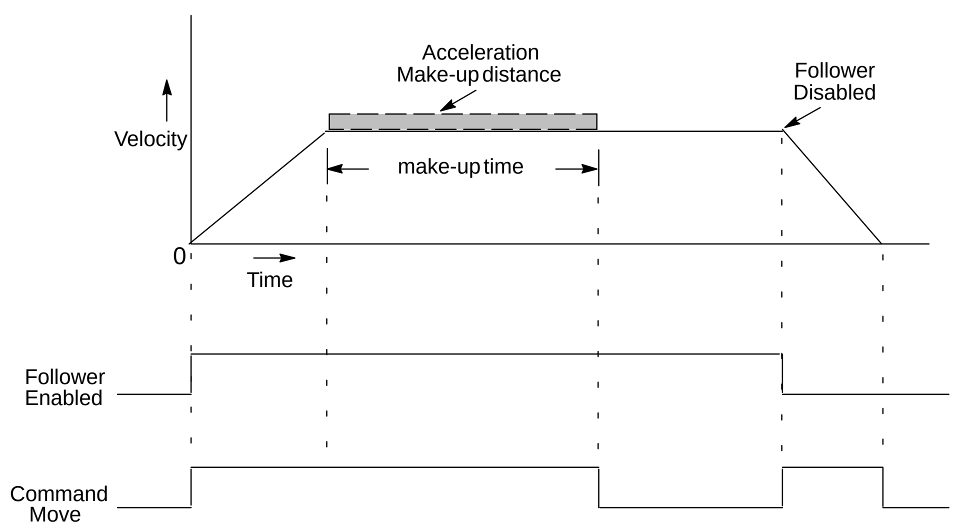


Figure 6-28. Follower Ramp Up/Ramp Down Cycle

Combined Motion

Combined motion consists of Follower motion commanded from a master axis combined with one of these internally commanded motions:

- %Q Jog Command
- %AQ Move at Velocity Command
- %AQ Move Command (Power Mate APM firmware version 2.10 or later)
- Stored Motion Program (Power Mate APM firmware version 2.10 or later)

Combined motions are additive. The slave axis motion is equal to the sum of the motion commanded by the master axis and the internally commanded motion.

Example 11: Follower Motion Combined with Jog

In this example, the *Enable Follower* %Q bit is set, causing the slave axis to follow the master input. While the slave axis is following, the *Jog Plus* %Q bit is set. The following axis accelerates from its master's velocity to it's master's velocity added to the current *Jog Velocity*. This acceleration will be just as if the axis was not following a master source at the time. When the *Jog Plus* %Q bit is cleared, the following axis decelerates to its master's velocity. In the velocity profiles below, the dotted lines indicate when the *Jog Plus* bit is turned ON and then OFF.

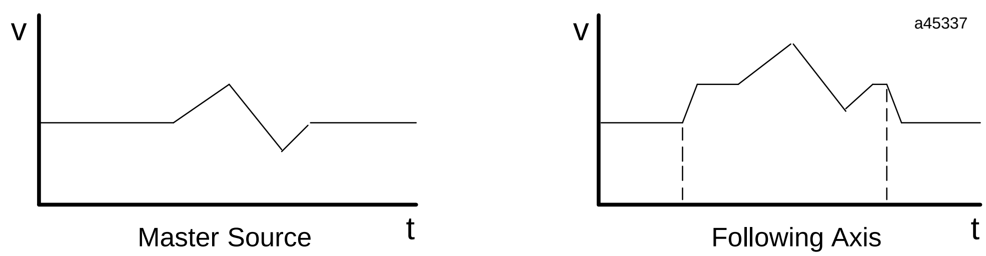


Figure 6-29. Combined Motion (Follower + Jog)

Follower Motion Combined with Motion Programs

Motion commands from stored programs or the %AQ *Move Command* can also be combined with the master command to drive the follower axis. These point-to-point move commands can come from one of the stored motion programs 1 through 6 and any stored subroutines they may call. The %AQ *Move Command* is treated as a single line motion program which uses the present Jog Velocity and Jog Acceleration. Program execution is started by the PLC setting a program select Q bit or sending a %AQ *Move Command*. All the existing program commands will be supported in the follower mode, however, if any PMOVEs and CMOVEs with S curve acceleration mode exist in the program, they will be processed as linear moves. **Only Linear acceleration moves should be programmed in the Follower mode.**

If there is no master command, the axis can be commanded solely from the stored motion program data. Thus, with no master input to axis 2 and Encoder 2 selected as the master source for axis 1, a stored program can be used to control axis 2 with axis 1 following per the designated ratio.

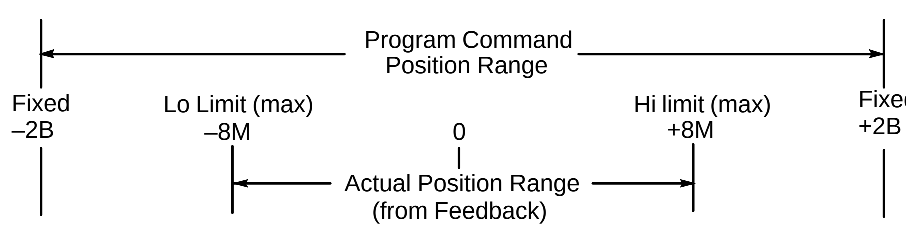
When PMOVEs are executed with Follower not enabled, the *in zone* %I bit must be set at the end of the move before programmed motion will continue. When Follower is enabled, since *in zone* may not turn on while also following a master command, the *in zone* indication will not be required to continue. The next move will take place when the Commanded distance for the previous move has completed. The *in zone* %I bit will always indicate the true *in zone* condition.

The active commanded position updated and used by the stored motion program is referred to as *Program Command Position*. Each time a program is selected for execution, this position register is initialized in one of the two ways listed below.

1. If the follower is **not enabled**, the Program Command Position is set to the current *Commanded Position = Actual Position + Position Error.*

2. If the follower **is enabled,** the Program Command Position is set to the Program Reference Position **(0).** Since the Program Command Position is only updated by internally generated commands (and not by the master command), it will then indicate the position commanded by the stored program. Absolute move commands from the stored program will be referenced to the Program Reference Position.

Position ranges (in counts) for the Actual and Program Command Position registers are indicated in the figure below.



With sustained commanded motion in the same direction, the Program Command Position will roll over at +2,147,483,647 or –2,147,483,648 counts

The Actual Position, however, will be confined by the configured high and low limits with maximum values of +8,388,607 and –8,388,608 counts.

Table 6-1 below indicates which source commands affect these position registers and the actual and commanded velocities. Program Command Position is updated only by internally generated move commands (program commands, Jog, Home, and Move at Velocity). The Commanded Velocity (returned in %AI data) also only indicates velocity commanded by these internally generated move commands. Actual Position and Actual Velocity %AI return data reflect the combination of the master input and the move commands. In other words, counts coming from the master source affect **only** the Actual Position and Actual Velocity, and if there are no internally generated move commands, the Commanded Velocity will be 0 and the Program Command Position will not change.

Table 6-1. Command Input Effect on Position Registers

COMMAND Input	Follower Enabled ?	Follower Registers Affected by input
MasterCommands (from selected Mastersource)	No	None affected
	Yes	Actual Position is Updated Commanded Position (%AI) is Updated (= Actual Psn + Psn Err) Program Command Position is **Not** Affected Actual Velocity is Updated Commanded Velocity is **Not** Affected
ProgramCommands	No	Actual Position is Updated Commanded Position (%AI) is Updated (= Actual Psn + Psn Err) Program Command Position is Updated Actual Velocity is Updated Commanded Velocity is Updated (by Prog cmd vel only)
	Yes	Actual Position is Updated (by Prog cmd + Master cmd) Commanded Position (%AI) is Updated (= Actual Psn + Psn Err) Program Command Position is Updated (by Prog cmd only) Actual Velocity is Updated (by Prog cmd vel + Master cmd vel) Commanded Velocity is Updated (by Prog cmd vel only)
Other Internally Generated Move Commands (Home, Jog, and Move at Velocity)	No	Actual Position is Updated Commanded Position (%AI) is Updated (= Actual Psn + Psn Err) Program Command Position is Updated but not used Actual Velocity is Updated. Commanded Velocity is Updated (by Internal cmd vel only)
	Yes (Home is not allowed)	Actual Position is Updated (by Internal cmd + Master cmd) Commanded Position (%AI) is Updated (= Actual Psn + Psn Err) Program Command Position is Updated but not used Actual Velocity is Updated (by Internal cmd vel + Master cmd vel). Commanded Velocity is Updated (by Internal cmd vel only)

The Program Command Position is synchronized to the Actual Position by the Home command execution or by the Set Position %AQ command or **if the follower is not enabled,** by the *Execute motion program* command. This is indicated in Table 6-2 below. Subsequent rollovers of the Program Command Position have no affect on the Actual Position or vice-versa.

Table 6-2. Actions Affecting Program Command Position

ACTION	Follower Enabled ?	Resulting Updates to Follower Position Registers
Home Found	No	Actual Position is set to Home Value Program Command Position is set to Actual Position + Position Error
	Yes	Home Command is Not Allowed Status Error is returned
%AQ Set Position	Not applicable	Actual Position is set to %AQ Value Program Command Position is set to Actual Position + Position Error Note: Set Position is not allowed if Command Move is ON.
Execute Program Select	No	Actual Position is NOT affected Program Command Position is set to Actual Position + Position Error
	Yes	Actual Position is NOT affected Program Command Position is set to Reference Position. (Reference Position is 0)

Program moves will execute in a rotary fashion such that incremental programmed commands past the limits will roll over at the limit and continue. Absolute programmed commands can also be used for applications that do not require going beyond the high / lowcountlimits.

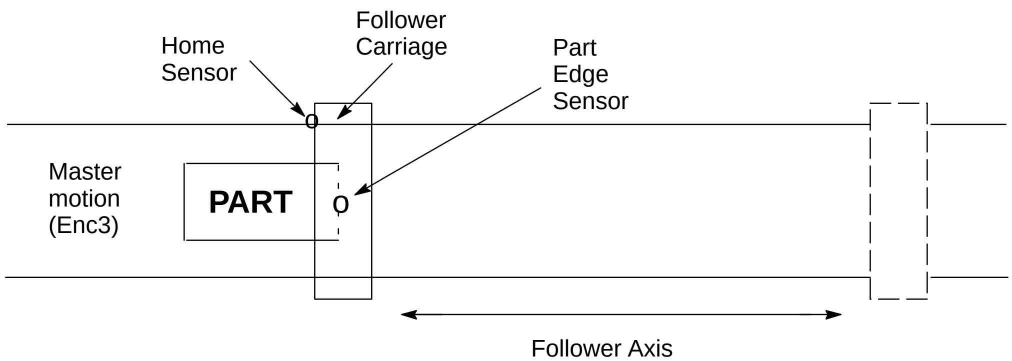
Any internally generated move command can be immediately terminated by the *Abort All Moves* command.

The %AI return data for Commanded Position (by default) will indicate *Actual Position + Position Error*. This can be changed to indicate the Program Command Position by using the %AQ command for Configuration Modes.

The following application example illustrates how a stored program can be used to control positioning operations relative to the detected edge of the moving object as it moves along at a rate indicated by the follower master command.

EXAMPLE 12: Follower Motion Combined with Motion Program

Applications that require modifying parts on the fly (such as notching, marking, riveting, spot welding, spot gluing, and so forth) would make use of the point-to-point moves superimposed on follower motion and enable follower at input features. A typical configuration and control sequence required for these applications is shown below.



Control Sequence:

1. With Enable Follower OFF, the PLC sends Follower axis to home position where Actual Position & Program Command Position are synchronized and set to Home Position value. *Position valid* indicates when this step is complete.
2. PLC sends the *External Input Enable Follower* %AQ command to select the part edge sensor input as the Follower enable trigger. The PLC also sets the *Enable Follower* %Q bit command.
3. When the Part edge sensor trips, the Power Mate APM enables the Follower axis to start following the master (Encoder 3) inputs. The *follower enabled* %I bit indicates when the axis is following the master command.
4. Once the follower is enabled, the PLC sends the *execute motion program* %Q bit to start execution of the selected program for the follower axis. At the time the program is selected, Program Command Position will be set to program reference position (0) because the follower is enabled. Program execution is then relative to the moving part edge as the follower axis tracks the part. Program Command Position now contains the position of the follower axis relative to the part edge and Actual Position indicates the total distance the follower axis has moved from home point (master +/– program commands).
5. At the end of program, the PLC turns Enable Follower OFF and loops back to step 1 to repeat for next part.

Note

Since the Power Mate APM saved the Follower enable input trigger command position in a parameter (# 253 for axis 1, #252 for axis 2), step 1 this time could be used to execute another program with an absolute move command back to the parameter value position and continuing with step 2. In this case, the *In Zone* indication could be used to indicate when step 1 is complete.

This method is possible because the Program Command Position is set to the Actual Position + *Position Error* when execute motion program is commanded with the follower disabled.

Appendix A
Error Codes

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

- 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., etc.)

Note

The Status LED on the faceplate of the module flashes slow (4 times / sec) for Status Only errors and fast (8 times / sec) for errors which cause the servo to stop.

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

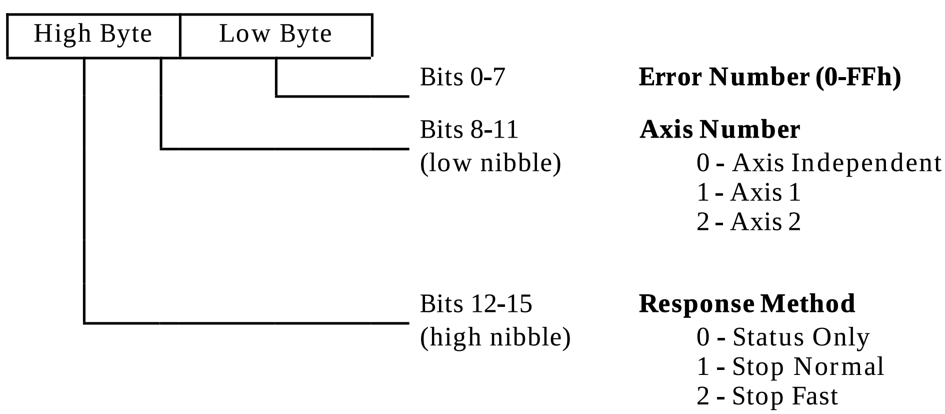


Figure A-1. Status Code Organization

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. The *Drive Enabled* and *Axis Enabled %I* bits are turned OFF after the configured *Drive Disable Delay.*
3. **Stop Fast Errors:** Instantly abort all motion by setting the analog output voltage to zero. The *Drive Enabled* and *Axis Enabled %I* bits are turned off after the configured *Drive Disable Delay.*

Table A-1. Status Word Error Codes

Error Number (Hexadecimal)	Response	Description
0	None	No Error
Configuration Parameter 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, defaults 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
2C	Status Only	Position Increment Error
Find Home Errors		
30	Status Only	Find Home while Drive Not Enabled 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 *
35	Status Only	Find Home while Follower Enabled error *
36	Status Only	Find Home while Abort bit set error *
Move at Velocity Errors		
39	Status Only	Move at Velocity while Drive Not Enabled error *
3A	Status Only	Move at Velocity while Program Selected error *
3B	Status Only	Move at Velocity while Home Cycle active error *
3C	Status Only	Move at Velocity while Jog error *
3D	Status Only	Move at Velocity while Abort All Moves bit is set error *
3E	Status Only	Move at Velocity Data greater than 8,388,607 uu / sec error *
3F	Status Only	Move at Velocity Data greater than 1 million cts / sec error *
Jog Errors		
40	Status Only	Jog while Find Home error *
41	Status Only	Jog while Move at Velocity error *
42	Status Only	Jog while Force D / A error *
43	Status Only	Jog while Program Selected and not Feedholding error *
Force D/A Errors		
47	Status Only	Force D / A while Jog error *
48	Status Only	Force D / A while Move at Velocity error *
49	Status Only	Force D / A while Program Selected error *
4A	Status Only	Force D / A while Follower Enabled error *

* Status error is reported, command is not executed.

Table A-1. Status Word Error Codes - Continued

Set Position Errors		
50	Status Only	Set Position while Program Selected error, *
51	Status Only	Set Position Data over-range error *
52	Status Only	Set Position while not In Zone error, OR 352 = * Set Position while Encoder 3 velocity > 128 error
End of Travel and Count Limit Errors		
56	Status Only	Commanded Position > High Count Limit *
57	Status Only	Commanded Position < 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 GE 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		
71	Status Only	Too many programs requested in same PLC sweep *
72	Status Only	Request Program 1-6 with multi-axis program active *
73	Status Only	Request two programs on same sweep with program active *
74	Status Only	Request two programs for same axis, lower number program executed *
75	Status Only	Empty or Invalid Program requested
Program Execution Conditions Errors		
80	Status Only	Execute Program while Home Cycle active *
81	Status Only	Execute Program while Jog *
82	Status Only	Execute Program while Move at Velocity *
83	Status Only	Execute Program while D / A Forced *
84	Status Only	Execute Program while Program Selected *
85	Status Only	Execute Program while Abort All Moves bit set *
86	Status Only	Execute Program while Position Valid not set *
87	Status Only	Execute Program while Drive Enabled not set *
Program Synchronous Block Errors		
8C	Status Only	Sync Block Error during CMOVE
8D	Status Only	Sync Block Error during Jump
EEPROM Errors		
90	Status Only	Flash EEPROM memory programming failure
Hardware Switch Errors		
A2	Stop Fast	Drive OK Switch error

* Status error is reported, command is not executed.

Table A-1. Status Word Error Codes - Continued

Error Number (Hexadecimal)	Response	Description
Hardware Errors		
A8	Stop Fast	Out of Sync error
A9	Stop Fast	Encoder Loss of Quadrature error
AA	Stop Normal	Analog Input Failure
Special Purpose Errors		
E0	Status Only	Custom Loop Type Mismatch
EF	Status Only	Firmware / Hardware Axis Number Mismatch (302 EPROM in 301 module)
Follower Errors		
F1	Status Only	Follower Position Error Limit condition encountered error
F2	Status Only	Follower Velocity Limit condition encountered error
F3	Status Only	Follower A / B ratio rejected because B = 0
F4	Status Only	Follower A / B ratio rejected because B < 0
F5	Status Only	Follower A / B ratio rejected because A:B or B:A greater than 32:1

Appendix B Parameter Download Using COMM_REQ

This appendix describes an alternate method of loading Power Mate APM Parameter Memory from the PLC using a the COMM_REQ function block with command code E501h. This function block can send up to 16 Power Mate APM 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)
```

```
|  (***********************************************************************)
|  (*  CLEAR THE REGISTER BLOCK, THEN LOAD THE COMM_REQ HEADER DATA       *)
|  (***********************************************************************)
|
|  SEND
| %T001   +-----+                 +-----+
+--] [----+ BLK_+-----------------+BLKMV+-
|         | CLR_|                 | INT |
|         | WORD|                 |     |
| CMREQST |     |          CONST -+IN1 Q+-%R0195
| %R0195  +IN   +          +00000 |     |
|         | LEN |                 |     |
|         |00045|          CONST -+IN2  |
|         |     |          +00004 |     |
|         +-----+                 |     |
|                          CONST -+IN3  |
|                          +00000 |     |
|                                 |     |
|                          CONST -+IN4  |
|                          +00008 |     |
|                                 |     |
|                          CONST -+IN5  |
|                          +00194 |     |
|                                 |     |
|                          CONST -+IN6  |
|                          +00000 |     |
|                                 |     |
|                          CONST -+IN7  |
|                          +00000 +-----+
|
|
|  (***********************************************************************)
|  (* PUT THE COMMAND TYPE (E501) IN  THE FIRST DATA WORD (R202)          *)
|  (* PUT THE BYTE COUNT OF THE DATA IN THE NEXT WORD (R203).  PUT THE    *)
|  (* MEMORY TYPE OF THE DATA (8=REG) IN THE NEXT WORD (R204). PUT THE    *)
|  (* STARTING LOCATION OF THE DATA BLOCK -1 IN THE NEXT WORD (R205)      *)
|  (* PUT THE STARTING PARAMETER NUMBER (1) IN THE LOWER BYTE OF (R206)   *)
|  (* AND THEN THE NUMBER OF PARAMETERS TO BE SENT (16) IN THE LOWER      *)
|  (* BYTE OF R207                                                        *)
|  (***********************************************************************)
|
|  SEND
| %T0001  +-----+                +-----+                 +-----+
+--] [----+MOVE_+----------------+MOVE_+-----------------+MOVE_+-
|         | WORD|                | INT |                 | INT |
|         |     |                |     |                 |     |
|         |     | CMDTYP         |     | BYTECNT         |     | MEMTYP
|  CONST -+IN  Q+-%R0202  CONST -+IN  Q+-%R0203   CONST -+IN  Q+-%R0204
|   E501  | LEN |         +00068 | LEN |          +00008 | LEN |
|         |00001|                |00001|                 |00001|
|         |     |                |     |                 |     |
|         +-----+                +-----+                 +-----+
|
|  SEND
| %T0001  +-----+                +-----+                 +-----+
+--] [----+MOVE_+----------------+MOVE_+-----------------+MOVE_+-
|         | INT |                | INT |                 | INT |
|         |     |                |     |                 |     |
|         |     | DATAST         |     | PAR_NO          |     | NO_VALS
|  CONST -+IN  Q+-%R0205  CONST -+IN  Q+-%R0206   CONST -+IN  Q+-%R0207
|  +00205 | LEN |         +00001 | LEN |          +00016 | LEN |
|         |00001|                |00001|                 |00001|
|         |     |                |     |                 |     |
|         +-----+                +-----+                 +-----+
|
```

```
|
|  (************************************************************************)
|  (* ADD LOGIC HERE TO MOVE THE PROPER CONSTANTS INTO THE REGISTERS      *)
|  (* (%R208 - %R239) SO THEY CAN BE SENT TO THE PM-APM PARAMETERS.        *)
|  (************************************************************************)
|
|  (************************************************************************)
|  (* NOW ACTIVATE THE COMM_REQ TO SEND THE PARAMETER DATA TO THE PM-APM   *)
|  (************************************************************************)
|
|  SEND
| %T0001   +-----+
+--] [-----+COMM_+-
|          | REQ |     NOTE:  SYSID HIGH BYTE = COMM_REQ RACK DESTINATION
|          |     |            SYSID  LOW BYTE = COMM_REQ SLOT DESTINATION
|  HDR_WDS |     |
|  %R0196 -+IN FT+-           TASK ALWAYS = 0 FOR PM-APM COMM_REQ
|          |     |
|   CONST -+SYSID|
|    0007  |     |
|          |     |
|   CONST -+TASK |
| 00000000 +-----+
```

Appendix C Specifications

This appendix provides specifications for the Power Mate APM module, and wiring information for the I/O connections.

Module Specifications

Power Supply Voltage:	5 Vdc from backplane
Power Supply Current:	800 mA + (1.4 x Encoder Current Drain) + (95 mA SNP Mini-converter Current, if used)
Available Encoder +5V Current/Module:	500 ma @40C derated to 300 ma @ 55C.
Maximum number of modules/system: **Model 311, 313, 323 PLCs :**	3 IC693APU301 in CPU baseplate (limited by power supply) 2 IC693APU302 in CPU baseplate (limited by %AI data)
Model 331 PLC :	8 IC693APU301 or 4 IC693APU302 in CPU, expansion and remote base-plates (limited by %AI data) Maximum 3 per baseplate (limited by power supply)
Model 341 PLC :	14 IC693APU301 or IC693APU302 in CPU, expansion and remotebaseplates Maximum 2 in CPU baseplate and 3 in each expansion / remote baseplate (limited by power supply)
Model 351, 352 * PLC :	23 IC693APU301 or IC693APU302 in CPU, expansion and remotebaseplates Maximum 2 in CPU baseplate and 3 in each expansion / remote baseplate (limited by power supply) *Note that a Model 351 or 352 system can have up to 8 baseplates (CPU baseplate and 7 expansion/remotebaseplates)*

Refer to GFK-0867B, or later version, for product standards and general specifications.

*** CPU model 352 will be available in 4th Quarter 1996.**

I/O Specifications

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

Velocity Command

Output of D/A converter with the following characteristics.

- Resolution: 13 bits including sign
- Linearity: .02% of full scale output
- Offset Voltage: " 500 µV maximum
- Maximum Output: 10.0 V , " 0.3 V
- Minimum Load Resistance: 2000 ohms
- Voltage Between Analog Common and Ground: " 1.0 V

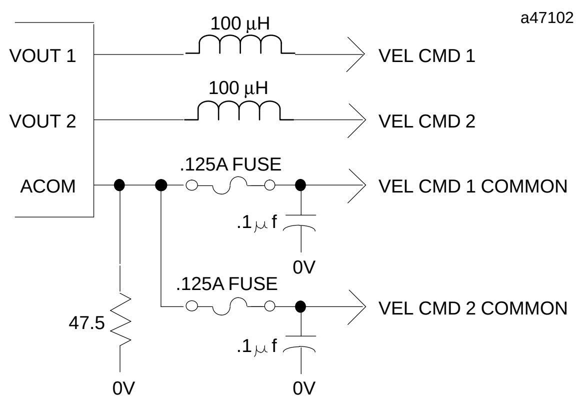


Figure C-1. Velocity Command Circuitry

Enable Relay Output

Normally-open dc solid state relay contact; Contacts rated at 30 V, 100 mA dc. Resistive load only. The off-state leakage current is 10 µA maximum.

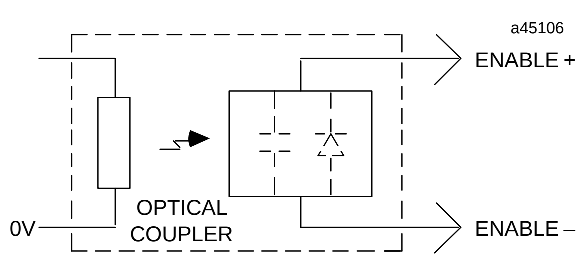
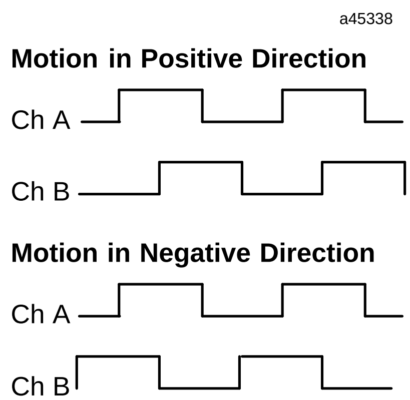


Figure C-2. Enable Relay Output Circuitry

Encoder Inputs

- Input Type: 5V differential or single ended
- Input Impedance: 4000 ohms (common mode)
- Input Threshold: Single ended: +1.5 V nominal (" 0.4V) , Differential: +0.5 V nominal (" 0.4V)
- Input Common Mode Range: " 15 V
- Signal Ended Input Voltage: + 15 V maximum
- Maximum Input Frequency: 175 kHz / Channel (X4 count rate = 700 kHz)
- Input Filtering: Noise pulses shorter than 1 Microsecond will be rejected.
- Encoder Power: Non-isolated current limited +5 V supply is available at the PM-APM front panel I / O connectors for use by one or more encoders. Maximum load must be limited to 500 mA @ 40°C, 300 mA @ 55°C.
- Quadrature Tolerance: 90 degrees " 30 degrees.
- Quadrature Error Detection: Simultaneous transitions on the Channel A and Channel B inputs will be detected as loss of quadrature.
- Z Channel (Marker Operation: A positive going edge on the marker channel will be used to latch the present encoder position. The minimum pulse width is 4 µs.
- Encoder direction of travel: Channel A leading Channel B indicates motion in positive direction:



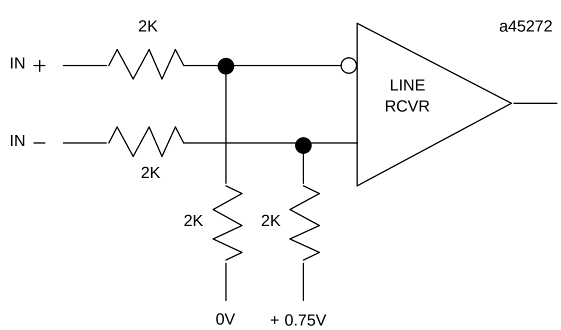


Figure C-3. Encoder Input Circuitry

Encoder Power

- Encoder Power: Non-isolated current limited +5 V supply is available at the PM-APM front panel I/O connectors for use by one or more encoders. Maximum load must be limited to 500 mA @ 40C, 300mA @ 55C

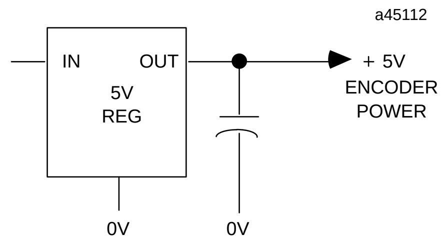


Figure C-4. Encoder Power Circuitry

General Purpose 24 V dc Digital Inputs

I/O CONNECTOR A:

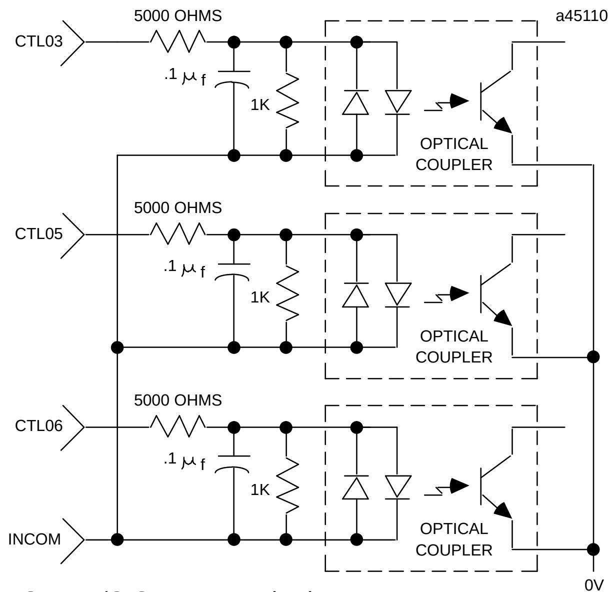
- Axis 1 Home Switch, CTL03
- Axis 1 Drive OK Switch, CTL05
- Axis (Encoder) 3 Home Switch , CTL06

I/O CONNECTOR B:

- Axis 2 Home Switch, CTL04
- Axis 2 Drive OK Switch, CTL07
- CTL08

Optically Isolated with the following specifications:

- Optically isolated DC source/sink (bidirectional input optocoupler)
- Input ON threshold: 18.0 V to 30.0 V
- Input OFF threshold: 0 V to 4.0 V
- Input resistance: 5000 ohms " 10%
- Input filtering: 5 ms nominal
- Isolation voltage: 1500 V peak transient



NOTE: I/O Connector A is shown.
INCOM on Connector A is isolated from INCOM on Connector B.

Figure C-5. General Purpose Input Circuitry

Analog Input

12-bit plus sign A / D converter with the following specifications:

- Input range: " 10.0 V
- Input impedance: 50 k ohms
- Common mode range: " 20 V
- Resolution: 12 bits magnitude plus sign bit
- Linearity: < 1 LSB
- Accuracy: 2 % of reading " 4 LSBs
- Scale Factor: 10.0 V = 32,000, –10V = –32,000
- Update Rate: 16 ms (does not include PLC scan time)

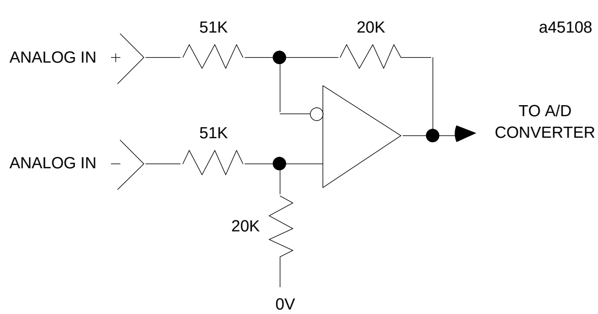


Figure C-6. Analog Input Circuitry

I/O Connector Cable Specifications

The cable that connects from the I/O connector to an external terminal block can be shortened to meet the requirements of your installation. Refer to Table 14 to correctly match cable wires with connector pins. Also, refer to Figures 2-4 through 2-7 and Tables 2-2 through 2-5 for specific wiring requirements.

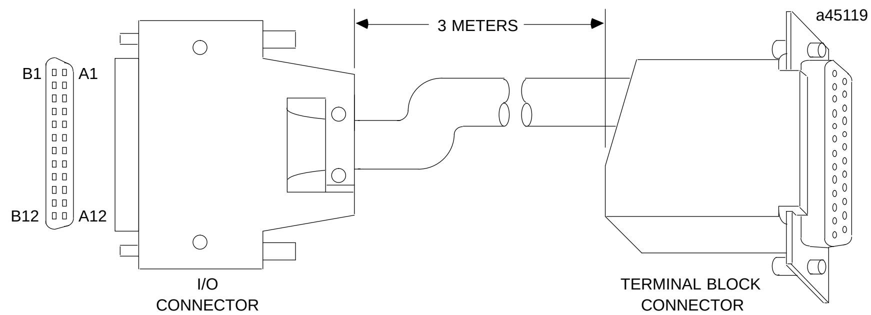


Figure C-7. I/O Connector Cable Specifications

I/O Cables

An I/O Cable assembly for the PM-APM modules consisting of a 24-pin connector, a cable, and a 25-pin D-type terminal block connector is available in two lengths. Catalog number IC693CBL311 is 10 feet (3 meters) in length and IC693CBL319 is 3 feet (1 meter) in length. The cable wire code list for these cables is provided in Table C-1.

An I/O cable assembly similar to IC693CBL311, but with the drain shield wire disconnected from pin B12 and brought outside of the cable housing through an 8" pigtail, is also available in two lengths. Catalog number IC693CBL317 is 10 feet (3 meters) in length and IC693CBL320 is 3 feet (1 meter) in length. *This cable improves the PM-APM's noise immunity*. The cable wire code list for these cables is provided in Table C-2.

Table C-1. I/O Cable Wire Coding for IC693CBL311 and IC693CBL319

I/O Connector Pin Number	Cable Wire Color Codes	25-Pin Connector Terminal Number*
no connection	Wire 1 Pair1(Brown / Black)	25
A1	Wire 2 Pair 1 (Brown)	12
B1	Wire 1 Pair2(Red / Black)	24
A2	Wire 2 Pair 2 (Red)	11
B2	Wire 1 Pair3(Orange / Black)	23
A3	Wire 2 Pair 3 (Orange)	10
B3	Wire 1 Pair 4 (Yellow / Black)	22
A4	Wire 2 Pair 4 (Yellow)	9
B4	Wire 1 Pair5(Green / Black)	15
A5	Wire 2 Pair 5 (Green)	2
B5	Wire 1 Pair6(Blue / Black)	14
A6	Wire 2 Pair 6 (Blue)	1
B6	Wire 1 Pair 7 (Violet / Black)	16
A7	Wire 2 Pair 7 (Violet)	3
B7	Wire 1 Pair8(White / Black)	17
A8	Wire 2 Pair 8 (White)	4
B8	Wire 1 Pair9(Gray / Black)	21
A9	Wire 2 Pair 9 (Gray)	8
B9	Wire 1 Pair10(Pink / Black)	20
A10	Wire 2 Pair 10 (Pink)	7
B10	Wire 1 Pair11(LightBlue / Black)	19
A11	Wire 2 Pair 11 (Light Blue)	6
B11	Wire 1 Pair12(LightGreen / Black)	18
A12	Wire 2 Pair 12 (Light Green)	5
B12	Drain Wire (Shield)	13

* Same as Terminal Block Terminal Number.

Table C-2. I/O Cable Wire Coding for IC693CBL317 and IC693CBL320

I/O Connector Pin Number	Cable Wire Color Codes	25-Pin Connector Terminal Number[1]
no connection	Wire 1 Pair1(Brown / Black)	25
A1	Wire 2 Pair 1 (Brown)	12
B1	Wire 1 Pair2(Red / Black)	24
A2	Wire 2 Pair 2 (Red)	11
B2	Wire 1 Pair3(Orange / Black)	23
A3	Wire 2 Pair 3 (Orange)	10
B3	Wire 1 Pair 4 (Yellow / Black)	22
A4	Wire 2 Pair 4 (Yellow)	9
B4	Wire 1 Pair5(Green / Black)	15
A5	Wire 2 Pair 5 (Green)	2
B5	Wire 1 Pair6(Blue / Black)	14
A6	Wire 2 Pair 6 (Blue)	1
B6	Wire 1 Pair 7 (Violet / Black)	16
A7	Wire 2 Pair 7 (Violet)	3
B7	Wire 1 Pair8(White / Black)	17
A8	Wire 2 Pair 8 (White)	4
B8	Wire 1 Pair9(Gray / Black)	21
A9	Wire 2 Pair 9 (Gray)	8
B9	Wire 1 Pair10(Pink / Black)	20
A10	Wire 2 Pair 10 (Pink)	7
B10	Wire 1 Pair11(LightBlue / Black)	19
A11	Wire 2 Pair 11 (Light Blue)	6
B11	Wire 1 Pair12(LightGreen / Black)	18
A12	Wire 2 Pair 12 (Light Green)	5
External Ring Terminal	Drain Wire (Shield)[2]	13

1 Same as Terminal Block Terminal Number.
2 16 gauge wire, green w / yellow tracer. 8” length (from back of connector), terminates with a #10 ring terminal.

The 24-pin I / O connector itself (which mates with the I / O Connector on the faceplate of the PM-APM) is available as three types and can be ordered as an accessory kit as listed below. Three types of connectors are available; solder pin, crimp pin, and ribbon cable. *Each accessory kit contains enough components (D-connectors, backshells, contact pins, etc.) to assemble ten single-ended cables of the type specified for each kit.*

IC693ACC316	FCN–361J024–A U	Solder eyelet receptacle
	FCN–360C024–B	Backshell (for above)
IC693ACC317	FCN–363J024	Crimp wire receptacle
	FCN–363J–A U	Crimp pin (for above; 24 needed)
	FCN–360C024-B	Backshell (for above)
IC693ACC318	FCN–367J024- AUF	IDC (ribbon) receptacle - closed cover
	FCN–367J024- AUH	IDC (ribbon) receptacle - open cover

Note that additional tools from Fujitsu are required to properly assemble the crimped contact and ribbon cable type connectors. The solder eyelet connectors (as provided in IC693ACC316) do not require any special tooling.

Crimped Contact Connectors (as provided in IC693ACC317) require :

Hand Crimping Tool	FCN-363T-T005 / H
Contact Extraction Tool	FCN-360T-T001 / H

Ribbon Cable Connectors (as provided in IC693ACC318) require :

Cable Cutter	FCN-707T-T001 / H
Hand Press	FCN-707T-T101 / H
Locator Plate	FCN-367T-T012 / H

These tools can be ordered from an authorized Fujitsu distributor. Three of the largest US distributors for Fujitsu connectors are Marshall at (800)522-0084, Milgray at (800)MILGRAY, and Vantage at (800)843-0707. If none of these distributors service your area, then contact Fujitsu Microelectronics in San Jose, California, USA via telephone at (408) 922-9000 or via fax at (408) 954-0616 for further information.

It is recommended that you order any necessary connector tooling with sufficient lead time to meet your assembly requirements for these connectors. These tools are generally not stock items and can have significant lead times from distribution. If you have any further questions about this issue, please feel free to contact the GE Fanuc PLC Hotline at 1-800-828-5747 or 804-978-5747.

Appendix D
Wiring to SS-90 Drives

When the Power Mate APM is used with the SS-90 Servo (Digital 100 / Digital 200) use the connection diagram below. The pin numbers shown correspond to the 25 pin female connector of the IC693CBL311A and also the terminals on the recommended terminal block. Machine inputs such as Overtravel and Home Switches are not shown in this diagram. Refer to Chapter 2, Installing the Power Mate APM for connection diagrams.

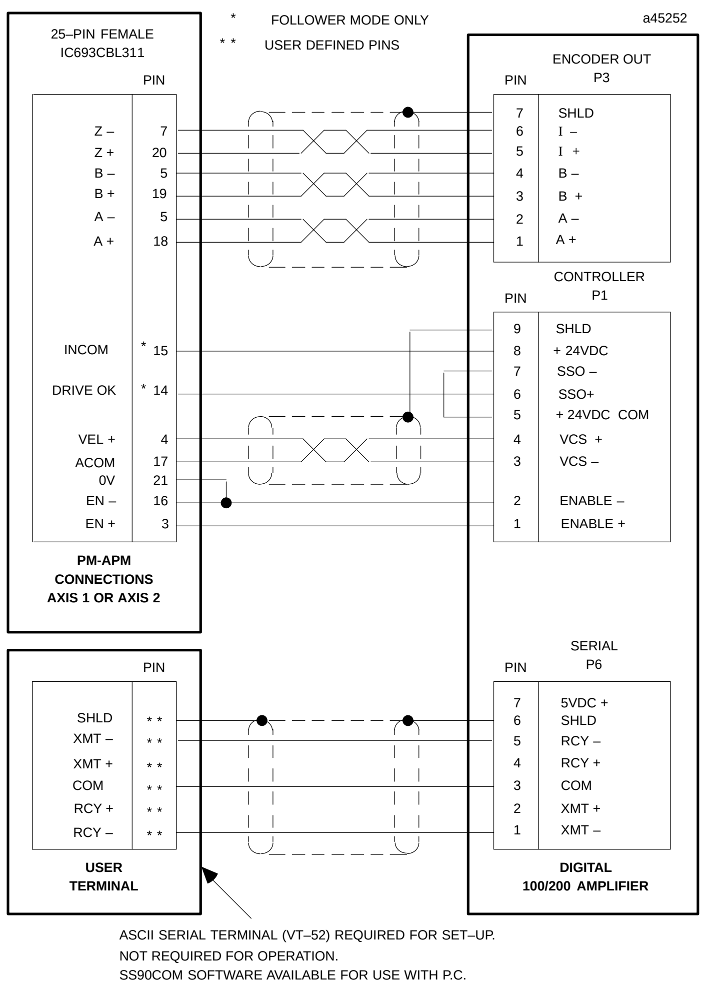


Figure D-1. SS–90 (Digital 100/Digital 200) Wiring

Note that the FAC+ and FAC- pins on the SS-90, as well as the RAC+ and RAC– pins, MUST be closed for the SS-90 to work.

Refer to the following documents for more information:

GFT-106 - SS-90 Servo System Product Selection Guide

GFK-0776 - Digital 100 / Digital 200 Brushless Servo Amplifier Instruction Manual

GFK-0777 - Analog 100 Brushless Servo Amplifier Instruction Manual

Appendix E Ordering Information

Ordering Information

Series 90-30 Power Mate APM	IC693APU301 (1 Axis with Follower 0ption) IC693APU302 (2 Axis with Follower option)
Motion Programmer Software	IC641SWP065
I/O Cable, 10 feet (3 meters) I/O Cable, 3 feet (1 meter) I/O Cable, 10 feet (3 meters) w/external shield wire I/O Cable, 3 feet (1 meter) w/external shield wire	IC693CBL311 IC693CBL319 (Consists of a 24-pin I/O connector, a cable, and a 25-pin, D-type terminal block connector). IC693CBL317 IC693CBL320 (Consists of a 24-pin I/O connector, a cable, and a 25-pin, D-type terminal block connector– also has an 8” external shield pigtail)
Terminal Block	Weidmuller RD25 910648 or equivalent (must be consistent with I/O cable IC693CBL311)
24-pin I/O Connector Accessory Kits (three types available, each kit has enough components to assemble ten single-ended cables)	IC693ACC316 (solder eyelet receptacle) IC693ACC317 (crimp wire receptacle) IC693ACC318 (IDC (Ribbon) receptacle)
Serial Communications Miniconverter Kit	IC690ACC901

Symbols

A

D

E

R

S

T

U

V