ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

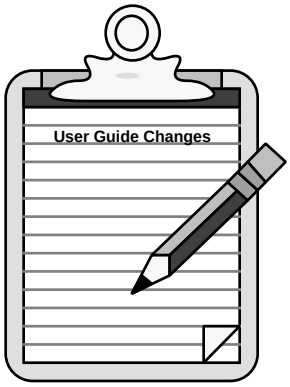
- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Parker / Compumotor CPHX* at our website: *Click HERE*

User Guide Change Summary



The following is a summary of the primary changes to this user guide since the last version was released. This user guide, version 88-007488-02**F**, supersedes version 88-007488-02**E**.

The entire user guide has been changed according to the new Compumotor user guide styles and illustration standards. Also, the chapters have been renumbered and reorganized.

Chapter ① Introduction

Changes to *Chapter 1, Introduction* are summarized as follows:

- ❏ There were no changes to this chapter

Chapter ② Getting Started

Changes to *Chapter 2, Getting Started* are summarized as follows:

- ❏ There were no changes to this chapter

Chapter ③ Installation

Changes to *Chapter 3, Installation* are summarized as follows:

- ❏ Minor revisions were made for clarity

Chapter ④ Application Design

Changes to *Chapter 4, Application Design* are summarized as follows:

- ❏ Minor revisions were made for clarity

Chapter ⑤ Software Reference

Changes to *Chapter 5, Software Reference* are summarized as follows:

- ❏ Several commands were added, including:
  BSP
  RIFS
  SP

Chapter ⑥ Hardware Reference

Changes to *Chapter 6, Hardware Reference* are summarized as follows:

- ❏ Minor corrections to the I/O Schematic
- ❏ Minor changes were made for clarity

Chapter ⑦ Maintenance & Troubleshooting

Changes to *Chapter 7, Maintenance & Troubleshooting* are summarized as follows:

- ❏ Minor changes were made for clarity

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C O N T E N T S

How To Use This User Guide iii
Process Overview iv
Conventions v
Related Publications v

① INTRODUCTION 1

Product Description 1
Elements of the System 2
Interface Options 2
Product Features 2

② GETTING STARTED 5

What You Should Have 5
Ship Kit Contents 5
Drive/Motor Configuration 6
Front Panel Description 6
Check-Out Procedure 6
Connect Power 7
Establish Communications 7
Turning the System On 8
Setting Motor Configuration 9
Setting Drive Current 9

③ INSTALLATION 11

Environmental Specifications 11
Complete System Configuration 11
Wiring Guidelines 12
connections 12
Motor Connections 13
Connector 13
RS-232C Connections (Serial Communications) 13
Using the Front-Panel Pushbuttons 14
RS-232C Daisy Chain Connections 14
Encoders 15
Fault Output 15
End-of-Travel Limits 15
Mounting
System 16
Mounting 16
Motor Mounting 16
Coupling the Motor to the Load 17
Setting Motor Resolution 18
Installation
Verifying 19
Bypassing Limit Switch Inputs for Test 19
Making the Motor Move 20
Homing The Motor 21
Testing Continuous Mode Moves 21
Final System Configuration 22
Inputs & Outputs 22

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

④ APPLICATION DESIGN 27

Defining Moves 27
Application Considerations 27
Positioning Modes 28
Programming Highlights 29
Interactive Programming 29
Programmable Delays 29
Program Branching 30
Input/Output 33
Programmable Outputs (POBs) 33
Event Completion Signals 34
Remote Jogging 35
Defining and Using Programs (Sequences) 35
Sequence Selection Methods 36
Standalone Operation 37
PLC Operation 38
Registration and Synchronization 40
Tuning Your System 42
Tuning Theory 42
General Tuning Considerations 43
Gain Descriptions 44
Tuning Your System Manually 44
Tuning Your System Automatically 46
Self Tuning Procedure 48
Tuning Problems 48

⑤ SOFTWARE REFERENCE 51

Command Format Description 51
Alphabetical Command List 54

⑥ HARDWARE REFERENCE 121

Environmental Specifications 121
Drive Electrical Specifications 121
Input power 121
Output Power 121
Connector Summary 122
I/O Diagram 123
I/O Schematics 123
I/O Specifications 124
Dimensional Drawings 126
Motor Dimensions 127
System Specifications 127
Accuracy & Repeatability 127
Specifications 127
Torque/Speed Curves 128

⑦ MAINTENANCE & TROUBLESHOOTING 129

Troubleshooting 129
Problem Isolation 129
Input/Output and Discrete Control Problems 130
Motor Control Problems 130
Electrical Noise 130
Diagnostic Codes 130
RS-232C 131
Debugging Your Motion Program 132
Common Problems 134

APPENDIX 135

Command Listing 135

INDEX 137

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

O V E R V I E W

How To Use This User Guide

This user guide is designed to help you install, develop, and maintain your system. Each chapter begins with a list of specific objectives that should be met after you have read the chapter. This section is intended to help you find and use information in this manual.

Assumptions

This user guide assumes that you have the skills or fundamental understanding of the following information.

- ❏ IBM (or IBM-compatible) computer experience
- ❏ Basic electronics concepts (voltage, switches, current, etc.)
- ❏ Basic motion control concepts (torque, velocity, distance, force, etc.)

With this level of understanding, you can effectively use this manual to install, develop, and maintain your system. You can get information about basic motion control concepts by referring to the current Parker Compumotor Motion Control Catalog.

User Guide Contents

This user guide contains the following information.

Chapter ① Introduction	This chapter provides a description of the product and a brief account of its specific features.
Chapter ② Getting Started	This chapter contains a list of items you should have received with your shipment. It will help you become familiar with the system and ensure that each component functions properly. You will configure the system properly in this chapter.
Chapter ③ Installation	This chapter helps you properly mount the system and make all electrical connections. Upon completion of this chapter, your system should be completely installed and ready to perform basic operations.
Chapter ④ Application Design	This chapter helps you customize the system to meet your application's needs. Important application considerations are discussed. Sample applications are provided.
Chapter ⑤ Software Reference	This chapter explains Compumotor's X Series programming language in detail. It describes command syntax and system parameters that affect command usage. An alphabetical list of all commands, with a syntax and command description for each command is included.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter ⑥ Hardware Reference

This chapter contains information on system specifications (dimensions and performance). This chapter is as a quick-reference tool for proper I/O connections.

Chapter ⑦ Maintenance & Troubleshooting

This chapter describes Compumotor's recommended system maintenance procedures. It also provides methods for isolating and resolving hardware and software problems.

Process Overview

To ensure trouble-free operation, you should pay special attention to the environment in which the Compumotor Plus equipment will operate, the layout and mounting, and the wiring and grounding practices used. These recommendations are intended to help you easily and safely integrate Compumotor Plus equipment into your facility. Industrial environments often contain conditions that may adversely affect solid state equipment. Electrical noise or atmospheric contamination, may also affect the Compumotor Plus System.

Developing Your Application

Before you develop and implement your application, there are several issues that you should consider and address.

- ❑ Clarify the requirements of your application. Clearly define what you expect the system to do.
- ❑ Assess your resources and limitations. This will help you find the most efficient and effective means of developing and implementing your application.
- ❑ Follow the guidelines and instructions outlined in this user guide. Do not skip any steps or procedures. Proper installation and implementation can only be ensured if all procedures are completed in the proper sequence.

Installation Procedures

Before you attempt to install this product, you should complete the following steps:

① Review this entire user guide. Become familiar with the manual's contents so that you can quickly find the information you need.

② Develop a basic understanding of all system components, their functions, and interrelationships.

③ Complete the basic system configuration and wiring instructions provided in *Chapter 1, Getting Started*.

④ Perform as many basic moves and functions as you can with the preliminary configuration. You can only perform this task if you have reviewed the entire user guide. You should try to simulate the task(s) that you expect to perform when you permanently install your application (however, do not attach a load at this time). This will give you a realistic preview of what to expect from the complete configuration.

⑤ After you have tested all of the system's functions and used or become familiar with all of the system's features, carefully read *Chapter 3, Installation*.

⑥ After you have read *Chapter 3, Installation and* clearly understand what must be done to properly install the system, you should begin the installation process. Do not deviate from the sequence or installation methods provided.

⑦ Before you begin to customize your system, check all of the system functions and features to ensure that you have completed the installation process correctly.

The successful completion of these steps will prevent subsequent performance problems and allow you to isolate and resolve any potential system difficulties before they affect your system's operation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Conventions

To help you understand and use this user guide effectively, the conventions used throughout this user guide are explained in this section.

Warnings & Cautions

Warning and caution notes alert you to possible dangers that may occur if you do not follow instructions correctly. Situations that may cause bodily injury are presented as warnings. Situations that may cause system damage are presented as cautions. Refer to the examples shown below.

CAUTION

Remember, electrical noise, a poorly designed enclosure, or improper grounding can affect system performance and safety. For more information about these factors, refer to *Chapter 7, Troubleshooting and Maintenance*.

Related Publications

The following publications may be helpful resources.

The current Parker Compumotor Motion Control Catalog. The catalog is available from Parker Compumotor at (800) 358-9068 or (707) 584-7558.

Schram, Peter (editor). The National Electric Code Handbook (*Third Edition*). Quincy, MA: National Fire Protection Association.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ①

Introduction

Chapter Objectives

The information in this chapter will enable you to:

- ❑ Understand the product's basic functions & features
- ❑ Understand basic motion control concepts and apply them to your application

Product Description

The Compumotor Plus Indexer/Drive is a powerful, *stand-alone*, motion-control device. It provides users with an indexer and driver in one package. A simple RS-232C communications interface allows users to implement Compumotor Plus' powerful command language with minimal setup and preparation. The system's ability to operate up to 16 Compumotor Plus Drives in a daisy chain configuration, from one RS-232C device, further simplifies the Compumotor Plus' implementation in more complex applications.

The Compumotor Plus is equipped with optically isolated CCW and CW end-of-travel limit inputs and a home limit input. The end-of-travel limits can be disabled using the Limit Disable (**LD3**) command.

In addition, the Compumotor Plus has three trigger inputs for general use with the Wait for Trigger (**TR**) command. These inputs can also be configured as jog inputs, stop inputs, or sequence select inputs.

The Compumotor Plus system provides high accelerations (up to 10 times higher than open-loop stepper systems) and high torque's in each frame size. The torque profiles have been chosen to provide high torque's at low to moderate speeds. This makes the Compumotor Plus system ideal for point-to-point positioning applications where increased throughput is important.

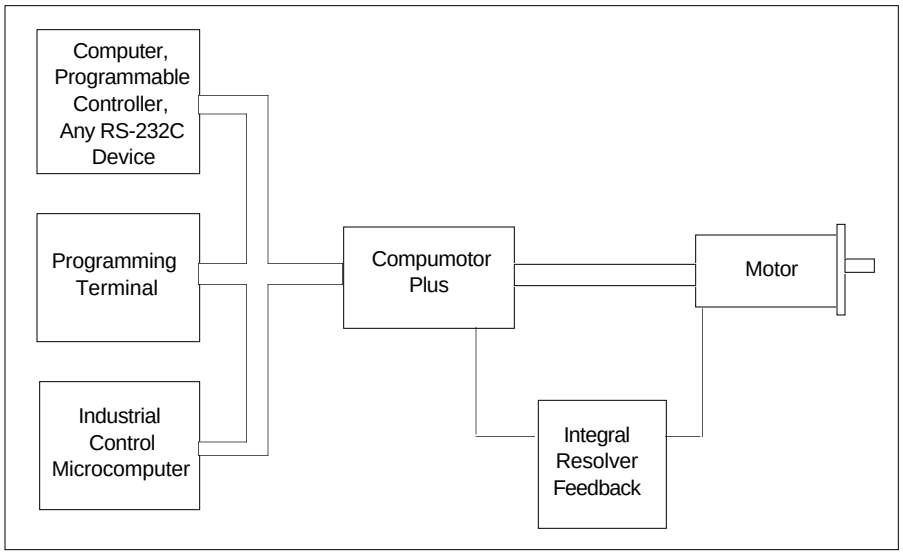
Since the Compumotor Plus system runs closed loop, current is only produced when torque demands require it. Consequently, the Compumotor Plus system runs much cooler than comparable open-loop steppers. This can be important in applications where motor heating is a factor (such as stage drives).

The Compumotor Plus system runs as a servo (i.e., the motor cannot stall in the same way that conventional open-loop steppers can). If torque demands exceed the capability of the system, the motor just slows down and tries to keep moving. If the load absolutely prohibits the motor from

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

moving, the drive can be programmed to abort the program or move, indicate a fault to the operator or control system, and display an error code on the LED.

Elements of the System



Interface Options

The Compumotor Plus system is offered with three different standard interfaces. All three versions have push-button and RS-232C tuning, Velocity Monitor and Fault outputs, and Hardware Enable input. The system also offers CW/CCW limit inputs, and a display that indicates set-up and fault information. The exact nature of any fault (e.g., excess average current, excess following error, loss of resolver signal, etc.) is displayed as a unique two-digit number.

The X version of the Compumotor Plus system contains the servo controller and a complete, RS-232C-based indexer that executes the Compumotor X programming language. Motion control parameters such as distance, velocity, and acceleration can be combined into sequences along with time delays, loops, and programmable outputs. These sequences can be controlled via RS-232C or via external inputs that can control and time program execution. Trigger inputs can be used to coordinate program execution with external events. Multiple motion-control programs can be stored in non-volatile memory and executed automatically or according to external sequence and trigger controls.

The X language is based on simple, readable commands that use ordinary decimal numbers. Any CRT, hand-held terminal, computer, or programmable controller with an RS-232C interface can be used to program the system.

Part Number Designation

	Low Power	High Power
X version	CPLX	CPHX

Product Features

The Compumotor Plus servo system is a brushless, digital, closed loop positioning system based on motor and drive technology developed by Compumotor. It offers many innovative features, including:

- ❑ Motor with integral brushless resolver

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- ❑ Industry standard 23, 34, and 42 stepper compatible frame sizes
- ❑ Digital velocity and position loops with simple PID tuning
- ❑ Front panel push-button tuning with LED diagnostic display
- ❑ Standard RS-232C interface provides on-line setup assistance
- ❑ Velocity monitor output provides real-time performance data
- ❑ Three interface options
  - ❍ Built-in RS-232C indexer executes Compumotor X language and offers non-volatile program memory and program sequence control
  - ❍ Step & Direction interface compatible with all Compumotor Indexer
- ❑ Integrated controller/amplifier has built-in power supply, low EMI filtered amplifier outputs, full optical isolation on all user connections, 2 digit diagnostic LED display

Theory of Operation

Compumotor has joined each motor with a new digital control system designed specifically to take advantage of the performance characteristics of the motor. Each drive is fully packaged with integral power supply, filtered power amplifier (for low EMI and low motor heating), and controller card. The controller is based on a 68,000 16-bit microprocessor and digitally closes the position and velocity loop. Loop gains (specified as proportional, integral, derivative, and velocity gains) are factory preset for stability with moderate frictional and inertial loads. These gains may be adjusted with either hidden front panel push-buttons (assisted by a 2 digit LED display) or with the unit's standard RS-232C interface. With the RS-232C interface the user can query the servo system about performance data such as following error, average and peak currents, set points, and gains. Each drive also has a Velocity Monitor output which provides a synthesized tachometer signal that can be used in conjunction with an oscilloscope to visually monitor system performance.

A brushless resolver made of the same rotor and stator components used in the motor was developed. This resolver is magnetically similar to the motor itself, and is manufactured as an integral part of the motor. The result is a sensor with the same number of poles as the motor (which makes control straightforward) that is always properly aligned with the motor and can be operated in high noise, high temperature environments.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ②

Getting Started

The information in this chapter will enable you to:

- ❏ Verify that each component of your system has been delivered safely
- ❏ Become familiar with system components and their interrelationships
- ❏ Ensure that each component functions properly
- ❏ Configure the system properly

What You Should Have

You should inspect your Compumotor Plus shipment upon receipt for obvious damage to its shipping container. Report any such damage to the shipping company as soon as possible. Parker Compumotor cannot be held responsible for damage incurred in shipment. Carefully unpack and inspect your Compumotor Plus shipment. The items listed in the following tables should be present and in good condition.

Ship Kit Contents

The following table lists the components included in the ship kit for the low-power drive, the CPLX. *The CPLX-Drive can only be used with Compumotor CPLX57-120 and CPHX83-150 motors.*

Part Number	Low Power Ship Kit Description	Qty
47-007386-01	36VAC Transformer	1
71-007819-10	10-Foot Motor Cable (for 57-120/83-150 motors)	1
71-007811-10	Resolver Cable	1
52-006007-01	Mounting Bracket	1
88-007488-02	User Guide	1

The following table lists the components included in the ship kit for the high-power drive, the CPHX. *The CPHX can only be used with Compumotor CPH83-150 and CPH106-220 motors.*

Part Number	High Power Ship Kit Description	Qty
71-007547-01	AC Cable (high-power only)	1
71-007817-10	Motor Cable (for 106-220)	1
71-007811-10	Resolver Cable (for all models)	1
52-006007-01	Mounting Bracket	1
88-007488-02	User Guide	1

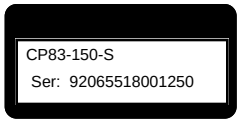
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drive/Motor Configuration

Compumotor configures the Compumotor Plus system according to the motor that you order. There are three motors that can be configured with the Compumotor Plus (57-120, 83-150, and 106-220). You can determine which motor you have by checking the Product label.

Drive Label

Motor Label

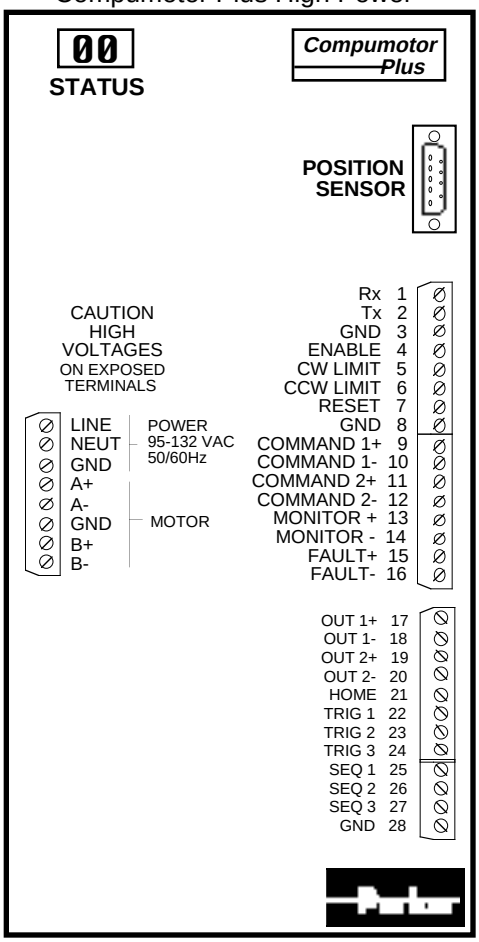


Serial #'s must match

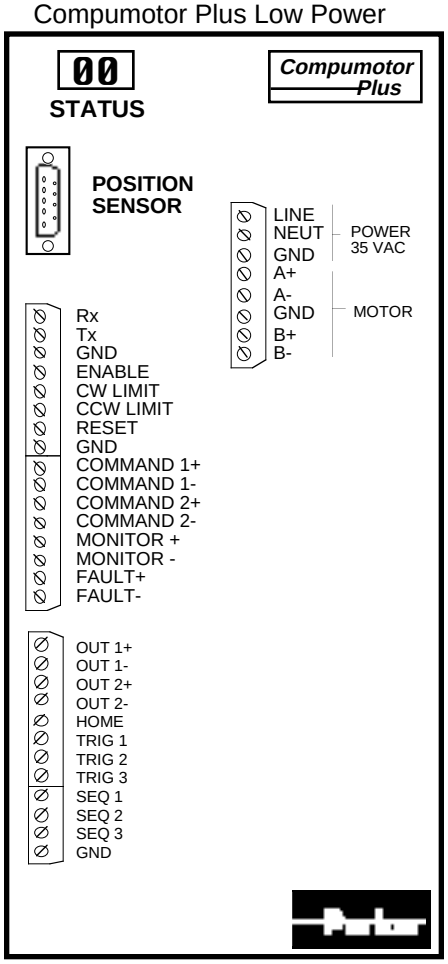
Front Panel Description

The front panels for the Compumotor Plus low and high power are different.

Compumotor Plus High Power



Compumotor Plus Low Power



Check-Out Procedure

This section provides a basic ***bench test*** of the Compumotor Plus. Compumotor recommends that you complete the steps in this chapter before you permanently install the Compumotor Plus and connect the motor to its intended load. The above figure illustrates the location of these connections.

Connect Power

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

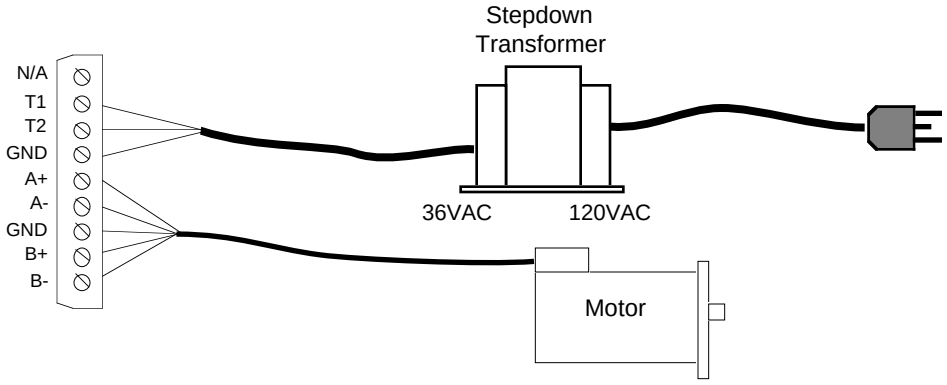
Connect the power cable to the Compumotor Plus and verify that the color codes are correct.

The CPLX57-120 and CPLX83-150 are provided with an input step-down transformer and the CPHX106-120 is provided with an AC power cord and no transformer.

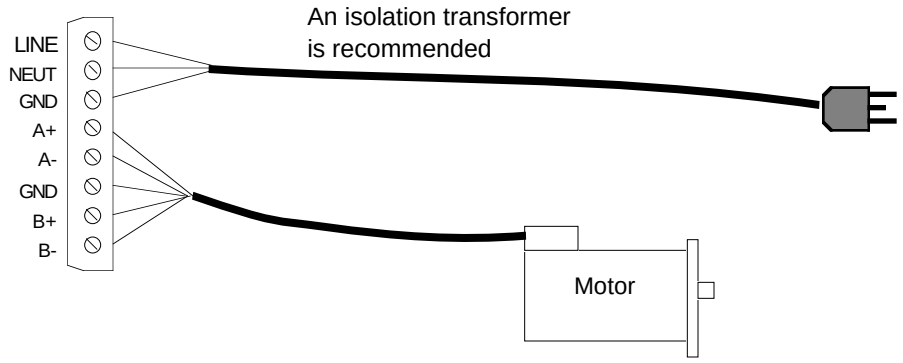
Compumotor Plus power connections are listed in the table below.

Pin #	CPL Function	Wire Color	CPH Function	Wire Color
1	N/A	No Connection	N/A	No Connection
2	T1	Brown	LINE	Black
3	T2	Blue	NEUT	White
4	GND	Green	GND	Green

The drawing below illustrates the power connections for the CPL (low-power) drive.



The drawing below illustrates the power connections for the CPH (high-power) drive.



Establish Communications

To communicate with the Compumotor Plus system, your computer or terminal must have an RS-232C serial port. If it does not, you can purchase one from your local computer dealer.

RS-232C Connections

The RS-232C connections are made via screw terminals. The Compumotor Plus Drive has a three-wire implementation of this interface and provides Receive Data (Rx), Transmit Data (Tx), and Ground (GND) signals on the connector. Refer *Chapter 3, Installation* for more information.

The communication parameters default to 9,600 baud, 8 data bits, 1 stop bit, and no parity (full-duplex).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Issuing Commands

The Compumotor Plus requires that commands be issued over the RS-232C connection with the following syntax. Refer to *Chapter 5, Software Reference* for more information.

<address>COMMAND<parameter><delimiter>

- The address is optional for all commands except those which require a response. You will not need to include an address for most of the examples in this section. Where an address is required, it will be provided with the command.
- The command itself is a sequence of one or several uppercase letters.
- A numeric parameter is sometimes required. It will be listed with the command in the examples in this section.
- A delimiter is always required to end the command. A delimiter is a space or carriage return.

Turning the System On

To familiarize yourself with the operation of the system, you may wish to go through these start-up procedures before final installation of the motor to your load. If so, you should first attach the motor, resolver, and RS-232C cables before applying AC power. Remember, the CPL drive (low-power) uses 36VAC and requires an isolated step-down transformer (the required transformer is supplied with the low-power systems). The CPH (high power) does not require a transformer to operate from 120VAC, but an isolation transformer is recommended. For more information, refer to *Chapter 3, Installation*.

CAUTION

Be careful to keep the motor shaft away from any cables or other loose objects that could get tangled when the shaft rotates. The motor should be firmly mounted to prevent it from moving while it is running.

Do not grab the motor shaft while it is turning. There may be sharp surfaces on the shaft. Also be sure to remove the shaft key (if there is one). The key may fly off when the motor is rotating.

Verify that all cables and wires are properly connected, and that the motor shaft is free from obstructions before applying power to the system. If the system has been wired properly and the drive is enabled, the diagnostic display will read zero.

If there is a fault, the LED display will flash a diagnostic code, refer to *Chapter 7, Maintenance & Troubleshooting* for an explanation of those codes.

Connection Overview

Operation of the Compumotor Plus requires a minimum of four sets of electrical connections.

- The motor and resolver
- Interface to an RS-232C serial communication device (which may be removed after programming)
- AC power to the drive
- Other optional connections include CW and CCW end-of-travel limit inputs, a home position limit input, trigger and sequence inputs, and programmable and fault outputs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setting Motor Configuration

The Compumotor Plus Drive has been configured to its motor for you at the factory, if they were ordered together as a system.

If you are using a motor other than the one that was ordered with the system, you will need to re-configure the Compumotor Plus for that size motor. The Configure Motor (**CMTR**) command should be used for this purpose (refer to *Chapter 5, Software Reference*).

Follow the steps listed below to configure your Compumotor Plus motor.

Step ①

Connect the motor to the drive. Connect the power cables (supplied). Connect an RS-232C terminal or terminal emulator. Refer to *Chapter 3, Installation* for the appropriate wiring diagrams. Apply power to the Compumotor Plus.

Step ②

Refer to the table below to select the appropriate **CMTR** command for your motor. This command selects the proper current values and factory defaults.

Issue the following commands to the Compumotor Plus to configure it for your motor. Be sure to type a space or carriage return after each command. Type only the commands in the left hand column, not the description in the right hand column.

Example

Command	Description
OFF	Turn off the current to the motor.
CMTRx	Select the appropriate **CMTR** command from the table below.
ON	Check the motor for stability. If it appears to have torque and no error messages are sent to the terminal, proceed to the next step.
SAVE	Save the configuration.

You must issue an **OFF** command prior to configuring the motor. You must issue a **SAVE** command after issuing a **CMTR** command. The configuration is not saved automatically.

The possible configurations are:

Command	Description
1CMTR	Reports the present set up as *MOTOR_TYPE=CPn, where n=1, 2, or 3
1CMTR1	Sets the drive up for the 57-120 motor
1CMTR2L	Sets the drive up for the 83-150 motor (with Low power amplifier)
1CMTR2H	Sets the drive up for the 83-150 motor (with high power amplifier)
1CMTR3	Sets the drive up for the 106-220 motor

Setting Drive Current

The drive current is set at the factory and normally does not need to be changed. If your application requires a different current the Configure Current Average (**CCA**) and Configure Maximum Current Peak (**CCP**) commands allow you to change the current settings (Refer to *Chapter 5, Software Reference*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ③

Installation

The information in this chapter will enable you to:

- ❏ Mount all system components properly
- ❏ Connect all electrical and non-electrical system inputs and outputs properly
- ❏ Ensure that the complete system is installed properly
- ❏ Perform basic system operations

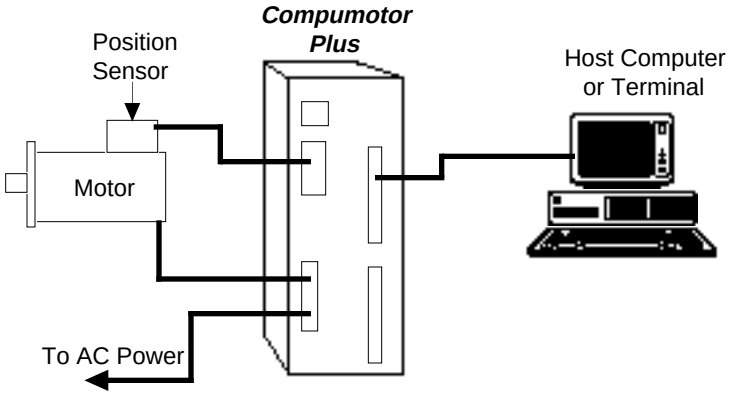
Environmental Specifications

The environmental temperature and humidity limitations for the Compumotor Plus are listed below.

Condition	Specification
Ambient (operating) temperature	0 to 50°C (32 to 122°F)
Max drive heatsink temperature	75°C (162°F) (Low Power only)
Max temperature source below drive	45°C (113°F)
Max motor case temperature	100°C (212°F)
Storage temperature	-40 - 85°C (-40 - 185°F)
Humidity	0 to 95% non-condensing

Complete System Configuration

In this section, you will go through complete set-up procedures to set the Compumotor Plus functions. The wiring requirements will be described and a check out procedure to verify the functionality of the interface is provided.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Wiring Guidelines

Proper grounding of electrical equipment is essential to ensure the safety of personnel. You can reduce the effects of electrical noise due to electromagnetic interference (EMI) by grounding. All Compumotor equipment should be properly grounded. A good source of information on grounding requirements is the National Electrical Code published by the National Fire Protection Association of Boston, Massachusetts.

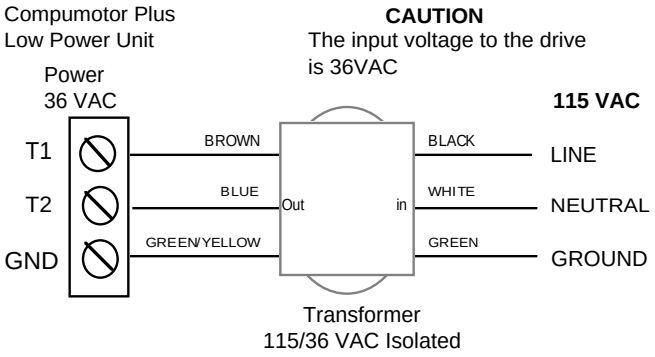
In general, all components and enclosures must be connected to earth ground through a grounding electrode conductor to provide a low impedance path for ground fault or noise-induced currents. All earth ground connections must be continuous and permanent. Compumotor recommends that you use a single-point ground.

One commonly used grounding method is to prepare components and mounting surfaces prior to installation so that good electrical contact is made between mounting surfaces of equipment and enclosure. Remove the paint from equipment surfaces where the ground contact will be bolted to a panel and use star washers to ensure solid bare metal contact. You should connect the case of the motor to the Motor Ground terminal on the Compumotor Plus Drive. (This is done for you with Compumotor supplied cables.)

For temporary installation, or when you cannot implement the grounding method described above, you must connect the Ground terminal on the AC power connector to the earth ground.

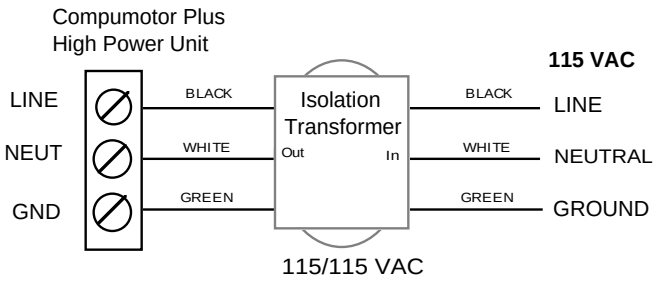
Line Power Connections

The CPLX57-120 and CPLX83-150 operate on 36VAC single-phase power. The 36VAC is supplied from a 115/36VAC isolated transformer that is provided with the unit (externally). The connections from the transformer to the indexer/drive are discussed below.



High Power Connections

The CPHX83-150 and CPHX106-220 operate on 120VAC single-phase power. AC power is connected to the screw terminal connector on the front panel. The 120VAC single-phase should be connected with 14-gauge or heavier, stranded wire. The wires should be connected to the screw terminals as shown in the figure below.



Note: The isolation transformer is provided by the user, the transformer is recommended but not required.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Isolation Transformer (CPH only)

The Compumotor Plus high-power indexer/drive does not provide line isolation for the input power. If your installation requires line isolation, use an isolation transformer with a 1KVA rating.

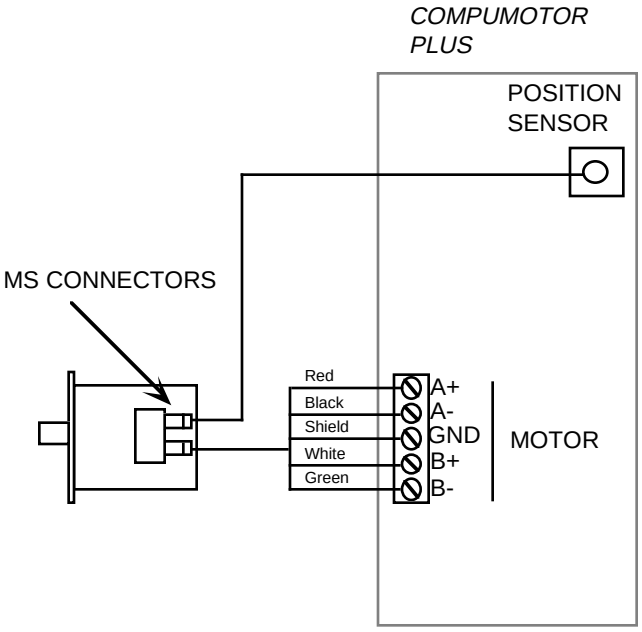
An isolation transformer with line filtering capabilities also protects sensitive equipment on the same power line from power line noise produced by the switch-mode amplifiers in the Compumotor Plus Drive.

The Compumotor Plus low-power indexer/drives (CPLX57-120 and CPLX83-150) are supplied with step-down, isolating transformers to step down 120VAC to 36VAC.

Motor Connections

The Compumotor Plus Motor is supplied with a preassembled motor cable. The cable has an MS-type connector on the motor end, and five leads with screw terminals on the drive end. This cable should be attached to both the motor and the drive before the AC power is connected. The motor connections on the Compumotor Plus Drive are made to a screw terminal block on the front panel. The terminals are marked as shown in the figure below.

Caution
Do not disconnect the motor cable while the drive is powered.



Resolver Connections

The resolver cable has an MS-style connector on the motor/resolver end and a 9-pin, D-type connector on the drive end. The cable cannot be installed on the wrong connector. Attach the MS-style connector to the motor an the D-type connector to the motor/drive.

RS-232C Connections (Serial Communications)

The RS-232C connector is a screw terminal connector and is electrically compatible with the EIA specifications for RS-232C communications. The Compumotor Plus Drive has a three-wire implementation of this interface and provides Receive Data (Rx), Transmit Data (Tx), and Ground signals on the connector.

To set the baud rate, expose the front panel pushbuttons by removing the Compumotor Plus logo on the front panel. Press the V and D buttons together and hold. Press the **+** or **—** buttons to select the next and previous baud rate, respectively. Refer to the section below on Tuning for a detailed description of button locations. The available baud rates are shown in the table below.

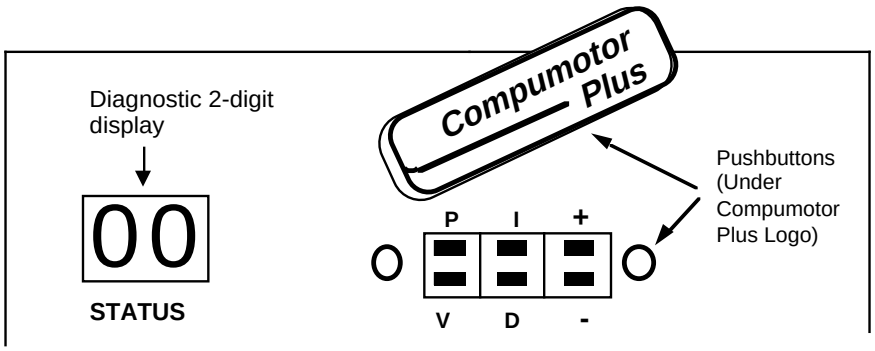
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Baud rates are selected through the front-panel push buttons

Baud rate	
300	
600	
1,200	
2,400	
4,800	
9,600	**Default setting**
19,200	

Using the Front-Panel Pushbuttons

The following section describes how to use the pushbuttons located under the Compumotor Plus logo on the unit's front. The pushbuttons are useful for setting gains, saving parameters, resetting the unit, setting the baud rate for RS-232 and setting the units device address.



Tuning pushbutton adjustments

Function	Show value	Adjust (press together)
Integral Gain	Press I	Press I and **+** or **-** .
Velocity gain	Press V	Press V and **+** or **-**
Proportional gain	Press P	Press P and **+** or **-**
Derivative gain:	Press D	Press D and **+** or **-**

Miscellaneous pushbutton adjustments

Function	Show value	Adjust (press together)
Device address:	Press V and P	Press V, P and **+** or **-**
Baud rate	Press D and V	Press D,V and **+** or **-**

Miscellaneous pushbutton functions

Function	Keys
Return to factory settings	Press I and P
Drive reset	Press **+** and **-**
Display software revision	Press D and I

RS-232C Daisy Chain Connections

You can daisy chain up to 16 Compumotor Plus Drives to a single RS-232C port on a computer or terminal. Use the figure below as a guide for daisy chain connections. You must establish a unique device address for each Compumotor Plus so that you can distinguish them when programming. The device address is set with the front panel pushbuttons (refer to *Chapter 4, Application Design* for appropriate settings).

To set the device address follow the steps shown below.

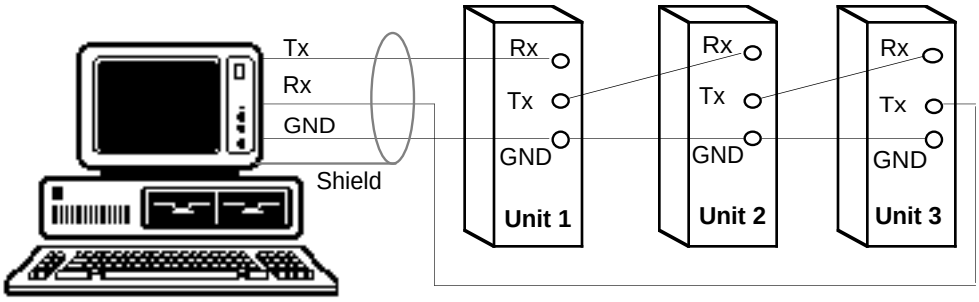
❑ Remove the Compumotor Plus logo on the front of the unit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

❑ Press the V and P buttons together and hold. The current device address will appear in the diagnostic display.

❑ Press and hold the V and P keys, the press + or — to increase or decrease the device address, respectively.

RS-232C Interface



The daisy chain feature is useful when you are setting up or monitoring the motor's performance from multiple units.

Encoders

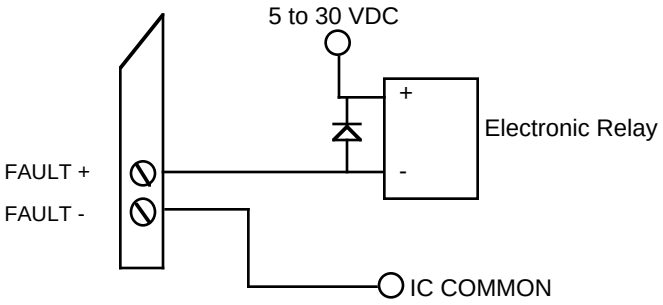
The Compumotor Plus is not capable of accepting encoder feedback. However, you can install an encoder on the back of the motor and use a separate indexer or encoder monitoring device to track motor position.

Fault Output

The system provides two complimentary outputs that signal when the amplifier has shut off motor current due to a fault (e.g., excessive following error). The ouputs, named **Fault+** and **Fault-**, are the collector and emitter, respectively, of a 60 mA, optically isolated transistor. A separate DC power supply is required to power the output. This assures the signal is valid even if the Compumotor Plus has not power.

☛ **Hint** *If you use a mechanical relay you must use a blocking diode.*

Connect the anode of the diode to the **Fault+** output and the cathode to your positive terminal of your DC power supply. Connect **Fault-** to the negative terminal of your DC power supply.



End-of-Travel Limits

The Compumotor Plus has two end-of-travel limit inputs, (named the **CW LIMIT** and the **CCW LIMIT**) they are connected to ground to allow motor movement. ***Note: If your motor does not respond to motion commands over RS-232C, be sure to check the end-of-travel limit inputs.*** Limits may be bypassed using the Limit Disable (**LD3**) command over the RS-232C interface. Normally closed switches, preferably electronic, must be used. These inputs draw 12 mA from the internal 12VDC supply.

The limit inputs are set up to function with normally closed contact switches in order to provide fail-safe operation. This way, if a limit switch wire is disconnected or broken, motion in the direction of the disabled limit is

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

inhibited. Refer to *Chapter 2, Getting Started* for a detailed description of how to disable the limits.

System Mounting

You should give special attention to the environment and location in which you will operate your Compumotor Plus Series system. Consider atmospheric contamination and excess heat before you install and operate your Compumotor Plus system. Refer to the table outlining the environmental specifications of the Compumotor Plus at the beginning of this chapter.

Drive Mounting

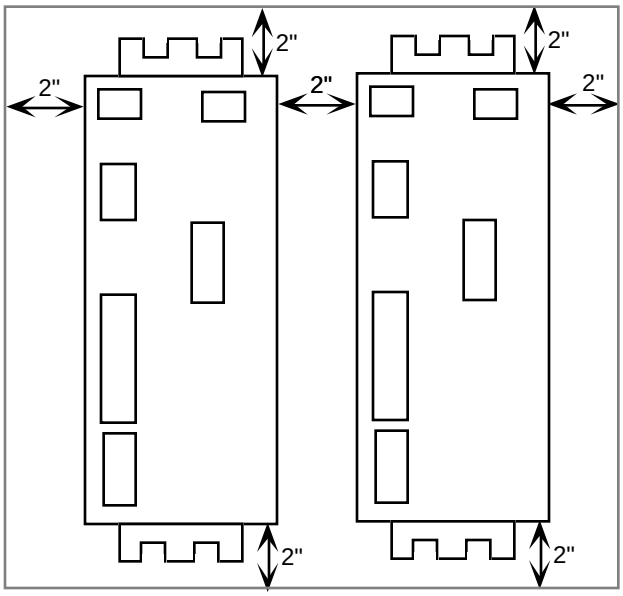
Safety is of prime importance when installing a motion-control system. This section outlines installation guidelines with the safety of the operator and the equipment in mind. You should become familiar with this section and with local and national codes that pertain to the installation of electrical equipment.

If you mount the Compumotor Plus in an enclosure, observe the following guidelines:

- ❏ The vertical clearance between the Compumotor Plus Drive and other equipment, or the top or bottom of the enclosure, should be no less than 2 inches.
- ❏ The horizontal clearance should be no less than 2 inches.
- ❏ Do not mount large, heat-producing equipment directly beneath the Compumotor Plus.
- ❏ Do not mount the Compumotor Plus Drive directly below an indexer (the Compumotor Plus produces more heat than an indexer).
- ❏ The maximum allowable ambient temperature directly below the Compumotor Plus is 45°C. Fan cooling may be necessary if adequate air flow is not provided with the low power units.

The following illustration describes suggested clearances around the Compumotor Plus upon installation.

☛ **Hint** *Leave at least a 2" space around the drive to provide adequate ventilation. Do not put heat sensitive units above the drive.*



Motor Mounting

The motor should be mounted using bolts through the flange and centered by the pilot on the front face. The pilot is the raised circular portion of the flange end of the motor. Refer to the *Mechanical Reference* section for motor dimensions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CAUTION

Remember, electrical noise, a poorly designed enclosure, or improper grounding can affect system performance and safety. For more information about these factors, refer to *Chapter 7, Troubleshooting and Maintenance*.

The motors used with the Compumotor Plus Drive can produce very large torques. These motors can also produce high accelerations. This combination can shear shafts and mounting hardware if the mounting is inadequate. High accelerations can produce shocks and vibrations that require much stronger hardware than required for similar static loads. The motor can set up low-frequency vibrations in its mounting hardware.

These vibrations will cause fasteners to loosen if they are not locked. This can be prevented by using a locking nut such as an aircraft grade NyLock™ nut. These vibrations can also cause metal fatigue in structural members if harmonic resonances are induced by the move profiles. A thorough mechanical engineering analysis of the machine should be performed to ensure the mounting structure is adequate.

Coupling the Motor to the Load

The coupling you choose to link the motor to the load affects the accuracy and efficiency of your application. A bad coupling can cause a large proportion of the motor's power to be lost before it gets to the load. This can be caused by poor shaft alignment between the motor and the load and by loose and/or poor fitting hardware.

Special couplings that accommodate different types of misalignment are available. The following are the three types of misalignments; they can exist in any combination.

- ❑ Parallel Misalignment. The offset of two mating shaft center lines, although the center lines remain parallel to each other.
- ❑ Angular Misalignment. When two shaft center lines intersect at an angle other than zero degree.
- ❑ End Float. A change in the relative distance between the ends of two shafts

Special couplings are used to accommodate the above misalignments and to transmit the desired torque. The coupling manufacturer should be consulted to ensure that the coupling is being used within its specified torque capacity and alignment ranges.

Shaft couplings may be divided into three types: single-flex, double-flex, and rigid. Like a hinge, single-flex couplings accept angular misalignment only. A double-flex coupling accepts both angular and parallel misalignments. Both single-flex and double-flex, depending on their design, may or may not accept end-play. Rigid couplings do not compensate for any misalignment.

CAUTION

Do not machine the motor shaft without consulting a Compumotor Applications Engineer at (800) 358-9070. Improper shaft machining can destroy the motor's bearings.

Single-Flex Coupling

When a single-flex coupling is used, one and only one of the shafts must be free to move in the radial direction without constraint. *Do not use a double-flex coupling in this situation because it will allow too much freedom and the shaft will rotate in an eccentric fashion; this will cause large vibrations and eventual coupling failure.*

Double-Flex Coupling

Use a double-flex coupling whenever two shafts are joined that are fixed in the radial and angular direction (angular misalignment). *Do not use a single-flex coupling with a parallel misalignment; this will bend the shafts, causing excessive bearing loads and premature failure.*

Rigid Coupling

Rigid couplings are generally not recommended. They should be used only if the motor is on some form of float mounting which allows for misalignment.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Coupling Manufacturers

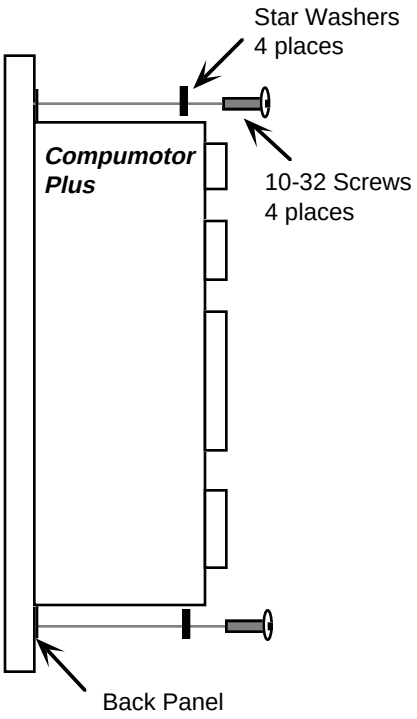
HELI-CAL
901 McCoy Lane
P.O. Box 1460
Santa Maria, CA
(805) 928-3851

ROCOM CORP
5957 Engineering Drive
Huntington Beach, CA 92649
(714) 891-9922

For unusual motor installations contact Compumotor Application Engineering for assistance.

Panel Mounting

The Compumotor Plus has L shaped mounting brackets. They are notched with slots to accept screws on either end to facilitate mounting to flat panel surfaces. Use 10-32 screws with captured nuts to mount drive. If a fan is used, it may create vibrations, so locking the screws is advisable. Since the drive is difficult to manage with one hand, it is advisable to have a power interlock that prevents the drive from being removed with the power on.



Setting Motor Resolution

The Compumotor Plus control hardware functions with a natural resolution of 12,800 steps/rev (12,800 = 50 poles per revolution times 256 steps/pole). To provide the most flexible system possible, the Compumotor Plus software allows user to define any motor resolution from 200 steps/rev to 25,600 steps/rev. This is called the user resolution. However, motor resolutions above 12,800 steps per revolution do not improve the accuracy of the system.

User resolutions may be better suited for your application than the natural resolution by allowing you to program in units which make sense in your application. For example, it may be desirable to specify distance in microns. By selecting appropriate mechanical components and a suitable user resolution you will be able to specify moves in increments of microns. This does not mean that each motor step will be one micron, the accuracy of the system is unchanged. It does mean that you may calculate distance using units familiar to you.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Truncation Errors

The Compumotor Plus servo algorithm converts the commanded move distance (in terms of the user resolution) into the equivalent number of steps at 12,800 steps/rev. This is done by multiplying the number of commanded motor steps by 12,800 divided by the user-resolution.

$$distance_{natural\ steps} = \frac{distance_{user\ steps} \times 12{,}800}{user\ resolution}$$

The fractional part of this product is discarded. Thus, there is a truncation error of 1/12,800 of a revolution for user resolutions that are not sub-multiples of 12,800. A sub-multiple is any resolution which can be multiplied by a whole number to get 12,800 (e.g., 6,400; 3,200 steps/rev). This error is not cumulative, however. The truncation error is same for one revolution as it is for many revolutions. User resolutions that are sub-multiples of 12,800 do not have a truncation error.

Verifying Proper Installation

The first time you power-up the Compumotor Plus, it will send the following ready prompt from the RS-232C port:

```
*READY
>
```

To verify that the Compumotor Plus is communicating, power up the terminal before you power up the drive. Once the drive is powered, the ready prompt shown above is sent to your terminal. If your terminal displays garbled characters, check the terminal's protocol set-up. It is likely that the baud rate is not set at 9,600. If the baud rate is set at 9,600, and nothing appears on the terminal's screen, switch the transmit and receive wires and try again.

After you receive the ready prompt, enter some characters (they should appear on your screen). If each character appears twice, your terminal is set to half-duplex and should be reset to full-duplex.

Enter the following on your terminal's keyboard.

Type This	Compumotor Plus Response
<space>E<space>	E >

Note: <space> *represents the space bar.*

If the correct response does not appear, type the commands again. If you are still unsuccessful, refer to *Chapter 6, Maintenance and Troubleshooting*.

If characters are being echoed on the screen, but no prompts (>) are being displayed, your system may not be set in Interactive mode (**SSI0** command). Interactive mode is the factory default mode. The Interactive mode is intended for initial testing and set-up. This mode is not generally used with applications that are controlled by a host computer.

Bypassing Limit Switch Inputs for Test

The limit switches are set-up to function with normally closed contact switches. This is a fail-safe configuration. If a wire is disconnected, motion in that direction is not allowed by the system.

If you wish to test your motor without connecting the limit switches you can defeat them by issuing the Limit Disable (**LD3**) command. Compumotor ships the drive from the factory with the limits enabled. This means that the motor will not move unless either the limit switches are properly connected, or the **LD3** command has been issued.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Testing Limit Switch Operation

To test the CW limit function in the Compumotor Plus Indexer/Drive, ground the CW limit input with piece of wire and enter the following commands.

Example

Command	Description
> LD0	Enable CW and CCW Limits
> H+	Set direction CW
> MC	Set to Continuous mode
> A10	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> G	Execute the move (Go)

The **LD0** command enables the limits so the motor will not turn unless the CW limit input is grounded. The motor begins turning in the CW direction at a constant velocity of one revolution per second.

CW limit test

To test the CW limit function, open the CW limit (remove the CW Limit wire). The motor comes to an immediate halt.

To test the CCW limit function in the Compumotor Plus Indexer/Drive, ground the CCW limit input with piece of wire and enter the following commands.

Example

Command	Description
> MC	Set to Continuous mode
> A10	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> H-	Set direction to CCW
> G	Execute the move (Go)

CCW limit test

The motor begins turning in the CCW direction at constant velocity of one revolution per second. To test the CCW limit function, open the CCW limit (remove the CCW Limit wire). The motor comes to an immediate halt.

*Note: If the motor fails to stop issue a Stop (**S**) or Kill (**K**) command over the RS-232 port (Rx and Tx connections).*

Making the Motor Move

To test the CW rotation function of the Compumotor Plus, enter the following string of commands on the terminal's keyboard.

Example

Command	Description
> LD3	Enable end-of-travel limits
> A10	Set acceleration to 10 rps^2
> V10	Set velocity to 10 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)

The motor shaft turns two revolutions in the CW direction. To test the CCW rotation function of the Compumotor Plus enter the following string of commands:

Example

Command	Description
> A10	Set acceleration to 10 rps^2
> V10	Set velocity to 10 rps
> D-10000	Set distance to -10,000 steps
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The motor shaft turns two revolutions in the CCW direction. If the motor moves the wrong distance, issue the Configure Motor Resolution (**1CMR**)<space> command (assuming a device address of 1). The response should be 5,000 steps/rev. If it is some other value, you may reset it (refer to the **CMR** command listing in *Chapter 5, Software Reference*).

Homing The Motor

You can initiate the Go Home function by issuing the Go Home (**GH**) command. When you issue the Go Home command, you must include the direction and velocity that the motor should use to search for home. The home limit input on the Compumotor Plus is optically isolated, and is normally off. A normally open, load activated switch to ground is the most common way of signifying the home position.

When you command the Compumotor Plus to go home, it begins to move in the direction and at the velocity you specify with the **GH** command (e.g., **GH+2** causes a go home move in the CW direction at two revolutions per second).

It performs this move at the last defined acceleration rate, and looks for the home limit input to become active. If the motor encounters an end-of-travel limit while it searches for home, it will reverse direction and look for the home limit input to go active in the opposite direction. If motor encounters the other limit before it detects the home signal, the go home move is aborted and the motor stops.

To test the functionality of the home input switch, manually open the switch and type the following:

1IS<space>	*000_011_000_0 >

Assuming your end-of-travel limits are active, you should get back the response shown. Close the home switch and type the following:

1IS<space>	*000_111_000_0 >

This verifies that the limit switch is functioning properly. Refer to the **IS** command in *Chapter 5, Software Reference* for more information.

Example

To test the Compumotor Plus Drive's homing function you will need a short piece of wire to ground the **HOME** input. Then, enter the following commands:

Command	Description
> 1GA5	Set go home acceleration to 5 rps^2
> 1GH+1	Instructs the motor to go home CW at 1 rps

The following events occur as a result of this command.

① The motor moves in the positive direction at a constant velocity at one revolution per second.

② You must create a **HOME** input signal by grounding the HOME input.

③ The motor decelerates to a stop.

When the **HOME** input goes on after a **GH** command, the system recognizes the location where the input became active as home. The drive decelerates the motor at the last rate specified. The absolute position counter is automatically reset at the end of the Go Home move. More elaborate home sequences are available (refer to *Chapter 5, Software Reference*.)

Testing Continuous Mode Moves

The Mode Continuous (**MC**) command accelerates the motor to the velocity that you last specified with the Velocity (**V**) command. The motor continues to move at the specified velocity until you issue the Stop (**S**) command or specify a velocity change.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To change velocity while the motor is moving, use the **V** command followed by the **G** command. Continuous mode is useful for applications that require constant movement of the load, when the motor must stop after a period of time has elapsed (rather than after a fixed distance), or when the motor must be synchronized to external events such as trigger input signals.

Example

Command	Description
> PS	Pause
> MC	Set to Continuous mode
> A2	Set acceleration to 2 rps^2
> V.1	Set velocity to 0.1 rps
> G	Execute the move (Go)
> T1	Wait 1 second after the motor reaches constant velocity
> V.5	Set velocity to 0.5 rev/sec
> G	Change velocity (Go)
> T1	Wait 1 second after the motor reaches constant velocity
> VØ	Set velocity to 0 rps
> G	Change velocity to 0 (Stop)
> C	Continue

This series of commands sets the indexer to the Continuous mode. The motor is accelerated to 0.1 revolution per second, waits one second, changes velocity to 0.5 revolution per second, waits one second, and then stops. Note that the commands **VØ G** stops the motor (the **S** command is not a buffered command and cannot be used in this type of situation).

Final System Configuration

You should now have tested all of your interfaces and should be prepared to install the Compumotor Plus using your final configuration.

Inputs & Outputs

This section presents information about the Compumotor Plus Drive's inputs and outputs. The following inputs and outputs are addressed:

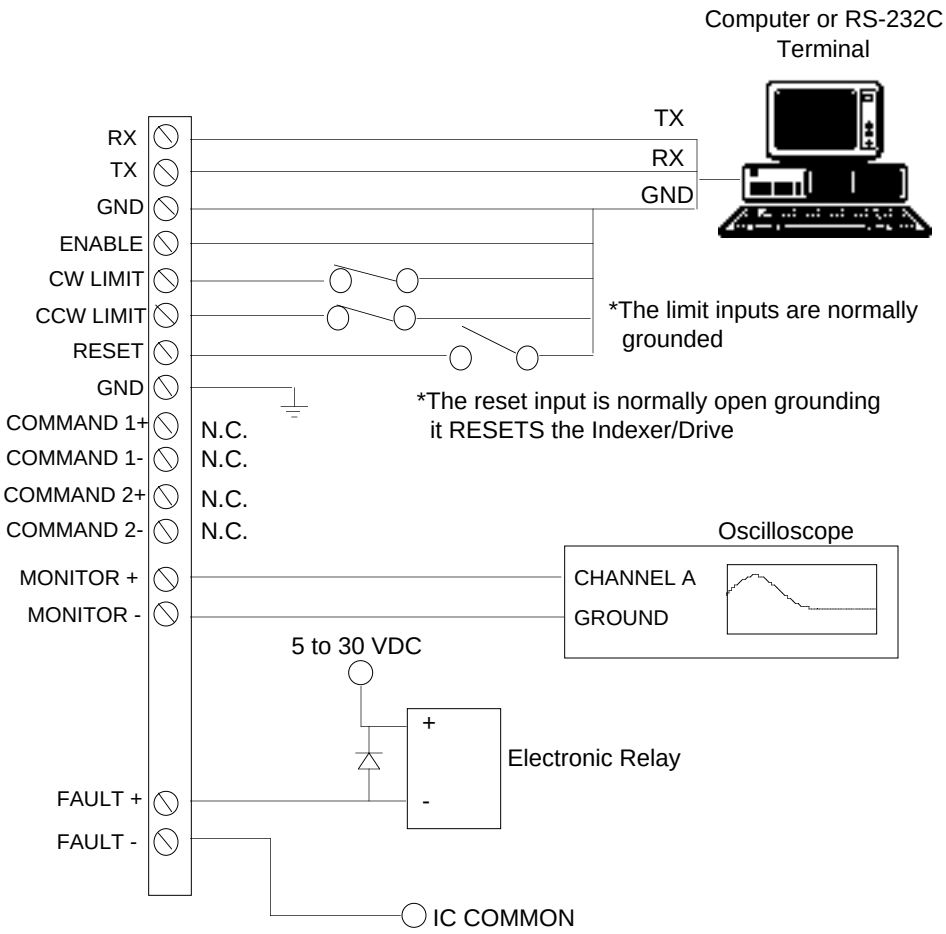
- ❏ Programmable Inputs
- ❏ Programmable Outputs and/or Faults

Input Functions	Output Functions
End-of-travel limits	Moving/Not moving
Home	Programmable outputs
Jog	Registration failed
Programmable trigger inputs	Registration occurred
Registration	Slip fault
Sequence select	
Start and stop	

Dedicated Inputs

End-of-Travel limits (CW & CCW limits) and the Home inputs are limited to those functions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Home Input

The home input is used to signal the indexer when the motor is at a home position. An external switch or signal is typically used to ground the input. The polarity of the home input is definable using the Define Active State of Home Switch (**OSC**) command.

Use the normally open switch if you want a grounded (low true) home. Use the normally closed switch if you want a high true (ungrounded) home. This input draws 12 mA through the internal 12VDC supply.

Command Inputs

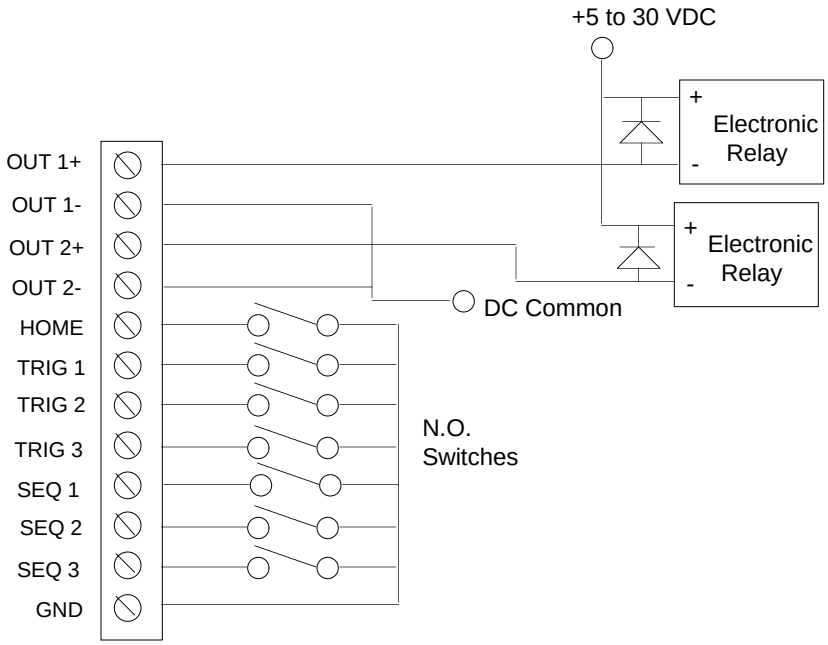
The command inputs (Command 1+, Command 1-, Command 2+, and Command 2-) are not functional on the Indexer version of Compumotor Plus.

Programmable Outputs

The programmable outputs require external power (5-30 VDC) to operate. When active, they conduct between the OUT+ and OUT- connections. The OUT+ terminal is the collector of an isolated transistor. The OUT- terminal is the emitter of the same transistor. Connect them as shown in the figure below.

These are open-collector outputs with the capability of supplying 60mA. (Refer to *Chapter 6, Hardware Reference* for electrical specifications).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Sequence Inputs

Six inputs may be user defined to provide different functions. Use the **IM** command to specify the different functions.

Sequence inputs are grounded (closed) to make them active. They may be left open if they are not used. This input draws 12 mA through the internal 12VDC supply.

Trigger Inputs

Trigger inputs are grounded (closed) to make them active. They may be left open if they are not used. This input draws 12 mA through the internal 12VDC supply.

Verifying Proper Trigger Input Wiring

Use the following steps to verify that you have wired the trigger inputs properly. You will need a short piece of wire to ground the trigger inputs.

Step ①

Check the state of the trigger inputs. Note: The 1TS command reports the state of the inputs only for the instant that the command is entered. Make sure you have a good connection between the input and ground before issuing the command.

Command	**Response**
1TS<space>	*000000 >

Step ② Ground TRIG 1 input to activate it.

Step ③ Check the state of the trigger inputs.

Command	**Response**
1TS<space>	***100000 >**

This verifies that TRIG 1 input is turned on.

Step ④ Repeat this process for each input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Verifying Proper Trigger Function

If the above test was successful you are ready to use the trigger inputs in a simple program You will require a short piece of wire to ground the TRIG1 input as in the above examples. To verify that the Trigger function is operating properly enter the following commands. Be sure to type a space or carriage return after each command.

Example

Command	Description
> PS	Pause
> LD3	Disable the limit function on trigger 3
> A2	Set acceleration to 2 rev/sec^2
> V.5	Set velocity to 0.5 rev/sec
> D10000	Set distance to 10,000 steps
> TR1XXXXX	Wait for Trigger Input 1 to turn on
> G	Execute the move (Go)
> C	Continue

The above example defines a move and then waits for a trigger input to occur. When you ground TRIG1 the motor makes a move of two revolutions.

*Note: The Compumotor Plus will wait forever for the trigger input, so if it does not respond to the trigger you will need to issue a STOP (**S**) command or a KILL (**K**) command to allow it to respond to further buffered commands.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ④

Application Design

The information in this chapter will enable you to:

- ❑ Recognize and understand important considerations that must be addressed before you implement your application
- ❑ Understand the capabilities of the system
- ❑ Customize the system to meet your requirements
- ❑ Use sample applications to help you develop your application

Defining Moves

The following section describes things you should consider when creating move profiles with the Compumotor Plus.

Application Considerations

This section describes some differences between theoretical and real-world performance in defining moves.

Position Accuracy Versus Repeatability

In positioning systems, some applications require high absolute accuracy. Others require repeatability. You should clearly define and distinguish these two concepts when you address the issue of system performance.

For many systems, the term accuracy is used when repeatability is required. When the motor always moves to the same distance from the same position, the primary positioning goal is not accuracy, but repeatability. Repeatability measures how accurately you can repeat moves to the same position. For example, a bottle labeling machine must rotate one revolution each time a label is applied. Since the motor always ends up in the same position and is always moving the same direction, repeatability is the dominant accuracy factor.

Accuracy on the other hand, is the error in finding a random position. For example, suppose the job is to measure the size of an object. The size of the object is determined by moving the positioning system to a point on the object and using the move distance required to get there as the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

measurement value. In this situation, basic system accuracy is important. The system accuracy must be better than the tolerance on the measurement that is desired.

Consult the technical data section of *The Compumotor Catalog* for more information on accuracy and repeatability.

Calculating Move Times

You can calculate the time it takes to complete a move by using the acceleration, velocity, and distance values that you define. However, you should not assume that this value is the actual move time.

There is calculation delay and motor settling time that make your move longer. After you issue the Go (**G**) command the indexer can take up to 100ms to calculate the move before the motor starts moving. You should also allow some time for the motor to settle into position. There is normally a delay of less than 30ms.

$$T_{Total} = T_{Calculation} + T_{Move} + T_{Settling}$$

You can virtually eliminate the calculation delay by using predefined moves (**GDEF**). This feature is available in Z3 and higher software revisions.

Positioning Modes

Incremental vs. Absolute Positioning

A preset move is a move with a distance that you specify (in motor steps). You can select preset moves by putting the Compumotor Plus into normal mode using the Mode Normal (**MN**) command. Preset moves allow you to position the motor in relation to the motor's previous stopped position (incremental moves) or in relation to a defined zero reference position (absolute moves). You can select incremental moves by using the Mode Position Incremental (**MPI**) command. You can select absolute moves using the Mode Position Absolute (**MPA**) command.

Incremental Moves (Preset Mode)

When you are in the Incremental mode (**MPI**), a preset move rotates the motor the specified distance from its starting position. For example, to move the Compumotor Plus motor 2 revolutions. You must specify a preset move with a distance of +10,000 steps, assuming a resolution of 5,000 steps per revolution. Every time the indexer executes this move, the motor moves 2 revolutions from its resting position. You can specify the direction of the move in one command. You specify the direction by using the optional sign (D+10,000 or D-10,000), or you can define it separately with the Set Direction (**H**) command (**H+** or **H-**).

Example

Command	Description
> MPI	Set to Incremental Position mode
> A2	Set acceleration to 2 rps^2
> V5	Set velocity to 5 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)
> G	Repeat the move (Go)
> H	Reverse direction of next move
> G	Execute the move (Go)

The motor moves 2 revolutions and stops. It then moves another 2 revolutions in the same direction and stops. The motor changes direction and moves 2 revolutions.

Absolute Moves (Preset Mode)

A preset move in the Absolute mode (**MPA**) moves the motor the distance that you specify (in motor steps) from the absolute zero position. You can set the absolute position to zero with the Position Zero (**PZ**) command, the Reset (**Z**) command, or by cycling the power to the drive. The absolute zero position is initially the power-up position.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The direction of an absolute preset move depends upon the motor position at the beginning of the move and the position you command it to move to. For example, if the motor is at absolute position +12,800, and you instruct the motor to move to position +5,000, the motor will move in the negative direction a distance of 7,800 steps to reach the absolute position of +5,000.

The Compumotor Plus powers up in Incremental mode. When you issue the Mode Position Absolute (**MPA**) command, it sets the mode to absolute. When you issue the Mode Position incremental (**MPI**) command the unit switches to Incremental mode. The Compumotor Plus Drive retains the absolute position, even while the unit is in the Incremental mode. You can use the Position Report (**PR**) command to read the absolute position.

Example

Command	Description
> MPA	Set to Absolute Position mode
> A2	Set acceleration to 2 rps
> V1Ø	Set velocity to 10 rps
> PZ	Set the current position to as home
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> D1ØØØØ	Set distance to 10,000 steps
> G	Move the motor to absolute position 10,000 (Go)
> DØ	Set the move distance to 0
> G	Execute the move (Go)
> MPI	Set indexer to Incremental Position mode

The motor moves 1 revolution and stops. It then moves another revolution and stops. The motor then moves in the opposite direction two revolutions.

Programming Highlights

This section contains information you will need when programming the Compumotor Plus.

Interactive Programming

You can operate the Compumotor Plus RS-232C interface in two modes based on the needs of your application. The two modes are *interactive* and *non-interactive*. In the interactive mode the Compumotor Plus returns a prompt (**>**) when it is ready for another command. Use the Enable Interactive Mode (**SSIØ**) command to make the Compumotor Plus interactive. You use the non-interactive mode when controlling the Compumotor Plus from a pre-programmed computer (i.e., not from terminal mode). You do this because it is easier for the computer to understand the Compumotor Plus without the prompt (**>**) character coming back after each command.

When you enable the interactive mode, the device address must be set to 1. The indexer responds with a **>** or a **?** when it receives a command. It responds with a **>** when the command has been successfully processed and a question mark (**?**) when it does not receive a valid command. If you enter a valid command, but enter an invalid range (e.g., **V70**), the Compumotor Plus responds with a **?**. These interactive responses are preceded with a carriage return and a line feed. You must enter and define entire loops and sequences before the system provides an interactive response.

Programmable Delays

You can use the Time (**T**) command to halt the operation of the indexer function for a preset time. If the Compumotor Plus is in Continuous mode, you may use the Time (**T**) command to run the motor at continuous velocity for a set time, then change to a different velocity.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In the Preset mode, the motor finishes the move before the indexer executes the time delay.

Example

Command	Description
> PS	Wait for the controller to receive a Continue (C) command before executing the next command
> G	Move motor 25,000 steps
> T5	Wait 5 seconds after the move ends
> H	Change motor direction
> G	Move motor 25,000 steps in the opposite direction
> C	Continue execution

Program Branching

This section discusses methods of changing the path of program execution through a conditional statement like **IF**, or through a fixed branch such as a loop or goto.

Looping

This section discusses methods of establishing loops in the program you write for your application. Loops are implemented with the Loop (**L**) command and the End Loop (**N**) command. Loops can be created individually or nested up to 16 levels deep.

The Loop (**L**) command repeats only buffered commands. Buffered commands are queued up and executed in order, so you can enter sequences of commands which are then executed one after the other. Place the commands to be executed between the **L** command and the **N** command. They are repeated the number of times indicated in the **L** command, for example, **L3** causes three loops to be executed.

You can use the Immediate Pause (**U**) command to pause execution of the loop while it is in progress. *The **U** command does not work in Continuous mode.* To resume loop execution, issue the Continue (**C**) command.

The example below shows a sample loop. In this example, the motor makes 2 moves with a half second delay between the moves.

Example

Command	Description
> L2	Loop twice
> G	Execute the move (Go)
> T.5	Wait 0.5 seconds
> N	End the loop

Loop

Nesting Loops

The example below shows how you can nest one loop inside another loop.

Example

Command	Description
> L	Loop indefinitely
> CR	Send a carriage return
> L2	Loop twice
> G	Execute the move (Go)
> T.5	Wait 0.5 seconds
> N	End the loop
> N	End the loop

Loop

Loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Branching with IF

You can perform conditional branching with the Compare Error Flag (**IFER**), Compare User Flag (**IFFL**), and Compare Trigger Status (**IFTR**) commands. All three of these commands are very similar to **IF_THEN** statement in *Basic* programming. If the condition evaluates true, the Compumotor Plus performs the commands that immediately follow the **IF** command. If the condition evaluates false, the Compumotor Plus skips all of the commands that follow the **IF** command until it reaches the End of **IF** Statement (**NIF**) command.

IF An Error Occurs

The **IFER** command checks to see if there is an error (such as slip fault). If there is an error, the commands immediately following the **IFER** are executed, otherwise the commands immediately following the End of **IF** statement are executed. For a detailed description of this command, refer to *Chapter 5, Software Reference*.

Example

Command	Description
> XE1Ø	Erase Sequence #10
> XD1Ø	Define Sequence #10
> IFER	If hardware error or limit encountered, then execute the following
> "SYSTEM_	Write to RS-232C port to inform user of system status
> "ERROR_	
> "OR_LIMIT_	
> "ENCOUNTERED_	
> ST1	Turns amplifier off
> ELSE	Otherwise
> "SYSTEM_	Writes to RS-232C Port
> "READY_	
> NIF	Ends IF statement
> XT	Ends sequence definition

IF User Flags

The Compare User Flag (**IFFL**), command compares the pattern set by the User Flag (**SFL**) command. If the patterns match the commands following the **IFFL** command are executed. If the patterns do not match the program continues after the next **NIF** command (End If).

This command is useful if you wish to make a decision based on previous program events that set or clear the user flag bits. For example, in an application with several sequences (or programs), at the end of each sequence, you can assign different bit patterns with the **SFL** command. If you select these sequences from the host computer, you may wish to make different moves depending on the sequence you ran. For a detailed description of this command, refer to *Chapter 5, Software Reference*.

Example

Command	Description
> PS	Wait for the controller to receive a Continue (C) command before executing the next command
> SFL1Ø1Ø	Set user flag bits 7 and 5 and clears bits 6 and 4, the remaining bits are not altered
> IFFL1Ø1Ø	If user flag bits 5 and 7 are set, and bits 6 and 4 are clear, perform the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End IF statement
> C	Continue execution

The **IFFL** pattern matches the SFL setting and the motor moves 25,000 steps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IF a Trigger Input

The **IFTR** command compares its pattern with the state of the programmable inputs defined as trigger inputs. If the patterns match the following commands are executed. It the patterns do not match, the commands following the next **NIF** command are executed. This command is useful for branching and performing conditional moves using the programmable inputs. For a detailed description of this command, refer to *Chapter 5, Software Reference*.

Example

Command	Description
> IFTRXXX1ØX	If sequence #1 is active and sequence #2 is not active, execute the following commands:
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End IF statement
> IFTRXXXØ1X	If sequence #1 is not active (open)and sequence #2 is active (closed), issue the following commands:
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps in the opposite direction
> G	Execute the move (Go)
> NIF	End IF statement
> IFTR1XX	If Trigger #1 is active, do the following command.
> 1"DONE	End message DONE
> NIF	End IF statement

Subroutines

When you use the Goto Sequence (**XG**) and the Execute a Sequence (**XR**) command sequences, you can execute different sequences from within a sequence.

These commands are similar to **GOTO** and **GOSUB** commands in *Basic* programming. If you use an **XG** command, the program will call or go to the sequence that you specified in the **XG** command. After executing the specified sequence, the system will not return to the original sequence. You could cause it to return to the original sequence by issuing another **XG** command, but there is a better way, the Run Sequence (**XR**) command. You use the **XR** command when you want to return to the calling (original) sequence upon completion of the called sequence when a Terminate Sequence (**XT**) command is encountered.

You can make as many as 16 calls with the **XR** command before returning to the calling sequence. There is no limit to the number of times you can use the **XG** command since the program need not return control to the original sequence.

Example

Command	Description
> XE2	Erase sequence #2
> XD2	Define sequence #2
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D2ØØØ	Set distance to 2,000 steps
> G	Execute the move (Go)
> XT	End sequence #2 definition
> XE3	Erase sequence #3
> XD3	Define sequence #3
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

> D-2ØØØ	Set distance to 2,000 steps (CCW)
> G	Execute the move (Go)
> XT	End sequence #3 definition
> XE1	Erase sequence #1
> XD1	Define sequence #1
> XR2	Execute sequence #2
> XR3	Execute sequence #3
> XT	End sequence #1 definition
> XR1	Execute sequence #1

In the previous example, when you execute sequence #1, the program moves to sequence #2. After executing sequence #2, the program returns to sequence #1. The program then moves to execute sequence #3.

Example

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> IFTR1	If TRIG 1 is active (Closed)
> XG2	Execute sequence #2
> NIF	End IF statement
> A1	Set acceleration to 1 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute move.
> XT	End sequence #1 definition
> XR1	Execute sequence #1

In the previous example, when you execute Sequence 1, the program checks the input pattern. If **TRIG1** is on, the program moves to execute Sequence 2. After executing sequence 2, program does not return to Sequence 1. If **TRIG1** is off, the program ignores the **XG2** command and makes the 25,000-step move.

Input/Output

The following section describes operation of the programmable inputs and outputs, and the move completion signals.

Programmable Outputs (POBs)

You can turn the programmable outputs (**OUT 1 - OUT 2**) on and off with the **O** command. Outputs **OUT 1** and **OUT 2** are factory set as programmable outputs. However, you can configure the outputs to perform different functions with the Output Mode (**OM**) command. Refer to the **OM** command in *Chapter 5, Software Reference* for descriptions of the available functions. You can use these outputs to turn on and off other devices (i.e., lights, switches, etc.).

Example

Command	Description
> PS	Pause command execution until the controller receives a **C**
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> OM1	Set OUT1 and OUT2 as programmable outputs
> O1Ø	Turns OUT1 on and OUT2 off
> G	Execute the move
> OØ1	Turn OUT1 off and OUT1 on
> C	Continue execution

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This example above defines **OUT1** and **OUT2** as programmable outputs. **OUT1** turns on and **OUT2** is turned off, the motor moves 25,000 steps. When the motor stops and **OUT1** turns off and **OUT2** turns on.

Event Completion Signals

You can program the Compumotor Plus to notify you when a move or other event is complete. *Note: You may only signal the completion of buffered events.*

- ❑ LF Line feed
- ❑ CR Carriage return
- ❑ O Output command
- ❑ " Quote command (literal string)

Example

	Command	Description
	> A2	Set acceleration to 2 rps^2
	> V.5	Set velocity to 0.5 rps
	> D125ØØ	Set distance to 12,500 steps
	> G	Execute the move (Go)
Line feed⇨	**> 1LF**	Send a line feed over the RS-232C interface

The motor moves 12,500 steps. When the move is complete, the Compumotor Plus sends a line feed to the host over the RS-232C interface.

Example

	Command	Description
	> A2	Set acceleration to 2 rps^2
	> V.5	Set velocity to 0.5 rps
	> D125ØØ	Set distance to 12,500 steps
	> G	Execute the move (Go)
Carriage return⇨	**> 1CR**	Send a carriage return

The motor moves 12,500 steps. When the move is complete, the Compumotor Plus sends a carriage to the host over the RS-232C interface.

Example

	Command	Description
	> A2	Set acceleration to 2 rps^2
	> V.5	Set velocity to 0.5 rps
	> D125ØØ	Set distance to 12,500 steps
	> G	Execute the move (Go)
Output⇨	**> O1**	Turn on Output #1

The motor moves 12,500 steps. When the move is complete, Output 1 is turned on.

Example

	Command	Description
	> A2	Set acceleration to 2 rps^2
	> V.5	Set velocity to 0.5 rps
	> D125ØØ	Set distance to 12,500 steps
	> G	Execute the move (Go)
Message⇨	**> 1"DONE**	Set the **DONE** message

The motor moves 12,500 steps. When the Compumotor Plus completes the move, the unit issues the **DONE** message from the Compumotor Plus to the host over the RS-232C interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Remote Jogging

In some applications, you may want to adjust the motor position by toggling a switch on and off. The motor moves when the switch is on and stops when the switch is off. This is called jogging the motor. You can configure the Compumotor Plus for jogging operation by doing the following.

- ❏ Define programmable inputs as jog inputs using the Input Mode (**IM**) command
- ❏ Define the jogging velocity with the Jog Velocity (**JV**) command
- ❏ Enable jogging with the **OSE1** command
- ❏ Attach a switch to the jog inputs. Use a two pole switch for jogging both directions
- ❏ Turn the switch on to jog the motor

The following example shows how you can define power-up sequence #40 to set up jogging.

Step ① Define a power up sequence:

Command	Description
> XE4Ø	Erase sequence #40
> XD4Ø	Define sequence #40
JA2	Set jog acceleration to 2 rps^2
OSE1	Enables jog function
JV5	Set jog velocity to 5 rps
IM3	Define TRIG 2 and TRIG 3 as jog CW and CCW lines
XT	End sequence definition
> SV	Saves sequence definitions into EEPROM
> Z	Resets the Compumotor Plus controller
>	

Step ② Turn on TRIG 2 input to move the motor in the CW direction at 5 rps (until you turn off TRIG 2).

Step ③ Turn on TRIG 3 input to move the motor in the CCW direction at 5 rps (until you turn off TRIG3).

Defining and Using Programs (Sequences)

Use the following commands to define, erase, and run programs. *In the Compumotor Plus language programs are referred to as sequences of commands or sequences.* Refer to *Chapter 5, Software Reference,* for detailed descriptions and syntax of the following commands.

Pertinent Commands

Command	Description
> XD	Start sequence definition
> XE	Delete sequence EEPROM
> XQ	Set/reset interrupted Run mode
> XRP	Run sequence with a pause
> XT	End sequence definition
> XU	Upload sequence
> XR	Run sequence
> SSJ1	Run a sequence defined by BCD sequence inputs
> XG	Exit current sequence and move to execute another sequence

A sequence is a series of commands. The commands are executed in order whenever the sequence is run. Only buffered commands may be used in a sequence. Immediate commands cannot be stored in a sequence, just as they cannot be stored in the command buffer.

The Compumotor Plus has 1,500 bytes of non-volatile memory to store 40 sequences. The sequences may have different lengths, so you may have one

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

long sequence or several short ones, as long as the total length does not exceed 1,500 characters.

To define a sequence do the following:

- ❑ Enter the Define Sequence (**XD**) command immediately followed by sequence identifier number (1 to 40) and a delimiter. For example, **XD1**.
- ❑ Enter the commands you want to execute when the sequence is run.
- ❑ Enter the Terminate Sequence (**XT**) command to end the sequence definition.
- ❑ Enter the **SAVE** command to save the sequence.

All commands that you enter after the **XD** command and before the **XT** command are executed when the sequence is run. An example is provided below.

Example

Command	Description
> **XE1**	Erase sequence #1
> **XD1**	Begin definition of sequence #1
A2	Set acceleration to 2 rps^2
V1Ø	Set velocity to 10 rps
D5ØØØ	Set distance to 5000 steps
G	Execute the move (Go)
H	Reverse direction
G	Execute the move (Go)
XT	End definition of sequence
> **XR1**	Runs sequence #1

Sequence #40 is reserved for power up execution.

You can run a sequence by entering the **XR** command and sequence identifier number (in the range of 1 to 39) and a delimiter.

You can also run the sequences by specifying the sequence number in BCD on the programmable inputs. To accomplish this do the following.

- ❑ Define some programmable inputs as sequence-select inputs using the Input Mode (**IM**) command
- ❑ Enable the Continuous Sequence scan mode (**SSJ1**), you can also execute a sequence by turning on Sequence inputs to indicate which sequence you wish to run

Once you define a sequence, it cannot be redefined until you delete it. You can delete a sequence by entering the **XE** command immediately followed by a sequence identifier (1 to 40) and a delimiter. You may then redefine that sequence.

Sequences that you define are not saved into the non-volatile memory, until a Save (**SAVE** or **SV**) command is issued.

Sequence Selection Methods

After you define the sequences from the RS-232C interface, you can execute the sequences by using one of the following methods.

- ❑ Standalone — Use thumbwheel switches to select and run the sequence
- ❑ Computer Interface — Use the Execute Sequence (**XR**) command to run the sequences
- ❑ PLC — Use the sequence select inputs to run a sequence

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Standalone Operation

This section explains and provides examples of how to store programs and run them with remote switches, and run them automatically when you power up the system. First, you will need to enter the programs into the Compumotor Plus. You will need a terminal or a computer with RS-232C communication capabilities for programming the Compumotor Plus controller.

Power-Up Sequence execution

Sequence #40 is always run on power up and after a Reset (**Z**) command. To run another sequence on power up, put an **XR** or **XG** at the end of sequence #40. If sequence #40 is empty, no sequence is run on power up. Refer to *Chapter 5, Software Reference*, for detailed descriptions and syntax of the following commands.

Example

Command	Description
> XE4Ø	Erase sequence #40
> XD4Ø	Begin definition of sequence #40
LD3	Disable limits if they are not connected
A2	Set acceleration to 2 rps^2
V5	Set velocity to 5 rps
D125ØØ	Set distance to 12,500 steps
G	Execute the move (Go)
XT	End sequence definition
> Z	Reset the controller and runs sequence #40

A power-up sequence is typically used to store set-up or initialization parameters that your application requires. Some of these commands are listed below.

- ❑ SSJ1 Continuous Sequence Scan mode
- ❑ SN Scan time
- ❑ JA Jog acceleration
- ❑ JVL Jog velocity low
- ❑ JVH Jog velocity high

You can put any buffered commands into Sequence #40 to have them executed during power-up. Immediate commands cannot be put into sequences.

Remote Sequence Execution

You can execute sequences remotely using a switch selection by doing the following.

Step ①

Define a power up sequence

Example

Command	Description
> XE4Ø	Erase sequence #40
> XD4Ø	Define sequence #40
JA2	Set jog acceleration to 2 rps^2
OSE1	Enables jog function
JV5	Set jog velocity to 5 rps
IM3	Define TRIG 2 and TRIG 3 as jog CW and CCW lines
XT	End sequence definition
>	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step ②

Define all sequences that your application may require. The following example defines sequence #1.

Command	Description
> **XE1**	Erase sequence #1
> **XD1**	Define sequence #1
A1	Set acceleration to 1 rps^2
V2	Set velocity to 2 rps
D1ØØØ	Set distance to 1,000 steps CW
G	Execute the move (Go)
XT	End sequence definition
>	

The following example defines sequence #2.

> **XE2**	Erase sequence #2
> **XD2**	Define sequence #2
A1	Set acceleration to 1 rps^2
V2	Set velocity to 2 rps
D-1ØØØ	Set distance to 1,000 steps CCW
G	Execute the move (Go)
XT	End sequence definition
>	

Step ③

Enter the **XR1** command to move the motor 1,000 steps CW.

Step ④

Enter the **XR2** command to move the motor 1,000 steps CCW.

PLC Operation

You can use a PLC to execute 39 of the 40 sequences that are stored in the Compumotor Plus controller (the 40th is reserved for power-up execution). You can configure up to eight inputs as sequence-select inputs. You can accomplish this by changing the BCD values of the PLC outputs

Scanning for Sequence Execution

Changing the BCD values of sequence input lines results in a new sequence being run that corresponds to the new value. The sum of the values for ach input determines which sequence the indexer will run. For example, if you set up all six inputs as sequence-select inputs, the lowest input (SEQ 3) will have the least significant value and the highest input (TRIG 1) will have the most significant value. Refer to the table below.

Do the following to set the Compumotor Plus inputs as sequence select lines.

- ❏ Enter the Input Mode (**IM**) command to set up the inputs as sequence-select inputs.
- ❏ Enter the **SSJ1** command to configure the Compumotor Plus controller to read sequences via BCD input.
- ❏ Optionally enter the **XQ1** command to configure the Compumotor Plus controller to use the Interrupted Run mode. See below for a detailed explanation.
- ❏ Optionally, enter the Scan (**SN**) command to tell the Compumotor Plus how long the inputs must be stable before accepting the input as valid. See below, or *Software Reference*, for a detailed explanation.

Once you issue the **SSJ1** command, the Compumotor Plus controller scans the sequence select inputs to find the sequence that you have specified for execution. If it finds a valid sequence number on the inputs (any number other than zero) it runs the sequence. The Compumotor Plus completes the execution of the desired sequence and the process begins again.

The Interrupted Run Mode (**XQ1**) command makes the Compumotor Plus wait until all of the sequence inputs are turned off (sequence zero) before selecting the next sequence to execute.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Scan (**SN**) command determines how long the sequence-select input must be maintained before the controller executes the program. This is also called the debounce time.

Sample PLC Applications and Commands

This section provides step-by-step procedures to run sequences from your PLC. First, you need to enter the programs into the Compumotor Plus. You will need a terminal or a computer with RS-232C communication capability. You need to define the sequences before you can execute them with your PLC's BCD outputs. The Compumotor Plus controller saves these sequences with the Save (**SV**) command.

Step ①

Define a power-up sequence

Command	Description
> XE4Ø	Erase sequence #40
> XD4Ø	Define sequence #40
> SSJ1	Execute sequences via PLC input
> SN2Ø	Set scan time to 20 ms
> XQ1	Set to Interrupted Run mode
> A1Ø	Set acceleration to 10 rps^2
> V2	Set velocity to 2 rps
> IM3	Define all six TRIG/SEQ inputs as sequence-select lines
> LD3	Disable the limits (if they are not connected)
> XT	End the sequence definition

Every time you power up the Compumotor Plus controller, it executes these commands and enables the Compumotor Plus to read up to 39 sequences from the sequence select inputs.

Step ②

Define any sequences that your application may require.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> D2ØØØ	Set distance to 2,000 steps (CW)
> G	Execute the move (Go)
> XT	End sequence #1 definition
> XE2	Erases sequence #2
> XD2	Defines sequence #2
> D4ØØØ	Set distance to 4,000 steps (CW)
> G	Execute the move (Go)
> XT	End sequence #2 definition
> XE3	Erase sequence #3
> XD3	Define sequence #3
> D8ØØØ	Set distance to 8,000 steps (CW)
> G	Execute the move (Go)
> XT	End sequence #1 definition
> XE39	Erase sequence #39
> XD39	Define sequence #39
> D-14ØØØ	Set distance to -14,000 steps (CCW)
> G	Execute the move (Go)
> XT	End sequence #39 definition

Step ③

Verify that your programs were stored properly by uploading each entered sequence (**XU**). If you receive responses that differ from what you programmed, re-enter those sequences.

Step ④

Save the sequences entered by typing the Save (**SV**) command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step ⑤ Run each program from the RS-232C interface with the Run Sequence (**XR**) command. Make sure that the motor moves the distance that you specify.

Step ⑥ Assuming your PLC accepts open collector outputs connect the inputs and outputs as shown in the following table. If not, you will need to add pull up resistors to the outputs.

PLC	Compumotor Plus
Output 6	Trig 6
Output 5	Trig 5
Output 4	Trig 4
Output 3	Trig 3
Output 2	Trig 2
Output 1	Trig 1
Ground	I/O Ground

Step ⑦ Refer to the user guide that accompanied your PLC unit to turn on the proper combination of outputs to execute one of the four sequences programmed and stored in the Compumotor Plus controller.

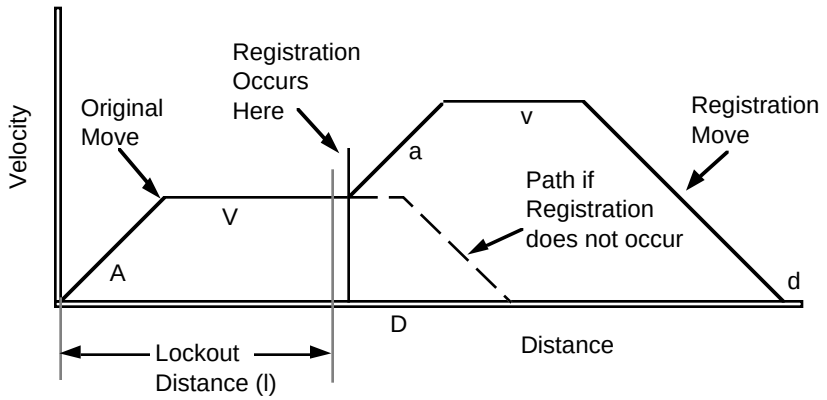
- ❏ Turning on only the PLC's Output 1 will execute sequence #1
- ❏ Turning on only the PLC's Output 2 will execute sequence #2
- ❏ Turning on only the PLC's Outputs 1 and 2 will execute sequence #3
- ❏ Turning on only the PLC's Outputs 1, 4, 5, and 6 will execute sequence #39

Step ⑧ Cycle power to the Compumotor Plus. The system will execute Sequence #40.

Step ⑨ Turn on the appropriate sequence-select input [set for the least Scan Time(**SN**)] to execute the proper sequences. Since the sequence feature (refer to the **XQ1** command) was enabled during the power-up sequence, your PLC program must turn off all of the sequence-select inputs before you can select another sequence.

Registration and Synchronization

The registration function allows you to make *a move within a move* based on an external event. This is done by responding to an external trigger input, which causes the existing move to blend into a new move (the registration move). The registration move causes the motor to accelerate or decelerate as required to get to the specified end point using the parameters defined for the registration move.



☛ Hint
Registration is used to synchronize materials which slip or change size when processed.

The typical use of registration moves is to synchronize web applications. For example, a sheet of pre-printed cardboard is to be cut up into paper plates. Registration moves synchronize the cutter with the blank space between the plates. This is done by using a sensor which picks up something unique about the web, such as a registration mark (hence the name registration moves). The web is then moved to a point a fixed distance from the registration mark and an operation such as a cut is made.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To specify the move, the acceleration, velocity, and distance of the move as well as the lockout distance are required. The lockout feature is designed to eliminate false triggering. The lockout distance is the distance to move before looking for the registration mark. Refer to the above figure. The lockout distance is specified from the beginning of the current move, the move that is aborted when the registration mark is detected. Lockout is required to prevent the registration sensor from triggering falsely on a pattern printed on the web.

To define the registration move do the following.

- ❑ Issue the registration command: REG*a, v, d, l* where *a* is the acceleration, *v* is the velocity, *d* is the distance, and *l* is the lockout distance of the registration move. This defines the registration move.
- ❑ Issue the **SSK1** command. This enables the registration feature.

Once the registration move has been defined and enabled, the Compumotor Plus begins looking for the registration mark during any move except jog and homing moves. This condition is referred to as being in registration mode.

If the parameters specified are impossible to execute the registration command is ignored. If the lockout distance is longer than the currently defined move the registration move is never executed. Refer to *Chapter 5, Software Reference* for information about how to check the move to see if it is impossible to execute.

If the currently defined move is a continuous move the registration feature operates normally, but *should registration never occur the motor will continue on just as it would if it were not in registration mode.* In contrast, when registration is used with preset moves, the move ends at the preset distance defined by the distance (**D**) command if a registration trigger never occurs. This is useful for indicating an error condition to the operator. It can also keep material waste to a minimum.

Input modes (**IM4** or **IM5**) define an input for use as the registration input. Additionally, two other inputs, may be defined as jog inputs. The jog inputs are used to position the load for initial setup (refer to *Software Reference*).

The Compumotor Plus records the motor's position as soon as the registration input is received. You should not use the **SN** command to debounce the registration input. The debounce process introduces a variable delay between receiving the registration signal and recording the motor position, reducing accuracy.

Example

Command	Description
> SSK1	Enable registration
> IM4	Set Input mode #4 as registration input
> A10	Set normal acceleration to 10 rps^2
> V1	Set normal velocity to 1 rps
> D50000	Set normal distance to 50000
> MN	Set to mode normal
> REG50,20,5000,40000	Set registration move to accel=50, vel = 20, distance from registration mark=5,000, wait until 40m000 steps from start of move to start looking for registration mark
***SETTING UP REG TABLE** ***REG TABLE COMPLETE**	
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

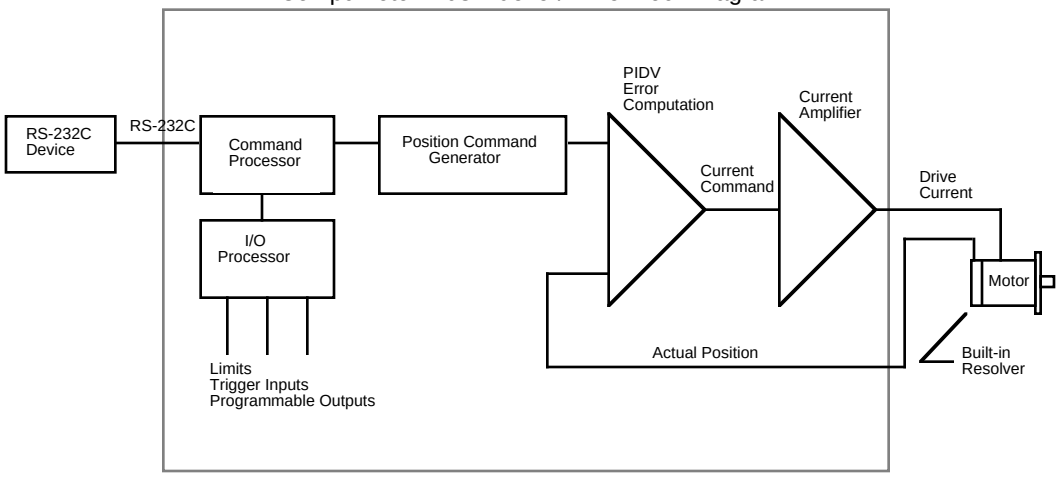
The Compumotor Plus is a servo system requiring a closed feedback loop to control the motor and thus the load. The Compumotor Plus employs a sophisticated algorithm to control the response of the motor and its load. This system is tunable: you can adjust its behavior to suit your needs. You tune the system by adjusting gains. The gains are adjusted by pushing buttons on the Compumotor Plus front panel or via commands over RS-232C. Below is a description of how the system works, how to use it and what to look out for.

Tuning Theory

The Compumotor Plus Drive can be divided into two major areas: the digital controller and the analog amplifier. All of the positioning compensation (Proportional, Integral and Derivative gains) and the velocity gains are set by the user and processed by the digital controller board. Once the values are set, they can be stored or saved in the EEPROM by issuing a Save (**SV**) command. The Compumotor Plus is preprogrammed at the factory with gains appropriate for the size of motor provided. Refer to the block diagram of the Compumotor Plus Drive servo system shown below.

The controller board sends two digitized waveforms from its digital-to-analog converters (DACs) to the analog amplifier board. These waveforms represent two commanded, motor-phase currents. The analog amplifier board measures the actual motor current to determine the correct voltage to apply to the motor windings. The controller commands a desired current to the amplifier board, the amplifier board then attempts to generate that desired current in the motor's windings. The position of the motor's shaft is sensed by the controller via the resolver attached to the motor. The controller uses this position information to generate the desired current command to the amplifiers.

Compumotor Plus Indexer/Drive Block Diagram



The current command to the amplifier is based on several quantities including the position of the servo motor's shaft, the desired position (which is generated by the indexer), and previous position errors.

The controller subtracts the motor's actual position from this desired position to determine the position error. The position error is put into an equation, along with previous position errors, to generate the current command for the amplifier.

In the Compumotor Plus, the equation is a digital approximation of an analog continuous-time PID (proportional, integral, and derivative) and velocity control network. The analog continuous-time PID is traditionally used to stabilize conventional servo systems. It consists of several

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

potentiometers, resistors, and capacitors. In the Compumotor Plus, these parts are replaced with digital logic controlled by a microprocessor.

The digital approximation referred to above is the discrete-time equivalent to a continuous-time PID network. It is called a discrete-time PID network because it operates on sampled digital data rather than on continuous analog data. The sampling rate of the Compumotor Plus controller is the rate at which the equation is evaluated and the current command to the amplifier is updated. The sample rate of the Compumotor Plus controller is 3,333 times per second, or every 300ms. This rate is sufficiently fast to provide excellent dynamic response.

Actual motor position information is provided by a resolver built into the motor. The resolver is read three times every update period. The desired position is compared to the actual position and an error or correction value is generated.

The velocity command is calculated using the following formula.

$$V_c = P_g d_{e_n} + I_g \sum (d_{e_1 \cdots e_n}) + D_g (d_{e_n} - d_{e_{n-1}})$$

where,

V_C is the commanded velocity
d_e is the position error
P_g is the proportional gain
I_g is the integral gain
D_g is the derivitive gain

General Tuning Considerations

The gains set at the factory are satisfactory for most applications, but if your application requires higher performance, you can change these gains.

If you adjust the gains and the system appears unstable, use the Return to Factory settings (**RFS**) command or pushbutton combination to return to the factory settings (pushbuttons P and I together).

Proportional, Integral, Derivative, and Velocity tuning (P, I, D, and V) can be implemented through RS-232C commands or through the front panel pushbuttons. The factory values of the tuning algorithm are based on the following.

- ❏ The ratio of load-to-rotor inertia is 10 to 1 or less
- ❏ The load is purely inertial (i.e., there is little or no friction impeding the load)
- ❏ It is directly and solidly coupled to the motor
- ❏ A 4% velocity overshoot is tolerable

If you find that your system cannot be tuned satisfactorily using the pushbuttons on the front panel, the load to rotor inertia ratio may be higher than the default factory gain **maximums** are capable of handling. This may also be the case if there is a significant amount of friction in the system, or the coupling to the load is torsionally flexible.

Tuning usually involves choosing between a high degree of stability accuracy, or responsiveness. A stable system wants to remain at rest and is therefore not responsive. It will maintain a small, repeatable, steady state error. An accurate system which is critically damped at the end of its moves will be somewhat stable, but the cycle time for the move will be somewhat higher than the best the system can do. Finally, a very responsive system will turn in very fast cycle times, but overshoot and ringing will be significant

You can increase the speed of your machine at the price of motor/drive heating, final position accuracy, and settling time. Or you can improve system accuracy at the price of final position stability. In other words, you can trade final position accuracy for system response time. All of the gains are interactive. It may take considerable experimentation to find the exact

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

combination of gain values required to get the best performance in your application.

As a general rule the velocity gain should be set first, followed by proportional, integral, and finally the derivative gain. The velocity gain affects the responsiveness of the system as a whole. The proportional gain affects stiffness. The derivative gain affects settling time and dynamic response. The integral gain affects the final position accuracy. All gains interact with each other.

Gain Descriptions

The following section describes the four gains found in the Compumotor Plus: proportional, integral, velocity and derivitive. The first letters of each gain type makes up the term PIDV which describes the tuning method used in the Compumotor Plus.

Velocity Gain

The velocity gain affects the overall responsiveness of the system. If the system is sluggish, increase the velocity gain. If the system overshoots unacceptably, or the motor rings at the end of each move, reduce the velocity gain. You can increase the derivative gain to compensate for post-move ringing as well.

Proportional Gain

Proportional gain affects system stiffness and accuracy. As the proportional gain increases, the influence of the feedback signal increases. If the gain is too high, the system oscillates. This happens because very small resolver changes are amplified into very large error signals. Eventually the motor begins to lag behind the feedback signal, causing the feedback and the command signals to be in phase. This is what causes oscillation.

Integral Gain

Integral gain allows the system to compensate for steady-state position errors (due, for example, to friction). Integral gain also reduces velocity ripple. It does this by slowing down the electronic response time so that it more closely resembles the response of the mechanical components of the system.

Derivative Gain

Derivative gain adds damping to the system. This damping helps reduce the oscillations at the end of moves, and ringing when the velocity is suddenly changed (at the end or beginning of an acceleration ramp). When this happens, increasing the derivative gain will reduce the oscillation. The derivative gain adds phase lead to compensate for the system's natural phase lag.

Tuning Your System Manually

The Compumotor Plus uses a conventional discrete-time PIDV servo algorithm. The algorithm is a recursive function using four user adjustable gains to control system responsiveness and accuracy. The four gains model static and dynamic characteristics of the motion control system and its load. The gains are adjusted to stabilize the system.

The four gains in a Compumotor Plus are normalized to a range of 0 to 100. This makes tuning the system much simpler due to the small numbers employed. To ensure that you have sufficient range in the gain settings a gain maximum is provided for each gain which allows you to change the meaning of the gain number from 0 to 100. Refer to the **CPM**, **CIM**, **CVM**, and **CDM** commands in *Chapter 5, Software Reference* for more information.

Tuning a Compumotor Plus Drive servo system usually requires adjusting only one controller gain (normally velocity). The other gains are predefined and, in most cases, require no further adjustment.

If you find that you are tuning a given parameter in the extreme range (i.e., 0-5% or 95-99%), you might consider changing the maximum gain value via the **CIM**, **CPM**, **CVM**, and **CDM** commands.

Once you have the system installed and the motor connected to its intended load, you can determine whether any fine tuning is required by observing the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

response of the system to commands from your indexer and by observing how stiff the system is when at rest.

With the motor at rest, try to deflect the shaft. You should not be able to easily turn the shaft away from its rest position. If it feels very soft, the system gains probably need to be increased. A very soft system does not respond very quickly to move commands.

If the shaft feels stiff, check that the system is not vibrating. If it is, the gain may be too high. This causes the drive to provide excess current, and can shorten the life of mechanical components. In extreme cases the vibration grows, producing violent motions that cause the drive to fault or possibly break the equipment it is attached to. This is called instability. For this reason, you should tune the Compumotor Plus Drive with some caution.

Pushbutton Tuning

There are two methods available to adjust an Compumotor Plus Drive's servo compensation network.

- The six pushbuttons on the Compumotor Plus Drive's front panel
- The RS-232C communications port

The buttons are labeled UP, DOWN, P, I, V, and D and are defined as follows. Refer to *Chapter 3, Installation* for more information on pushbuttons.

- P selects the PROPORTIONAL gain
- I selects the INTEGRAL gain
- V selects the VELOCITY gain
- D selects the DERIVATIVE gain

Select one of the P, I, V, or D buttons and hold it down. When you do, the two-digit display displays the present value for the gain you have selected (from 0-99%). You should note this value in case you wish to return to it. While holding the button for the gain you selected, push the UP button to increase the gain. Push the DOWN button to decrease the gain. Continue to hold the UP or DOWN button and the proportional gain changes automatically.

☛ Hint
Don't forget to ***SAVE****!*

The value displayed when you release the **UP** or **DOWN** button becomes the new gain setting. This value is used by the Compumotor Plus immediately. However, you must issue a **SAVE** or **SV** command to make the change permanent. Because the gains must be manually saved you may return to your previous settings by resetting the drive, by cycling power, or issuing the **Z** command. This recalls the last setting in non-volatile memory.

If you wish to retrieve the factory default values, all you need to do is push both the P and I gain buttons together. This causes the factory values for each gain to be reloaded.

After you have completed the pushbutton tuning procedure, it will be necessary to save the values you have selected into non-volatile EEPROM. This is done by pressing all four gain buttons (P, I, D, & V).

Tuning Via RS-232

The tuning process of the front-panel switches can be duplicated via the RS-232C communication port.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You use the following commands to set the gains.

- CPG sets the proportional gain
- CIG sets the integral gain
- CDG sets the derivative gain
- CVG sets the velocity gain

To set the gain maximums use the following commands.

- CPM sets the proportional gain maximum
- CIM sets the integral gain maximum
- CDM sets the derivative gain maximum
- CVM sets the velocity gain maximum

Tuning Your System Automatically

With the Z5 revision level software (see the **RV** command), your Compumotor Plus Drive/Indexer is equipped with the ability to tune itself. Self-tuning refers to the procedure by which the Compumotor Plus operating system automatically determines the proper servo gains for your application.

You use the **TUNE** command to implement self-tuning. Once self-tuning is complete you have two sets of gains you can use: PIDV and Pole-placed. You use the **GAINX** command to select between the two sets of gains.

The gains computed with the **TUNE** command are high performance gains. This means that they are fairly high for typical loads. Consequently, this implementation of self-tuning is intended for loads below ten times the rotor inertia. When large loads are used with this system, the gains computed are typically too large to be of use. You may be able to use option four to produce stable gains for large loads. See below for a description of the **TUNE** options.

If you need to drive loads in the 10:1 rotor inertia range and above, you should use PIDV tuning. If you find that the resulting gains produce excessive noise in the motor, you may want to filter out this noise with the **FILT** command. The **FILT** command sets the digital torque filter's time constant. Since larger loads typically result in lower system bandwidth, you may be able to increase the digital torque filter's time constant from the default value of 5 ms to a value as high as 10 ms.

The algorithm used in the Compumotor Plus is referred to as a self-tuning regulator, and it consists of two main stages. Stage one is system identification. When the **TUNE** command is invoked, the Compumotor Plus executes a brief predefined move sequence. This sequence consists of a series of short step moves. Step moves are moves with infinitely high acceleration and deceleration. The move sequence involves both the motor and your application load. Since some applications have restrictions on how they can move, the **TUNE** command has three move options described below.

- Move one (default) consists of back-and-forth, 512 step moves (at 12,800 steps/rev)
- Move two consists of CW 512 step moves
- Move three consists of CCW 512 step moves

During the move sequence, the Compumotor Plus records the value of current sent to the motor, and the position of the motor for 1500 time intervals. This data is then used to estimate the performance characteristics of the motor and load. In particular, the system determines the total load of the motor and the application.

☛ **Hint** *Secure the motor to the load*

For system identification to be as accurate as possible, it is important that you secure the motor to its mount and firmly couple the shaft to the load. If the motor body moves around or the coupler slips during the preset move, the resulting data will be flawed and the system will not be able to compute the proper gains.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

After the data has been acquired, the operating system shuts the amplifier off and analyzes the data. The Compumotor Plus issues the following message.

```
*TUNING_BEGUN
>
```

Because of the large amount of data being processed, this analysis takes approximately two minutes to complete. At the end of this time, the system uses the estimated total system load to compute the appropriate gains. It then issues the following message,

```
*TUNING_COMPLETE
>
```

The Four Options

In order to compute the proper gains for an application, there has to be some desired final closed-loop performance specified. Different sets of gains result in different final performances. To make this choice easier for you, the **TUNE** command provides four optional final motor responses as describe below.

- ❏ Option one specifies responses with very low overshoot but limited in position stiffness.
- ❏ Option two specifies responses with some overshoot on fast accelerations but greater in position stiffness.
- ❏ Option three specifies responses with high in-position stiffness and greater overshoot on fast accelerations than options one and two.
- ❏ Option four specifies responses using a lower set of gains for very large loads.

Tuning a servo system always involves trade-offs in selecting desired performance characteristics. These four options offer you flexibility in tuning your Compumotor Plus.

After the self-tuning command has calculated the new gains, the Compumotor Plus re-energizes the amplifier. The system then waits approximately five seconds with the new gains in place to confirm that the system is stable.

Do not touch the rotor at this time because this will be interpreted as instability in the system. When the calculated gains are found to be unstable, the system will send the message **TUNING_COMPLETE** to the terminal and return the system to the PIDV control scheme.

The **TUNE** command is designed to catch all instances of unstable gains and default to PIDV when they occur. Nonetheless, it is a good idea to make sure that the maximum following error is set to a reasonable value (e.g. 1 rev) with the **CPE** command before self-tuning. This causes the Compumotor Plus to fault if the motor moves more than one revolution away from the commanded position.

The gains computed with the **TUNE** command form a control structure known as a *Pole-Placed Controller*. These gains are not directly compatible with the PIDV control structure. This means that either you use the self-tuning gains or you use the PIDV gains. The **GAINX** command allows you to switch between these different control schemes at any time. **GAINX:1** implements the self-tuning gains previously computed; **GAINX:0** implements the PIDV gains. This command is a buffered command so it can be used in sequences. This command is also independently saved so you can determine which control scheme will be in effect when the system is powered up.

*Do not issue the **GAINX:1** command when self-tuning has not been performed and do not issue this command if the results of the self-tuning were unstable gains. Issuing **GAINX:1** under these conditions can result in motor instability.*

Self Tuning Procedure

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step ① Mount the motor and connect it to the load. Motor should be firmly mounted and there should be no slippage in the coupling.

Step ② The **TUNE** command takes two parameters: tuning-move and tuning-option. In the command **TUNE:m,o** the **m** refers to the tuning-move and the **o** refers to the tuning-option.

a. Determine which tuning move to use. If the load cannot move in a particular direction or if there is known backlash in the system, use **TUNE:2,o** or **TUNE:3,o**. Otherwise use **TUNE:1,o**.

b. Determine what closed-loop response you want for your application. If in-position stiffness is important, use **TUNE:m,2** or **TUNE:m,3**. If low overshoot is important, use **TUNE:m,1**. If you are using a larger load.

Step ③ Issue the **TUNE** command. If you want to use tuning-move 1 with tuning-option 2 type the following:

```
> TUNE:1,2
```

If the motor does not move, reset the system and check your limit status. The motor must move for self-tuning to work properly.

Step ④ Wait for the system to compute new gains. When the amplifier is re-energized, do not touch the rotor until the system has responded with the following.

```
>*TUNING_COMPLETE
```

Step ⑤ If no error message has been sent to the terminal, the new self-tuned gains are now in place. If you wish to save these gains in non-volatile memory, issue the **SV** command. Once saved, this control structure is automatically put in use when the system is reset or powered up. To return to PIDV control, issue the **GAINX:0** command. To have PIDV control implemented on power-up, issue **SV** after **GAINX:0** .

Tuning Problems

The following section describes common problems you may run into while tuning your system

Ringing and Overshoot

If the system exhibits excessive overshoot (when making the transition from acceleration to constant velocity or from deceleration to a stop), it may be caused by a commanded acceleration that is much higher than the system can actually achieve. If this occurs, reduce the move acceleration with the **A** command until the overshoot is at an acceptable level. Reducing the integral gain also limits overshoot in most cases.

Position Errors Due to Load Inertia

Load is the inertia (mass times the radius of rotation squared) seen by the shaft of the motor (measured in oz/in/in). The amount of inertia affects the torque required. The torque required is a function of acceleration and inertia. If you find that your load inertia is larger than 10 times the motor inertia, you may want to trade some speed performance for accuracy at the final position. In this case, reduce the velocity gain and increase the integral gain.

Slow Response Due to Load Inertia

In a system where the load inertia is larger than the factory setting supports, it may be possible to improve the system response by allowing the overshoot to increase. In this case, increase the velocity gain, increase the proportional gain, and decrease the integral gain.

Position Errors Due to Friction

If load friction represents a large percentage of the motor's torque, the end-of-move position may have an unacceptable error. In this case, you can trade off system response for final position accuracy. To do this, increase the integral gain, increase the derivative gain, and if necessary, decrease proportional or velocity gains.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Shaft and Coupling Vibration	If the load is coupled to the shaft through a non-rigid coupler, it is possible for the shaft and coupler to oscillate at a frequency greater than the system's natural frequency. In this case, it may be possible to trade system response for system stability. To do this increase the integral and derivative gains, while decreasing velocity or proportional gains.
Inadequate Response Time (Frequency Response)	Some systems may involve very little inertia, but need very high acceleration. In this case, you may want to narrow the range of values that the system can handle to optimize the move profile for a light load. To do this, increase the proportional and velocity gains while decreasing the integral gain.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑤

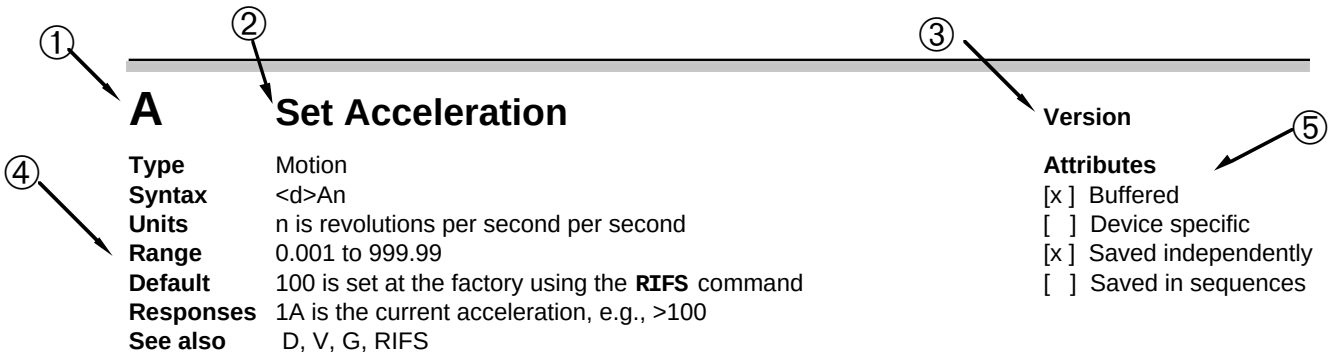
Software Reference

The information in this chapter will enable you to:

- ❑ Identify the four types of commands in Compumotor's X-Series Language
- ❑ Use this chapter as a reference for the function, range, default, and sample use of each command

Command Format Description

The following section describes the format of the command descriptions used in this chapter. The numbered arrows refer to the numbered sections below the drawing.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

① Command Identifier

The letter or letters used to represent the command.

② Command name

This name used to refer to the command. For example, Acceleration for the A command.

③ Version

The revision of software in the Compumotor Plus when the described command was first introduced or last modified. If the revision level of the software you are using is equal to or greater than the revision level listed here, the command is available in your unit. You can determine the level of software in your Compumotor Plus by issuing the Revision Level (RV) command.

④ Characteristics

The following sections describe the main characteristics of the command.

Type

This portion of the box contains the command's type. The four command types are listed below.

Set-Up: These commands define Set-Up conditions for the application. Set-Up commands include the following types of commands:

- Homing (go home acceleration and velocity, etc.)
- Input/Output (limits, scan time, in-position time, etc.)
- Tuning (servo or position tracking)
- General (set switches, return to factory settings, etc.)

Programming: Programming commands affect programming and program flow. For example, trigger, output, all sequence commands, quote, time delays, pause and continue, enable and front-panel, loop and end loop, line feed, carriage return, and backspace.

Status: Status commands respond (report back) information.

Motion: Motion commands affect motor motion (for example, acceleration, velocity, distance, go home, stop, direction, mode, etc.)

Syntax

This field shows the syntax for the command. Compumotor Plus commands use the following generic syntax: acspd

Variable a: This variable is the device address. If the address is optional it is shown in angle brackets: **<d>**. Only commands which require the Compumotor Plus to send a response require a device address. All commands may use a device address to designate which unit on a daisy chain the command is intended for.

Variable c: This variable is the command identifier, which is one or more letters.

Variable s: This variable represents a sign. A sign is not allowed for all commands. The **s** is not shown in the syntax if not allowed.

Variable p: This variable represents the parameters the command requires. There may be zero or more parameters. If the number of parameters is zero **n** is not shown in the syntax

Variable d: This variable is the end of command delimiter. This is always required and is not shown in the following descriptions for clarity. The delimiter may be a space character or a carriage return.

Units

This field describes what unit of measurement the parameter in the command syntax represents.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Range

This is the range of valid values that you can specify for **n** (or any other parameter specified).

Default

The default setting for the command is shown in this box. A command will perform its function with the default setting if you do not provide a value.

Response

The response to the command is shown in this box. Status commands report a condition in the indexer. Status commands do not affect the status they read.

Commands that set parameters report the parameters when the command is issued without a parameter. For example, **A1ØØ** sets the acceleration to 100 rps, but **1A** returns the current setting. ***Note**: To receive a response, a device address is required.*

See Also

Commands that are related or similar to the command described are listed here.

⑤ Attributes

Each command has attributes as shown below.

Attributes
[X] Buffered
[ ] Device specific
[X] Independently saved
[ ] Saved in sequences

Buffered

If the Buffered box is checked the command is buffered. If it is not checked the command is acted on immediately. Buffered commands are executed in the order they are received. An internal buffer, or storage area, holds the commands in a queue until the previous command has been executed.

Immediate commands are executed as they are received. Immediate commands are executed even if the command buffer has commands in it. For example, the Stop (**S**) command is immediate. When a Stop command is received the motor is stopped as soon as the command is received. The Compumotor Plus does not process the commands in its command buffer before stopping the motor.

Device specific

If the Device specific box is checked the command requires a device identifier. If it is not checked the command may be used with or without a device identifier. Commands which are device specific are normally Status commands. Device specific commands have a syntax description with a d by itself before the command. If it is not device specific the command syntax description has a **<d>** in angle brackets before the command.

Saved always

If the Independently saved box is checked the parameter controlled by the command is always saved. This differs from commands which may only be saved in sequences and those which are never saved. If neither the Saved always nor the Saved in sequences box is checked the command is never saved.

Saved in sequences

If the Saved in sequences box is checked the command will be saved only if it is in a sequence and you issue the Save command (**SV**). If neither the Saved always nor the Saved in sequences box is checked the command is never saved.

⑥ Description

A description of the command appears in this area along with any special considerations you should know about.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

⑦ Example

An example of how to use the command appears in this area. The left column contains the commands you would issue to the Compumotor Plus. The right column contains descriptions of what the commands do in the program.

Alphabetical Command List

" Quote

Version Z5

Type Programming
Syntax a"x
Units Any printable character
Range Any printable ASCII character (maximum of 11 characters)
Default None
Response SAMPLE
See also None

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

Any characters entered after the quotation mark (") are transmitted, exactly as they were entered, over the RS-232C link. A space (entered by the space bar) indicates the end of the command. A space is transmitted after the last character in the string. This command is used during buffered moves or sequences, or to command other Compumotor devices to move.

Command	Description
> MN	Set to Normal mode
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D12500	Set distance to 12,500 steps
> G	Execute the move (Go)
> "MOVE_DONE	After motor finished the move, the Compumotor Plus will send the message MOVE_DONE out from the RS-232C port

Command	Description
> MN	Set to Normal mode
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D12500	Set distance to 12,500 steps
> G	Execute the move (Go)
> "2XR1	Once the move is done, run sequence #1 is commanded on a unit with device address #2

Single Step

Version Z5

Type Programming
Syntax <a>#n
Units None
Range None
Default None
Response None
See also XST, XTR

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command controls the execution of a sequence when the Single Step Mode (XST) is enabled. Each time you enter the " command followed by a delimiter (carriage return or space), one command in the sequence buffer will be executed. You can run in Single Step mode only if you have RS-232C interface connected to a host. If you issue a Kill (K) command, while you are in Single Step mode, the sequence execution will be aborted, but the Single Step mode is retained. When you cycle power, the indexer will no longer be in Single Step mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> A5	Set acceleration to 5 rps^2
> V2	Set velocity to 2 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	Ends sequence definition
> XST1	Enables Single Step mode
> XR1	Executes sequence #1
> XTR1	Enables trace mode
#	Execute a command
*SEQUENCE_001_COMMAND_A5	Reports executing the A5 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_V2	Reports executing the V2 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_D10000	Reports executing the D1000 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_G	Reports executing the G command in sequence #1 The motor moves 10,000 steps
#	Execute the next command
*SEQUENCE_001_COMMAND_XT	Reports executing the XT command in sequence 1

A Set Acceleration

Version Z5

Type Motion
Syntax <a>An
Units rps
Range 0.001 to 2147483.687
Default 10
Response current acceleration, e.g., 10
See also D, V, G

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The acceleration command specifies the acceleration rate to be used upon executing the next Go (G) command. The acceleration remains set until you change it. You do not need to reissue this command for subsequent Go (G) commands. Accelerations outside the valid range cause the acceleration to remain in previous valid acceleration setting. The Compumotor Plus uses the same value for deceleration.

Though there is no effective limit for the acceleration definition, values above 24,000 rps^2 will all appear instantaneous.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)

B Buffer Status

Version Z5

Type Status
Syntax aB
Units None
Range None
Default None
Response *R or *B
See also BS

Attributes
[ ] Buffered
[X] Device specific
[ ] Saved independently
[ ] Saved in sequences

The buffer status command will report the status of the command buffer. If the command buffer is empty or less than 90% full, the controller will respond with a ***R**[cr].

The command buffer is 2,000 bytes long. A *B[cr] response will be issued if less than 10% of the command buffer is free.

*R = More than 10% of the buffer is free

*B = Less than 10% of the buffer is free

You may want to use this command when you load a long series of commands remotely. If the buffer size is exceeded, the extra commands will not be received by the system.

Command	Description
> 1B	*R (more than 10% of the Buffer is free)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCCA Buffered Configure Current Average

Version Z5

Type Set-Up
Syntax aBCCA
Units Amps
Range 0 to 7.5 amps
Default None
Response *AVERAGE_CURRENT_LIMIT=7.5_AMPERES
See also BCCP, RSE, CCA

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines a new maximum average current. If the average current commanded exceeds the value defined by the BCCA command (while the motor is turned on), the controller disables the amplifier and indicates an error. This command is identical to the CCA command, except that this command is buffered and may be executed within a sequence, where the CCA command is an immediate command. It is useful when the average current must be changed within a sequence.

Command	Description
> XD1	Define sequence #1
> BCCA2.0	Set average current limit to 2.0 amps
> A10	Set acceleration to 10 units/sec^2
> V10	Set velocity to 10 units/sec
> D25000	Set distance to 25,000 steps
> G	Execute the move (Go)
> BCCA5.0	Set average current limit to 5.0 amps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	End sequence definition

The average current limit is set to 2 amps before the first 5,000 step move and to 5.0 amps before the second 5,000 stop move.

BCCP Buffered Configure Current Peak

Version Z5

Type Set-Up
Syntax aBCCP
Units Amps
Range 0 to 8.5 amps
Default 8.5 amps
Response *PEAK_CURRENT_LIMIT=8.5_AMPERES
See also BCCA, CCA, RSE

Attributes
[X] Buffered
[X] Device specific
[X] Saved always
[X] Saved in sequences

This command defines the maximum peak current that can be sent to the motor. The value entered becomes the new peak current limit. If the current going to the motor exceeds the value defined by the BCCP command, the controller disables the amplifier and indicates an error. This command is identical to CCP except that it is buffered. It is a useful command when you want to change the peak current in a sequence.

Command	Description
> XD1	Define sequence #1
> A10	Set acceleration to 10 rps^2
> BCCA5.0	Set average current limit to 5.0 amps
> V10	Set velocity to 10 rps
> D25000	Set distance to 25,000 steps
> G	Execute the move (Go)
> 1BCCP7.0	Set peak current limit to 7.0 amps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	End sequence definition

The peak current limit is set to five (5) amps before the first 10,000 step move and to eight (8) amps before the second 10,000 step move.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCDB Buffered Configure Dead Band

Version Z5

Type Set-Up
Syntax aBCDB
Units Steps
Range 0 to 167772160
Default 0
Response Current setting, e.g., 0
See also CDB, SSCI

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

The buffered configure dead band command defines the dead band value. If a parameter follows the command, the value will become the new dead band value in motor steps. Output #2 may be used to indicate when the absolute value of the following error is outside of the dead band region. When the following error exceeds the dead band, the in-position output will turn ON. When the following error is within the dead band region set with this command, the in-position output will turn OFF.

Command	Description
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D50000	Set distance to 10,000 steps
> SSCI	Turn on enable output #2 when *In Position*
> BCDB50	Configure dead band to 50 steps

BCDG Buffered Configure Derivative Gain

Version Z5

Type Set-Up
Syntax aBCDG
Units Percent of maximum value
Range 0 to 99
Default 30
Response *DIFFERENTIAL_GAIN=30_PERCENT
See also CDB, BCDB, BCDM, CDG

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the derivative gain to be used for tuning. When you tune the drive using the derivative gain (BCDG) command, you are actually setting a percentage of the derivative gain maximum (CDM).

This command is identical to the CDG command except that it is buffered. It is useful when you must change the derivative gain in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D32000	Set distance to 32,000 steps
> BCDG44	Set the differential gain to 44% of the maximum value
> G	Execute the move (Go)
> BCDG20	Set the differential gain to 20% of the maximum value
> G	Execute the move (Go)

BCDM Buffered Configure Derivative Maximum

Version Z5

Type Set-Up
Syntax aBCDMn
Units None
Range 0 to 32767
Default 400
Response *DERIVATIVE_GAIN_MAXIMUM=400
See also CDM, BCDG, BCIM, BCPM

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the maximum derivative gain you may use for tuning purposes. If a valid number is entered, it will become the new maximum derivative gain. When you tune the drive using the Configure Derivative Gain (CDG), you will actually set a percentage of the maximum derivative gain. This command is identical to the Configure derivative Maximum (CDM) command except that it is buffered.

Command	Description
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D32000	Set distance to 32,000 steps
> BCDM15000	Set derivative maximum gain to 15,000
> G	Execute the move (Go)
> BCDM5000	Set derivative maximum gain to 5,000
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCIG Buffered Configure Integral Gain

Version Z5

Type Set-Up
Syntax aBCIGn
Units Percent
Range 0 to 99
Default 50
Response *INTEGRAL_GAIN=50_PERCENT
See also CIG, BCDG, BCIM

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

If a value is supplied, the specified value will become the new value for the integral gain. A new value for the integral gain is calculated using the entered value as a percent of maximum I value. The BCIG command differs from the Configure Integral Gain (CIG) command only in that it can be stored in a sequence.

This command defines the new integral gain to be used by the drive. If a valid number is entered the new integral gain is calculated using the following equation:

```
CIG * CIM ÷ 100 = Integral gain
```

A new value for the integral gain will be calculated using the entered value as a percent of maximum I value. The BCDM command differs from the CDM command only in that it is a buffered command and can be executed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps^2
> D32ØØØ	Set distance to 32,000 steps
> G	Execute the move (Go)
> BCIG5Ø	Set integral gain to 50% of maximum value
> G	Execute the move (Go)

BCIL Buffered Configure Maximum Integral Sum Limit

Version Z5

Type Set-Up
Syntax aBCILn
Units None
Range 0 to 214748347
Default 65535
Response *INTEGRATOR_LIMIT=65536
See also CIL

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the maximum integral sum limit. The value for the sum of the integral value cannot exceed the value set by the BCIL command. This command is useful if you want to move into a final position quickly. If you have a small BCIL value, the system will not allow the sum of the errors to exceed the n value resulting in the motor moving into position faster. If you make the BCIL value very large, the response of the final move will be slow. The Save (SV) command must be used to retain the value in nonvolatile memory.

Command	Description
> BCIL7ØØØØ	Set the maximum sum of the integral error to 70,000

BCIM Buffered Configure Integral Maximum

Version Z5

Type Set-Up
Syntax aBCIM
Units None
Range 0 to 32767
Default 9000
Response *INTEGRAL_GAIN_MAXIMUM=9000
See also CIM, CIG

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command is identical to the Configure Integral Maximum (CIM) command except that it is buffered.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps^2
> D32ØØØ	Set distance to 32,000 steps
> BCIM2ØØØ	Set maximum integral gain to 2000
> G	Execute the move (Go)
> BCIM5ØØØ	Set maximum integral gain to 5000
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCPE Buffered Configure Position Error

Version Z5

Type	Set-Up
Syntax	aBCPEn
Units	Steps
Range	0 to 336544320
Default	5000
Response	*MAXIMUM_POSITION_ERROR=5000_STEPS
See also	CDE, CDB, CPE

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

The BCPE command specifies the value for the maximum position (or following) error. The BCPE command differs from the CPE command only in that it can be executed sequentially. If no parameter is specified, the BCPE command reports the current value for CPE.

This command is identical to Configure Position Error (CPE) command except that it is buffered. It is useful when the position error must be changed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> BCPE4ØØØ	Set maximum following error to 4,000
> D32ØØØ	Set distance to 32,000
> G	Execute the move (Go)
> BCPE6ØØØ	Set maximum following error to 6,000
> G	Execute the move (Go)

BCPG Buffered Configure Position Gain

Version Z5

Type	Set-Up
Syntax	aBCPGn
Units	Percent
Range	0 to 99 percent
Default	15
Response	*PROPORTIONAL_GAIN=15_PERCENT
See also	CPG, CPM

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command is identical to the Configure Positional Gain (CPG) command except that it is buffered. It is useful when the proportional gain needs to be changed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D32ØØØ	Set distance to 32,000
> BCPG2Ø	Set proportional gain to 20
> G	Execute the move (Go)
> BCPG5Ø	Set proportional gain to 50
> G	Execute the move (Go)

BCPM Buffered Configure Positional Maximum

Version Z5

Type	Set-Up
Syntax	aBCPMn
Units	Steps
Range	0 to 32767
Default	100
Response	*PROPORTIONAL_GAIN_MAXIMUM=100
See also	CPG, CPM

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

This command causes the Compumotor Plus to calculate a new value for integral gain. It will be calculated using the entered value as a percent of maximum value. The BCPM command is a buffered version of the CPM command.

You may want to use this command when you must change the positional maximum within a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D25ØØØ	Set distance to 25,000 steps
> BCPM5ØØ	Set maximum proportional gain to 500
> G	Execute the move (Go)
> BCPM6ØØ	Set maximum proportional gain to 600
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCVG Buffered Configure Velocity Gain

Version Z5

Type	Set-Up
Syntax	aBCVG
Units	Percent of maximum
Range	0 to 99
Default	25
Response	*VELOCITY_GAIN=25_PERCENT
See also	None

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

The Buffered Configure Velocity Gain (BCVG) command is identical to the Configure Velocity Gain (CVG) command except that it is buffered. BCVG is useful when you need to change the velocity gain in a sequence.

The velocity gain is related to the error in the motor speed with respect to the velocity commanded by the PID control loop. If a valid parameter is entered, the velocity gain will be recalculated using the new percentage of the maximum.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØ	Set distance to 5,000 steps
> BCVG5Ø	Set velocity gain to 50
> G	Execute the move (Go)
> BCVG1Ø	Set velocity gain to 10
> G	Execute the move (Go)

BCVM Buffered Configure Velocity Maximum

Version Z5

Type	Set-Up
Syntax	<a>BCVMn
Units	Velocity
Range	0 to 32767
Default	2000
Response	VELOCITY_GAIN_MAXIMUM=n
See also	CVG, CVM

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the maximum velocity gain to be used for tuning. When you tune the drive using the Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command. This command is identical as Configure Velocity Maximum (CVM) except that it is buffered.

Changes made with this command will not be permanent (saved in non-volatile memory) until a Save (SV) command is issued.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 10 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØ	Set distance to 5,000 steps
> BCVM5Ø	Set velocity maximum to 50
> G	Execute the move (Go)
> BCVM1Ø	Set velocity maximum to 10
> G	Execute the move (Go)

BS Buffer Space

Version Z5

Type	Status
Syntax	aBS
Units	Bytes (characters)
Range	None
Default	None
Response	*100
See also	B

Attributes
[ ] Buffered
[X] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Buffer Space command reports the amount of space available in the command buffer. The report is in bytes (characters). When entering long string commands, check the buffer status to be sure it will not be overflowed by subsequent commands. All characters sent to the Compumotor Plus use space in the buffer including delimiters such as carriage returns and spaces.

Command	Description
> 1BS	Request buffer space available
> *1ØØ	Space for 100 characters is remaining in the command buffer

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BSP Buffer Space

Version Z5

Type	Set-Up
Syntax	aBSP
Units	Steps
Range	±838860800
Default	0
Response	None
See also	SP, PI, MPA, PR, PZ, D

Attributes
[x] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command is a buffered version of the **SP** command, it allows you to set the buffered absolute counter (value n). The **BSP** command is useful for labeling a position value at a certain point. For example, you can set the zero reference point (home) to some location other than that of the physical hardware home.

C Continue

Version Z5

Type	Programming
Syntax	aC
Units	None
Range	None
Default	None
Response	None
See also	PS, U

Attributes
[ ] Buffered
[X] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Continue (C) command ends a pause state. It enables your indexer to continue executing buffered commands. After you initiate a pause with the Pause (PS) command or the Pause and Wait for Continue (U) command, you can clear it with a Continue (C) command. This command is useful when you want to transmit a string of commands before you actually need to execute them.

Command	Description
> PS	Pause execution until the indexer receives a C command
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)
> T1Ø	Wait 10 seconds after the move
> VØ	Set velocity to zero
> G	Decelerates the motor to zero velocity
> C	Start executing commands in buffer

CCA Configure Current Average

Version Z5

Type	Set-Up
Syntax	aCCAn
Units	Amperes
Range	0 to 7.5
Default	7.5
Response	*AVERAGE_CURRENT_LIMIT=n_AMPERES
See also	BCCA, CCP, RSE

Attributes
[ ] Buffered
[X] Device specific
[X] Saved independently
[ ] Saved in sequences

The CCA command defines a new maximum average current. If the average current commanded exceeds the value defined by CCA command, while the motor is turned on, the controller will disable the amplifier and indicate an error.

Command	Description
> CCA5.0	Set average current to 5 amps

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CCP Configure Maximum Current Peak

Version Z5

Type	Set-Up
Syntax	<a>CCPn
Units	Amperes
Range	0 to 8.5
Default	8.5
Response	*PEAK_CURRENT_LIMIT=n_AMPERES
See also	BCCP, CCA

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the maximum peak current that will be sent to the motor. The value entered will become the new peak current limit. If the current going to the motor ever exceeds the value defined by CCP command, the controller will disable the amplifier and indicate an error.

Command	Description
> CCP4	Set peak current limit to 4 amps

CDB Configure Dead Band

Version Z5

Type	Set-Up
Syntax	<a>CDBn
Units	Steps
Range	0 to 167772160
Default	0
Response	*SLIP_FAULT_DEADBAND=n_STEPS
See also	BCDB, OM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

The Configure Dead Band (CDB) command defines the new dead band value in motor steps. If dead band is exceeded the slip fault output to the indexer connector will be active. When the slip fault line is off it indicates that the absolute value of the following error is within the dead band region. This is useful when you need to know if the motor rotor is within a certain tolerance range with respect to the commanded position.

Command	Description
> 1CDB50	Set the dead band to 50 steps
> 1CDB	Response *SLIP_FAULT_DEADBAND=50 STEPS

CDG Configure Differential Gain

Version Z5

Type	Set-Up
Syntax	<a>CDGn
Units	Percent of maximum (CDM)
Range	0 to 99
Default	30
Response	*DIFFERENTIAL_GAIN_MAXIMUM=n_PERCENT
See also	BCDG, CDM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the differential gain be used for tuning. When you tune the drive using the Differential Gain (CVG) command, you are setting a percentage of the Differential Gain Maximum (CDM) command.

Command	Description
> 1CDG30	Set the differential gain to 30% of the maximum value

CDM Configure Differential Maximum

Version Z5

Type	Set-Up
Syntax	<a>CDMn
Units	None
Range	0 to 32,767
Default	400
Response	*DIFFERENTIAL_GAIN_MAXIMUM=n
See also	BCDM, CDG

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the maximum differential gain you may use for tuning purposes. If you enter a valid number, it will become the new maximum differential gain. When you tune the drive using the Configure Derivative Gain (CDG) command, you will actually set a percentage of the maximum differential gain.

Command	Description
> CDM500	Set the maximum differential gain to 500
> CDG35	Set the differential gain to 35% of the maximum value

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CGS Configure Gain Switching

Version Z5

Type Set-Up
Syntax <a>CGSn
Units Mode
Range 0 to 4
Default 1 and 4
Response *DERIVATIVE_SWITCHING_xx_TORQUE_SWITCHING_yy
See also CDG, CDM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines and reports the current gain switching mode. Torque and derivative gain switching are affected by this command.

Derivative gain switching linearly scales the derivative gain value (defined by the CDM and CDG commands) down to zero as the motor speed drops from 2.0 rps to 0.0 rps.

Torque gain switching provides a high-resolution torque command to the amplifier when the torque command computed by the PIDV software falls below a certain threshold. The purpose of this mode is to provide high in-position stability.

CGS1	Derivative switching off
CGS2	Derivative switching on
CGS3	Torque switching off
CGS4	Torque switching on

The default setting is derivative (switching off) and torque (switching on).

CIG Configure Integral Gain

Version Z5

Type Set-Up
Syntax <a>CIGn
Units Percent of maximum (CIM)
Range 0 to 99
Default 50
Response *INTEGRAL_GAIN=n_PERCENT
See also BCIG, CIM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command configures the integral gain to be used for tuning. When you tune the drive using Configure Integral Gain (CIG) command, you are actually setting a percentage of the Maximum Integral Gain (CIM) command.

Command	Description
> CIG70	Set the integral gain to 70% of the maximum integral gain

CIL Configure Integral Sum Maximum

Version Z5

Type Set-Up
Syntax <a>CIMn
Units Sum limit
Range 0 to 2147483647
Default 65535
Response *INTEGRATOR_LIMIT=n
See also BCIG, BCIL, BCIM, CIG, CIM, SV

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command defines the maximum integral sum limit. The value for the sum of the integral value is set by the CIL command. This command is useful if you want to move into a final position quickly. If you have a small CIL value, the system will not allow the sum of the errors to exceed the n value. This will result in the motor moving into position faster. If you make the CIL value very large, the response of final move will be slow. The Save (S) command must be used to retain the value in non-volatile memory.

Command	Description
> CIL70000	Set the maximum sum of the integral error to 70,000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CIM Configure Integral Maximum

Version Z5

Type	Set-Up
Syntax	<a>CIMn
Units	Max integral gain
Range	0 to 32,767
Default	9000
Response	*INTEGRAL_GAIN_MAXIMUM=n
See also	BCIM, CIG

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the maximum integral gain you may use for tuning purposes. If you enter a valid number, it will become the new maximum integral gain. When you tune the drive using the Configure Integral Gain (CIG) command, you are actually setting the percentage of the maximum integral gain.

Command	Description
> CIM1000	Set the maximum integral gain to 1,000 steps
> CIG45	Set the integral gain to 45% of the maximum integral gain

CIP Configure In-Position Time

Version Z5

Type	Set-Up
Syntax	<a>CIPn
Units	ms
Range	1 to 32767
Default	2
Response	*IN_POSITION_WAIT=n
See also	CDB, IM

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The CIP command specifies the time period that the motor must be within the dead band region before the in-position output is turned on. The n value is the number of 10ms periods to be used as the testing time frame.

The least amount of time the motor can be stopped and within the dead-band region before the motor is considered in-position is one 10ms time period. If at any point during that 10 ms the motor is out of the dead-band region, the in-position output will not turn on (the factory default setting is two 10ms periods, or 20 ms).

Note: Be sure to divide the number of milliseconds you want by 10.

The In-Position output provides a signal that can be used by a PLC to indicate when the motor has completed its move and settled into the user-specified position tolerance. The position tolerance is specified the the Configure Deadband (CDB) command.

Command	Description
> CIP5	In-position will be on if the motor positions its shaft within a dead-band region and stays in that position for at least 50 ms

CMR Configure Motor Resolution

Version Z5

Type	Set-Up
Syntax	<a>CMRn
Units	Steps per revolution
Range	200 to 25,600
Default	5000
Response	*n
See also	OFF, ON, ST1, ST0, SV

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

The Compumotor Plus motor actually functions at 12,800 steps per revolution (resolver resolution12,800). If you choose a multiple or submultiple of 12,800 as your motor resolution, you will not get any truncation error. Compumotor recommends that you use a multiple or submultiple of 12,800 as the motor resolution.

If you enter a CMR value of 5,000 steps per revolution, you would compute the scale factor as:

(12,800 • 65,536)/5,000 = 167,772.16

Since there is 16-bit precision, the 0.16 is truncated.

If you set move distance with D5ØØØØØ, the conversion from user-defined revs back to resolver revs is done as follows:

(500,000 • 167,772)÷65,536 = 1,279,998.779

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Since you can't move to a fractional position, the motor actually moves to 1,279,998 counts of the resolver. You might expect the motor to go exactly 100 revolutions (500,000/5,000). For this move, the motor will actually go 1,279,998/12800 = 99.99984375 revolutions. This error does not accumulate, because if you give a second move of same distance, the calculation will use the absolute distance requested by the indexer to calculate the next move.

If truncation error is a problem you can choose a resolution that divides evenly into (12,800 • 65,536) or 2,147,483,648. For example, 4,096; 8,1912; 16,384; etc. Remember to save (SV) any changes you wish to retain before cycling power to the drive.

Note; The CMR command can only change motor resolution when the amplifier if off.

Command	Description
> OFF	Turn amplifier off
> CMR4Ø96	Define a motor resolution of 4,096 steps per revolution
> ON	Turn amplifier on
> SV	Save the new resolution to EEPROM

CMTR Configure Motor

Version Z5

Type Set-Up
Syntax <a>CMTR
Units None
Range 1, 2L, 2H or 3
Default None
Response None
See also None

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

The CMTR command sets a number of internal parameters appropriate for the motor connected to the drive. There are two types of CPX drives. The low-power CPX drive can run CP57-120 and CP83-150 motor sizes. The high-power CPX can drive CP83-150 and CP106-210 motors. The CP83-150 motor is the only motor that runs on both high- and low-power drives.

Note: Be sure to use the appropriate command parameter for the type of drive you have. Refer to the following table.

Low Power Drive (CPL)	High Power (CPH)
CMTR1 for the CP57-120 motor	CMTR2H for the CP83-150 motor
CMTR2L for the CP83-150 motor	CMTR3 for the CP106-220 motor

Command	Description
> 1CMTR1	Set the CPX Drive for the CP57-120 motor

CPB Configure Pushbutton

Version Z5

Type Set-Up
Syntax <a>CPBn
Units None
Range 0 or 1
Default 1
Response None
See also None

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command allows you to control user access to the front-panel pushbuttons. Since the PID & V tuning parameters can be modified via the front control panel, and changing these values can effect move time and final position accuracy, it may be desirable to disable this interface. The CPB1 command enables the front-panel pushbuttons. CPBØ disables the front panel.

Command	Description
> L	Loop infinite times
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 unit/sec
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> CPB1	Enable pushbuttons
> TR1XX	Wait for Trigger input 1
> CPBØ	Disable pushbuttons
> N	End loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CPE Configure Position Error

Version Z5

Type	Set-Up
Syntax	<a>CPEn
Units	Steps
Range	0 to 336544320
Default	5000
Response	None
See also	BCPE, RSE

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

The response to this command defines or reports the maximum following error. If the absolute position error is greater than this number, the amplifier will shut itself off and generate an error code #20 on the LED display. If a valid number in steps is entered, it will become the new maximum following error. Otherwise, the current setting is reported. Exceeding the maximum following error is an error condition that will cause the amplifier to be shutdown. If the maximum following error is defined as zero, the shutdown motor on following error exceeded function is disabled and no amount of following error will generate an error condition or shutdown the motor. The factory default setting is one revolution of the motor.

The value of the following error is only calculated when the CPE command is given. The stored number is in terms of motor revolution. Changes of the motor resolution will leave the following error actual distance unchanged unless a new CPE command is issued.

This command differs from the Configure Dead Band (CDB) command, since being outside of deadband region only affects the slip fault output. The CPE settings will shut off the drive. Save your settings to the EEPROM if you wish them to be permanent.

Command	Description
> 1CMR5000	Set motor resolution to 5,000 steps/rev
> 1CPE5000	Set position error to 1 revolution

CPG Configure Proportional Gain

Version Z5

Type	Set-Up
Syntax	<a>CPGn
Units	Percent
Range	0 to 99
Default	15
Response	*PROPORTIONAL_GAIN=n_PERCENT
See also	BCPG, CPM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the proportional gain to be used for tuning. When you tune the drive using the Configure Proportional Gain (CPG) command, you are actually setting a percentage of the Proportional Gain Maximum (CPM). This command is useful in tuning out the positional error during moves.

Command	Description
> 1CPG4	Configure the proportional gain to 4% of the maximum value

CPM Configure Proportional Maximum

Version Z5

Type	Set-Up
Syntax	<a>CPMn
Units	None
Range	0 to 32,767
Default	100
Response	*PROPORTIONAL_GAIN_MAXIMUM=n
See also	CPG

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the maximum of the term that amplifies the position error. If a valid number is entered, it will become the new proportional gain maximum. Otherwise, the current setting is reported.

Command	Response
>1CPM	*PROPORTIONAL_GAIN_MAXIMUM=n

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CR Carriage Return

Version Z5

Type	Programming
Syntax	<a>CR
Units	None
Range	None
Default	None
Response	[cr]
See also	LF, " (Quote)

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Carriage Return (CR) command determines when the indexer has reached a particular point in the execution buffer. When the indexer reaches this command in the buffer, it responds by issuing a carriage return (ASCII 13) over its interface back to the host computer. If you place the CR command after a Go (G) command, it indicates when a move is complete. If you place the CR command after a Trigger (TR) command, it indicates when the trigger condition is met.

Command	Description
> MPA	Set to Absolute Position mode
> A5Ø	Set acceleration to 50 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> CR	Sends a carriage return

CVG Configure Velocity Gain

Version Z5

Type	Set-Up
Syntax	<a>CVGn
Units	Percent
Range	0 to 99
Default	25
Response	None
See also	BCVG

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the velocity gain to be used for tuning. When you tune the drive using the Configure Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command. This command is useful in determining how well the motor/drive servos or maintains a commanded velocity during moves.

Command	Description
> 1CVG6Ø	Set the velocity gain to 60% of the maximum velocity gain

CVM Configure Velocity Maximum

Version Z5

Type	Set-Up
Syntax	<a>CVMn
Units	Steps
Range	0 to 32,767
Default	3000
Response	aCVM IS *VELOCITY_GAIN_MAXIMUM=n
See also	CVG, BCVM

Attributes
[ ] Buffered
[ ] Device specific
[X] Saved independently
[ ] Saved in sequences

This command defines the maximum velocity gain to be used for tuning. When you tune the drive using the Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command.

Command	Description
> 1CVM3ØØØ	Set velocity gain maximum to 3,000.steps
> 1CVG6Ø	Set velocity gain to 60% of the maximum value

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D Distance

Version Z5

Type	Motion
Syntax	<a>Dn
Units	Steps
Range	-838860800 to +838860800
Default	5000
Response	None
See also	A, G, MN, MPA, MPI, V

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Distance (D) command specifies the end position for subsequent moves. The position is interpreted as incremental or absolute depending on the current positioning mode as set by the Motor Position Absolute (MPA) and Motor Position Incremental (MPI) commands.

In Incremental mode (MPI), the motor moves the distance specified by this command. The absolute position is changed by the same amount.

In Absolute mode (MPA), the motor moves to the absolute position specified by this command. If the current absolute position is the same as the specified position the motor does not move. It is possible for the motor to move in the negative direction for a positive move and vice versa. This happens when the absolute position specified is less than the current absolute position even though both positions may be positive.

The absolute position can be set to zero with a PZ or a Z (Reset) command.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØØ	Set distance to 50,000 steps
> G	Execute the move (Go)

DCA Display Current Average

Version Z5

Type	Status
Syntax	<a>DCA
Units	None
Range	None
Default	None
Response	*AVERAGE_CURRENT=n_AMPERES
See also	DCI, DCP

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The DCA command periodically reports the average current flowing through the motor in amperes. This information is reported and repeatedly updated until any character is sent to the Compumotor Plus. On a terminal, the character can be sent by pressing any key, such as the spacebar.

The Compumotor Plus calculates the average current by recording the instantaneous current several thousand times each second. It always keeps the previous 10 seconds of readings (about 60,000 readings). It then averages the readings every 10ms, and reports the average to the user upon receipt of the CDA command.

Command	Response
> 1DCA	*AVERAGE-CURRENT=n_AMPERES

DCI Display Current Instantaneous

Version Z5

Type	Status
Syntax	<a>DCI
Units	None
Range	None
Default	None
Response	aDCI = *CURRENT=n_AMPERES
See also	DCA, DCP

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The DCI command periodically reports motor current on an instantaneous basis. This number is reported in amperes and is repeatedly updated until a key is pressed. This number is a single sampling of the current.

Command	Response
> 1DCI	*CURRENT=+2.83_AMPERES

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DCP Display Current Peak

Version Z5

Type Status
Syntax <a>DCP
Units None
Range None
Default None
Response *CURRENT=n_AMPERES
See also DCI, DCA

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Display Current Peak command will periodically display/report the largest instantaneous current value commanded to the motor since the command was issued. This value is reported in amperes and is repeatedly updated until any character is sent to the controller. The controller samples the instantaneous current at 300 microsecond intervals. Each reading is compared to the largest previous reading. If the new value is larger, it will become the new value. This reading accumulates from the time the command was sent, so that the highest instantaneous current ever seen by the motor over long periods of time may be captured.

DFS Display Flags for Servo Parameters

Version Z5

Type Status
Syntax <a>DFS
Units None
Range None
Default None
Response *bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb
See also DFX

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command returns all the drives status flag as a 32 bit response. Each bit refers to a status flag as shown on the chart below. The associated number for each flag corresponds to each binary (1 or 0) response is in the following order: bit 31, bit 30...bit 0.

Bit	Function	Bit	Function
31	Reserved	15	Reserved
30	Reserved	14	Reserved
29	Reserved	13	Reserved
28	Reserved	12	Reserved
27	Reserved	11	enable circuit (enabled=0, disabled=1)
26	Reserved	10	Reserved
25	Reserved	9	Reserved
24	Reserved	8	Failed crc check (no = 0, yes = 1)
23	Reserved	7	Reserved
22	Reserved	6	Average current exceeded (no = 0, yes = 1)
21	Reserved	5	Max position error exceeded (no = 0, yes = 1)
20	Reserved	4	Reserved
19	Reserved	3	Driver internal error (no error=0, pwm hardware shutdown=1)
18	Reserved	2	Reserved
17	Reserved	1	Overcurrent (no=0, yes (shutdown error)=1)
16	Reserved	0	RS-232C CMD (on (ST0)=0, off (ST1)=1)

DFX Display Indexer Flags

Version Z5

Type Status
Syntax <a>DFX
Units None
Range None
Default None
Response *bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb
See also DFS

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The display Indexer Flags command returns all the indexers status flags as a 32 bit response. Each bit refers to a status flag as shown on the chart below. The associated number for each flag corresponds to each binary (1 or 0) response is in the following order: bit 31, bit 30...bit 0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit	Function	Bit	Function
31	Reserved	15	U command waiting for continue (no=0, yes=1)
30	Reserved	14	Waiting for a trigger (no=0, yes=1)
29	Reserved	13	Home to resolver position (no=0, yes=1)
28	Reserved	12	Back up to home limit (no=1, yes = 1)
27	Reserved	11	High speed portion of home move (not in progress=0, in progress=1)
26	Reserved	10	Execute a sequence (no=0, yes=1)
25	Reserved	9	Waiting on a timer (no=0, yes=1)
24	Reserved	8	Hit a CCW limit (no=0, yes=1)
23	Homing to resolver position (no=0, yes=1)	7	Hit a CW limit (no=0, yes=1)
22	Hit a soft CW limit (no=0, yes=1)	6	PS command waiting for continue (no=0, yes=1)
21	Hit a soft CCW limit (no=0, yes=1)	5	Absolute move direction (positive=0, negative=1)
20	Registration move in progress (no=0, yes=1)	4	Positioning mode (incremental/absolute) (MPI=0, MPA=1)
19	Home limit (not found=0,found=1)	3	Mode (preset/continuous) (MN=0, MC=1)
18	Jog (disable=0, enable=1)	2	Commanded move direction (positive=0, negative=1)
17	Queued for RM mode (no = 0, yes = 1)	1	Preset move in progress (not moving=0, moving=0)
16	Run sequence on power up (no = 0, yes=1)	0	Continuous move in progress (not moving=0, moving=0)

DPA Display Position Actual

Version Z5

Type Status
Syntax <a>DPA
Units None
Range None
Default None
Response ±nnnnnnnn where n is a digit from zero to nine.
See also DPE

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Display Position Actual (DPA) command periodically displays the actual absolute position in motor steps. The function is canceled by sending any character to the controller. The value displayed is a cumulative count based on the shaft position when the drive was enabled and is scaled by the Configure Motor Resolution (CMR) command.

Resulting decimals are truncated. This can cause a small non-cumulative error in position if the CMR resolution does not evenly divide into 12,800

DPE Display/Report Position Error

Version Z5

Type Status
Syntax <a>DPE
Units None
Range None
Default None
Response ±nnnnnnnn where n is a digit from zero to nine
See also None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command reports the difference between set point and actual position in steps. The position control algorithm uses this number to determine what sort of current should be sent to the motor. The difference between the command set point and the actual position is also used to determine if the motor is within the dead band specified in the Configure Dead Band (CDB) command. This number is reported in motor steps and is repeatedly updated until a key is pressed.

DPR Display/Report Position Resolver

Version Z5

Type Status
Syntax <a>DPR
Units None
Range None
Default None
Response None
See also None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command continuously reports the resolver position in motor steps within a single pole of the motor shaft. This position is reported in steps and is repeatedly updated until a key is pressed. The Compumotor Plus motor has 50 poles, and the resolver built onto the motor is absolute within one motor pole. The resolver divides each pole into 256 steps.

The drive has a natural resolution of 12,800 steps/rev (256 steps/pole * 50 poles/rev). The report from the DPR command can be useful in diagnosing resolver cable problems.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DPS Display/Report Position Set Point

Version Z5

Type	Status
Syntax	<a>DPS
Units	None
Range	None
Default	None
Response	None
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command reports the absolute number of pulses sent to the drive from the indexer since the drive was enabled (or reset). This number is reported in motor steps and is repeatedly updated until a key is pressed.

DTP Display Tuning Parameters

Version Z5

Type	Status
Syntax	<a>DTP
Units	None
Range	None
Default	None
Response	*---------P-----I-----D-----V *PERCENT__20____50____50____60 *MAXIMUM__100___9000__400___2000 *INTEGRATOR_LIMIT=10000 *MINIMUM_POSITION_ERROR=0_STEP *MOTOR_TYPE=CP1 *MOTOR_RESOLUTION=5000_STEP/REV *LIMIT_DISABLED
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command displays all pertinent tuning parameters at once. This is useful when tuning several parameters at once.

DVA Display/Report Actual Velocity

Version Z5

Type	Status
Syntax	<a>DVAn
Units	None
Range	None
Default	None
Response	None
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command reports the velocity of the motor in rpm. The instantaneous velocity is stored every millisecond. The average of the readings is reported every 128ms. Velocity is reported until a key is pressed.

DVS Display/Report Velocity Setpoint

Version Z5

Type	Status
Syntax	<a>DVSn
Units	None
Range	None
Default	None
Response	*VELOCITY_SETPOINT=n
See also	None

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The DVS command continuously reports the current target velocity. The velocity is reported in steps per second and is repeatedly updated until a key is pressed. The target velocity is the velocity used in the velocity portion of the servo loop by the PID algorithm. For more information on PID refer *Chapter 4, Application Design* section.

Command	**Description**
> 1DVS	Velocity setpoint is reported repeatedly until a number key is pressed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E Enable Communications Interface

Version Z5

Type	Programming
Syntax	<a>E
Units	None
Range	None
Default	None
Response	None
See also	F

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Enable Communications Interface (E) command allows the indexer to accept commands over the communications interface. You can re-enable the communications interface with this command if you previously disabled the interface with the Disable Communications Interface (F) command. If several revolutions are using the same communications interface, the E and F commands can help streamline programming.

Command	Description
> F	Disables all revs (axes) on the communications interface
> 1E	Enables device #1
> 4E	Enables device #4
> G	Executes the move (Go — only axes 1 and 4 will move)

ELSE Else

Version Z5

Type	Programming
Syntax	<a>ELSE
Units	None
Range	None
Default	None
Response	None
See also	IF, NIF, IFTR, IFFL, IFER

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The ELSE command is used in conjunction with the four IF commands and the end of IF...NIF statement.

If the IF condition evaluates **true**, then the commands between the IF and ELSE commands are executed and the commands between the ELSE and the NIF command are skipped.

Conversely, if the IF condition evaluates **false**, then the commands between the IF and the ELSE are ignored and the commands between the ELSE and NIF are executed.

The ELSE command is optional and does not have to be included in the IF statements. IF...ELSE...NIF statements take the general form show below.

IF(condition) commands ELSE commands NIF

Command	Description
>IF(INXXX1_OR_VARI>20)	If input status is XXX1 (input #1 is active) or variable 1 greater than 20 then execute the command following the ELSE command
> GD1	Execute predefined move #1
> ELSE	Else
> GD2	Execute predefined move #2
> NIF	End of it

F Disable Communications Interface

Version Z5

Type	Programming
Syntax	<a>F
Units	None
Range	None
Default	None
Response	None
See also	E

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The F command causes the Compumotor Plus to ignore all commands over the RS-232C interface except the Enable (E) command. The command is useful when you are programming multiple units on a single RS-232C interface. You send an F command to axes that you want to exclude from global commands. This allows you to program other units on the daisy chain without including a device identifier with every command.

This command is also useful when uploading programs (XU) from a unit on a daisy chain. If you do not disable the other units in a daisy chain, uploading programs may cause other units on the daisy chain to execute uploaded commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> 1F	Disables the communications interface on units with device address #1
> 3F	Disables the communications interface on units with device address #3
> G	All of the indexers (not 1 and 3) will execute a move (Go)

FILT Enable Digital Torque Filter

Version Z5

Type Set-Up
Syntax <a>FILTn
Units ms
Range 0.3 to 100
Default 5
Response None
See also TUNE, GAINX

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The FILT command implements a digital filter on the torque command prior to sending this command out to the amplifier. The parameter specifies the speed of the filter in milliseconds. This corresponds to the bandwidth of the filter (e.g., FILT5 implements a 5mS filter which has a bandwidth of 1/5mS or 200Hz). The higher the number the lower the bandwidth of the torque command.

The presence of the filter significantly reduces audible noise during motion and at standstill. This allows you to use much higher gains than previously possible. The slower the filter the quieter the system, but performance becomes more sluggish. You must assess the dynamic needs of your system in order to decide what filter value to use.

A low bandwidth application, such as a continuous move with low acceleration and deceleration rates can use, for instance, a 100Hz filter (FILT10) with high gains for stiff but quiet action. For applications with high accelerations, filters above 200Hz are more appropriate. FILTØ turns off the filter entirely.

G Go

Version Z5

Type Motion
Syntax <a>G
Units None
Range None
Default None
Response None
See also A, D, MC, MN, S, V

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Go (G) command instructs the motor to make a move using motion parameters that you have previously defined. You do not have to re-define Acceleration (A), Velocity (V), Distance (D), or the current mode (MN or MC) commands with each Go (G) command.

In Continuous mode (MC) the Go command initiates a move which must be terminated by an explicit stop action such as the Stop command or stop on trigger. In the Continuous mode (MC), you only need to enter the A and V commands prior to the G command. The system ignores the D command in this mode.

In Preset mode (MN) the distance the motor moves is affected by the current positioning mode: incremental or Absolute.

In Incremental mode (MPI), a G command causes the motor to move the distance you specified with the D command.

In Absolute mode (MPA), a G command interprets the distance as absolute with reference to absolute zero. Refer to the Distance (D) command for more information.

If the motor does not move in response to a G command the problem could be:

- ❑ The indexer is in Absolute Positioning mode and the motor is already at the absolute position specified an active end-of-travel limit switch may be on. Check the limit switches.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> A1	Set acceleration to 1 unit/sec^2
> G	Execute the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GA Go Home Acceleration

Version Z5

Type	Motion
Syntax	<a>GAn
Units	rps/sec
Range	0.001-2147483.647 rps/sec
Default	99
Response	*nnnnnnnnn
See also	A, GH, OS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The Go Home Acceleration(GA) command sets the linear acceleration value that the motor will use during any subsequent Go Home (GH) moves.

Command	Description
> GA5	Set go home acceleration to 5 rps^2
> GH-5	The motor accelerates at 5 rps^2 to 5 rps in the CCW direction and searches for home

GAINX Set Gain Type

Version Z5

Type	Set-Up
Syntax	aGAINX:n
Units	0
Range	0 or 1
Default	None
Response	None
See also	FILT, TUNE

Attributes
[X] Buffered
[X] Device specific
[X] Saved independently
[X] Saved in sequences

The GAINX command enables self-tuning or PIDV tuning.

The GAINX command is a buffered command which allows you to switch between the pole-placement control scheme resulting from self-tuning and the PIDV gain scheme. This command can be put in a sequence or its result can be saved in non-volatile memory. Since the two gain schemes are not directly compatible, this command can be used to take advantage of both self-tuning and the flexible PIDV control scheme. The format of this command is as follows:

GAINX:0	Enable PIDV tuning
GAINX:1	Enable self-tuning

The Compumotor Plus allows you to select whether you want to set the systems gains yourself using the CIG, CDG, CPG and CVG commands or use the gains the Compumotor Plus has chosen in response to the TUNE command. Refer to the TUNE and FILT commands for details.

Do not use GAINX:1 unless you have gone through the self-tuning procedure. Failure to do so will lead to unpredictable results. It is always a good idea to make sure your maximum position error is set to a relatively low value (e.g., 1 rev) while doing any sort of tuning.

Command	Description
GAINX:1	Select self tuning gains
A99	Set acceleration to 99 rps/s
V5Ø	Set velocity to 50 rps
D1ØØØØØ	Set distance to 100,000 steps
G	Execute the move
GAINX:Ø	Select PIDV gains

GD Go Defined

Version Z5

Type	Motion
Syntax	<a>GDn
Units	Move number
Range	1 to 32
Default	None
Response	None
See also	GDEF, G

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Go Defined (GD) command executes a move that has been predefined and precalculated with the move definition (GDEF) command.

The command sequence An Vn Dn G is compressed to GD. You set the acceleration, velocity and distance with the GDEF command. The values specified by the A, V, and D commands are not changed by the GDEF command. The delay from the time the command is processed until motion occurs is approximately 4ms which is significantly faster than executing the series: An, Vn, Dn, G.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> A1 V1 D1000	Set accel, vel and distance
> GDEF10,A25,V20,D500	Define and calculate move #10
> GDEF16,A2,V1,D200	Define and calculate move #16
> GD10	Execute predefined move #10
> GD16	Execute predefined move #16
> G	Use A1, V1, D1000 from above and move

GDEF Move Definition

Version Z3

Type Motion
Syntax <a>GDEFn,Ab,Vc,De
Units b = acceleration, c = velocity, e = distance
Range 1 to 10
Default None
Response None
See also GD

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The GDEF command defines move number n with parameters A (acceleration), V (velocity), and D (distance). The parameters for the move are calculated when the GDEF command is issued. The move may be executed using the GD command. Because the GDEF command is precalculated there is virtually no delay from the issuance of the GD and motion start.

You may define up to 32 predefined moves. The moves may be used any number of times in any sequence.

Command	Description
> GDEF1,A5,V5,D250000	Define and calculate move #1
> GDEF2,A20,V10,D100000	Define and calculate move #2
> GD1	Execute predefined move #1
> GD2	Execute predefined move #2

GH Go Home

Version Z5

Type Motion
Syntax <a>GHn
Units rps
Range ±0-49.99 rps
Default None
Response *nnnnnnnn
See also RC, OS, GA, GHF

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Go Home (GH) command instructs the Compumotor Plus to search for an externally specified reference position. The external position is specified using the home input. Once home, the Compumotor Plus resets the absolute position to zero. If another position is desired, you can use the Buffer Set Absolute Position (BSP) command to specify the position when home.

The Compumotor Plus searches for home as follows:

① Search using the direction and velocity specified in the GH command.
② When the home input changes state, the Compumotor Plus stops the motor using the Go Home Acceleration (GA).
③ If the home input is inactive as defined by the OSC command the motor reverses direction to re-enter the *home region*.
④ If the OSB command is set to 0 the motor stops, and considers itself at home. If the OSB command is set to 1 the motor moves to find the home edge defined by the OSH command. It does this at the Go Home Final velocity (GHF).
⑤ The motor then moves to approach home from the direction defined in the OSG command.
⑥ If the OSA1 command has been issued the motor then moves to the resolver position as defined by the GHP command.

If the system encounters an end-of-travel limit during the homing sequence, the motor reverses direction. If a second limit is encountered during homing, the homing sequence is aborted. You can use the RB command to find out if the home attempt was successful.

Command	Description
> GH-20	The motor moves in the negative direction at 20 rps and looks for the home limit input to go active

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GHF Go Home Final Velocity

Version Z5

Type	Motion
Syntax	<a>GHFn
Units	rps
Range	0.00 to 9.99
Default	None
Response	*nn.nn
See also	GH, OS, GA

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command sets the velocity for the final approach to home in the go home sequence of the GH command.

Command	Description
> GHF.1	The velocity of the final approach of the next go home move will be 0.1 rps
> GH2	Execute go home at 2 rps^2 in CW direction

GHP Define Resolver Home Position

Version Z5

Type	Motion
Syntax	<a>GHPn
Units	motor steps
Range	0 to 255
Default	128
Response	None
See also	GH, GA, GHF, OSA, OSB, OSC, OSG, OSH

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the resolver position the motor moves to at the end of a Go Home sequence (GH). The Compumotor Plus does not move to a resolver position unless the OSA command is set to 1 and the home limit is active as defined by the OSC command.

Compumotor Plus motors have 50 poles and a built in resolver which can resolve 256 steps within each pole. Therefore, a given resolver position is only unique with 7.2° of shaft rotation. This command provides an accurate and repeatable home position at the end of a Go Home sequence defined by OSB1 and the OSG and OSH commands. It is necessary that the GHF definition be such that the home limit switch causes the motor to stop repeatably within ±3.6° for the Go Home to Resolver position to be repeatable.

Command	Description
> GHP15	Defines the home resolver position to be 15

^H Backspace (Control-H)

Version Z5

Type	Programming
Syntax	<a>^H
Units	None
Range	None
Default	None
Response	None
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Backspace command (Control-H) allows you to delete the last character that you issued to the Compumotor Plus. (^H indicates that the Ctrl key is held down as the H key is pressed.)

Note: Once a command has begun execution the Backspace command has no effect.

The backspace (^H) command does not prevent execution of immediate commands after you send a delimiter (i.e., space or carriage return). This command tells the indexer to backup one character in its input buffer, regardless of what may be showing on a terminal. On some terminals, the Ctrl and the left arrow (←) key issues a backspace character.

H Set Direction

Version Z5

Type	Motion
Syntax	<a>H<n>
Units	Direction
Range	Nothing, + or -
Default	None
Response	None
See also	D

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Set Direction (H) command changes or defines the direction of the next move that the system will execute.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This command does not effect moves already in progress.

H+ = Set move to CW direction
H– = Set move to CCW direction
H = Reverses direction from the previous setting

In Preset moves, a Distance command (D) entered after the Set Direction (H) command overrides the direction set by the H command. In Continuous mode (MC) only the Set Direction (H) command can set the direction of motion.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D25000	Set distance to 25,000 steps
> G	Execute the move (Go) in CW direction
> H	Reverse direction
> G	Execute the move (Go) in CCW direction
> MC	Set to Continuous mode
> H+	Set direction to CW
> G	Move continuously in CW direction

IFER If Error Condition

Version Z5

Type Status
Syntax <a>IFER
Units None
Range None
Default None
Response None
See also IFER, IFIN, IFTR, NIF, RSE, XFR

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command checks to see if any fault condition or limit condition exists. If one does, the Compumotor Plus executes a conditional command. This command is very similar to the IF_THEN statement in basic programming.

If any hard or soft limit has been hit or if an excessive position error, excessive average current error, drive enable disconnect error, or EEPROM error is detected, the commands between IFER and NIF will be executed. At present this command does not distinguish between error conditions in the Compumotor Plus.

An IFxx command (IFTR, IFER, IFFL) cannot be used within another IF command.

Command	Description
> L	Loop forever
> IFER	If error condition do the following commands
>"ERROR!	Send the string "ERROR!" over RS-232C
> ST1	Turn off amplifier
> NIF	End of IF statement
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D1000	Set distance to 1000 steps
> G	Execute the move (Go)
> N	End of loop

IFFL Compare User Flag

Version Z5

Type Programming
Syntax <a>IFFLnnnnnnnn
Units None
Range None
Default None
Response None
See also NIF, SFL, IFER, IFIN

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The IFFL command uses the pattern set by the Set User Flag (SFL) command to determine if it should execute commands between the IFFL command the the next NIF command. This command is similar to the IF...THEN compound statement found in Basic programming.

The 8 bits set by the SFL command are compared to the 8 flags following the IFFL command. If the patterns match, the commands following the IFFL command up to the next NIF are executed. If the bit patterns do not match the program skips to the command immediately following the next NIF command. An IFxx command (IFTR, IFER, IFFL) cannot be used within another IF command.

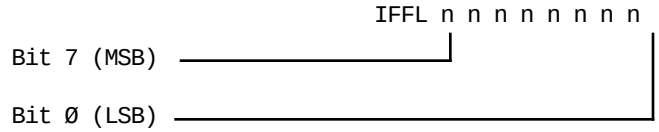
Each flag may have a one of three values shown below.

0 = not set,
1 = set

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

X = don't care (i.e., don't look at this flag)

This command is useful if you wish to make a decision based on previous sequence executions than will set or clear user flag bits. Also you can use a host computer to generate the SFL (User Flag) pattern depending on the condition, then you can use IFFL to perform branching type of operations.



Command	Description
> SFL101	Set user flag bits 7 and 5 and clear bit 6, the remaining bits are left unaltered
> IFFL1010	If user flag bits 5 and 7 are set and bits 6 and 4 clear, do the following commands
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25000	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End of statement
	If the IFFL pattern matches the SFL setting the motor moves 25,000 steps.

Command	Description
> SFLX1X1	Set bits 4 and 6
> IFFL1011XX10	If bits 1, 4, 5, and 7 are set and bits Ø and 6 are not set do the following commands
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5000	Set distance to 5,000 steps
> G	Execute the move (Go)
> NIF	End of statement
> IFFLXXX1	If bit 4 on SFL is set, do the following commands
> A10	Set acceleration to 10 rps^2
> V2	Set velocity to 2 rps
> D-25000	Set distance to 25,000 steps in the CCW direction
> G	Execute the move (Go)
> NIF	End of statement

There are 2 IF_THEN statements in this example. The first one does not match the user flag, and skips to the command after NIF. The second IFFL pattern does match the SFL (User flag) pattern, therefore the motor moves -25,000 steps.

IFIN If Input

Version Z5

Type Programming
Syntax <a>IFINnnnnnnnnn
Units None
Range None
Default None
Response None
See also IFER, IFTR, IFFL

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The IFIN command checks to see if the inputs match the parameters listed after the command. If they match, sequence execution continues with the command immediately after the IFIN command. If they do not match, sequence execution contues after the next ELSE or NIF statement.

This command differs from the IFTR command in that this command looks at all the inputs (including CW, CCW, and Home) and the IFTR command looks only at the trigger inputs as defined with the IM command.

IFTR If Trigger

Version Z5

Type Status
Syntax <a>IFTRnnnnnn
Units trigger condition
Range 0, 1, or X
Default None
Response None
See also IFTR, IFFL, IFER, NIF, TR

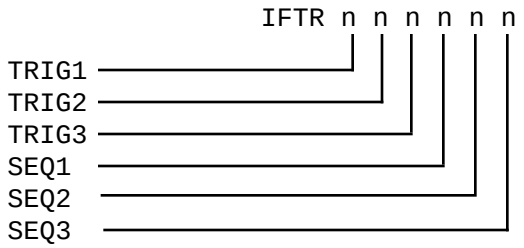
Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command uses the input pattern (6 inputs) of the Compumotor Plus Controller to execute the conditional commands. This command is very similar to the IF_THEN statement in basic programming. The difference is, the 6 input bits will determine the next command to be executed. If the hardware input patterns match the IFTR input pattern, then the commands between IFIN and NIF will be executed. If patterns do not match, the command following the NIF will be executed. For this command, 1 means active and 0 means not active. An X means the Compumotor Plus doesn't care whether that particular input is active or not. An IFxx command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(IFTR, IFER, IFFL) cannot be used within another IF command. *Note: IF commands may not be nested.*

This command is useful for branching and performing conditional moves. For example, depending on where you are, you may want to perform different moves.



Command	Description
> IFTR1ØX	If TRIG1 is active and TRIG2 is not active, issue the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)
> NIF	End of IF statement
> IFTRXXX1ØX	If SEQ1 is active and SEQ2 is not active, issue the following commands
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> G	Execute the move (Go)
> NIF	End of IF statement

IM Input Mode Configure

Version Z5

Type Set-Up
Syntax <a>IMn
Units Input number
Range 1 to 5
Default 1
Response n
See also SN, SS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command configures the auxiliary inputs for different functions. By reconfiguring the inputs, the same input may be used for different functions such as trigger inputs, sequence select inputs, jog inputs, registration input, or stop input.

This command specifies how you may define the three sequences and three trigger inputs as follows.

Mode 1 The inputs are used to select sequences 1 through 7. (Note the BCD weighting of the sequence lines listed below.)

Mode 2 The inputs are used to select sequences 1 through 39, using BCD weighting for the input lines.

Mode 3 Dedicates the triggers for use as Stop, Jog+, and Jog-.

The Trigger Pause (TR) command may still be used to halt sequence execution on any input line even though it is defined as a sequence select by this mode. Because of this, if you are not using all of your input lines for sequence selection you may use the unused inputs as trigger inputs even though they have been defined as sequence selection inputs.

Sequence 40 may not be selected via the sequence select inputs. Sequence 40 is always run on power up and may be run via the XR4Ø command over the RS-232C interface.

In input modes 1 through 3, the Continuous Sequence Scan (SSJ1) command allows sequences to be selected by grounding the appropriate sequence select inputs to obtain a desired BCD value. If the Sequence Hold (XQ1) command is issued, the input lines must be released (ungrounded) before another sequence may be run. Grounding all sequence select lines causes no sequence to be selected.

Changing the BCD values of sequence input lines results in a new sequence being run that corresponds to the new value after the current sequence is completed. The BCD sum of the values of the grounded lines determines which sequence is run. For example, grounding SEQ1 and SEQ2 results in sequence 6 being run (refer to the table below).

The SN command is used to determine how long the input must be stable before running a new sequence.

The following table shows the different input modes available.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Input Mode 1	Front Panel Designation	IM1	BCD
	TRIG 1	Trigger 1	Ø
	TRIG 2	Trigger 2	Ø
	TRIG 3	Trigger 3	Ø
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 2	Front Panel Designation	IM2	BCD
	TRIG 1	Sequence Select	2Ø
	TRIG 2	Sequence Select	1Ø
	TRIG 3	Sequence Select	8
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 3	Front Panel Designation	IM3	BCD
	TRIG 1	STOP	Ø
	TRIG 2	JOG +	Ø
	TRIG 3	JOG -	Ø
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 4	Front Panel Designation	IM4	BCD
	TRIG 1	Registration	0
	TRIG 2	JOG +	0
	TRIG 3	JOG -	0
	SEQ 1	Sequence select	4
	SEQ 2	Sequence select	2
	SEQ 3	Sequence select	1

Input Mode 5	Front Panel Designation	IM5	BCD
	TRIG 1	Registration	0
	TRIG 2	Trigger	0
	TRIG 3	Trigger	0
	SEQ 1	Sequence select	4
	SEQ 2	Sequence select	2
	SEQ 3	Sequence select	1

Command	Description
> XE4Ø	Erases sequence #40
> XD4Ø	Defines sequence #40
> IM1	Changes to input mode #1
> XR1Ø	Runs sequence #10
> XT	Ends sequence definition
	Indexer will run sequence #10 upon power up.

IO Immediate Output

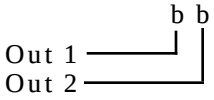
Version Z5

Type Set-Up
Syntax <a>IObb
Units State of an output
Range 0 or 1
Default None
Response None
See also O

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command immediately sets the output bits as specified in the pattern. Set the output mode (O) command for other uses of the outputs.

This command allows output to be used as general-purpose outputs, which can be controlled through RS-232C (independent of motor movement).



Command	Description
> MN	Set to Normal mode
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> D9999	Set distance to 9,999 steps
> IO11	Turn outputs 1 and 2 on before beginning of move
> G	Execute the move (Go)
> IOØØ	Turn outputs 1 and 2 off after the move

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IS Input Status

Version Z5

Type	Status
Syntax	<a>IS
Units	None
Range	None
Default	None
Response	*bbb_bbb_bbb_b
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

Reports the status of all hardware inputs. Including trigger, sequence, select lines, and limits.

This is not a software status. It will report the actual hardware status of the inputs. This command is useful for troubleshooting an application, to verify that Limit switches, Trigger inputs and home switches work.

Command	Description
> 1IS	Input status of device 1 is reported

JA Jog Acceleration

Version Z5

Type	Motion
Syntax	<a>JAn
Units	rps/sec
Range	0.001 to 99.999 rps/sec
Default	99
Response	None
See also	JV, IM, OS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The jog acceleration command allows you to specify the linear acceleration rate that will be applied to the motor until it reaches constant jog velocity. The jog acceleration remains set until you change it.

Command	Description
> JA3Ø	Set jog acceleration to 30 rps^2
> JV2	Set jog velocity to 1 unit/sec

JV Jog Velocity

Version Z5

Type	Motion
Syntax	<a>JVn
Units	rps
Range	0.00 to 50.00
Default	1
Response	None
See also	JA, IM, OS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command allows you to set the maximum jog velocity. Jogging allows you to manually position the motor. When you jog the motor, it will accelerate to the velocity specified and remain at this speed until you turn off the jog input.

Command	Description
> JA1Ø	Set jog acceleration to 10 rps^2
> JV5	Set jog velocity to 5 rps

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

K Kill Motion

Version Z5

Type	Motion
Syntax	<a>K
Units	None
Range	None
Default	None
Response	None
See also	S

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Kill (K) command is an emergency stop command. This command causes indexing to cease immediately. The Compumotor Plus attempts to stop the motor instantaneously.

> ***CAUTION***
>
> **Because of the high deceleration rate commanded by the K command, the motor may lose torque, particularly when driving large inertial loads being moved at high speeds. The load could drive itself past limit switches and cause damage to the mechanism and or personnel.**

The K command clears all indexer flags (e.g., Display Flags of Indexer (DFX) command will report all zeros.)

In addition to stopping the motor, the K command terminates any loops, time delays or down-loads in process. It also clears the command buffer.

Command	Description
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V2	Set velocity to 2 rps
> G	Execute the move (Go)
•	
•	
•	
> K	Commands the motor to stop instantly

KILL Kill Motion

Version Z5

Type	Motion
Syntax	KILL
Units	None
Range	None
Default	None
Response	None
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The KILL command is identical to the K command. See the K command for a complete description.

L Begin Loop

Version Z5

Type	Programming
Syntax	<a>Ln
Units	number of loops
Range	0 to 65,535
Default	None
Response	None
See also	N, Y

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

When you combine the Loop (L) command with the End-of-Loop (N) command, all of the commands between L and N will be repeated the number of times indicated by n. If you enter the L command without a value specified for n, or with a 0, subsequent commands will be repeated continuously.

The End-of-Loop command prompts the indexer to proceed with further commands after the designated number of loops have been executed. The Stop Loop (Y) command indicates where execution will stop. The Immediate Pause (U) command allows you to temporarily halt loop execution. You can use the Continue (C) command to resume loop execution.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Loops may be nested up to 16 levels deep. Possible structures are shown below:

```
L N
L L N N
L L N L N N
L L L L N N N N
L L L N  L N N L N N
```

Command	Description
> L5	Loops 5 times
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D1ØØØØ	Set distance to 10,000 steps
> G	Execute the move (Go)
> N	Specifie the above 10,000-step move to be repeated five times

LA Acceleration Limit

Version Z5

Type Set-Up
Syntax <a>LAn
Units rps/sec
Range 0.001 to 2147483.647
Default 900
Response None
See also A, LD

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The Limit Acceleration (LA) command allows you to define the deceleration rate that should be used when an end-of-travel limit is encountered. This command is useful if you do not want an abrupt stop upon encountering a limit. However, you should be careful to specify a deceleration rate that will stop the load before it can do any damage. Normally, limit switches are placed so that the motor has room to safely decelerate the load.

Command	Description
> LA5Ø	The motor decelerates at 50 rps^2 when it encounters an end-of-travel limit.

LD Limit Disable

Version Z5

Type Set-Up
Syntax <a>LDn
Units State of limit enable
Range 0 to 3
Default 0
Response None
See also None

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The Limit Disable (LD) command allows you to enable/disable the end-of-travel limit switch protection. The LDØ condition does not allow the motor to turn without properly installing the limit inputs. If you want motion without wiring the limits, you must issue the LD3 command.

When you disable limit inputs, they are reconfigured as trigger inputs.

Enable CCW and CW limits	0 (Default)
Disable CW limits	1
Disable CCW limits	2
Disable CCW and CW limits	3

Command	Description
> 1LDØ	Enables CW and CCW limits. The motor will move only if the limit inputs are bypassed or connected to limit switches.
> 1LD3	Allows you to make any move, regardless of the limit input state

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LED Show Number on LED Display

Version Z5

Type	Programming
Syntax	<a>LDn
Units	Number to display
Range	0 to 99
Default	None
Response	None
See also	None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command allows the user to set a number to the 2 digit LED display on the Compumotor Plus. This command is useful when you want the Compumotor Plus to interact with the operator. For example, you might define number 90 as the moving status, or number 95 as stopped status.

Command	Description
> PS	Pause execution of following commands until the Continue (C) command is issued
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25000	Set move distance to 25,000 steps
> LED90	Turn on 2 digit LEDs to 90 (Moving)
> G	Execute the move (Go)
> LED95	Turn on 2 digit LEDs to 95 (stopped)
> C	Continue execution

LF Line Feed

Version Z5

Type	Programming
Syntax	aLF
Units	None
Range	None
Default	None
Response	[lf] where [lf] is a line feed character (ASCII 10)
See also	CR, "

Attributes
[X] Buffered
[X] Device specific
[ ] Saved independently
[X] Saved in sequences

The LF command causes the Compumotor Plus to send a line feed character (ASCII 10) when the command is processed. Since the command is buffered it may be executed interactively or in a sequence.

If you place the LF command after a Go (G) command, the line feed is sent at the completion of the move. This is because both the LF and G commands are buffered. Placing a LF command after a Trigger (TR) command, causes the line feed to be sent when the trigger condition is met.

Command	Description
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D15000	Set distance to 15,000 revs
> G	Execute the move (Go)
> LF	Transmits a line feed character over the communications interface after the move is completed

MC Mode Continuous

Version Z5

Type	Motion
Syntax	<a>MC
Units	None
Range	None
Default	None
Response	None
See also	A, D, MN, MPA, MPI, V

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Mode Continuous (MC) command causes subsequent moves to ignore any distance parameter and move continuously. You can clear the MC command with the Move Normal (MN) command. The MC command will only change the mode when the motor is not moving.

The Indexer uses the Acceleration (A) and Velocity (V) commands to reach continuous velocity.

Using the Time Delay (T), Trigger (TR), and Velocity (V) commands, you can achieve basic velocity profiling.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)

MN Mode Normal

Version Z5

Type Motion
Syntax <a>MA
Units None
Range None
Default None
Response None
See also MC, D, A, V, MPA, MPI

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Mode Normal (MN) command sets the positioning mode to preset. In Normal mode, the motor will move the distance specified with the Distance (D) command. To define the complete move profile, you must define Acceleration (A), Velocity (V), and the Distance (D). The MN command is used to change the mode of operation from Mode Continuous (MC) back to normal or preset. The MN command can only have an effect when the motor is not moving.

Command	Description
> MN	Set to Preset Positioning mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D1000	Set distance to 1,000 steps
> G	Execute the move (Go)

MPA Mode Position Absolute

Version Z5

Type Set-Up
Syntax <a>MPA
Units None
Range None
Default None
Response None
See also D, MC, MN, MPI, PZ

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command sets the positioning mode to absolute. In this mode all move distances are referenced to absolute zero. Note that in Absolute mode (MPA) and Normal mode (MN), giving two consecutive Go (G) commands will cause the motor to move once, since the motor will have achieved its desired absolute position at the end of the first move.

Position Absolute mode (MPA) is most useful in applications that require moves to specific locations.

You can set the absolute counter to zero by cycling power or issuing a Position Zero (PZ) command.

Command	Description
> MN	Set to Normal mode
> MPA	Set to Position Absolute mode
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D25000	Set distance to 25,000 steps
> G	Motor will move to absolute position 25,000
> D12500	Set absolute position to +12,500 steps
> G	Motor will move to absolute position +12,500 steps

MPI Mode Position Incremental

Version Z5

Type Set-Up
Syntax <a>MPI
Units None
Range None
Default None
Response None
See also MPA, D, MN

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command sets the positioning mode to incremental. In incremental mode all move distances specified with the Distance (D) command will be referenced to the current position. The MPI command is most useful in applications that require repetitive movements, such as feed to length applications.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> MN	Set to Normal mode
> MPI	Set to Positioning Incremental mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D1ØØØØ	Set distance of move to 10,000 steps
> G	Move 10,000 steps CW
> G	Move another 10,000 steps CW

N End of Loop

Version Z5

Type Programming
Syntax <a>N
Units None
Range None
Default None
Response None
See also L

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command marks the end of loop. You can use this command in conjunction with the Loop (L) command. All buffered commands that you enter between the L and N commands are executed as many times as the number that you enter following the L command.

You may nest loops 16 levels deep.

Command	Description
> PS	Pauses execution of buffered commands until the indexer receives a Continue (C) command
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D1ØØØØ	Set move distance to 10,000 steps
> L5	Loops the above series of commands five times
> G	Execute the move (Go)
> N	Ends the loop
> C	Clears pause and executes all the buffered commands

NIF End of IFxx Command

Version Z5

Type Programming
Syntax <a>NIF
Units None
Range None
Default None
Response None
See also IFFL, IFER, IFIN, IFTR

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command marks the end of IF compound command. The Compumotor Plus has four IF commands.

IFFL If the User Flag pattern matches, execute the following commands
IFER If certain error conditions exist, execute the following commands
IFIN If the input pattern matches, execute the following command
IFTR If the trigger input pattern matches, execute the following command

When you use any of the four commands, listed above, you must complete the command by placing the NIF command after the commands you want to execute should the IFxx command test true.

In case the IF condition does not match, the commands between IF and NIF are not executed, and the program will skip to the command following the NIF command.

Command	Description
> IFTRXXX1Ø	If SEQ 1 is active and SEQ 2 is not active issue the following commands and go to command following the NIF command
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End of statement
> IFTRXXXØ1	If SEQ 1 is not active and SEQ 2 is active, issue the following commands, skip to the command following NIF
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps in the opposite direction
> G	Execute the move (Go)
> NIF	End of statement
> IFTRXXX1	If SEQ 1 is active, issue the following command, then skip to the command following the **NIF** command
> 1"DONE	Transmit message saying done
> NIF	End of statement

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

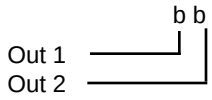
O Output

Version Z5

Type	Programming
Syntax	<a>Obb
Units	Output—on or off
Range	0, 1 or X
Default	None
Response	None
See also	IO, OM

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Output (O) command turns the programmable output bits on and off. This is used for signaling remote controllers, turning on LED's, or sounding whistles. The output can indicate that the motor is in position, about to begin its move, or is at constant velocity, etc.



Command	Description
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D20000	Set move distance to 20,000 steps
> O01	Set programmable output 1 off and output 2 on
> G	Execute the move (Go)
> O00	After the move ends, turn off output 2

OFF Amplifier Off

Version Z5

Type	Programming
Syntax	<a>OFF
Units	None
Range	None
Default	None
Response	None
See also	ON, ST0, ST1

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command allows you to shut the amplifier off and remove current from the motor. You must issue an ON command to re-energize the motor. The contents of the buffer will be cleared when you execute this command.

The OFF command removes motor torque, allowing you to move the motor manually. This command is similar to the Shutdown (ST1) command. However, this command is immediate, and ST1 is buffered.

Command	Description
> OFF	Powers down the motor (no torque)

OM Output Mode

Version Z5

Type	Set-Up
Syntax	<a>OMn
Units	Mode
Range	1 to 6
Default	1
Response	None
See also	IO, O, CDB

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command allows you to set different output modes.

Mode	Output 1	Output 2
OM1	Programmable	Programmable
OM2	Programmable	Slip Fault (In Position)
OM3	Motor Moving	Programmable
OM4	Motor Moving	Slip Fault (In Position)
OM5	Motor Moving/Not Moving	Registration Occurred in the current or previous move
OM6	Motor Moving/Not Moving	Registration Failed. The previous mode was not terminated by a registration move

Programmable means the output may be turned on and off with the Output (O) and Immediate Output (IO) commands.

Motor Moving means the output is on when the internal indexer is commanding the motor to move. It is off when the internal indexer is commanding the motor to hold position. Note: The servo loop can cause motion

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

which will not turn on the motor moving output. For example, if the motor is stationary and you move the motor shaft with your hand and the motor makes a move to correct its position, the motor moving output will not turn on, because the indexer did not command the motion.

Slip Fault is turned on when the deadband specified is exceeded (refer to the CDB command for more information). It is turned off when a slip fault is detected. The mode you use will depend upon your application. If you need to control other devices using programmable outputs, run your application in OM2 mode. If you are only interested in detecting a slip fault or monitoring the motor movement, use the OM4 mode.

Registration Occured turns on Out 2, which becomes active as soon as registration is detected and stays active until the next move is begun. Thus it stays active **after** the registration move is over. This feature was added to allow signalling of an error condition. If the output never comes on, registration has not occured during the previous move. This gives you a way to detect if the registration sensor is functioning properly, and that your mechanical system is aligned properly.

Registration Failed turns on Out 2, which is used to signal an error condition. This output comes on if, during the previous move, a registration signal was not detected. (Registration must be enabled to detect the registration sensor.) This allows external controllers to let the registration moves occur continuously in a sequence while an external controller monitors Out 2. If the output ever comes on the controller knows that a move has occured in which no registration mark was detected. This may be an error condition in some systems, warranting stopping the system in order to find the cause of the error.

Command	Description
> OM1	Both outputs may be programmed to turn on and off using the O and IO commands.
> O1X	Turns on Output #1 and leaves Output #2 at itsprevious setting.

ON Amplifier On

Version Z5

Type Programming
Syntax <a>ON
Units None
Range None
Default None
Response None
See also OFF, ST1, STØ

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command turns the amplifier back on from the off state. If you issue the OFF command to shut the drive down, issuing the ON command re-enables the current to the motor, restoring motor torque. The ON command stops existing motion. This command is similar to the Shutdown (STØ) command. The ON command, however, is immediate and STØ is buffered.

Command	Description
> ON	Turns the motor on

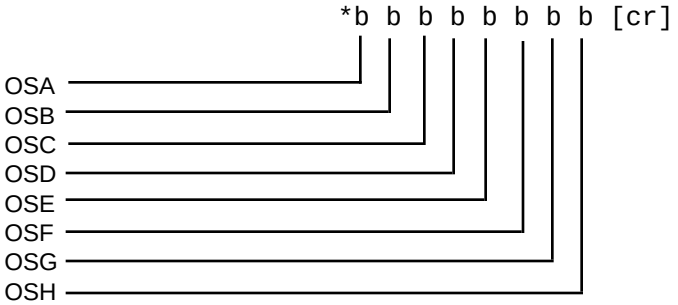
OS Function Setup Report

Version Z5

Type Status
Syntax aOS
Units None
Range None
Default None
Response *bbbb_bbbb
See also OSA, OSB, OSC, OSC, OSE, OSG, OSH

Attributes
[X] Buffered
[X] Device specific
[ ] Saved independently
[X] Saved in sequences

The OS command reports the status of the commands OSA, OSB, OSC, OSC, OSE, OSG, and OSH.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following table briefly describes each of the OS commands.

Command	Description
OSA	Go home to resolver position
OSB	Back-up to home switch
OSC	Define active edge of home switch 1 = Active high signal
OSD	Not defined
OSE	Enable Jogging
OSF	Not defined
OSG	Define final home approach direction 1=CCW
OSH	Define active edge of home switch to stop on 1 = CCW
OSI	Save Sequence Scan mode on stop

OSA Go Home to Resolver Position

Version Z5

Type Set-Up
Syntax <a>OSAb
Units None
Range 0 or 1
Default 0
Response None
See also GH, GHP, OSA, OSB, OSC, OSE, OSG, OSH

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The OSA command enables positioning to a resolver position at the end of a go home sequence.

OSAØ Do not go home to resolver position

OSA1 Go home to resolver position

The OSA1 command causes the motor to creep to the resolver position specified by the GHP command at the end of a go home move (GH). This command is used to produce a highly accurate, repeatable go home move. See the GHP command for more details. The OSA0 command causes, the motor to go home in accordance with settings of the OSB, OSG, and OSH commands.

OSB Backup to Home Switch

Version Z5

Type Set-Up
Syntax <a>OSBb
Units Mode
Range 0 or 1
Default 0
Response None
See also OSG, OSH

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The OSB command affects the way the motor goes home in a Go Home sequence (GH) . It causes the motor to reverse direction and find the edge of the home region if set to 1, or stop as soon as the motor enters the home region if set to 0.

OSBØ Do not back up to home switch
OSB1 Back up to home switch

If this command is set to 0, the Compumotor Plus considers the motor at home if the home input is active, after encountering the active edge of home region.

If this command is set to 1, the Compumotor Plus decelerates the motor to a stop after encountering the active edge of the home region. The motor moves in the opposite direction of the initial go home move at the velocity set by the GF command until the active edge of the home region is encountered. The Compumotor Plus then considers the motor to be at home.

Refer to the GH command for more detailed information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OSC Define Active State of Home Switch

Version Z5

Type	Set-Up
Syntax	<a>OSCb
Units	Mode
Range	0 or 1
Default	0
Response	None
See also	GH, OSG, OSH, SS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The OSC command sets the active state of the home input. It enables you to use either a normally closed or a normally open switch for homing. OSCØ requires that a normally open switch be connected to the home limit input. OSC1 requires that a normally closed switch be connected to the home limit input.

OSCØ Active state of home input is n = 0 (closed)
OSC1 Active state of home input is n = 1 (open)

Command	Description
> OSC1	Set the active state of the home input to open

OSE Jog Enable

Version Z5

Type	Set-Up
Syntax	<a>OSEb
Units	Enabled or disabled
Range	0 or 1
Default	0
Response	None
See also	SS, GH, IM, JA, JV

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The OSE command enables jogging the motor via the trigger inputs. Use the IM command to select the inputs to be used for jogging the motor.

OSEØ Jog disable
OSE1 Jog enable

OSG Final Homing Direction

Version Z5

Type	Set-Up
Syntax	<a>OSGn
Units	Direction
Range	0 or 1
Default	0
Response	None
See also	OSB, OSH, GH

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The OSG command allows you to set the direction from which the Compumotor Plus approaches the final home position. By specifying the direction for approaching home you get a more repeatable home position, and you can minimize the number of moves required to establish the home position.

OSGØ Sets the final Home approach direction to CW
OSG1 Sets the final home approach direction to CCW

This command will have no effect unless the OSB1 command has been issued, telling the Compumotor Plus to back-off from home.

OSH Reference Edge of Home Switch

Version Z5

Type	Set-Up
Syntax	<a>OSHn
Units	Direction
Range	0 or 1
Default	0
Response	aOSGn
See also	OSG, OSB, GH

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The OSH command allows you to select the side of the home area you want to use as the home position. This command allows you to specify a more repeatable home position.

OSHØ Selects the CW side of the home signal as the *edge* on which the final approach will stop
OSH1 Selects the CCW side of the home signal as the *edge* on which the final approach will stop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The CW edge of the home switch is defined as the side of home closest to the CW limit. Conversely, the CCW edge of the home switch is defined as the side of home closest to the CCW limit. If 1, the CCW edge of the home switch is the home position.

This command will have no effect unless the OSB1 command has been issued, telling the Compumotor Plus to back-off from home.

OSI Save Sequence Scan Mode On Stop

Version Z5

Type Set-Up
Syntax <a>OSIn
Units Mode
Range 0 or 1
Default 0
Response aOSIn
See also SSJ,XQ, IM

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

OSI1 is used to preserve the Sequence Scan mode (SSJ1) during and after execution of a stop command. If OSI0 is issued, the Sequence Scan mode (SSJ) is cleared after execution of a STOP.

Care must be taken in the use of OSI1 to ensure that no sequences are unintentionally executed after a stop. The Sequence Interrupted-Run (XQ) command can help with this problem.

PR Absolute Position Report

Version Z5

Type Programming
Syntax <a>PR
Units None
Range None
Default None
Response *snnnnnnnnn (where s is the sign and 2,147,483,647 steps)
See also D, MN, MPI, MPA, PZ, SP

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The PR command reports the motor's current position with respect to the absolute zero position.

The absolute zero position is established as the current position during power up and when an PZ or Z command is issued. The absolute position counter can track up to $\pm 2^{31}$ - 1 or 2,147,483,647 steps. If the counter is overrun in the relative position mode (by running the motor continuously for long periods of time, (e.g., 24 hours at 20 revolutions per second and 5,000 steps per revolution), the absolute position reported is invalid.

Command	Description
> PZ	Resets the absolute counter to zero
> LD3	Disable both CW and CCW limits
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5000	Set move distance to 5,000 steps
> G	Execute the move
> 1PR	Request absolute position report
> *+0000005000	Indexer reports position is 5,000

PS Pause

Version Z5

Type Programming
Syntax <a>PS
Units None
Range None
Default None
Response None
See also C, U

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The PS command pauses processing of the command buffer or active sequence until the Compumotor Plus receives a Continue (C) command. This command is useful if you want to enter a series of commands but not execute them until you have finished entering them.

This command is useful for interactive tests and in synchronizing multiple indexes that have long command strings.

Command	Description
> PS	Pauses execution of following commands until the Compumotor Plus a C
> A5	Sets acceleration to 5 rps^2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

> V5	Sets velocity to 5 rps
> D25ØØØ	Sets move distance to 25,000 steps
> G	Execute the move (Go)
> T2	Delays the next move for 2 seconds
> G	Execute the move (Go)
> C	Continues execution

PZ Set Absolute Position to Zero

Version Z5

Type Programming
Syntax <a>PZ
Units None
Range None
Default None
Response None
See also MN, MPI, MPA, PR, D

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The PZ command sets the absolute position counter to zero. Subsequent absolute moves are made with reference to the zero position established with this command. The absolute position is also set to zero on initialization and during a reset (Z), or after a successful home move.

Command	Description
> PZ	Zero the absolute counter
> MPA	Make all preset moves with respect to absolute zero position
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØ	Set move distance to 2,500 steps
> G	Execute the move (Go)
> 1PR	Report Absolute Position (Response = *+2500)
> PZ	Zero the absolute counter
> 1PR	Report absolute position (Response = *0)

Q Set Velocity Profiling Mode

Version Z5

Type Programming
Syntax <a>QØn
Units None
Range None
Default None
Response None
See also RM

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Q command controls whether the Compumotor Plus is in or out of the Velocity Profiling mode. In this mode you use RM commands to control the instantaneous velocity of the motor. There is no acceleration ramping other than that provided by frequent changes in velocity using the RM command in this mode.

QØ exits the velocity profiling mode
Q1 enters the velocity profiling mode

Note: The motor stops when QØ is issued.

Command	Description
> Q1	Enter Velocity Profiling mode
> RMØØ11	Go to RM velocity of (11 hex) RM units/sec
> RMØØ55	Go to RM velocity of (55 hex) RM units/sec
> RMØ1ØØ	Go to RM velocity of (100 hex) RM units/sec
> RMØØ55	Go to RM velocity of (55 hex) RM units/sec
> RMØØ11	Go to RM velocity of (11 hex) RM units/sec
> QØ	Exit Velocity Profiling mode

R Report Indexer Status

Version Z5

Type Status
Syntax aR
Units None
Range None
Default None
Response *x (where x is one of four letters described below)
See also RA, RB, RC, RSE

Attributes
[ ] Buffered
[X] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Request Indexer Status (R) command is used to get the general status of the indexer. Possible responses are shown below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Response Character	Definition
*R	Ready
*S	Ready, Attention Needed
*B	Busy
*C	Busy, Attention Needed

The following conditions will cause a response indicating the indexer is busy:

* Performing a preset move
* Accelerating/decelerating during a continuous move
* A time delay is in progress
* In RM mode
* Paused
* Waiting on a Trigger
* In Jog mode
* Going home
* In Power-On Sequence mode
* Running a sequence
* Executing a loop

The following conditions indicate attention is required.

* A servo error exists
* Go home failed
* Limit has been encountered
* Sequence execution was unsuccessful

When the response indicates that attention is required, more details on the condition are available from the RA, RB, and RE commands.

Do not repeatedly issue this command with little or no time between status requests. Doing so could result in overloading the Compumotor Plus. If you need to closely monitor the status of the Compumotor Plus, use a short delay before re-issuing the command.

This command is not intended to be used to determine if a move is complete. Rather it should be used after the move is complete to determine if there were errors or faults. Use a buffered status request command or a programmable output to indicate move completion.

Command	Description
> 1R	Request status
> *R	The indexer is ready to accept commands and no error conditions require attention

RA Limit Switch Status Report

Version Z5

Type Status
Syntax <a>RA
Units None
Range None
Default None
Response *x (where x is a letter listed below.)
See also R, RB

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Limit Switch Status Report (RA) command responds with the status of the end-of-travel limits during the last move as well as the present condition. This is done by responding with one of 16 characters representing the conditions listed below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Response	Last Move Terminated By CW Limit	CCW Limit	Limit Switch Active CCW	CW
*@	no	no	no	no
*A	yes	no	no	no
*B	no	yes	no	no
*C	yes	yes	no	no
*D	no	no	yes	no
*E	yes	no	yes	no
*F	no	yes	yes	no
*G	yes	yes	yes	no
*H	no	no	no	yes
*I	yes	no	no	yes
*J	no	yes	no	yes
*K	yes	yes	no	yes
*L	no	no	yes	yes
*M	yes	no	yes	yes
*N	no	yes	yes	yes
*O	yes	yes	yes	yes

The RA command is useful when the motor will not move in either or both directions. The report back will indicate whether or not the last move was terminated by one or both end-of-travel limits.

Command	Response
1RA*@	*@ The last move was not terminated by a limit and no limits are currently active

RB Loop, Pause, Sutdown and Trigger Status

Version Z5

Type Status
Syntax <a>RBn
Units None
Range None
Default None
Response *x (where x is a letter listed below)
See also RA, RC

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The RB command reports the status of currently executing loops, if the command buffer is currently paused, if the amplifier is shutdown, and if a trigger is active. The command responds with the form *x[cr] where x is listed in the response column of the following table.

Response	Loop	Pause	Shutdown	Triggered
*@	no	no	no	no
*A	yes	no	no	no
*B	no	yes	no	no
*C	yes	yes	no	no
*D	no	no	yes	no
*E	yes	no	yes	no
*F	no	yes	yes	no
*G	yes	yes	yes	no
*H	no	no	no	yes
*I	yes	no	no	yes
*J	no	yes	no	yes
*K	yes	yes	no	yes
*L	no	no	yes	yes
*M	yes	no	yes	yes
*N	no	yes	yes	yes
*O	yes	yes	yes	yes

Command	Response
> 1RB*	*A The indexer is currently executing a move.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REG Set Registration Parameters

Version Z5

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

Type Motion
Syntax <a>REGa,v,d,l
Units a = rps/sec
v = rps
d = steps
l = steps
Range a = 0.001 to 247483
v = 0.00 to 60.00
d = 0 to 838860800
l = 0 to 838860800
Default None
Response None
See also SSK, IM

The distance specified is the incremental distance the motor must traverse from the point at which the registration input goes active. The acceleration and velocity specified are parameters for the registration move. The velocity may be higher, lower, or the same as the move in progress. The lockout distance is the distance which the motor must traverse before recognizing the registration input (Trigger #1).

Registration is allowed in either preset or continuous moves, and can occur at any point in the move. Registration is disabled during jogging, homing, and when the motor is stopped. The registration input is recognized within 300 mS and is not debounced, the stopping position will vary by a small amount proportional to the speed of the motor. The motor's position will vary no more than the distance the motor can travel in 300 mS. Using a resolution of 5,000 steps per revolution, the motor can overshoot its desired position by no more than 1.5 steps/revolution/second. For example, at five revolutions per second, the motor could overshoot the desired position by 7.5 steps.

Registration is enabled with the SSK1 command, and disabled with the SSK0 command. When enabled, the registration move is performed upon receipt of an input on trigger #1 regardless of the input mode, but input mode #'s 4 and 5 dedicate Trigger #1 for registration input, and Trigger #'s 2 and 3 as JOG+ and JOG-, respectively. Input mode #5 selects trigger #1 as the registration input and leaves trigger #'s 2 and 3 designated as triggers. Output mode #5 allows the programmable outputs to be used to signal motion in progress on output #1 and registration move in progress on output #2.

When the registration command is issued in the interactive mode the indexer responds with *CALCULATING_REG_TABLES. If the amplifier is on the calculation takes about five to six seconds, if the amplifier is off the calculation takes less than a second. When finished, the indexer sends the message, *REG_TABLES_COMPLETE.

It is possible to define a registration move which is impossible to complete. When defining the registration distance, you need to obey the following relationship.

$$s \geq \frac{V_i^2 - V_o^2}{2a}$$

where,

s is the registration distance in revolutions
v_i is the initial velocity in rps
v_o is the target velocity in rps
a is the registration acceleration in rps/sec

If the registration move occurs during a continuous move the mode is switched to Normal mode. The next Go command will cause a preset or mode normal move to be executed. A Continouous mode (MC) command must be issued to return the indexer to its previous mode.

Command	Description
> MN	Set to Normal mode
> A10	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> D50000	Set distance to 50,000 steps
> G	Execute the move (Go)
> REG50,20,5000,40000	Set registration move to accel = 50, vel = 20, distance from registration mark = 5000, wait until 40000-steps from start of move to start looking for registration mark

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RFS Return Drive Parameters to Factory Setting

Version Z5

Type Set-Up
Syntax <a>RFS
Units None
Range None
Default None
Response None
See also Z

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

Return the drive to factory setting. The following settings are changed:

Deadband (CDB)	0
Gains (CIG, CDG, CVG, CPG)	Factory Default
Motor Resolution (CMR)	5000
Average Current (CCA)	Factory Default
Peak Current (CCP)	Factory Default
Following Error (CPE)	Factory Default

RG Go Home Status Request

Version Z5

Type Status
Syntax <a>RG
Units None
Range None
Default None
Response *x (where x is a letter listed below)
See also RA, RC, R, RB, GH

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

Go Home Status Request responds with either *@ or *A indicating success or failure of last go home attempt.

Response	Go Home Successful
*@	No
*A	Yes

RIFS Return Indexer to Factory Settings

Version Z5

Type Set-Up
Syntax <a>RIFSn
Units None
Range None
Default None
Response None
See also RFS

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

Execution of this command will cause all of the indexer and servo parameters to return to factory default settings, with the exception of the device address and the drive's motor configuration. The Compumotor Plus must have the amplifiers disabled when this command is executed (STO).

Command	Default
LD	**LDØ**
LA	900 rps^2
JA	99 rps^2
JV	rps^2
GA	rps^2
GHF	0.1 rps
GHP	128
SS and OS	cleared
IM	IM1
CIP	2 (20 milliseconds)
SN	20 milliseconds
OM	OM1

The device address and the motor type are not changed. The sequences are cleared and the run time variables are initialized to the following settings:

A = 10 rps^2
D = 1 rps
V = 1 rps/sec
R = RS-232C enabled
REG = 99 rps2/5 rps, rps, 1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RM Set Rate Multiplier Velocity

Version Z5

Type	Motion
Syntax	<a>RMh
Units	rps in hexadecimal
Range	00000 - FFFFF
Default	00000
Response	None
See also	Q1, Q0

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

RM command specifies the immediate velocity in Velocity Streaming mode.

The command is followed by 5 hexadecimal digits which represent the desired velocity. The velocity resolution is 0.0008 to 800 rps.

The velocity change is essentially instantaneous; there is no acceleration or deceleration ramp between velocities. A limit switch input stops movement while in velocity profiling mode, but does not cause the indexer to exit velocity streaming mode. RM mode is unidirectional. The direction will be the last direction either from an actual move or from a D or H command.

Bidirectional moves using this mode can be made by returning to zero velocity, switching out of RM mode, changing the direction (with the H command), and re-entering RM mode. The overhead required by this approach should be acceptable given the time required to stop when changing directions.

Use the following formula to generate the hexadecimal numbers required by the RM command.

① Calculate the decimal velocity number.

$$v_d = v_{rps} \bullet 1{,}310.72$$

where,

V_d is the decimal velocity number
V_{rps} is the decimal velocity in rps

② Convert V_d to hexadecimal.

Example: 1 rps on the Compumotor Plus equates to a decimal velocity number of 1,310.72 decimal. Now convert 1,310.72 to hexadecimal: 051E hexadecimal.

Applications requiring non-linear accelerations may use the QØ, Q1 and RM commands. Q1 is used to enter the Velocity Profiling mode, and QØ is used to exit the Velocity Profiling mode. While in this mode the RM command is used to generate velocity values that are immediately implemented, even while the motor is moving. This means that the RM command must be sent to the Compumotor Plus at the time the change in velocity is required. This creates a stair-step effect in velocity change. By implementing a large number of very small instantaneous velocity changes, a smooth, non-linear acceleration ramp can be achieved.

Command	Description
> Q1	Enter Velocity Streaming mode
> RMF3	Accelerate to 0.25 units/sec
> RMØ1E6	Accelerate to 0.5 units/sec
> RMØ3D7	Accelerate to 1 units/sec
> RMØ7AE	Accelerate to 2 units/sec
> T1	Run at 2 units/sec for 1 sec
> RMØ3D7	Decelerate to 1 units/sec
> RMØ12E6	Decelerate to 0.5 units/sec
> RMØØF3	Decelerate to 0.25 units/sec
> RMØØØØØ	Decelerate to 0 units/sec
> QØ	Exit Velocity Streaming mode

RS Status of Sequence Execution

Version Z5

Type	Status
Syntax	<a>RSn
Units	None
Range	None
Default	None
Response	*x (where x is a letter listed below)
See also	XR

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The RS command reports the status of the currently executing sequence or of the last sequence executed. It also reports if an invalid loop has been encountered in a sequence. This command responds in the form *x[cr] where x is a response character listed below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Response Character	Sequence Started	Sequence Ended	Bad Loop
*@	No	No	No
*A	Yes	No	No
*B	No	Yes	No
*C	Yes	Yes	No
*D	N/A	N/A	Yes

Whenever a sequence is started, the sequence start bit is set and the sequence end bit is cleared (this only occurs if the sequence is valid and is actually run). Whenever a sequence is completed, the start bit is cleared and the end bit is set.

*D is reported when there is an unbalanced number of loops and loop terminators inside a sequence. Starting a loop in one sequence and terminating it in another sequence is not allowed. Nested loops require complete closure before execution will begin.

Sequence started is true when: An XR, XRP or power-up successfully starts a sequence.
Sequence started is false when: A STOP or a KILL command is received, or limits are hit.
Sequence ended is true when: An XT is encountered, when a STOP or KILL is executed, or when end-of-travel limit is encountered.
Sequence ended is false when: A sequence is successfully started.

RSE Report Servo Errors

Version Z5

Type Status
Syntax <a>RSE
Units None
Range None
Default None
Response None
See also R, XFK

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

You can find out what error condition exists in the servo drive using this command. If a red LED' on the front panel illuminates, you should troubleshoot the unit using the RSE command. The possible error messages are:

Response	Description	Shown On Front Panel
00	No Errors	Yes
20	Maximun position error exceeded	Yes
22	Maximun average current exceeded	Yes
23	Drive enable plug not inserted	Yes
30	EEPROM Checksum error	Yes
60	RS-232C commanded shutdown (using the **ST1** or **0FF** commands)	No
61	Indexer incoming pulses (non-indexer version only)	Yes

RV Revision Level

Version Z5

Type Status
Syntax <a>RV
Units None
Range None
Default None
Response *92-nnnnnn-nn<xn> (where n is a digit from 0 to 9, and x is a letter)
See also None

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Revision (RV) command responds with the software part number and its revision level. The response is in the form shown below:

*92-nnnnnn-nn<xn>[cr]

The part number may be used to identify which product the software is written for, as well as any special features that the software may have. The revision level also identifies when the software was written. You may want to record this information in your records for future use. This information is useful when you consult Parker Compumotor's Applications Department.

Command	Description
> 1RV	*92-007730-01Z5

The product is identified by 92-007730. The revision level is identified by 01Z5.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

S Stop Motion

Version Z5

Type	Motion
Syntax	<a>S
Units	None
Range	None
Default	None
Response	None
See also	A, K, QØ, STOP, SSH

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command decelerates the motor to a stop using the last defined Acceleration (A) command. This command normally clears any remaining commands in the command buffer, unless prevented from doing so by the Clear/Save The Command Buffer On Stop (SSH1) command. When the SSH1 command is present, the S command stops only the current move. The indexer executes the next command in the buffer. The Stop (S) command does not stop the motor in Velocity Streaming or Rate Multiplier (RM) mode. If you are in the RM mode, issue an Exit Velocity Profiling Mode (QØ) command to stop the motor.

Command	Description
> MC	Sets to Continuous mode
> A1	Sets acceleration to 1 units/sec^2
> V1Ø	Sets velocity to 10 units/sec
> G	Execute the move (Go)
> A5	Sets acceleration to 5 units/sec^2
> S	Stops motor (motor comes to 0 units/sec at a deceleration rate of 5 units/sec^2)

SAVE Save Set-Up and Sequences

Version Z5

Type	Set-Up
Syntax	<a>SAVE
Units	None
Range	None
Default	None
Response	None
See also	SV

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The SAVE command is identical to the SV command. See the SV command for a complete description.

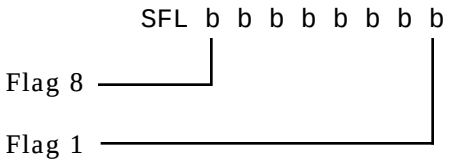
SFL Set User Flag

Version Z5

Type	Programming
Syntax	<a>SFLbbbbbbbb
Units	Flag
Range	0 or 1
Default	0
Response	None
See also	IFFL, NIF

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command sets a condition of 8 user flags. You can define the state of flags 1 through 8 using this command to be either a Ø, 1, or X. A Ø clears the corresponding flag, 1 sets the corresponding flag, and an X retains the flag's state. You need not specify the state of all of the flags in an SFL command: SFL11 sets flags 8 and 6 while leaving the remaining flags unaltered. Thus SFL11 is equivalent to SFL11XXXXXX.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Once you set flags 1 though 8, you can use the IFFL command to check the state of the flags and execute different commands based on the check.

Command	Description
> SFL1Ø1Ø	Set bits 5 and 7, and clear bits 6 and 4
> IFFL1Ø1Ø	If user flag bits 5 and 7 are set, then issue the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End of IFFL
	If the IFFL pattern matches the SFL pattern, the motor will move 25,000 steps

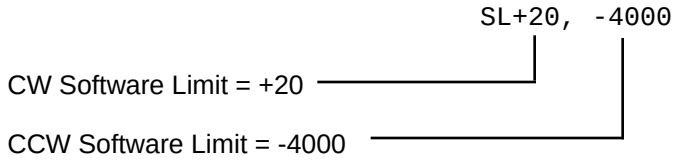
SL Set Soft Limits

Version Z5

Type Set-Up
Syntax <a>SLn,n
Units Steps
Range ± 838860800
Default 0
Response *CW_SOFTWARE_LIMIT_n_CCW_SOFTWARE_LIMIT_n
See also OUT, SLD

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command defines the absolute positions for the CW and CCW soft end-of-travel limits. Once you define the CW and CCW soft limits, the Compumotor Plus does not allow the motor to pass outside the specified absolute positions, unless the software limits are disabled by the SLD3 command. You may use the OUT command to configure one or more outputs to signal when a software end of travel limit is encountered. When the motor reaches a software limit, it comes to an immediate stop, using the acceleration specified with the Limit Acceleration (LA) command.



Command	Description
> SL+1ØØØØ,-26944	Set CW software limit to +10000 steps, set CCW software limit to -26944 steps
> SLDØ	Enable both CW and CCW software limits
> PZ	Set absolute position to zero
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D4ØØØØ	Set distance to 40,000 steps
> G	Execute the move (Go)
> SL,-29ØØØ	Set CCW limit to -29000
> SL+29ØØØ	Set CW limit to +29000

SLD Soft Limit Disable

Version Z5

Type Set-Up
Syntax <a>SLDn
Units Enable/disable state
Range 0 to 3
Default 3
Response Ø_CW_CCW_SOFTWARE_TRAVEL_LIMITS_ENABLED
1_CCW_SOFTWARE_TRAVEL_LIMIT_ENABLED
2_CW_SOFTWARE_TRAVEL_LIMIT_ENABLED
3_NO_SOFTWARE_TRAVEL_LIMITS_ENABLED
See also OUT, SL

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

This command enables or disables the soft end-of-travel limits defined by the SL command. This command is very similar to the Hardware Limit Disable (LD) command, except that it specifies soft limits instead of hard limits.

SLDØ Enables both CW and CCW software limits. Motion will not be allowed to go past the software limits.
SLD1 Disables CW Software limit. Can travel past CW software limit.
SLD2 Disables CCW Software Limit. Can travel past CCW software Limit.
SLD3 Disable both CW and CCW Software limits. Motor will ignore both CW and CCW software limits.

You can use the OUT command to configure an output to indicate that the software limit has been reached. However, if you disable the limits, the output will not be activated even if the motor passes the software limit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> SLØ,Ø	Set CCW software limit to 0 and CW software limit to 0
> PZ	Set absolute position to zero
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> SLD3	Disable both CW and CCW software limits
> G	Execute the move (Go)

SN Scan

Version Z5

Type Set-Up
Syntax <a>SNn
Units ms
Range 1 to 200
Default 12
Response None
See also None

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The Scan (SN) command allows you to define the debounce time (in milliseconds) for external sequence selection inputs. The debounce time is the amount of time that the sequence inputs must remain constant for a proper reading from a remote controller, such as a programmable logic controller (PLC). If you are using a PLC you should change the debounce time to match the *on time* of the PLC outputs.

Command	Description
> SN15Ø	Sets scan time of sequence select inputs to 150 mS

SP Set Position Absolute

Version Z5

Type Set-Up
Syntax <a>SPn
Units Counter
Range 0, 1
Default None
Response aSPn = aSP*n
See also SSA, SSB, SSD, SSF, SSG, SSH, SSJ, SSL

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The SS command reports the status of the SS commands (SSA - SSL) settings, b represents a zero or a one as specified by the respective SS commands.

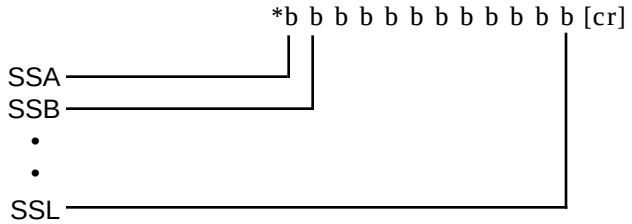
SS Set-Up Report

Version Z5

Type Status
Syntax aSS
Units None
Range None
Default None
Response aSS = *bbbbbbbbbbbb
See also SSA, SSD, SSG, SSH, SSI, SSJ, SSL, SSN, SSO, SSP

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The SS command reports the status of the SS commands (SSA - SSL) settings, b represents a zero or a one as specified by the respective SS commands.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following table briefly describes the function of each SS command.

Command	Description	Status
SSA	RS-232C Echo	0 = on, 1 = off
SSB	TRIG1 clears pause	0 = no, 1 = yes
SSC	Not defined	
SSD	TRIG1 dedicated stop line	0 = no, 1 = yes
SSE	Not defined	
SSF	TRIG1 sequence strobe	0 = no, 1 = yes
SSG	Clear/Save buffer on limit	0 = clear 1 = Save
SSH	Clear/Save buffer on stop	0 = clear 1 = Save
SSI	Enable/Disable Interactive Mode	0 = enable, 1 = disable
SSJ	Enable/Disable Continuous Scan mode	0 = enable, 1 = disable
SSK	Enable Registration mode	0 = no, 1 = yes
SSL	Resume execution enable	0 = no, 1 = yes

SSA RS-232C Echo Control

Version Z5

Type Set-Up
Syntax <a>SSAn
Units Enabled/disabled
Range 0 or 1
Default None
Response None
See also None

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The SSA command enables and disables echoing. When echoing is enabled the Compumotor Plus retransmits all characters sent to it over the RS-232C interface. When echoing is disabled, characters are not echoed from the Compumotor Plus. You must enable echoing when your Compumotor motion controllers are in a daisy chain.

SSAØ = Echo enabled (default)
SSA1 = Echo disabled

Command	Description
> SSA1	Turns echo off (characters sent to the Compumotor Plus are not echoed back to the host)

SSB TRIG1 Clears Pause

Version Z5

Type Set-Up
Syntax <a>SSBn
Units Enabled/disabled
Range 0 or 1
Default 0
Response None
See also C, SSD, SSF, SSK, SSL, PS, U

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

If you issue a SSB1, TRIG1 will clear a pause when connected to ground. The result of activating TRIG1 is identical to sending a C command. When SSB is set to 1, SSD, SSF, SSK, and SSL are all forced to 0.

SSD TRIG1 Dedicated Stop Line

Version Z5

Type Set-Up
Syntax <a>SSDn
Units enable/disable
Range 0 or 1
Default 0
Response None
See also S, SSB, SSF, IM

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

If SSD is set to 1, TRIG1 is dedicated as a stop line in input modes IM1 and IM2. This input is treated like a Stop command. In IM3 mode, TRIG1 is automatically dedicated as a stop input.

When the stop input is closed to ground, any active sequence is aborted, and the move in progress is stopped using the last specified acceleration. When SSD is set to 1, SSB, SSF, SSK, and SSL are all forced to 0.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SSD1 = stop motion immediately
SSDØ = stop motion at end of current cycle

Command	Description
> SSD1	Set Alternate mode stop to 1
> MA	Set Aternate mode
> G	Execute the move
> S	Stops motion immediately

SSF TRIG1 Dedicated as Sequence Strobe

Version Z5

Type Set-Up
Syntax <a>SSFn
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also SSB, SSD, SN, SSJ, IM

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

When SSF is set to 1, TRIG1 is dedicated as a sequence select strobe in IM1 and IM2. When TRIG1 is dedicated as a sequence select strobe, it must be connected to ground, held there for the period of time specified by the SN command, and then released in order for the sequence specified by the sequence select inputs to be run. When SSF is set to 1, SSB, SSD, SSK, and SSL are all forced to 0.

SSG Clear/Save Command Buffer on Limit

Version Z5

Type Set-Up
Syntax <a>SSGn
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also SSH, LS

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The SSG command controls whether the command buffer is saved or discarded upon encountering an end-of-travel limit.

In most cases, it is desirable that upon activating an end-of-travel limit, all motion should cease until the problem causing the limit is rectified. For this reason the command buffer is normally discarded upon encountering a limit. By specifying SSG1 you are telling the Compumotor Plus to save the command buffer when a limit is encountered. When you do this the Compumotor Plus attempts to execute the commands which occur after the command which caused the motor to trip the limit.

CAUTION

Encountering an end-of-travel is a potentially dangerous situation if the limit was encountered due to a machine fault.

In the example below, if an end-of-travel limit becomes active during the move (refer to the G command), the move is aborted but the outputs will still be turned on.

Command	Description
> SSG1	Save buffer on limit
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> Ø11	Activate programmable outputs #1 and #2

SSH Clear/Save Command Buffer on Stop

Version Z5

Type Set-Up
Syntax <a>SSHn
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also SSG, S

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The SSH command enables or disables saving the command buffer upon receiving a stop command (either from the RS-232C port or from an input dedicated as a stop input).

SSHØ = Clears command buffer

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SSH1 = Saves command buffer

In Normal Operation (SSHØ) the Stop (S) command or a dedicated stop input causes any commands in the command buffer to be cleared. If you select the Save Command Buffer On Stop (SSH1) command, a remote stop input or Stop (S) command will only stop execution of a move in progress. It continues executing commands in the command buffer.

Command	Description
> SSHØ	Clears buffer on stop
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> L5Ø	Loops 50 times
> G	Execute the move (Go)
> T.5	Pauses for 500 msec
> N	Ends loop
> S	Stops execution

SSI Enable/Disable Interactive Mode

Version Z5

Type Set-Up
Syntax <a>SSIn
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also None

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The SSI command enables or disables the Compumotor Plus' interactive mode.

SSIØ enables the interactive mode. The device address must be set to one
SSI1 disables the interactive mode.

In the interactive mode the Compumotor Plus responds with a (>) when it understands a command and a (?) when it doesn't. Both responses are preceded with a line feed, carriage return sequence. In the interactive mode, the Compumotor Plus transmits a *READY and a (>) when energized or when a Reset (Z) command is executed.

When you define a loop or sequence, the Compumotor Plus will not send back a (>) until you complete the loop or sequence.

The SSI1 command disables the interactive mode. The Compumotor Plus does not respond with (>) or (?) when the interactive mode is disabled.

SSJ Enable/Disable Continuous Scan Mode

Version Z5

Type Set-Up
Syntax <a>SSJn
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also IM, INL, OSI, SN, SS, XD, XR, XT, XQ

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

When SSJI is enabled the Compumotor Plus continuously scans the inputs designated as sequence select inputs by the IM command and executes the sequence represented by the BCD number presented on the inputs. If Interrupted Run Mode (XQ) is active, then all the sequence input lines must go inactive prior to scanning the next sequence. A stop command discontinues continuous sequence scanning unless the OSI command switch has been turned on (OSI1).

When SSJØ is disabled the Compumotor Plus does not scan the BCD numbers for sequence execution. In this mode, you could execute sequences using the RS-232C interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SSK Enable Registration

Version Z5

Type Set-Up
Syntax <a>SSKb
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also IM, REG

Attributes
[X] Buffered
[ ] Device specific
[X] Saved independently
[X] Saved in sequences

The SSK command allows you to enable and disable registration mode. Before using registration you must define a valid registration move with the REG command and you must define an input to be used for a registration trigger signal using the IM command.

SSK1 enables registration
SSKØ disables registration

When enabled, registration can occur in any move. The SSK1 command supercedes the SSB1, SSD1, and SSF1 commands, which use trigger #1 for clearing a pause, stopping, and a sequence strobe, respectively.

SSL Resume Execution Enable

Version Z5

Type Set-Up
Syntax <a>SSLb
Units Enable/disable
Range 0 or 1
Default 0
Response None
See also S, C

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The SSL command allows you to use a Stop command as a Pause. Normally, a Stop command causes the current move to terminate, the current sequence to abort, if any, and the command buffer to be flushed. The SSL1 command instructs the Compumotor Plus to interrup motion and program execution upon a Stop command, and to continue with the program when a Continue is received.

SSL1 Resume execution of the sequence or commands in the buffer when a Continue (C) command is entered
SSLØ Disable resume feature

You can stop a program or a move using the Stop (S) or Remote Stop input. If SSL1 is enabled, the move will resume as soon as you enter the Continue (C) command.

Command	Description
> SSL1	Enable resume function
> XE1	Erase sequence 1
> XD1	Define sequence 1
> A1	Set acceleration to 1 rps^2
> V1	Set velocity to 1 rps
> D2ØØØØØ	Set distance to 200,000 steps
> G	Execute the move (Go)
> T2	Wait 2 seconds
> G	Execute the move (Go)
> XT	End sequence definition
> XR1	Run sequence 1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ST Shutdown the Amplifier

Version Z5

Type	Programming
Syntax	<a>STb
Units	Energize/Shutdown
Range	0 or 1
Default	0
Response	None
See also	OFF, ON

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Shutdown (ST1) command commands the drive to de-energize. The motor has no torque when the drive is de-energized, allowing it to be moved by hand if not restricted by the mechanism it is mounted to. The Compumotor Plus ignores move commands that you issue while the drive is de-energized

The STØ command energizes the drive. Once you enable the drive, you can execute moves. The absolute position counter is set to zero when you enter an STØ command so that an Absolute Position Report (PR) reports 0 position and absolute moves are made with respect to the current position.

This command alleviates motor heating.

Command	Description
> ST1	Shuts off current to the motor

STOP Stop Motion

Version Z5

Type	Motion
Syntax	<a>STOP
Units	None
Range	None
Default	None
Response	None
See also	S, SSH, SSL

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The STOP command is identical to the S command except that it is buffered and can be put into sequences. Refer to the S command for a description.

SV Save Set-Up and Sequences

Version Z5

Type	Programming
Syntax	<a>SV
Units	None
Range	None
Default	None
Response	None
See also	SAVE

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The SV command saves any parameters you have defined since the last save (SV). Values you have not changed, are not changed by the Compumotor Plus. When you define a sequence, you must issue an SV command to save the sequence into permanent memory.

The following parameters are restored to their default values by the RFS command:

CCA, CCP, CDB, CDM, CDG, CGS, CIM, CIG, CIL, CPE, CPG, CMR, CPM, CVG, CVM.

Command	Description
> SV	Save all the changes
> *	Save successful

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

T Time Delay

Version Z5

Type Programming
Syntax <a>T
Units Seconds
Range 0.00 - 53687.09
Default None
Response None
See also None

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command performs a timed wait for the number of seconds specified by its parameter. No further buffered commands are executed during this wait. You may specify times as small as 0.01 second.

You may use this command in Continuous Mode (MC) to generate variable move profiles.

Command	Description
> MN	Sets to Normal mode
> A5	Sets acceleration to 5 rps^2
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> T10	Pauses for 10 seconds
> G	Executes the move (Go)
> T5	Pauses for 5 seconds after the move ends
> G	Executes the move (Go)

TR Wait for Trigger

Version Z5

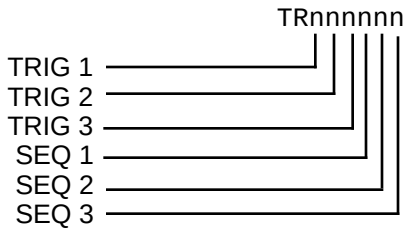
Type Programming
Syntax <a>TRb<bbbbb>
Units Off, on, or don't care
Range 0, 1 or X
Default None
Response None
See also IFTR, TS

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

Triggers are used to synchronize indexer operations with external events. They can be used to implement a handshaking function with other devices. Three symbols that are used for b are listed below.

Symbol	State
0	Open
1	Grounded
X	Don't care

You may omit trailing X's if you wish. Parameters not specified with the command are assumed to be X's. For example, TR11 is the same as TR11XXX.



All six triggers are available in all input modes.

When TR command is used in a buffer, the indexer will get to this command and wait until the input pattern is matched before going on to the next command.

If you omit trailing X's in the TR command, those trigger lines are ignored. The command TR100 is equivalent to TR100XXX.

Command	Description
> TR10XXXX	Wait for input #1 to be active and input #2 to be opened before going on to the next command. Inputs 3, 4, 5, & 6 are ignored.
> A10	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> G	Executes the move (Go)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TS Trigger Input Status

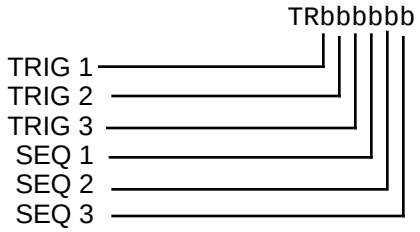
Version Z5

Type Status
Syntax <a>TS
Units None
Range None
Default None
Response <a>TS = *bbbbbb
See also IN, TR

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command retrieves the state of the trigger inputs. Response is in the form bbbbbb where b can be any of the following.

Symbol	State
0	Open
1	Grounded
X	Don't care



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TS command is useful for checking the status of the trigger inputs when it appears as though execution is being halted by a TR command. To make sure that your trigger pattern is met, you can check the TS command.

Command	Description
> 1TS	*1010000[cr] Trigger bits #1 and #3 are closed and Bits #2, #4 and #6 are opened

TUNE Self Tune

Version Z5

Type	Set-Up
Syntax	aTUNE:m,o
Units	m = move type, o = options
Range	m = 1 to 3, o = 1 to 4
Default	m = 1, o = 1
Response	1TUNE = *TUNING_BEGUN/*TUNING_COMPLETE
See also	GAINX, FILT

Attributes
[ ] Buffered
[X] Device specific
[X] Saved independently
[ ] Saved in sequences

The TUNE command initiates the self-tuning sequence in the Compumotor Plus. For additional information on self-tuning, consult *Chapter 4, Application Design*.

Note: Before using the TUNE command you should set the maximum position error to a relatively low value (for example, 1 revolution). Also, ensure the motor is firmly mounted, and the motor shaft is coupled to the load.

The TUNE command initiates a sequence of small fast moves and collects information on the system performance. The amplifier then shuts down, and the Compumotor Plus analyzes the data. The analysis takes approximately two minutes.

When the analysis is complete the system computes appropriate gains based on the desired performance based on your setting of parameter b (explained below). Once the gains are computed, the amplifier is re-energized with these new gains in place. The system waits for about five seconds to make sure that the resulting gains are stable. If you move the shaft during this time, an error is reported. If the resulting gains are for some reason unstable, the Compumotor Plus reports an error. When the Compumotor Plus determines that the gains are suitable, it reports the following.

```
>*TUNING_COMPLETE
```

The gains resulting from self-tuning form a control scheme called *pole-placement* which is distinct from PIDV control. At present there is no direct translation between the pole-placed gains of self-tuning and the user accessible PIDV gains. To switch between these control schemes, use the GAINX command. The gains resulting from self-tuning are typically much higher than the default PIDV gains. If you find that the resulting gains produce excessive audible noise in the motor, use the FILT command to remove this noise.

The command format of of this self-tuning command is:

```
TUNE:m,o
```

Parameter m determines the type of initial tuning move to use. If your system cannot move in both directions or there is known backlash in the system, use move 2 or move 3.

m=1	512 step (at 12800 steps/rev) back and forth move
m=2	512 step hops in CW direction
m=3	512 step hops in the CCW direction

Parameter o determines the desired closed-loop performance:

o=1	low overshoot response, low in-position stiffness
o=2	fast response, some overshoot, higher stiffness
o=3	faster response, more overshoot, very stiff
o=4	moderate response for large loads

The results of the self-tuning procedure can be saved to non-volatile memory with the SV command.

U Pause & Wait for Continue

Version Z5

Type	Programming
Syntax	<a>U
Units	None
Range	None
Default	None
Response	None
See also	C, PS, SSL

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This command causes the Compumotor Plus to complete the command in progress, then wait until it receives a Continue (C) to resume processing. Since the buffer is saved, the indexer continues to execute the program (at the point where it was interrupted). The indexer continues processing when it receives the C command. This command is typically used to stop a machine while it is unattended.

This following program fragment pauses at the point where the U command is entered. A Continue (C) command causes execution to resume at the point where it was paused. In this example, the loop stops at the end of a move, and resumes when the Compumotor Plus receives the C command. There may be a 10 second delay before motion resumes after the C command is executed, depending on when the Pause and Wait for Continue (U) command is completed.

Command	Description
> MN	Sets move to Normal mode
> A5	Sets acceleration to 5 rps^2
> V5	Sets velocity to 5 rps
> LØ	Loops indefinitely
> D25600	Sets distance to 25,600 steps
> G	Executes the move (G)
> T1Ø	Waits 10 seconds after the move
> N	Ends loop
> U	Halts execution until the Compumotor Plus receives the Continue command

V Velocity

Version Z5

Type Motion
Syntax <a>Vn
Units rps
Range 0.00 to 60.00
Default 1
Response *nnnnnnnn (where n is a number from 0 to 9)
See also A, C, D

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Velocity (V) command defines the maximum speed at which the motion will occur when given the Go (G) command.

The velocity specified may not be reached if the move is too short. In this case the move will be triangular instead of trapezoidal. In Continuous mode (MC) the indexer executes the next command in the buffer only after the motor has reached the set speed. In Normal mode (MN) the indexer waits for the move to be complete before executing the next command in the buffer.

Command	Description
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)

W1 Signed Binary Position Report

Version Z5

Type Status
Syntax aW1
Units None
Range None
Default None
Response cccc (where c is a single character (8 bits))
See also PR, W2, W3

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The W1 command returns the distance in steps the motor has moved since it last started motion. The reportback is a signed binary number with a range of ±2,147,483,648.

If the motor is moving, the reportback is the distance the motor has moved since starting motion. If the motor is not moving, it reports the distance of the previous move. The format of the response is a four byte response (cccc) that is interpreted as a 32-bit binary number. The information is transmitted in the order of high-byte to low-byte. The number must then be interpreted by the host device to give a decimal position in steps. The response is in 2's complement notation. Moves in the negative direction will report back negative number (bit 31 is set to 1).

The transmission is not preceded with an asterisk (*) like most numeric reportback commands in order to give the fastest possible response time. *Do not use the W1 command when multiple Compumotor Plus units are daisy-chained. Doing so will have unpredictable results.*

Interpreting Binary Position Reports

This form of position report (cccc), consisting of four bytes. The four bytes must be linked together to form a 32-bit binary number. Most terminals and communications programs for PCs expect to handle characters

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

rather than binary numbers and will not correctly display this kind of response. For example, a binary 7 will not display on a terminal, but will cause the terminal to beep.

The following table gives the weight of each byte:

Byte	Weight (decimal)	Weight (power of 2)
First	16,777,216	2^{24}
Second	65,536	2^{16}
Third	256	2^{8}
Fourth	1	2^{0}

To convert the reportback into a decimal, multiply each byte by the appropriate value in the table (i.e., the first byte by 16,777,216 etc...) and then add the decimal values for each to get the total.

Suppose you tell the motor to move 25,000 steps. You then issue the `W1` command. The response from the Compumotor Plus is interpreted as shown below. Note: only one of the bytes is displayable on most terminals.

Byte	Binary Value	Decimal Value		Decimal Weight	Decimal Weighted Value
First	00000000	0	x	16,777,216	0
Second	00000000	0	x	65,536	0
Third	01100001	97	x	256	24,832
Fourth	10101000	168	x	1	168
				Total	25,000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

W2 Hexadecimal Unsigned Position Report

Version Z5

Type Status
Syntax <a>W2
Units None
Range None
Default None
Response hhhhhhhh (where h is a hexadecimal number from 0 to F)
See also PR, W1, W3

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The immediate hexadecimal character position report back (normally used during motor motion) indicates the position in relation to the start of the current move. The format of the response is 8 hexadecimal characters. You must convert these characters into a usable format. This command does not report direction (i.e., it reports an unsigned number).

Digit	Base 16 Weight	Decimal Weight
h (MSD)	16^7	268,435,456
h	16^6	16,777,216
h	16^5	1,048,576
h	16^4	65,536
h	16^3	4,096
h	16^2	256
h	16^1	16
h (LSD)	16^0	1

The hexadecimal number h, can be one of the following values:

Decimal Value	Hexadecimal Value (h)	Decimal Value	Hexadecimal Value (h)
Ø	Ø	8	8
1	1	9	9
2	2	1Ø	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

To convert a hexadecimal number to a decimal number, multiply each digit by its decimal weight and add the value of each digit. For example, if you tell the motor to move 25,000 steps and issue the W2 command the Compumotor Plus responds with 000061A8. To convert this hexadecimal number to decimal:

Digit		Decimal Weight	Decimal Weighted Value
0	x	268,435,456	0
0	x	16,777,216	0
0	x	1,048,576	0
0	x	65,536	0
6	x	4,096	24,576
1	x	256	256
A	x	16	10
8	x	1	8
		Total	25,000

Command	Description
> 1W2 *ØØØØFAØ4	A distance of FA04 (64,004 steps) has occurred since the start of the last move

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

W3 Hexadecimal Signed Position Report

Version Z5

Type	Status
Syntax	aW3
Units	None
Range	None
Default	None
Response	*hhhhhhhh (where h is a hexadecimal number from 0 to F)
See also	W1, W2, PR

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command provides a position report in signed hexadecimal format while the motor is moving. The report indicates position in steps relative to the start of the current move.

This form of position report (*nnnnnnnn) is generated by the W3 command. It consists of an asterisk followed by eight hexadecimal characters; 0 through 9 and A through F. The position report is followed by a carriage return.

If the response is positive use the conversion process explained above for the W2 command. You can tell is the response is positive by looking at the first digit of the reponse after the asterisk. If it is 8 or greater (8, 9, A, B, C, D, E, F) the response is negative.

If the response is negative you must be careful when converting it to decimal.

The Standard Approach

① Convert the number to decimal using the method described for the W2 command.
② Subtract 4,294,967,296 from the number.

The Binary Approach

① Convert the hexadecimal response to binary form.
② Complement the binary number.
③ Add 1 to the binary result.
④ Convert the binary result to decimal value with a minus sign placed ahead of the decimal value.

The Computer Approach

Subtract the hexadecimal number from 16^8 (2^{32} or 4,294,967,296)

The Easy Way

① Ignore all the leading F's, then convert the hexadecimal number to decimal.
② Subtract the next largest power of 16.

Example

The indexer responds to W3 as follows: *FFFF9E58

① Chop off the F's: 9E58 hex = 40,536
② Subtract from 16^4 10000 hex = 65,536
③ Subtraction Result = -25,000

XC Sequence Checksum Report

Version Z5

Type	Status
Syntax	aXC
Units	None
Range	None
Default	None
Response	*nnnnn (where n is a number from 0 to 9)
See also	XD, XE

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command reports the nonvolatile memory checksum. After the unit has been programmed, the response can be used for system error checking. The number reported does not indicate the number of bytes programmed. This response is designed to be used for comparison. As long as the sequences are not reprogrammmed, the checksum response should always be the same.

Command	**Description**
> 1XC	*ØØ149

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XD Sequence Definition

Version Z5

Type Programming
Syntax <a>XDn
Units Sequence number
Range 1 to 40
Default None
Response None
See also SAVE, XE, XR, XT, XU

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command begins sequence definition for a specific sequence. All the commands between the XD command and the Sequence Termination (XT) command will be defined as a sequence. If a sequence you are trying to define already exists, you must erase that sequence before defining it. ***Immediate commands cannot be entered into a sequence***. Be sure to save your sequence with the SAVE command.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> MN	Sets to Normal mode
> A10	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining sequence #1
> XR1	Execute sequence #1

XDIR Sequence Directory Report

Version Z5

Type Status
Syntax aXDIR
Units None
Range None
Default None
Response None
See also XBS, XT, XE, XD

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Compumotor Plus lists the sequence number and amount of memory used for all sequences defined when this command is issued. The response is in the following format:

If no sequences are defined:

```
> 1XDIR        *NO_SEQUENCES_DEFINED
>
```

If sequences are defined:

```
> 1XDIR
*SEQUENCE_A_USES_X_BYTES
*SEQUENCE_B_USES_Y_BYTES
>
```

Command	Description
> 1XDIR *SEQUENCE_5_USES_251_BYTES *SEQUENCE_39_USES_45_BYTES >	Reports the number of bytes used by sequences 5 and 39.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XE Erase Sequence

Version Z5

Type Programming
Syntax <a>XEn
Units Sequence number
Range 1 to 40
Default None
Response None
See also XD, XR, XT

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue this command. Caution should be used when executing this command as the sequence is irretrievable.

Command	Description
> XE1	Deletes sequence #1

XFK Set Fault or Kill Sequence

Version Z5

Type Motion
Syntax <a>XFKn
Units Sequence number
Range 0 to 39
Default 0
Response *n (where n is a number from 0 to 39)
See also RSE, XR, K

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command selects the sequence that will be executed if a fault condition occurs. No sequence is executed if you select sequence zero. This command is useful if you want to reset an output in case a fault condition exists. You can also use this command to send a message through the RS-232C interface when a fault or kill condition exists. The fault conditions are listed below.

Response	Description	Shown On Front Panel
00	No Errors	Yes
16	Non-commanded drive off	Yes
20	Maximun position error exceeded	Yes
22	Maximun average current exceeded	Yes
23	Drive enable plug not inserted	Yes
30	Battery Backed RAM Checksum error	Yes
60	RS-232C commanded shutdown (Using the ST1 or OFF command)	No
61	Idexer incoming pulses (non-indexer version only)	Yes
66	User defined fault	Yes
88	Microprocessor fault	Yes

Command	Description
> XFK5	Execute sequence #5 when fault or kill condition exists
> XE5	Erase sequence #5
> XD5	Define sequence #5
1"FAULT_OR_KILL	Send the message
> XT	End sequence definition

XG Goto Sequence

Version Z5

Type Programming
Syntax <a>XGn
Units Sequence number
Range 1 to 39
Default None
Response None
See also XR

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command causes your program to jump to a designated sequence for execution. Once you jump to a sequence using the XG command, you cannot return to the sequence from which the XG originated (unless another XG command is executed). To jump to a sequence and return (GOSUB operation), you must use the XR command. There are no limitations on the number of XG commands as there is no nesting involved.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A2	Sets acceleration to 2 rps^2
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XG5	Go to sequence #5
> XT	End defining sequence #1
> XE5	Erase sequence #5
> XD5	Define sequence #5
> 1PR	Absolute position report
> XT	End sequence #5 definition
> XR1	Execute sequence #1

XQ Set Sequence Interrupted-Run Mode

Version Z5

Type Set-Up
Syntax <a>XQ
Units Mode
Range 0 or 1
Default 0
Response None
See also TR, V, IM, SSJ

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The XQ command allows you to specify that the programmable inputs must all be taken to their inactive state before another sequence may be run. When the unit is in the continuous run (SSJ) mode.

Set interrupted run mode (1)
Clear interrupted run mode (Ø)

This command can only be used in conjunction with continuous mode (SSJ), and external sequence select lines. If XQ1 is executed, the Compumotor Plus will not accept a sequence selected from the inputs, until all sequence select lines have been brought inactive. After all lines have simultaneously been brought inactive, the Compumotor Plus will then read the sequence select lines and execute the sequence whose number appears there. This interrupted mode will continue until an XQØ command is executed. You may use S or K commands to stop sequence execution.

Command	Description
> XE4Ø	Erase sequence #40
> XD4Ø	Define sequence #40
> LD3	Disable CW & CCW limits
> SSJ1	Set to Continuous mode
> XQ1	Set to Interrupted mode
> XT	End sequence #100

XR Run a Sequence

Version Z5

Type Programming
Syntax <a>XRn
Units Sequence number
Range 1 to 39
Default None
Response None
See also XD, XE, XT, XRP

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command is used to execute sequence n. Once sequence n finishes, the program resumes execution with the command following the XR command. This command is similar to a subroutine or GOSUB statement in Basic. The maximum number of nested XR commands is 16.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining sequence #1
> XR1	Execute sequence #1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XRP Sequence Run With Pause

Version Z5

Type	Programming
Syntax	<a>XRPn
Units	Sequence number
Range	1 to 40
Default	None
Response	None
See also	C, XD, XE, XR, XT

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

The Pause and Run Sequence (XRP) command pauses execution of the current command buffer or sequence and waits for a Continue (C) command. It then excecutes the sequence specified.

The pause condition is asserted only if the sequence is valid. This command is normally used interactively because it allows you to execute a sequence without the delay of buffering that sequence. An XRP command can be used within one sequence to start execution of another sequence (in this respect an XRP acts like a GOSUB). The maximum number of nested XRP commands is 16.

Command	Description
> XE5	Erases sequence #5
> XD5	Defines sequence #5
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D1ØØØØ	Sets distance to 10,000 steps
> G	Execute the move (Go)
> XT	Ends defining sequence #5
> XRP5	Runs sequence #5 with a pause
> C	Compumotor Plus executes sequence #5

XSD Sequence Status Definition Report

Version Z5

Type	Status
Syntax	aXSD
Units	None
Range	None
Default	None
Response	aXSD = *n (where n is a number from 0 to 3)
See also	None

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

This command reports the status of previous sequence definition (refer to the XD and XT commands). The response is in the form *n[cr]. The valid values and descriptions of n are shown below:

Ø = Download O.K.
1 = A sequence already exists with the number you specified
2 = Out of memory. The sequence buffer is full
3 = EEPROM write

The XSD command is useful for verifying that the last sequence definition attempt was successful. To retain the sequence, you must save it with the Save (SV) command.

Command	Response
> 1XSD *1	**1**

XSR Sequence Run Status

Version Z5

Type	Status
Syntax	aXSR
Units	None
Range	None
Default	None
Response	aXSR = *n (where n is a number from 0 to 2)
See also	XR, XRP

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The XSR command allows you to check whether the last sequence issued was started successfully. The valid values and descriptions for n are shown below.

Ø = The last attempt to run the sequence was successful
1 = The last attempt to run the sequence was unsuccessful
2 = An invalid sequence number was requested

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command	Description
> XR2	Runs sequence #2
> 1XSR	Request sequence status
*Ø[cr]	The sequence started OK

XSS Sequence Status

Version Z5

Type Status
Syntax aXSSn
Units Sequence number
Range None
Default None
Response *n (where n is a number from 0 to 2 as listed below)
See also XD, XE, XT

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command reports whether the sequence is empty or programmed. The valid values and descriptions of x are shown below:

Ø = Empty
1 = Bad Checksum
2 = O.K.
This command is useful to see if the particular sequence exist and if that portion of memory has been corrupted or not.

Command	Description
> 1XSS1	*Ø [cr] Nothing programmed in that sequence

XST Set Sequence Single-Step Mode

Version Z5

Type Set-Up
Syntax aXSTn
Units Mode
Range 0 or 1
Default 0
Response *STEP_MODE_ACTIVE or *STEP_MODE_INACTIVE
See also #, XR, XTR

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command puts the Compumotor Plus into Sequence Step mode. This command can only be used with the Step (#) command. When you run a sequence with the Sequence Step mode active, every time you issue a Step (#) command, the controller executes n commands in the sequence buffer.

XST1: Sequence Step mode active
XSTØ: Sequence Step mode inactive

Since you need to send a # command over the RS-232C interface, this command cannot be run in stand alone mode. You must be executing the sequence in RS-232C mode. You must enter a delimiter after the Step (#) command to execute the command. If you are in the Trace (XTR) mode, the controller will display n commands every time you enter the #n command. This command is useful for troubleshooting your program to see where you are in the program and what takes place with each command. You can use the Kill (K) command to abort the sequence execution.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A5	Sets acceleration to 5 rps^2
> V2	Sets velocity to 2 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining Sequence #1
> XST1	Enable Single Step mode
> 1XTR1	Enable Trace mode
> XR1	Execute sequence #1
> #	Execute the 1st command
*SEQUENCE_001_COMMAND_A5	Displays the 1st command executed
> #	Execute the 2nd command
*SEQUENCE_001_COMMAND_V2	Displays the 2nd command executed
> #	Execute the 3rd command
*SEQUENCE_001_COMMAND_D10000	Displays the 3rd command executed
> #	Execute the 4th command
*SEQUENCE_001_COMMAND_G	Displays the 4th command executed motor should have moved 10,000 steps
> #	Execute the 5th command
SEQUENCE_001_COMMAND_XT	Displays the last command executed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XT Terminate (End) Sequence

Version Z5

Type	Programming
Syntax	<a>XT
Units	None
Range	None
Default	None
Response	None
See also	XD, XE, XR

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The XT command is a sequence terminator. This command flags the end of the sequence currently being defined. Sequence definition is not complete until this command is issued.

Command	Description
> XD1	Define sequence #1
> MN	Set to Normal mode
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> G	Executes the move (Go)
> XT	End sequence definition

XTR Set Trace Mode

Version Z5

Type	Set-Up
Syntax	aXTRn
Units	Mode
Range	0 or 1
Default	0
Response	*TRACE_MODE_ACTIVE or *TRACE_MODE_INACTIVE
See also	XR, XST

Attributes
[X] Buffered
[ ] Device specific
[ ] Saved independently
[X] Saved in sequences

This command transmits the command that is being executed from the Compumotor Plus to the host via RS-232C interface. This command only works if you are running a sequence.

XTR1: Enables Trace mode
XTRØ: Disables Trace mode

Enabling trace mode transmits the commands and the sequence number being executed. If you have a Loop (L), REPEAT, or WHILE command in a sequence, it will also display the iteration count.

This command is useful if the user wishes to see where you are in the program as the program is being executed.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> L2	Loop 2 times
> G	Executes the move (Go)
> N	End Loop
> XT	End defining sequence #1
> 1XTR1	Enable Trace mode
> XR1	Execute sequence #1

After turning on the Trace mode, as you run the sequence (XR1), the controller will display the current command being executed. A sample of Trace mode output is shown below:

```
*SEQUENCE_ØØ1_COMMAND_A1Ø
*SEQUENCE_ØØ1_COMMAND_V5
*SEQUENCE_ØØ1_COMMAND_D25ØØØ
*SEQUENCE_ØØ1_COMMAND_L2
*SEQUENCE_ØØ1_COMMAND_G_LOOP_COUNT_ØØ1
*SEQUENCE_ØØ1_COMMAND_N_LOOP_COUNT_ØØ1
*SEQUENCE_ØØ1_COMMAND_G_LOOP_COUNT_ØØ2
*SEQUENCE_ØØ1_COMMAND_N_LOOP_COUNT_ØØ2
*SEQUENCE_ØØ1_COMMAND_XT
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XU Upload Sequence

Version Z5

Type	Programming
Syntax	aXUn
Units	Sequence number
Range	1 to 40
Default	None
Response	*x (where x is the contents of sequence n)
See also	XD, XE, XT

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The XU command sends all the characters in the requested sequence out the RS-232C port. The response is preceded with an asterisk (*) and terminated by an extra carriage return. All command delimiters in the sequence (spaces and carriage returns) are converted to underscores. If the sequence is empty, *_ is sent.

Command	Description
> 1XD1 A10 V10 D5000 G XT	Define sequence #1
> 1XU1	Upload sequence #1
1XD1_A10_V10_D5000_G	Response from Compumotor Plus
>	

Y Stop Loop

Version Z5

Type	Programming
Syntax	<a>Y
Units	None
Range	None
Default	None
Response	None
See also	L, LRD, N

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Stop Loop (Y) command exits a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the Compumotor Plus reaches the last command of the current loop. At that time, the Compumotor Plus executes the command following the End Loop (N) command. You cannot restart the command loop unless you enter the entire command structure, including the Loop (L) and End Loop (N) commands, or restart the sequence.

Command	Description
> L	Begin infinite loop
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25000	Set distance to 25,000 steps
> T2	Wait 2 seconds
> G	Execute the move (Go)
> N	End loop
> Y	Stop loop upon receipt of the command

Z Reset

Version Z5

Type	Programming
Syntax	<a>Z
Units	None
Range	None
Default	None
Response	None
See also	K, S

Attributes
[ ] Buffered
[ ] Device specific
[ ] Saved independently
[ ] Saved in sequences

The Reset (Z) command is equivalent to cycling power to the Compumotor Plus. This command returns all internal settings to their power-up values. It clears the command buffer and sets all position counters to zero.

When you use the Reset command, the Compumotor Plus is busy for 2.5 seconds and ignores all commands during that time.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑥

Hardware Reference

The information in this chapter will enable you to:

- ❑ Use this chapter as a quick reference tool for most system specifications
- ❑ Use this chapter as a quick reference tool for most switch settings
- ❑ Use this chapter as a quick reference tool for proper I/O connections

Environmental Specifications

Parameter	Range
Operating temperature	0 to 40°C (32 to 104°F) ambient
Drive temperature	Maximum heatsink temperature 75°C (162°F)
Motor temperature	Maximum motor case temperature 100°C (212°F)
Storage temperature	-40° to 85°C (-40 to 185°F)
Humidity	0 to 95% non-condensing

Drive Electrical Specifications

Input power

Parameter	Low Power	High Power
Voltage	110-130VAC*	100-130VAC, single phase
Frequency	47-66 Hz	47-66 Hz
Current	8.5 amps max continuous (RMS)	8.5 amps max continuous (RMS)

110VAC-130VAC is supplied to the transformer included with the low power Compumotor Plus.

Output Power

Parameter	Low Power	High Power
Voltage	50VDC peak	170VDC peak
Frequency	20 kHZ (PWM)	20 kHZ (PWM)
Current Continuous	7 amps continuous	7 amps continuous
Current Peak	8.5 amps peak	8.5 amps peak

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connector Summary

Motor & Power Connector, Low-Power Drive

Signal	Name	In/Out	Type	Current	Voltage	Frequency	Wire Color
1	T1	Input	Power	4A	36VAC	50/60 HZ	Brown
2	T2	Input	Power	4A	36VAC	50/60 HZ	Blue
3	GND	Input	Power	4A	Earth	N/A	Grn/yel
4	A+	Output	Motor	8A	50VAC	N/A	Red
5	A-	Output	Motor	8A	50VAC	N/A	Black
6	GND	Output	Motor	8A	0VAC	N/A	Shield
7	B+	Output	Motor	8A	50VAC	N/A	White
8	B-	Output	Motor	8A	50VAC	N/A	Green

Motor & Power Connector, High-Power Drive

Signal	Name	In/Out	Type	Current	Voltage	Timing	Wire Color
1	LINE	Input	Power	6A	115VAC	50/60 HZ	Black
2	NEUT	Input	Power	6A	115VAC	50/60 HZ	White
3	GND	Input	Power	6A	Earth	N/A	Green
4	A+	Output	Motor	10A	150VDC	N/A	Red
5	A-	Output	Motor	10A	150VDC	N/A	Black
6	GND	Output	Motor	10A	Earth	N/A	Shield
7	B+	Output	Motor	10A	150VDC	N/A	White
8	B-	Output	Motor	10A	150VDC	N/A	Green

Control Connector, Low Power & High Power Drives

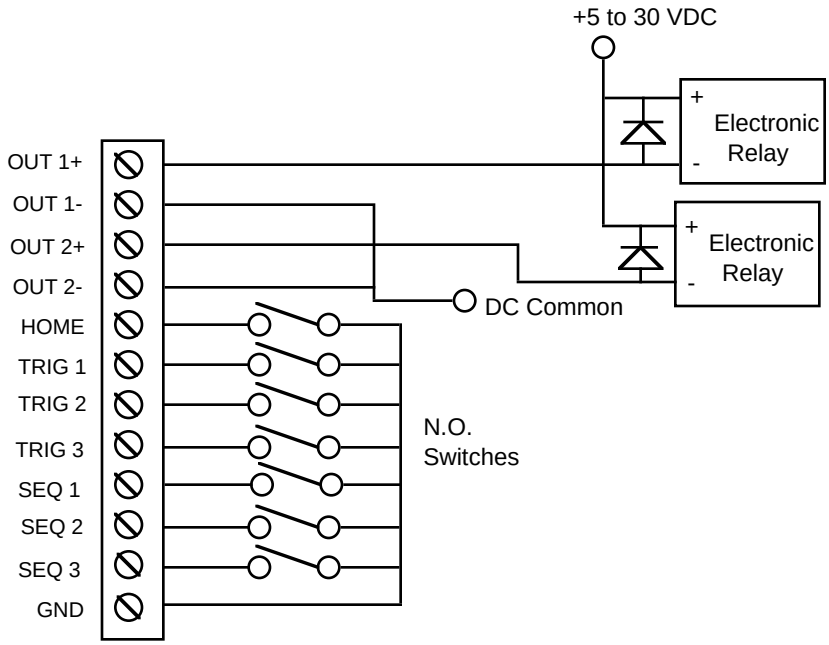
Signal	Name	In/Out	Type	Current (Min)	Voltage (Max)	Timing
1	Rx	Input	RS-232C	-	±15V	9600 Baud (adjustable)
2	Tx	Output	RS-232C	-	±15V	9600 Baud (adjustable)
3	GND	-	GROUND	-	Earth	-
4	ENABLE	Input	SRC	12 mA	12V	DC
5	CW LIMIT	Input	SRC	12 mA	12V	DC
6	CCW LIMIT	Input	SRC	12 mA	12V	DC
7	RESET	Input	SRC	12 mA	12V	DC
8	GND	Input	Ground	-	-	-
9	COMMAND 1+	Input (Step+)				
10	COMMAND 1-	Input (Step -)	Non-indexing			
11	COMMAND 2+	Input (Dir +)	Version Only			
2	COMMAND 2-	Input (Dir --)				
13	MONITOR +	Output	Analog	10mA	±10V	
14	MONITOR -	Output				
15	FAULT +	Output	OC	50mA	30V	DC
16	FAULT -	Output	OE			

I/O Connector, Low and High Power Drives

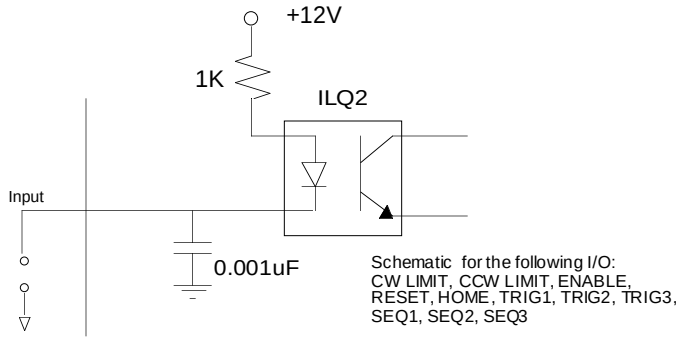
Signal	Name	In/Out	Type	Current	Voltage
1	OUT 1+	Output	OC	50 mA (Max)	30V
2	OUT 1-	Output	OE		
3	OUT 2+	Output	OC	50 mA (Max)	30V
4	OUT 2-	Output	OE		
5	HOME	Input	SRC	12 mA (Min)	12V
6	TRIG 1	Input	SRC	12 mA (Min)	12V
7	TRIG 2	Input	SRC	12 mA (Min)	12V
8	TRIG 3	Input	SRC	12 mA (Min)	12V
9	SEQ 1	Input	SRC	12 mA (Min)	12V
10	SEQ 2	Input	SRC	12 mA (Min)	12V
11	SEQ 3	Input	SRC	12 mA (Min)	12V
12	GND	Input	GND		

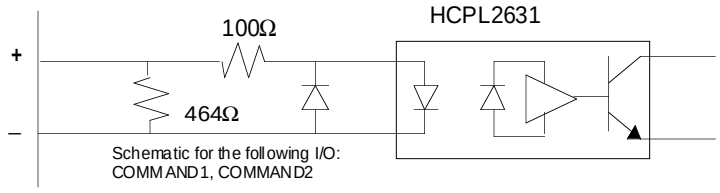
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

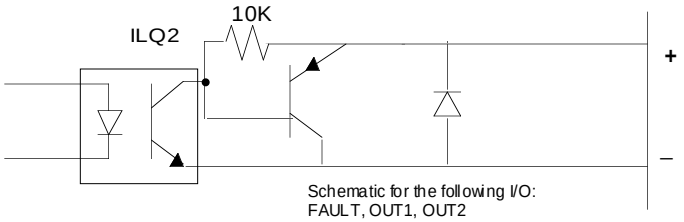
I/O Diagram



I/O Schematics







Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Position sensor: Accepts position information from the motor's built-in resolver. The cable between the motor and drive is pre-wired at the factory.

> ***CAUTION***
>
> **The position sensor connector on the low and high power drives are not normally accessed by user. Un-isolated hazardous voltages are present in high power units. Various length cables are available (consult the factory).**

Input power: Input power for the drive-high-power: 110VAC, low-power drive: 36VAC. The low-power drive is provided with a pre-wired 110VAC input, 36VAC output transformer.

T1	One phase of the input power (Brown)
T2	Second phase of the input power (Blue)
GND	Earth ground (Green/Yellow)

Motor connections: Output power to the motor. Up to 150VDC for the high-power drive and 50VDC (pulsed) for the low power drive.

A+	Plus side of motor phase A (Red)
A-	Minus side of motor phase A (Black)
GND	Shield ground
B+	Plus side of motor phase B (White)
B-	Minus side of motor phase B (Green)

RS-232C Connections

The RS-232C transmits and receives connections. It accepts standard EIA RS-232C signals from +15V to -15V. The baud rate is selectable. Eight data bits, no parity, and one stop bit are fixed parameters. The unit is optically isolated.

Tx, Rx and Gnd — The RS-232 connections are labeled Tx for transmit, Rx for receive and Gnd for ground.

Discrete Control Connections

These input and output signals control various dedicated functions of the Compumotor Plus as explained below. The unit is optically isolated.

Enable	Enable Input. Enables the drive. Must be connected to isolated signal ground for drive operation.
CW LIMIT	CW end-of-travel limit. This input must be connected to ground to be inactive.
CCW LIMIT	CCW end-of-travel limit. This input must be connected to ground to be inactive.
RESET	When the reset line is connected to ground, the reset line of the microprocessor is pulled low and the system resets.
GND	Isolated signal ground or common
COMMAND 1+	Step Input: The step pulse input to the step/direction Compumotor Plus Drive. Not used in the Indexer version.
COMMAND 1-	Step Return: The step pulse input return for the Step/Direction Compumotor Plus Drive. Not used in the Indexer version.
COMMAND 2+	Direction Input: The direction signal input for the Step/Direction Compumotor Plus Drive. Not used in the Indexer version.
COMMAND 2+	Direction Input: The direction signal input for the Step/Direction Compumotor Plus Drive. Not used in the Indexer version.
MONITOR +	Bipolar analog output (+10V) providing analog velocity report. -10V equates to -50 revolutions per second, 0V equates to zero speed, and +10V equates to +50 revolutions per second.
MONITOR -	Isolated ground for monitor output.
FAULT+	Fault indication output plus. Turns off (no current) to indicate fault condition. (Collector of optically isolated transistor.)
FAULT-	Fault indication output minus. (Emitter of optically isolated transistor.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Discrete Input/Output connections

Inputs and outputs that affect program control. They are optically isolated.

OUT 1+	General purpose output #1 plus. Controlled by O and IO command (collector of optically isolated transistor).
OUT 1-	General purpose output #1 minus (emitter of optically isolated transistor).
OUT 2+	General purpose output #2 plus. Controlled by O and IO command (collector of optically isolated transistor).
OUT 2-	General purpose output #2 minus (emitter of optically isolated transistor).
HOME	Home input. Indicates home position. Used in conjunction with the Go Home command (GH).
TRIG 1	General purpose input #1. Used in conjunction with the trigger commands (TR, TS) and sequence inputs in IM2.
TRIG 2	General purpose input #2. Used in conjunction with the trigger commands (TR, TS) and sequence inputs in IM2.
TRIG 3	General purpose input #3. Used in conjunction with the trigger commands (TR, TS) and sequence inputs in IM2.
SEQ 1	Sequence control input. Commands a specific sequence number to be executed. Refer to the Execute (X) commands.
SEQ 2	Sequence control input. Commands a specific sequence number to be executed. Refer to the Execute (X) commands.
SEQ 3	Sequence control input. Commands a specific sequence number to be executed. Refer to the Execute (X) commands.
GND	Logic ground.

Input Output Specifications Quick Reference

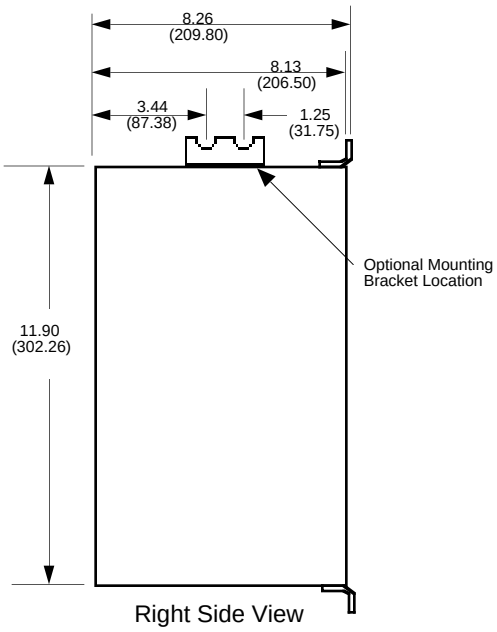
The types of Compumotor Plus inputs and outputs are listed in the table below.

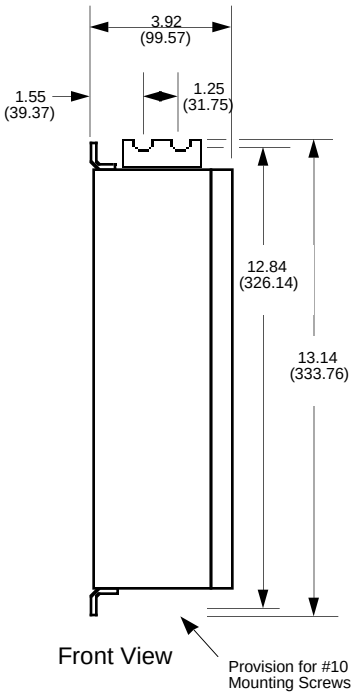
Type	Description
BLNCD	Balanced input. Requires both a plus and minus to activate
GROUND	Isolated ground for logic signals
RS-232C	Standard RS-232C I/O. Optically isolated.
SNK	Sinking input. Optically isolated, requires a ground to activate
OC	Open collector output. Optically isolated
OE	Open emitter output. Optically isolated. Used with open collector output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Dimensional Drawings

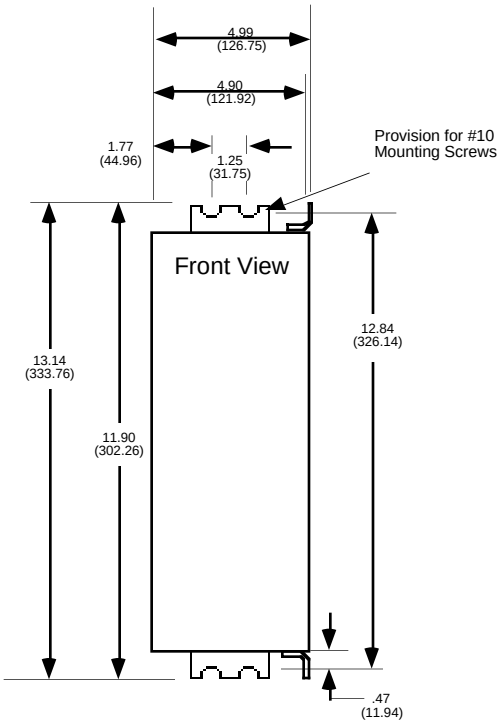
CPL, CPLX

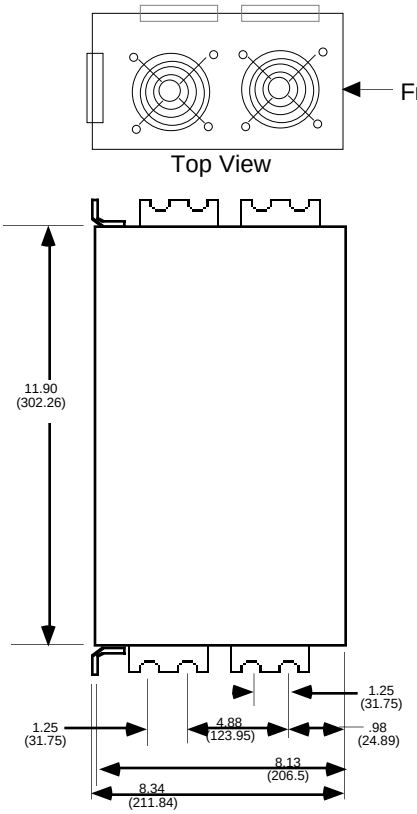




Dimension	Low Power
Drive Height	11.9 (302.3)
Drive Width	3.92 (99.6)
Drive Depth	8.13 (206.5)
Weight	11.5 lbs (5.2 Kg)

CPH & CPHX



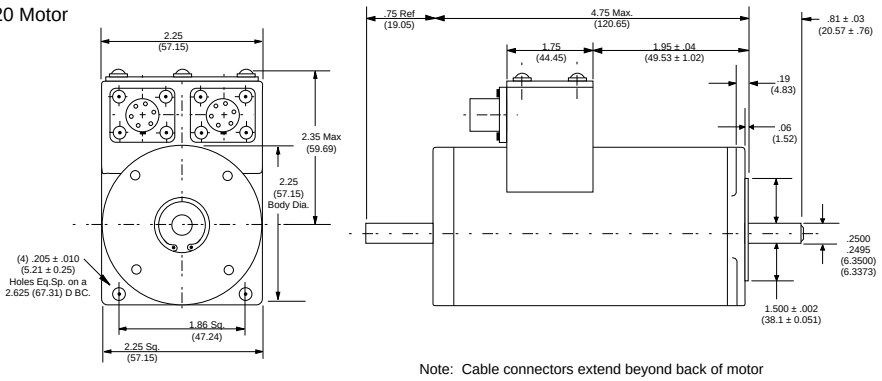


Dimension	High Power
Drive Height	11.9 (302.3)
Drive Width	4.787 (121.6)
Drive Depth	8.13 (206.5)
Weight	15.2 lbs (16.9 Kg)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

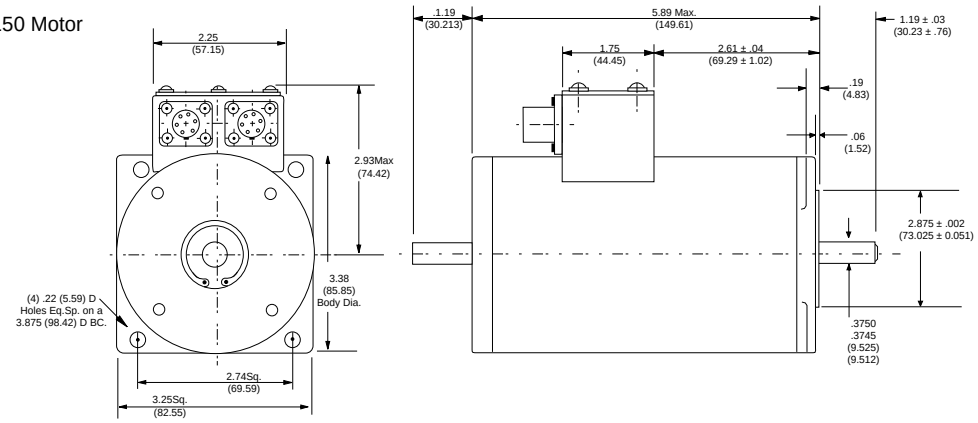
Motor Dimensions

CP57-120 Motor

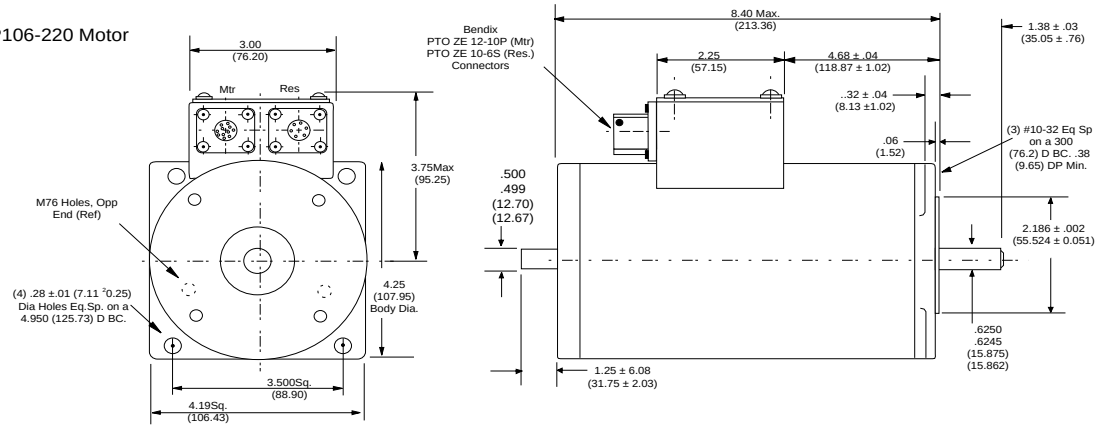


Note: Cable connectors extend beyond back of motor

CP83-150 Motor



CP106-220 Motor



Motor Specifications

Specification	CPL57-120	CPL83-150	CPH83-150	CPH106-210
Torque (continuous)	130	370	370	1100 in-oz
Torque (peak)	160	420	420	1400 in-oz
Max continuous speed	3600	3600	3600	3600 rpm
Rotor Inertia	0.0042	0.0239	O.0239	0.1393 oz-in-sec^2

System Specifications

Accuracy & Repeatability

Description	Value
Repeatability	± 2 arc min. (0.0334°)
Accuracy	± 12 arc min. (0.200°) Bi-directional, loaded at 80% of total torque. (With proper tuning)

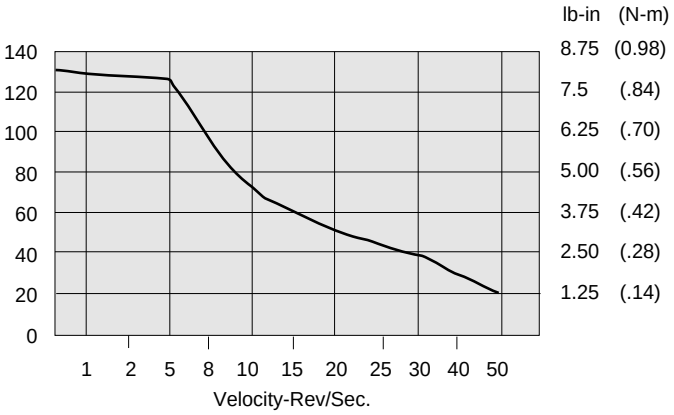
Physical

Description	CPL57-120	CPL83-150	CPH83-150	CPH106-210
Motor Weight	3.2	7.7	7.7	21.7
Shipping Weight	20.0	25.0	25.0	45.0

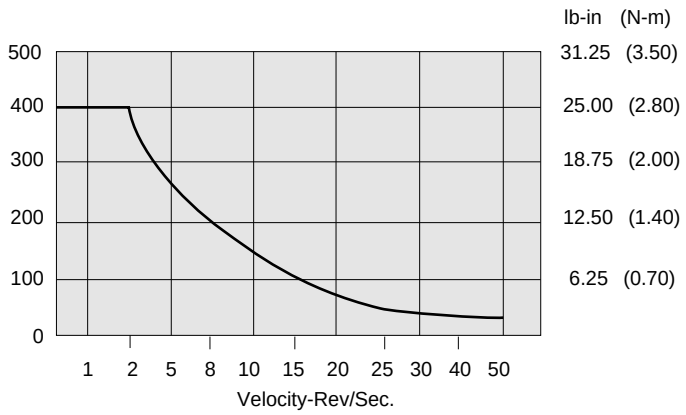
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Torque/Speed Curves

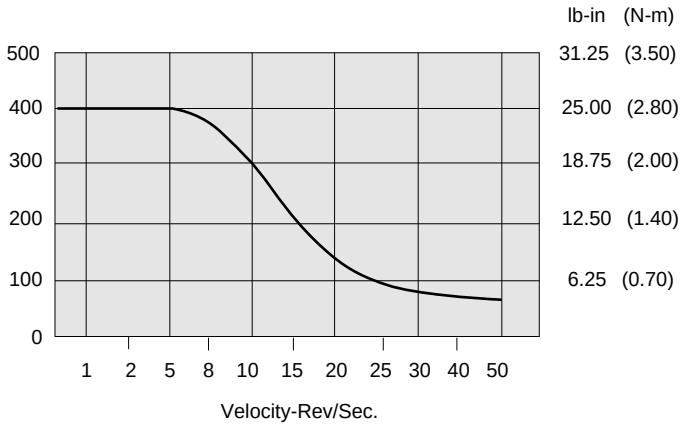
CPL57-120
or
CPLX57-120



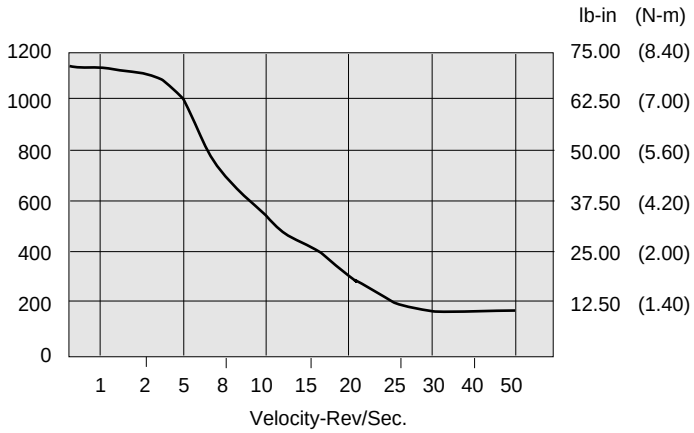
CPL83-150
or
CPLX83-150



CPH83-150
or
CPHX83-150



CPH106-220
or
CPHX106-220



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑦

Maintenance & Troubleshooting

The information in this chapter will enable you to:

- ❑ Isolate and resolve system hardware and software problems

Troubleshooting

This section discusses methods to identify, isolate, and resolve problems that may occur in the use of your Compumotor Plus Drive.

Problem Isolation

When your system does not function properly (or as you expect it to operate), the first thing that you must do is identify and isolate the problem. When you accomplish this, you can effectively begin to resolve the problem.

The first step is to isolate each system component and ensure that each component functions properly when it is run independently. You may have to dismantle your system and put it back together piece by piece to detect the problem. If you have replacement or additional units available, you may want to use them to replace existing components in your system to help identify the source of the problem.

Try to determine if the problem is mechanical, electrical, or software-related. Can you repeat or re-create the problem? Do not attempt to make quick rationalizations about problems. Random events may appear to be related, but they are not necessarily contributing factors to your problem. You must carefully investigate and decipher the events that occur before the subsequent system problem.

You may be experiencing more than one problem. You must solve one problem at a time. Log (document) all testing and problem isolation procedures. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

Once you have isolated the problem, take the necessary steps to resolve it. Refer to the problem solutions contained in this chapter. If your system's problem persist, contact Parker Compumotor's Applications Department.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Input/Output and Discrete Control Problems

This section describes how to solve problems relating to the use of the input and output lines on the Compumotor Plus.

Limits, Homing, Triggers & Sequence Selection

If you are having problems using the Trigger (**TR**), Home (**GH**), CW, CCW, or Sequence Select inputs, you must first check your wiring for proper installation. Use a multimeter to verify proper connection of the switches and inputs. If the hardware connections appear to be correctly made, use the Input Status (**IS**) command to monitor the state of each input manually and see if the Compumotor Plus recognizes the input change. You do this by changing the input state manually and issue the Input Status (**IS**) command. The **IS** command reports the hardware status of the Compumotor Plus inputs.

Remote Sequencing (BCD Inputs)

Start by checking your wiring as described in the section above for finding problems with limits, homing, triggers and sequence selection.

Ensure that your BCD input pattern is correct. Check *Chapter 6, Hardware Reference* for the Sequence Select Table. If it is not, the wrong sequence will be called. If you have a problem running a sequence from the remote input, try running the sequence using the **XR** command before attempting to run it using BCD input.

You may also put the indexer into Trace mode using the **XTR** command. You must have an RS-232C terminal or terminal emulator to use Trace mode. In Trace mode the indexer reports the current sequence and command being executed over the RS-232C port.

Motor Control Problems

The following section describes how to address problems with motor operation.

Motor Fails to Run at High Speeds

The motor may not produce enough force to move your load at the velocities you require. Check the speed/torque curve and make sure you are using the motor in the proper range.

Motor is Jerky or Weak

Check that there are no mechanical problems at the load causing a variable loading condition. Disconnect the motor from the load and run it without a load connected.

You can determine if the motor is developing its full holding torque by using a torque wrench on the shaft of the motor to measure the motor's torque capability.

Check to see if the resolver cable is connected properly. A damaged resolver cable may result in erratic motion.

Electrical Noise

For detailde information on reducing electrical noise in your system, refer to the current Compumotor catalog.

Diagnostic Codes

The following table lists all of the diagnostic codes which appear in the two digit status display on the front of the Compumotor Plus and that can be reported with the **RSE** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error Code	Description	Possible Solutions
00	No Errors	
20	Actual position lags commanded position	Check for excessive load on the motor. Verify the maximum following error value is appropriate. Use the **CPE** command to set the following error maximum value. Allow for a larger following error if appropriate.
22	The maximum average current has been exceeded	Check the load to ensure the torque required is less than the torque available from the motor at the speed required. Verify that the average current setting is appropriate. Set the average current with the CCA command.
23	The drive enable input is inactive	This normally occurs when the driven enable plug is missing. ENABLE (Pin 4) should be grounded.
30	EEPROM Checksum error	The Compumotor Plus calculates a checksum on reset and when power is cycled. When the new checksum does not match the previously calculated checksum an Error 30 is issued. Error 30 occurs in the following situations: The drive is not properly configured for its motor. You can correct this problem by issuing a **CMTR** command using the code for the motor being used. When you change EPROMs integrated circuits. If memory is corrupted either during operation or when the Compumotor Plus is off. If the error occurs each time you power up your unit, consult Compumotor Application Engineering (800-358-9070).
61	Indexer incoming pulses (non-indexer version only)	The drive was receiving step pulses while powering up or after a reset. Ensure that step pulses are disabled during power up or reset. *This is a safety feature which prevents the motor from suddenly moving immediately following a reset or power failure if the motion controller continues to issue a position command.*
88	The microprocessor is not running	The drive may be faulty. Try replacing the drive. There may be excessive noise in the environment. Following the anti-electrical noise recommendations elsewhere in this chapter. Ensure the microprocessor has not fallen from its socket.

RS-232C Communication Problems

Use the following procedure to troubleshoot communication problems that you may have with the Compumotor Plus.

① Be sure the host computer's transmit (TX) wire is wired to the peripheral's receive (RX) connection, and the host computer's receive (RX) wire is wired to the peripheral's transmit (TX) connection. Switch the receive and transmit wires on either the host or peripheral if the problem persists.

② Confirm that the host and peripheral are configured for the same baud rate, 8 data bits, 1 stop bit, and no parity.

③ If you receive double characters, for instance typing A and receiving AA, the computer is set for half duplex mode. Change the terminal setup to full duplex mode.

④ To test the terminal or terminal emulation software and the RS-232C cable for proper three-wire communication, unhook the Compumotor Plus and enter a character. You should not see a character on the screen. If you do, your terminal is in half duplex mode. Connect the host's transmit and receive lines together and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin outs.

⑤ Use DC common or signal ground as a reference, not earth ground.

⑥ Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to shield the cable-to-earth ground at one end only.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This section offers some helpful tips for debugging your programs or to understand why something may be happening in a way you do not expect. The Compumotor Plus has several tools that can be used to aide in the determination of problems in the system design. Those tools are listed below.

Command	Tool
DFS	Display servo status
DFX	Display indexer status
OS	Display homing/jog status
SS	Display set-up report
XDIR	Display programmed sequences
XSS	Sequence execution status
XST	Single step execution
XTR	Trace mode

DFS Servo Status

The **DFS** command reports the status of the servo control switches. The table below describes the function of each of the switches.

Bit	Function	Bit	Function
31	Reserved	15	Reserved
30	Reserved	14	Reserved
29	Reserved	13	Reserved
28	Reserved	12	Reserved
27	Reserved	11	Enable circuit — enabled=0, disabled=1
26	Reserved	10	Reserved
25	Reserved	9	Reserved
24	Reserved	8	Failed CRC check — no = 0, yes = 1
23	Reserved	7	Reserved
22	Reserved	6	Average current exceeded — no = 0, yes = 1
21	Reserved	5	Max position error exceeded —no = 0, yes = 1
20	Reserved	4	Reserved
19	Reserved	3	Driver internal error — no error=0, pwm hardware shutdown=1
18	Reserved	2	Reserved
17	Reserved	1	Overcurrent —no=0, yes (shutdown error)=1
16	Reserved	0	RS-232C CMD — on (ST0)=0, off (ST1)=1

DFX Command Report Back

The **DFX** command reports the Compumotor Plus indexer's current states and conditions. The information is described in the following table.

Bit	Function	Bit	Function
31	Reserved	15	U command waiting for continue no=0, yes=1
30	Reserved	14	Waiting for a trigger no=0, yes=1
29	Reserved	13	Home to resolver position no=0, yes=1
28	Reserved	12	Back up to home limit no=1, yes = 1
27	Reserved	11	High speed portion of home move not in progress=0, in progress=1
26	Reserved	10	Execute a sequence no=0, yes=1
25	Reserved	9	Waiting on a timer no=0, yes=1
24	Reserved	8	Hit a CCW limit no=0, yes=1
23	Homing to resolver position no=0, yes=1	7	Hit a CW limit no=0, yes=1
22	Hit a soft CW limit no=0, yes=1	6	PS command waiting for continue o=0, yes=1
21	Hit a soft CCW limit no=0, yes=1	5	Absolute move direction positive=0, negative=1
20	Registration move in progress no=0, yes=1	4	Positioning mode (incremental/absolute) MPI=0, MPA=1
19	Home limit not found=0,found=1	3	Mode (preset/continuous) MN=0, MC=1
18	Jog disable=0, enable=1	2	Commanded move direction positive=0, negative=1
17	Queued for RM mode no = 0, yes = 1	1	Preset move in progress not moving=0, moving=0
16	Run sequence on power-up no = 0, yes=1	0	Continuous move in progress not moving=0, moving=0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Homing and Jog Status

The **OS** command is used to report the status of homing, jog, and other setup functions. Below is a summary of the commands and their uses. Refer to the **OS** command in the *Chapter 5, Software Reference* for more information.

Command	Description
OSA	Go home to resolver position
OSB	Back-up to home switch
OSC	Define Active edge of Home switch 1=Active high signal
OSD	Not defined
OSE	Enable Jogging
OSF	Not defined
OSG	Define Final Home approach direction 1=CCW
OSH	Define Active edge of home switch to stop on 1=CCW
OSI	Save Sequence Scan Mode On Stop

Function Setup Reports

The **SS** command reports the state of the following setup switches.

Command	Description
SSA	RS232 Echo, 0 = on, 1 = off
SSB	TRIG1 clears pause, 0 = no, 1 = yes
SSC	Not defined
SSD	TRIG1 dedicated stop line, 0 = no, 1 = yes
SSE	Not defined
SSF	TRIG1 sequence strobe, 0 = no, 1 = yes
SSG	Clear/Save buffer on limit 0 = clear 1 = Save
SSH	Clear/Save buffer on stop 0 = clear 1 = Save
SSI	Enable/Disable Interactive Mode, 0 = enable, 1 = disable
SSJ	Enable/Disable Continuous scan mode, 0 = enable, 1 = disable
SSK	Enable registration mode, 0 = no, 1 = yes
SSL	Resume execution enable, 0 = no, 1 = yes

Sequence Directory

The Compumotor Plus has the ability to display the contents of all currently defined sequences at one time. You use the **XDIR** command to instruct the Compumotor Plus to do this. The display of sequences is useful to establish which sequences have been defined as desired. It also makes a compact reference list for storing or referring to your programs.

Sequence Status

The Compumotor Plus can report the status of any sequence. The report tells you if the sequence is defined, empty or bad. Refer to the **XSS** description in *Chapter 5, Software Reference*. The **XSS** command is useful for determining if a sequence is usable. You normally issue this command immediately after attempting a download to ensure the download was successful. You may also use the Sequence Directory (**XDIR**) command to verify the contents of sequences.

Single Stepping

It can be beneficial to execute a program one command at a time. You can do this in the Compumotor Plus by using the Single Step (**XST**) command . If you single step in conjunction with the Trace mode (described above) you have a powerful problem solving tool at your disposal.

Trace Mode

The Trace mode is used to display what is occurring as you execute your sequence. By running the Trace mode you can see what commands are being executed and if the program stops running you can see what command was last executed. The Trace mode along with the Interactive mode (**SSI**) will help you to find commands that the indexer may not recognize. The Trace mode is enabled and disabled using the **XTR** command. When enabled you will execute sequences as you normally would using the **XR** command. As the sequence is running the commands are displayed on the screen. **XTR1** enables the Trace mode, **XTRØ** disables it. See *Chapter 4, Application Design* for a more thorough explanation of the Trace mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

General Status

You can report back the value set by most of the Compumotor Plus commands by typing the device address followed by the command and a delimiter (carriage return or a space bar). In this way, you can find out what values you have entered for different commands.

Common Problems

This section describes some common problems which can occur then using the Compumotor Plus and possible reasons for the problems.

Problem	Possible Error
A move is commanded but no motion occurs.	A limit may be enabled and active. Use the **LD** command to find out which limits are enabled and the **IS** command to find out which limits are active. The Compumotor Plus may be in Absolute mode and is already at the commanded absolute position. Use the **PR** command to get the absolute position. If you wish to move incrementally, issue the **MPI** command.
The Compumotor Plus appears to be ignoring commands.	If you defined a sequence and never issued an **XT** command, the Compumotor Plus still thinks you are defining a sequence. Issue an **XT** command at the end of the sequence to end sequence definition.
The motor move sluggishly?	When the motor is easy to turn by hand over one-eighth or one-quarter of a revolution, and it moves into position without overshooting, the motor may have its gains set too low. Try increasing the velocity gain until the motor is as stiff as necessary. Be sure to lower the gains if the motor begins to resonate.
The motor vibrates when it's supposed to be standing still?	When the motor vibrates, resonates or makes a buzzing noise when it is holding a fixed position the gains main be set too high. The vibrations are not always a problem. Very high performance systems exhibit this behavior in some cases. The vibration is a signal that increasing the gains further could cause the motor to become unstable (see above). You can lower the velocity or derivative gains to stop the vibration if required.
I set the motor resolution to the maximum and my positioning isn't any more accurate. Why?	The accuracy of the system is determined by the motor's natural resolution, among other factors (see above). The user resolution has little to do with positioning accuracy unless a very coarse resolution is chosen.
My throughput is lower than I need.	You can frequently improve throughput by optimizing the gains in the Compumotor Plus. Sometimes you can get an improvement in throughput by using a motor with more torque. However, larger motors have larger inertias, making the gain in throughput less than you might think.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A P P E N D I X

Command Listing

Single Step
" Quote
A Acceleration
B Buffer Status
BCCA Buffered Configure Current Average
BCCP Buffered Configure Current Peak
BCDB Buffered Configure Dead Band
BCDG Buffered Configure Derivative Gain
BCDM Buffered Configure Differential Maximum
BCIG Buffered Configure Integral Gain
BCIL Buffered Configure Maximum Integral Limit
BCIM Buffered Configure Integral Maximum
BCPE Buffered Configure Position Error
BCPG Buffered Configure Positional Gain
BCPM Buffered Configure Positional Maximum
BCVG Buffered Configure Velocity Gain
BCVM Buffered Configure Velocity Maximum
BS Buffer Space
C Continue
CCA Configure Current Average
CCP Confogure Maximum Current Peak
CDB Configure Deadband
CDG Configure Differential Gain
CDM Configure Differential Maximum
CGS Configure Gain Switching
CIG Configure Integral Gain
CIL Configure Integral Sum Maximum
CIM Configure Integral Maximum
CIP Configure in Position Time
CMR Configure Motor Resolution
CMTR Configure Motor
CPB Configure Pushbuttons
CPE Configure Positional Error
CPG Configure Proportional Gain
CPM Configure Proportional Maximum
CR Carriage Return
CVG Configure Velocity Gain
CVM Configure Velocity Maximum
D Distance
DCA Display Current Average
DCI Display Current Instantaneous
DCP Display Current Peak
DFS Display Flags for Servo Parameters
DFX Display Flags of Indexer Commands
DPA Display Position Actual
DPE Display/Report Position Error
DPR Display/Report Position Resolver
DPS Display/Report Position Set Point
DTP Display Tuning Parameters
DVA Display/Report Velocity Actual
DVS Display/Report Veloccity Setpoint
E Enable Communications
ELSE Else
F Disable Communications
FILT Engage Digital Torque Filter
G Go
GA Go Home Acceleration
GAINX Select Gain Set
GD Go Defined
GDEF Move Definition
GH Go Home
GHF Go Home Final Velocity
GHP Define the Go Home to Resolver Position
^H Backspace
H Set Direction
IFER If Error Condition
IFFL Compare User Flag
IFIN IF Input
IFTR If Trigger
IM Input Mode
IO Immediate Output
IS Input Status
JA Jog Acceleration
JV Jog Velocity
K Kill
KILL Kill
L Loop
LA Limit Acceleration
LD Limit Disable
LED Illuminate Number on LED Display
LF Line Feed
MC Mode Continuous
MN Mode Normal
MPA Mode Position Absolute
MPI Mode Position Incremental
N End of Loop
NIF End of IF commands
O Output
OFF Off
OM Output Mode
ON On
OR Homing Status
OS Report State of OS Switches
OSA Home to Resolver Position
OSB Backup to Home Switch
OSC Define Active State of Home
OSE Jog Enable
OSG Final Homing Direction

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OSH Reference Edge of Home
OSI Save Sequence Scan Mode on Stop
PR Absolute Position Report
PS Pause
PZ Set Absolute Position to Zero
QØ Exit Velocity Profiling Mode
Q1 Enter Velocity Profiling Mode
R Request Indexer Status
RA Limit Switch Status Report
RB Loop, Pause, Shutdown, Trigger Status Request
REG Registration
RFS Return Drive Parameters to Factory Settings
RG Go Home Status
RM Rate Multiplier in Velocity Streaming Mode
RS Request Sequence Status
RSE Report Servo Errors
RV Revision
S Stop
SAVE Save
SFL Set User Flag
SL Software Limits
SLD Software Limit Disable
SN Scan
SS Function Set-Up Report
SSA RS-232C Echo Control
SSB TRIG1 Dedicated to Clear a Pause
SSD TRIG1 Dedicated as a Stop Line
SSF TRIG1 Dedicated as a Sequence Sequence Strobe
SSG Clear/Save the Command Buffer On Limit
SSH Clear/Save the Command Buffer on Stop
SSI Interactive Mode
SSJ Continuous Sequence Scan Mode
SSK Enable/Disable Registration
SSL Resume Execution Enable
ST Shutdown
STOP Stop
SV Servoing Parameter
T Time
TR Trigger
TS Trigger Input Status
TUNE Tune automatically
U Pause and Wait for Continue
V Velocity
W1 Signed Binary Position Report
W2 Hexadecimal Position Report
W3 Signed Hexadecimal Position Request
XBS Sequence Availability
XC Sequence Checksum
XD Sequence Definition
XDIR Sequence Directory
XE Sequence Erase
XFK Set Fault or Kill Sequence
XG GOTO Sequence
XQ Sequence Hold
XR Sequence Run
XRP Sequence Run with Pause
XSD Sequence Status
XSR Sequence Status Run
XSS Sequence Specified Status
XST Sequence Step Mode
XT Sequence Termination
XTR Set Trace Mode
XU Sequence Upload
Y Stop Loop
Z Reset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I N D E X

A

ABSOLUTE POSITION COUNTER 21
AC POWER 8, 12
ACCURACY & REPEATABILITY 127
AMBIENT TEMPERATURE 16
APPLICATION CONSIDERATIONS 27
ATMOSPHERIC CONTAMINATION 16

B

BAUD RATE 13
BAUD RATES 14
BENCH TEST 6
BRANCHING 30
BRANCHING WITH IF 31

C

CALCULATING MOVE TIMES 28
CCW LIMIT TEST 20
CHECK-OUT PROCEDURE 6
COMMAND
- ABSOLUTE POSITION REPORT 91
- ACCELERATION LIMIT 83
- AMPLIFIER OFF 87
- AMPLIFIER ON 88
- ATTRIBUTES 53
- BACKSPACE 76
- BACKUP TO HOME SWITCH 89
- BEGIN LOOP 82
- BUFFER SPACE 60, 61
- BUFFER STATUS 55
- BUFFERED CONFIGURE CURRENT AVERAGE 56
- BUFFERED CONFIGURE CURRENT PEAK 56
- BUFFERED CONFIGURE DEAD BAND 57
- BUFFERED CONFIGURE DERIVATIVE GAIN 57
- BUFFERED CONFIGURE DERIVATIVE MAXIMUM 57
- BUFFERED CONFIGURE INTEGRAL GAIN 58
- BUFFERED CONFIGURE INTEGRAL MAXIMUM 58
- BUFFERED CONFIGURE MAXIMUM INTEGRAL SUM LIMIT 58
- BUFFERED CONFIGURE POSITION ERROR 59
- BUFFERED CONFIGURE POSITION GAIN 59
- BUFFERED CONFIGURE POSITIONAL MAXIMUM 59
- BUFFERED CONFIGURE VELOCITY GAIN 60
- BUFFERED CONFIGURE VELOCITY MAXIMUM 60
- CARRIAGE RETURN 67
- CHARACTERISTICS 52
- CLEAR/SAVE COMMAND BUFFER ON LIMIT 103
- CLEAR/SAVE COMMAND BUFFER ON STOP 103
- COMPARE USER FLAG 77
- CONFIGURE CURRENT AVERAGE 61
- CONFIGURE DEAD BAND 62
- CONFIGURE DIFFERENTIAL GAIN 62
- CONFIGURE DIFFERENTIAL MAXIMUM 62
- CONFIGURE GAIN SWITCHING 63
- CONFIGURE IN-POSITION TIME 64
- CONFIGURE INTEGRAL GAIN 63
- CONFIGURE INTEGRAL MAXIMUM 64
- CONFIGURE INTEGRAL SUM MAXIMUM 63
- CONFIGURE MAXIMUM CURRENT PEAK 62
- CONFIGURE MOTOR 65
- CONFIGURE MOTOR RESOLUTION 64
- CONFIGURE POSITION ERROR 66
- CONFIGURE PROPORTIONAL GAIN 66
- CONFIGURE PROPORTIONAL MAXIMUM 66
- CONFIGURE PUSHBUTTON 65
- CONFIGURE VELOCITY GAIN 67
- CONFIGURE VELOCITY MAXIMUM 67
- CONTINUE 61
- DEFAULT 53
- DEFINE ACTIVE STATE OF HOME SWITCH 90
- DEFINE RESOLVER HOME POSITION 76
- DESCRIPTION 54
- DISABLE COMMUNICATIONS INTERFACE 72
- DISPLAY CURRENT AVERAGE 68
- DISPLAY CURRENT INSTANTANEOUS 68
- DISPLAY CURRENT PEAK 69
- DISPLAY FLAGS FOR SERVO PARAMETERS 69
- DISPLAY INDEXER FLAGS 69
- DISPLAY POSITION ACTUAL 70
- DISPLAY TUNING PARAMETERS 71
- DISPLAY/REPORT ACTUAL VELOCITY 71
- DISPLAY/REPORT POSITION ERROR 70
- DISPLAY/REPORT POSITION RESOLVER 70
- DISPLAY/REPORT POSITION SET POINT 71
- DISPLAY/REPORT VELOCITY SETPOINT 71
- DISTANCE 68
- ELSE 72
- ENABLE COMMUNICATIONS INTERFACE 72
- ENABLE DIGITAL TORQUE FILTER 73
- ENABLE REGISTRATION 105
- ENABLE/DISABLE CONTINUOUS SCAN MODE 104
- ENABLE/DISABLE INTERACTIVE MODE 104
- END OF IFXX COMMAND 86
- END OF LOOP 86
- ERASE SEQUENCE 114
- EXAMPLE 54
- FINAL HOMING DIRECTION 90
- FUNCTION SETUP REPORT 88
- GO 73
- GO DEFINED 74
- GO HOME 75
- GO HOME ACCELERATION 74
- GO HOME FINAL VELOCITY 76
- GO HOME STATUS REQUEST 96
- GO HOME TO RESOLVER POSITION 89
- GOTO SEQUENCE 114
- HEXADECIMAL SIGNED POSITION REPORT 112
- HEXADECIMAL UNSIGNED POSITION REPORT 111
- IDENTIFIER 52
- IF ERROR CONDITION 77
- IF INPUT 78
- IF TRIGGER 78
- IMMEDIATE OUTPUT 80
- INPUT MODE CONFIGURE 79
- INPUT STATUS 81
- JOG ACCELERATION 81
- JOG ENABLE 90
- JOG VELOCITY 81
- KILL MOTION 82
- LIMIT DISABLE 83
- LIMIT SWITCH STATUS REPORT 93
- LINE FEED 84
- LOOP, PAUSE, SUTDOWN AND TRIGGER STATUS 94
- MODE CONTINUOUS 84
- MODE NORMAL 85
- MODE POSITION ABSOLUTE 85
- MODE POSITION INCREMENTAL 85
- MOVE DEFINITION 75
- NAME 52
- OUTPUT 87
- OUTPUT MODE 87
- PAUSE 91

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PAUSE & WAIT FOR CONTINUE 109
QUOTE 54
RANGE 53
REFERENCE EDGE OF HOME SWITCH 90
REPORT INDEXER STATUS 93
REPORT SERVO ERRORS 98
RESET 119
RESUME EXECUTION ENABLE 105
RETURN DRIVE PARAMETERS TO FACTORY SETTING 96
RETURN INDEXER TO FACTORY SETTINGS 96
REVISION LEVEL 98
RS-232C ECHO CONTROL 102
RUN A SEQUENCE 115
SAVE SEQUENCE SCAN MODE ON STOP 91
SAVE SET-UP AND SEQUENCES 99, 106
SCAN 101
SELF TUNE 108
SEQUENCE CHECKSUM REPORT 112
SEQUENCE DEFINITION 113
SEQUENCE DIRECTORY REPORT 113
SEQUENCE RUN STATUS 116
SEQUENCE RUN WITH PAUSE 116
SEQUENCE STATUS 117
SEQUENCE STATUS DEFINITION REPORT 116
SET ABSOLUTE POSITION TO ZERO 92
SET ACCELERATION 55
SET DIRECTION 76
SET FAULT OR KILL SEQUENCE 114
SET GAIN TYPE 74
SET POSITION ABSOLUTE 101
SET RATE MULTIPLIER VELOCITY 97
SET REGISTRATION PARAMETERS 95
SET SEQUENCE INTERRUPTED-RUN MODE 115
SET SEQUENCE SINGLE-STEP MODE 117
SET SOFT LIMITS 100
SET TRACE MODE 118
SET USER FLAG 99
SET VELOCITY PROFILING MODE 92
SET-UP REPORT 101
SHOW NUMBER ON LED DISPLAY 84
SHUTDOWN THE AMPLIFIER 106
SIGNED BINARY POSITION REPORT 110
SINGLE STEP 54
SOFT LIMIT DISABLE 100
STATUS OF SEQUENCE EXECUTION 97
STOP LOOP 119
STOP MOTION 99, 106
SYNTAX 52
TERMINATE (END) SEQUENCE 118
TIME DELAY 107
TRIG1 CLEARS PAUSE 102
TRIG1 DEDICATED AS SEQUENCE STROBE 103
TRIG1 DEDICATED STOP LINE 102
TRIGGER INPUT STATUS 108
TYPE 52
UNITS 52
UPLOAD SEQUENCE 119
VELOCITY 109
VERSION 52
WAIT FOR TRIGGER 107
COMMANDS
BUFFERED 25
ISSUING 8
COMMUNICATION PARAMETERS 7
CONNECTION
DAISY CHAIN 14
MOTOR 13
CONNECTION OVERVIEW 8
CONNECTIONS
LINE POWER 12
POWER 7
RS-232 7
CONNECTOR
MS-TYPE 13
RESOLVER 13
RS-232C 13
SCREW TERMINAL 12
CONTINUOUS MODE 29
CONTINUOUS MODE MOVES 21
COUPLING *17*
ANGULAR MISALIGNMENT 17
DOUBLE-FLEX 17
END FLOAT 17
PARALLEL MISALIGNMENT 17
RIGID 18
SINGLE-FLEX 17
CW LIMIT TEST 20

D

DEBUGGING YOUR MOTION PROGRAM 132
DELIMITER 8
DEVICE ADDRESS 14
DFS SERVO STATUS 132
DFX COMMAND REPORT BACK 132
DISCRETE CONTROL CONNECTIONS 124
DISCRETE INPUT/OUTPUT CONNECTIONS 125
DRIVE DIMENSIONS
CPH & CPHX 126
CPL, CPLX 126
DRIVE/MOTOR CONFIGURATION 6

E

EEPROM 35
ELECTRICAL CONNECTIONS 8
ENCODER FEEDBACK 15
ENCODERS 15
ENVIRONMENTAL SPECIFICATIONS 11
ERRORS
DIAGNOSTIC CODES 130
ESTABLISH COMMUNICATIONS 7
EVENT COMPLETION SIGNALS 34

F

FACTORY DEFAULT 19
FACTORY DEFAULTS 9
FAULT OUTPUT 15
FAULTS 22
FINAL SYSTEM CONFIGURATION 22
FREQUENCY RESPONSE 49
FRONT PANEL
LOGO 14
FRONT PANEL DESCRIPTION 6
FUNCTION SETUP REPORTS 133

G

GAIN
DERIVATIVE 44
INTEGRAL 44
PROPORTIONAL 44
VELOCITY 44
GENERAL STATUS 134
GROUNDING
EMI 12

H

HIGH POWER CONNECTIONS 12
HOME INPUT SWITCH 21
HOMING AND JOG STATUS 133
HOMING THE MOTOR 21

I

I/O SCHEMATICS 123
I/O SPECIFICATIONS 124
IMMEDIATE COMMANDS 35
INCREMENTAL VS. ABSOLUTE POSITIONING 28
INPUT OUTPUT SPECIFICATIONS 125
INPUT POWER 124
INPUTS
COMMAND 23
HOME 23
LIMIT 15
LIMIT SWITCHES 19
PROGRAMMABLE 22
SEQUENCE 24
INPUTS & OUTPUTS 22
INSTABILITY 45
INSTALLATION
VERIFYING 19
INTERACTIVE MODE 19
INTERACTIVE PROGRAMMING 29
ISOLATION TRANSFORMER 13

L

LIMITS
END-OF-TRAVEL 15, 21
LIMITS, HOMING, TRIGGERS & SEQUENCE SELECTION 130
LINE FILTERING 13
LOOPING 30
LOOPS
NESTING 30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

M

MAXIMUM AVERAGE CURRENT 56
MAXIMUM PEAK CURRENT 56
MODES
  TRACE 133
MOTOR CONFIGURATION 9
MOTOR CONNECTIONS 124
MOTOR DIMENSIONS 127
MOTOR FAILS TO RUN AT HIGH SPEEDS 130
MOTOR IS JERKY OR WEAK 130
MOTOR RESOLUTION 18
MOTOR SPECIFICATIONS 127
MOTORS
  CPH106-220 5
  CPH83-150 5
  CPHX83-150 5
  CPLX57-120 5
MOUNTING
  DRIVE 16
  HORIZONTAL CLEARANCE 16
  PANEL 18
  SYSTEM 16
  VERTICAL CLEARANCE 16

N

NATIONAL ELECTRICAL CODE 12

O

OUTPUTS
  PROGRAMMABLE 22, 23

P

PEAK CURRENT LIMIT 56
PLC OPERATION 38
POSITION SENSOR 124
POSITIONING MODES 28
POWER CABLES 9
POWER-UP SEQUENCE EXECUTION 37
PRESET MODE) 28
PRODUCT FEATURES 2
PRODUCT LABEL 6
PROGRAMMABLE DELAYS 29
PROGRAMMABLE OUTPUTS (POBS) 33
PROPER GROUNDING 12
PUSHBUTTON
  TUNING 13
PUSHBUTTONS
  FRONT PANEL 13

R

REGISTRATION AND .I.SYNCHRONIZATION 40
REMOTE JOGGING 35
REMOTE SEQUENCE EXECUTION 37
REMOTE SEQUENCING (BCD INPUTS) 130
REPEATABILITY 27
RS-232C 131
RS-232C COMMUNICATION 39
RS-232C CONNECTIONS 124
RS-232C TERMINAL 9

S

SELF TUNING PROCEDURE 48
SEQUENCE
  40 36
SEQUENCE DIRECTORY 133
SEQUENCE SELECTION METHODS 36
SEQUENCE STATUS 133
SETTING DRIVE CURRENT 9
SHIP KIT CONTENTS 5
SINGLE STEPPING 133
SPECIFICATIONS
  PHYSICAL 127
STANDALONE OPERATION 37
SUBROUTINES 32
SWITCH-MODE AMPLIFIERS 13
SYSTEM CONFIGURATION 11
SYSTEM SPECIFICATIONS 127

T

TERMINAL EMULATOR 9
TESTING LIMIT SWITCH OPERATION 20
TORQUE/SPEED CURVES 128
TRANSFORMER 8, 12
  STEP DOWN 13
  STEP-DOWN 7
TRIGGER INPUTS 24
TROUBLESHOOTING 129
TRUNCATION ERRORS 19
TUNING
  RS-232C 45
TUNING PROBLEMS 48
TUNING THEORY 42

V

VIBRATION 49

W

WIRING
  VERIFICATION 24
WIRING GUIDELINES 12

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP