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 6201* at our website: *Click HERE*

Compumotor

6201
2-Axis Motion Controller
User Guide

Compumotor Division
Parker Hannifin Corporation
p/n 88-013966-01B June 1994

Parker

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

Important User Information

To ensure that the equipment described in this user guide, as well as all the equipment connected to and used with it, operates satisfactorily and safely, all applicable local and national codes that apply to installing and operating the equipment must be followed. Since codes can vary geographically and can change with time, it is the user's responsibility to identify and comply with the applicable standards and codes. **WARNING:** ***Failure to comply with applicable codes and standards can result in damage to equipment and/or serious injury to personnel.***

Personnel who are to install and operate the equipment should study this user guide and all referenced documentation prior to installation and/or operation of the equipment.

In no event will the provider of the equipment be liable for any incidental, consequential, or special damages of any kind or nature whatsoever, including but not limited to lost profits arising from or in any way connected with the use of this user guide or the equipment.

© Compumotor Division of Parker Hannifin Corporation, 1994
— All Rights Reserved —

Motion Architect is a registered trademark of Parker Hannifin Corporation.
CompuCAM and Servo Tuner are trademarks of Parker Hannifin Corporation.
Microsoft is a registered trademark and Windows is a trademark of Microsoft Coporation.

The information in this user guide, including any apparatus, methods, techniques, and concepts described herein, are the proprietary property of Parker Compumotor or its licensors, and may not be copied, disclosed, or used for any purpose not expressly authorized by the owner thereof.

Since Parker Compumotor constantly strives to improve all of its products, we reserve the right to change this user guide and equipment mentioned therein at any time without notice.

For assistance in the United States, contact:
Compumotor Division of Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070
Fax: (707) 584-8015

For assistance in Europe, contact:
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Telephone: 0202-690911
Fax: 0202-600820

Parker **Compumotor**

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

TABLE OF CONTENTS

Overview

Assumptions i
Contents of This User Guide i
Installation Process Overview ii
Conventions ii
Clockwise (CW, +) & Counter-clockwise (CCW, -) Directions ii
6000 Series Commands ii
Related Publications ii

Chapter 1. Introduction

6201 Description 1
System Hardware Block Diagram 2
6201 Features 2

Chapter 2. Getting Started

Inspect The Shipment 5
Pre-Wired Connections 6
Bench Test 6
① Select Motor Current 6
② Connect Motors 7
③ Establish RS-232C Communications 7
④ Connect Power Cable 8
⑤ Test Procedure 9
What's Next? 10

Chapter 3. Installation

Installation Precautions 11
Protective Circuits 11
Heat & Humidity 11
Electrical Connections 12
Electrical Noise 12
Airborne Contaminants 12
Follow Installation Procedure 12
① Mount the 6201 12
Airflow 13
Panel Mounting 13
② System Connections 14
Active High/Active Low Conventions 14
End-of-Travel Limit Connectionsconnections 14
Home Limit Connections 15
Encoder Connections 15
Auxiliary +5V Output Connection 16
Pulse Cut (P-CUT) Connection 16
Programmable Inputs & Outputs Connections 17
Fast Trigger Input Connections 18
RP240 Remote Operator Panel Connections 19
Joystick and Analog Input Connections 19
Extending 6201 System Cables 21
③ Motor Mounting 22
④ Installation Verification 22
⑤ Attaching the Load 24
⑥ Tuning 24
What's Next? 26
Determine Your Application Requirements First 26

Chapter 4. Feature Implementation

Before You Proceed With This Chapter 27
6000 Series Software Reference Guide 27
Compumotor Bulletin Board Service 27
Basic Motion Control Concepts 28
Support Software 28
6000 DOS Support Disk 28
Motion Architect® 28
System Configuration 28
Number of Axes 29
Memory Allocation 29
Drive Fault Level 29
Scaling 29
Acceleration & Deceleration Scaling (SCLA/PSCLA) 30
Velocity Scaling (SCLV/PSCLV) 30
Distance Scaling (SCLD/PSCLD) 31
End-of-Travel Limits 32
Homing 33
Positioning Modes 35
Preset Mode 36
Continuous Mode 37
Closed-Loop Operation (Using an Encoder) 38
Motor & Encoder Resolutions 38
Encoder Step Mode 39
Position Maintenance 39
Position Maintenance Deadband 39
Stall Detection & Kill-on-Stall 39
Stall Backlash Deadband 39
Encoder Set Up Example 40
Counter 40
User Interface Options 40
Programmable Inputs and Outputs 41
Output Functions 42
Input Functions 44
Thumbwheel Interface 51
PLC Interface 53
Analog Inputs 53
Joystick Control 53
Feedrate Override 55
RP240 Remote Operator Panel Interface 57
Operator Interface Features 57
Using the Default Mode 58

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

Chapter 4 *(continued)*

Host Computer Operation 60
Variables 61
Converting Between Binary and Numeric Variables 61
Performing Operations with Numeric Variables ... 61
Performing Operations with Binary Variables 63
X-Y Linear Interpolation 64
Contouring (Circular Interpolation) 65
Path Definition 65
Participating Axes 66
Path Acceleration, Deceleration, and Velocity 66
Segment End-point Coordinates 66
Line Segments 68
Arc Segments 68
Segment Boundary 70
Outputs Along the Path 70
Paths Built Using 6000 Commands 71
Compiling the Path 71
Executing the Path 71
Possible Programming Errors 71
Programming Examples 72
Teach Mode 73
Teach Mode Basics 73
Summary of Related 6000 Series Commands 75
Teach Mode Application Example 76
RS-232C Daisy-Chaining 78
Daisy-Chaining from a Computer or Terminal 80
Daisy-Chaining from a Master 6201 80
Daisy-Chaining and RP240s 81

Chapter 5. Hardware Reference

General Specifications 83
I/O Pin Outs & Circuit Drawings 84
Optional DIP Switch & Jumper Settings 87
Accessing the DIP Switches and Jumpers 87
Motor Specifications 89
Rotor Inertia 89
Speed/Torque Curves 89
Motor Dimensions 90
Using Non-Compumotor Motors 91
Terminal Connections 92
Setting Motor Current – Non-Compumotor Motors 93

Chapter 6. Troubleshooting

Troubleshooting Basics 95
Reducing Electrical Noise 96
Error Messages and Debug Tools 96
Test Program 96
Common Problems & Solutions 96
RS-232C Troubleshooting 98
Returning the System 99

Appendix A

Alphabetical Command List 101

Index

..... 105

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

O V E R V I E W

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 the information in this user guide.

Assumptions

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

- Basic electronics concepts (voltage, switches, current, etc.)
- Basic motion control concepts (torque, velocity, distance, force, etc.)
- Basic programming skills with any high-level language such as BASIC, FORTRAN, or Pascal

With this basic level of understanding, you will be able to effectively use this user guide to install, develop, and maintain your system.

Contents of This User Guide

This user guide contains the following information.

Chapter 1: *Introduction*
Introduction describes the product and provides a brief account of its features.

Chapter 2: *Getting Started*
This chapter contains a list and description of items you should have received with your 6201 shipment. A *bench test* procedure is provided to verify the functionality of the system's primary functions.

Chapter 3: *Installation*
Installation provides instructions for mounting, wiring (electrical connections), and testing the 6201 system. Upon completion of this chapter your system should be ready to perform basic operations.

Chapter 4: *Feature Implementation*
This chapter will help you customize the system to meet your application's needs. Feature implementation is discussed. Sample applications are provided. Refer to the ***6000 Series Software Reference Guide*** for programming guidelines and detailed descriptions of the 6000 Series commands.

Chapter 5: *Hardware Reference*
Use the *Hardware Reference* as a quick-reference tool for 6201 electrical specifications, optional DIP switch and jumper settings, and I/O signal descriptions and circuit drawings.

Chapter 6: *Troubleshooting*
This chapter describes methods for isolating and resolving hardware and software problems.

Appendix A
At the end of this user guide you will find an alphabetical listing of the 6000 Series Commands and an index.

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

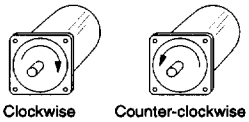
Installation Process Overview

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

1. Review this entire user guide. Become familiar with the user guide's contents so that you can quickly find the information you need. At times you may want to refer to the ***6000 Series Software Reference Guide*** for programming guidelines and detailed descriptions of the 6000 Series commands used in this user guide.
2. Read Chapter 1, *Introduction*, and the user documentation for all peripheral system components to develop a basic understanding of all system components, their functions, and interrelationships.
3. Read Chapter 2, *Bench Test*, and verify that you have received all the proper components for your system, and that all the items in your shipment have arrived without damage. Follow the step-by-step bench test procedures to verify the basic functionality of the 6201 Two-Axis Motion Controller, as well as the host computer (or terminal) and the drive/motor packages.
4. Complete the system configuration, mounting, and wiring instructions provided in Chapter 3, *Installation*. **Do not deviate from the sequence or installation methods provided.**
5. While in Chapter 3, be sure to use the *Installation Verification* procedures to check all the system functions and features to ensure that you have completed the installation process correctly.
6. After you successfully complete the *Installation Verification* procedures, you will be ready to proceed to Chapter 4 to implement the appropriate 6201 features in your application.

Conventions

Clockwise (CW, +) & Counter-clockwise (CCW, -) Directions



Throughout this user guide and the ***6000 Series Software Reference Guide***, you will find references to the *clockwise* (CW) and *counter-clockwise* (CCW) direction of motion. The CW or CCW direction is determined either by the direction of the motor shaft rotation (see illustration at left), or by the sign (+ or -) of the commanded position (e.g., the D+8ØØØ distance command indicates a 8,000-unit move in the clockwise direction). *This convention is accurate only if you connect the motor as instructed in Chapter 3.*

6000 Series Commands

The command language conventions are provided in the ***6000 Series Software Reference Guide***. *Because some 6000 Series products have four-axis capability, the syntax of the example commands in the* ***Reference Guide*** *shows data fields for all four axes; ignore the third and fourth data fields when entering commands or reading status commands for the 6201.*

Related Publications

The following publications may be helpful resources:

- ***6000 Series Software Reference Guide***, Parker Hannifin Corporation, Compumotor Division; part number 88-012966-01 (included with 6201)
- ***Motion Architect User Guide***, Parker Hannifin Corporation, Compumotor Division; part number 88-013056-01 (included with 6201)
- Current Parker Compumotor Motion Control Catalog
- User guide for the computer or terminal that you may use with the 6201
- Schram, Peter (editor). *The National Electrical Code Handbook*. 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

The information in this chapter will help you understand the 6201's basic functions & features.

6201 Description

The Compumotor 6201 is a stand-alone, microprocessor-based, two-axis motion controller with integrated power supply, microstepper controller, and drives.

Using the 6000 Series Programming Language, you can program the 6201 via a PC or a dumb terminal. User programs are stored in the 6201's battery-backed RAM.

The 6201 also provides remote operator interface capabilities when used with the Compumotor RP240 Operator Panel.

The 6201 comes standard with support software for the Microsoft® Windows™ and DOS operating environments:

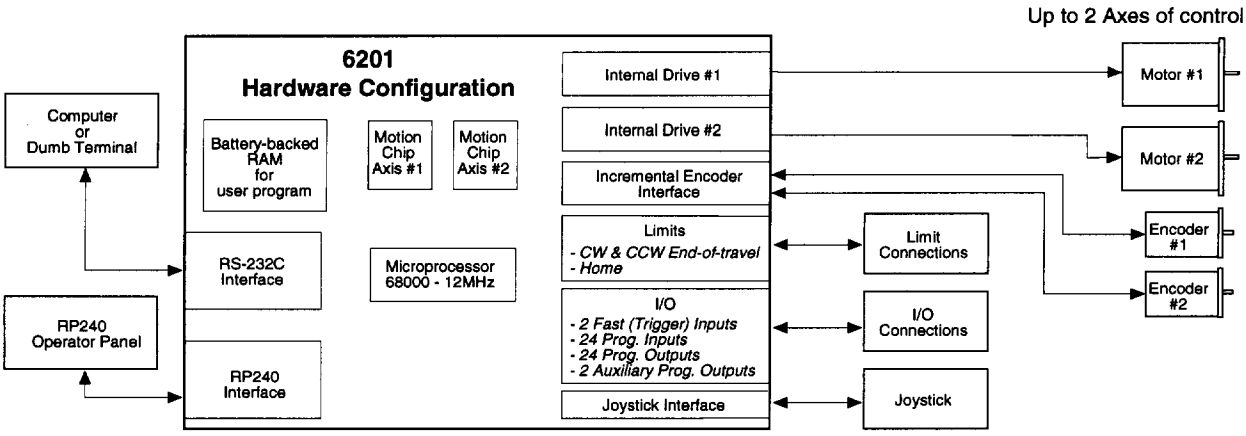
- *Motion Architect*® is an innovative, easy-to-use Microsoft® Windows™ based programming aide to help you easily create and implement complex motion programs. For more information, refer to the ***Motion Architect User Guide***.
- The *6000 DOS Support Disk* contains a DOS-based program editor and terminal emulator package. Also included are sample 6000 Command Language programs.

CompuCAM™ is also available. CompuCAM is a CAD-to-Motion (CAM) software tool that allows you to translate DXF, HP-GL, and G-Code files into 6000 Series Language motion programs.

Refer to the *6201 Features* below for a list of the 6201's I/O capability.

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

System Hardware Block Diagram

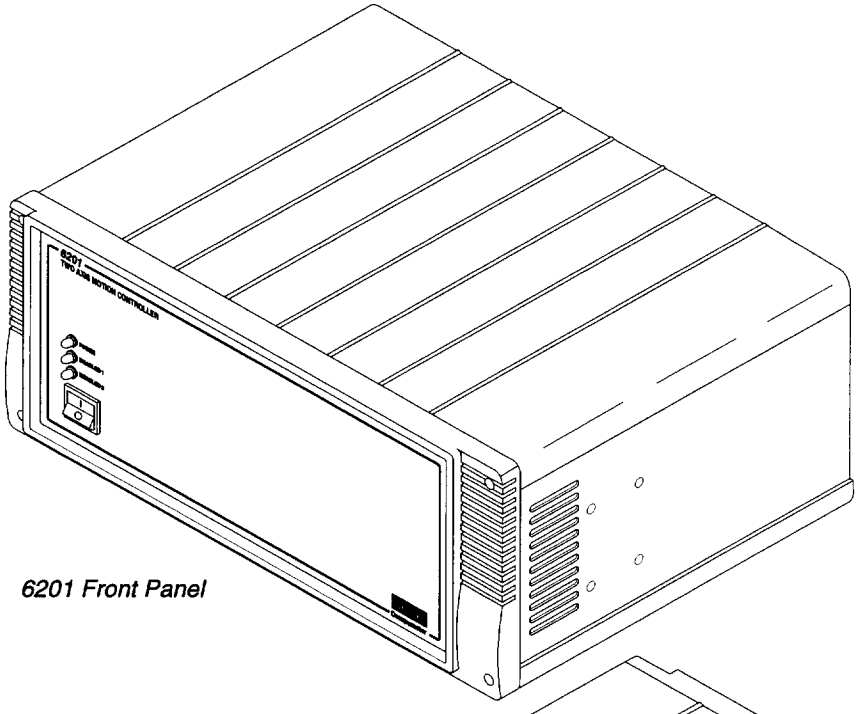


6201 Features

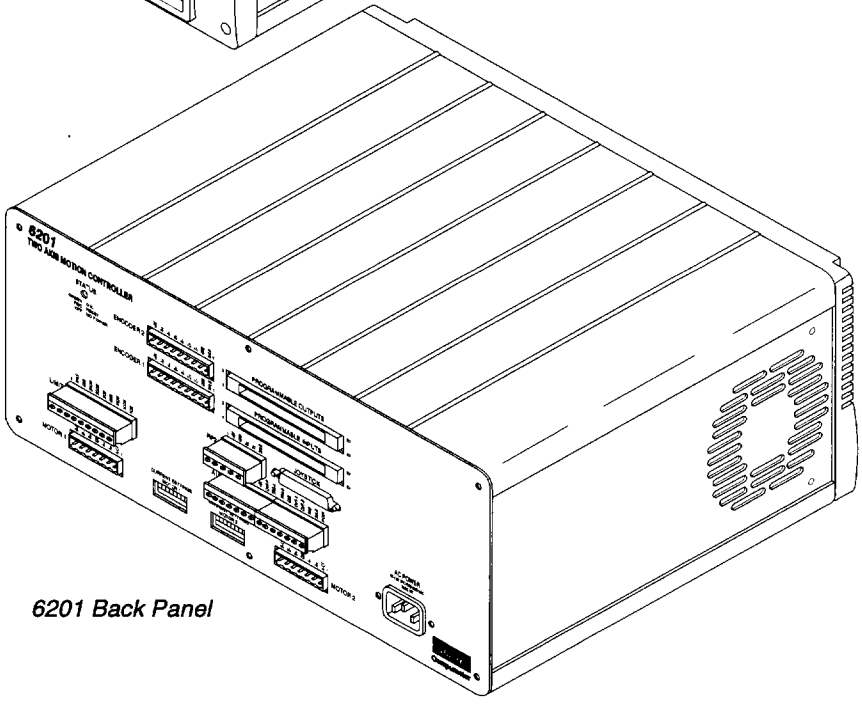
- Integrated controller, drives, and power supply in one chassis
- DOS Support Disk and Motion Architect® provided
- CompuCAM™ CAD-to-Motion software available
- 40,000 bytes of non-volatile memory for storing programs & paths; expansion to *150,000* bytes available with –M option
- Interrupt program execution on error conditions
- Capture encoder and motor positions (using the trigger inputs)
- Registration (using the trigger inputs)
- 2-axis linear interpolation
- 2-axis circular interpolation (*contouring*)
- Multi-axis teach mode
- Variable storage, conditional branching, and math capability
- Program debug tools — single-step and trace modes, breakpoints, error messages, and simulation of I/O
- Direct interface to RP240 Remote Operator Panel
- Operates stand-alone or interfaces to PCs, PLCs, and thumbwheels
- Communication with PC or dumb terminal via 3-wire RS-232C interface
- Encoder channels configurable as hardware up/down counters
- I/O capabilities (*all inputs and outputs are optically isolated*):
  - Incremental encoder input (up to 2 axes)
  - CW & CCW end-of-travel limit inputs (per axis)
  - Home limit input (per axis)
  - 3 analog inputs for joystick control and variable input
  - 2 *fast* (trigger) inputs – use with position capture and registration
  - 24 programmable inputs (Opto-22 compatible)
  - 24 programmable outputs (Opto-22 compatible)
  - 2 auxiliary programmable outputs
- Status/fault LEDs to confirm proper operation

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

- Protective circuits:
  - Motor short circuits phase-to-phase and phase-to-ground
  - Overtemperature of internal drives and power supply
  - Power dump (dissipates excess power caused by load regeneration)
- Phase offset for improved smoothness
- Drive resolution is fixed at 25,000 steps/rev
- Operates with Compumotor 6201 Series motors sizes 57-51 to 83-135 or user supplied motors operating from 2.5A to 7.1A



6201 Front Panel



6201 Back Panel

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

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 6201 system has been delivered safely and configured properly
- Bench test the system's primary components

Inspect The Shipment

If you need to return any or all of the 6201 system components, use the return procedures in Chapter 6, Troubleshooting.

You should inspect your 6201 shipment upon receipt for obvious damage to its shipping container. Report any damage to the shipping company as soon as possible. Parker Compumotor cannot be held responsible for damage incurred in shipment. The items listed below should be present and in good condition.

Part		Part Number
6201 main unit (w/ship kit)		6201
Ship kit:	6201 User Guide	88-013966-01
	6000 Series Software Reference Guide	88-012966-01
	6000 DOS Support Disk	95-012266-01
	6000 DOS Support Disk Quick Reference	88-013258-01
	Motion Architect diskettes: Disk 1	95-013070-01
	Disk 2	95-013070-02
	Motion Architect User Guide	88-013056-01
	8-foot AC power cord	44-000054-01

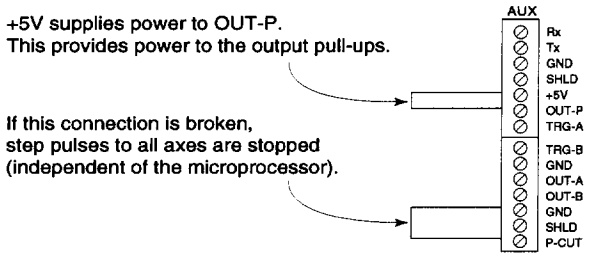
The following motors and options may be used with the 6201. Compare your order with the products shipped.

Options/Accessories	Part Number
6201 Series Motors	
Size 23 – 1 Stack Step Motor	6201-57-51-MO
Size 23 – 2 Stack Step Motor	6201-57-83-MO
Size 23 – 3 Stack Step Motor	6201-57-102-MO
Size 34 – 1 Stack Step Motor	6201-83-62-MO
Size 34 – 2 Stack Step Motor	6201-83-93-MO
Size 34 – 3 Stack Step Motor	6201-83-135-MO
Encoder Option (motor with E57 or E83 encoder)	6201-xx-xx(x)-MO-E
Single Shaft Motors	6201-xx-xx(x)-MO-S
19" Rack-Mount Brackets	6201-RMK-KIT
Expanded Memory Option (-M)	6201-M

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

Pre-Wired Connections

You should receive your 6201 with the following connections on the **AUX** connector pre-wired (from the factory):



Bench Test

This section leads you through step-by-step instructions to configure and bench test your 6201 system. *This is a temporary* (bench top) *configuration; the permanent installation will be performed in Chapter 3,* Installation. In this section, you will complete the following tasks:

① Select Motor Current
② Connect Motors
③ Establish RS-232C Communications
④ Connect Power Cable
⑤ Perform Test Procedure

Using Non-Compumotor Motors?

If you are using non-Compumotor motors, refer to the *Using Non-Compumotor Motors* section in Chapter 5, *Hardware Reference*, for details regarding required motor specifications, current selection options (DIP switch settings), and wiring and connection instructions.

Once you have selected the proper motor, configured the current setting DIP switches, and connected the motor to the 6201, skip steps ① and ② and proceed to step ③, RS-232C Communications.

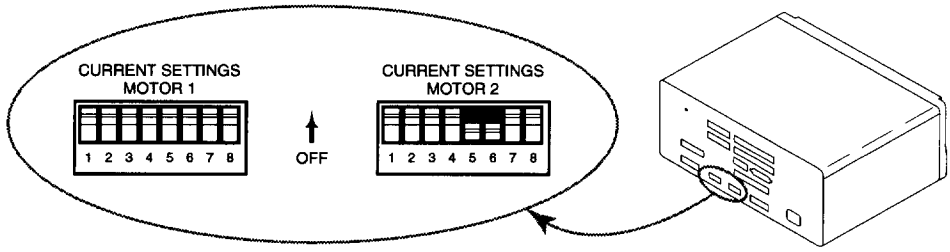
① Select Motor Current

WARNING

Make sure there is no power applied to the 6201 when adjusting motor current DIP switches.

Using Non-Compumotor Motors? Refer to Using Non-Compumotor Motors *section in Chapter 5*

Current for each motor is controlled by a set of DIP switches located on the back of the 6201 (see illustration below).



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

Verify which size motor you have, then set the appropriate current for that motor. The table below shows motor current settings for Compumotor 6201 Series motors. (DIP switches #7 and #8 are reserved.)

Motor Size	Current	Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	Switch 6
6201-57-51	2.5 A	**ON**	off	off	off	off	off
6201-57-83	3.1 A	off	**ON**	off	off	off	off
6201-57-102	3.5 A	off	off	**ON**	off	off	off
6201-83-62	4.4 A	off	off	off	**ON**	off	off
6201-83-93	5.6 A	off	off	off	off	**ON**	off
6201-83-135	7.0 A	off	off	off	off	off	**ON**

② Connect Motors

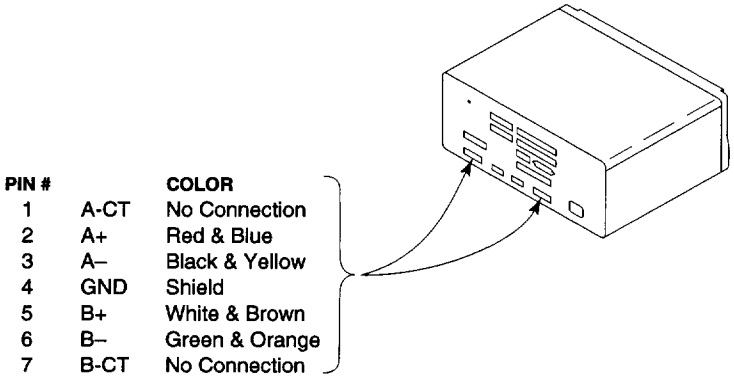
WARNING

Make sure there is no power applied to the 6201 when connecting or disconnecting motors.

The 6201 Two-Axis Motion Controller can run motors from Compumotor and other manufacturers. This section provides instructions for connecting Compumotor's 6201 Series motors.

Compumotor 6201 Series motors are pre-wired in parallel, with 10-foot cables and screw terminal connectors attached. The motors require no further setup. Simply plug the cable directly into the **MOTOR 1** or **MOTOR 2** connectors (see illustration below).

Using Non-Compumotor Motors? Refer to Using Non-Compumotor Motors *section in Chapter 5*



PIN #		COLOR
1	A-CT	No Connection
2	A+	Red & Blue
3	A–	Black & Yellow
4	GND	Shield
5	B+	White & Brown
6	B–	Green & Orange
7	B-CT	No Connection

③ Establish RS-232C Communications

To communicate with the 6201, your computer or terminal must have an RS-232C serial port.

The 6201 uses a three-wire implementation of standard EIA RS-232C signals.

Computer-to-Terminal Conversion

If you are using an IBM/compatible computer, you must use a terminal emulator software package to communicate with the 6201. The 6201 comes standard with Motion Architect® for Windows™ and the 6000 DOS Support Disk; both provide a terminal emulator and program editor (refer to the ***Motion Architect User Guide*** or the ***6000 DOS Support Disk Quick Reference*** for installation and other user information). You may also use communication programs such as Crosstalk™, PC-Talk™, and Procomm™.

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

Set Communication Parameters

Make sure your computer or terminal is set to the following communication parameters. You can configure these parameters by using one of the terminal emulation software packages listed above in *Computer-to-Terminal Conversion*. If you are using Motion Architect® or the 6000 DOS Support Disk, verify that the baud rate, data bit, parity, and stop bit parameters are set as follows:

- Baud Rate: 9600*
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Full Duplex
- XON/XOFF: Enabled

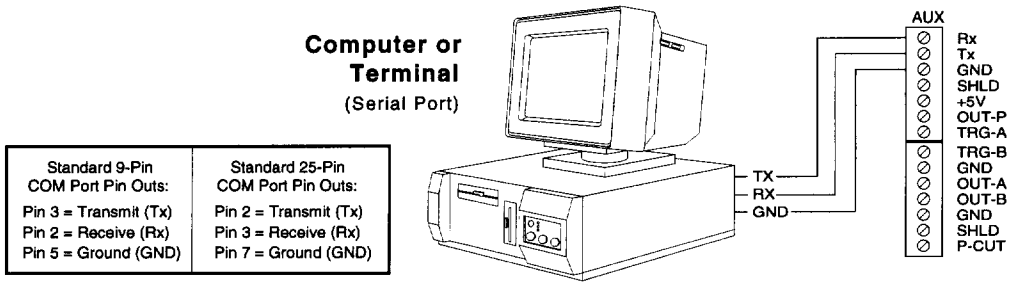
* If your terminal is not capable of 9600 baud, use the 6201's *auto-baud* function to automatically set the 6201's baud rate equal to the terminal's baud rate. Refer to *Optional DIP Switch & Jumper Settings* in Chapter 5 for instructions.

Terminal Connections

Connect your RS-232C cable to the 6201. The Receive Data (**Rx**), Transmit Data (**Tx**), and Ground (**GND**) signals are on the 6201's **AUX** connector (shown below). *The ground (**GND**) connection on the connector is signal ground or common as opposed to earth ground (**SHLD**).*

NOTE

If you intend to daisy chain multiple 6201 indexers, **do not attempt the daisy-chain connections now.** Daisy-chain instructions are provided in Chapter 4, *Feature Implementation.*

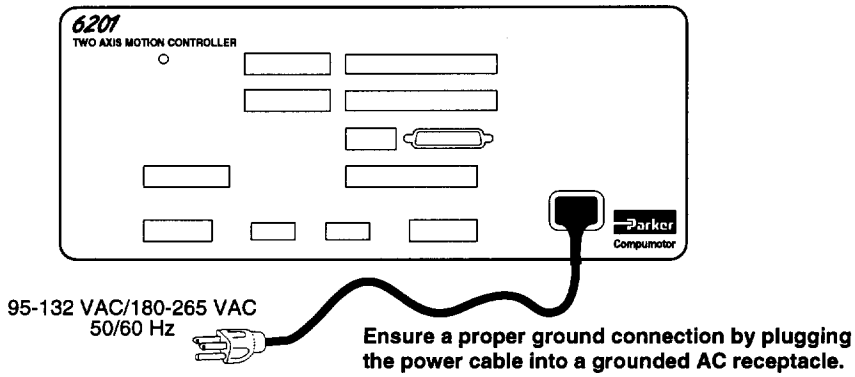


④ Connect Power Cable

The 6201 is shipped with an 8-foot power cable (p/n 44-000054-01). Attach the power cable to the 6201's **POWER** connector as illustrated below.

WARNING

DO NOT APPLY POWER TO THE 6201 UNTIL INSTRUCTED TO DO SO IN STEP ② OF THE FOLLOWING TEST PROCEDURE.



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

WARNING

The following test program, when initiated, will override the end-of-travel limits. To ensure safety of personnel and equipment, make sure the motor is secured in place and the shaft is allowed to rotate freely (**without being coupled to the load**).

⑤ Test Procedure

Use the following procedure to test the 6201 system. In Chapter 3, *Installation*, you will test the interface to encoders, RP240, joystick, and I/O.

Step 1 Make sure the Pulse Cut (**P-CUT**) input on the AUX connector is connected to **GND**.

Step 2 Apply power to the 6201 by plugging the power cable into a grounded power source and turning on the power switch on the front panel (see previous illustration).

Step 3 Watch the LEDs on the 6201.

Back Panel LED: The **STATUS** LED should be green, indicating the internal controller is ready for operation. If the LED is red, or if the LED does not illuminate, check your power source and connections. If these connections seem correct, disconnect power and consult Chapter 6, *Troubleshooting*.

Front Panel LEDs: The **POWER** LED should be green and illuminated, indicating that AC power is connected and the 6201 is turned on. The red LEDs labeled **DISABLED 1** and **DISABLED 2** should be OFF (not illuminated). If the **POWER** LED is off or the **DISABLED** LEDs are illuminated, disconnect power and consult Chapter 6, *Troubleshooting*.

Step 4 If you are using the 6000 DOS Support Disk, go to the **Set-up** menu and cursor down to **CHECK OUT** and press the ENTER key. The program will automatically check-out the communication interface to the 6201.

If the interface is not successful (`DEVICE NOT READY` message will flash on the screen), refer to the RS-232C troubleshooting procedures in Chapter 6.

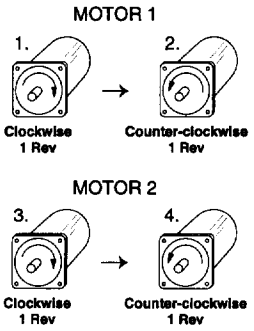
Step 5 Initiate the terminal emulator in Motion Architect or in the 6000 DOS Support Disk (refer to the ***Motion Architect User Guide*** or the ***6000 DOS Support Disk Quick Reference*** if necessary). You can also use your own terminal emulator package.

Press the RETURN key. The cursor should move down one or two lines each time you press the RETURN key. If the cursor does not move as described, refer to the RS-232C troubleshooting procedures in Chapter 6.

WARNING

The following test program, when initiated, will override the end-of-travel limits. To ensure safety of personnel and equipment, make sure the motor is secured in place and the shaft is allowed to rotate freely (**without being coupled to the load**).

Step 6 Type the `TEST` command at the prompt (`> TEST`) and press the ENTER or RETURN key. The motor on axis 1 should move clockwise one rev at one rev/sec, then move counter-clockwise the same distance at the same velocity, coming back to the starting position. The motor on axis 2 will then repeat the same pair of moves.



The primary objective of the `TEST` program is to verify the 6201 is functioning properly and that the motor is connected properly.

If the motor shaft moves in the opposite direction from that illustrated above, reverse the motor phase connections to the 6201 (swap the **A+** and **A–** connections), and then reissue the `TEST` command.

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

If the motor shaft does not move at the distance and velocity noted above, this simply means that the drive resolution has been changed from its default setting of 25,000 steps/rev. Issue the `DRES25ØØØ,25ØØØ` command to select the proper resolution setting.

What's Next?

At this point, you should have successfully configured the 6201's current selection DIP switches for the motors you are using, and verified that all system components are properly working together. You may now proceed with the permanent installation. See Chapter 3 *Installation* for instructions.

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 inputs and outputs properly
- Verify that the complete system is installed properly

To ensure proper installation, you should perform all the bench test procedures in Chapter 2, Getting Started, ***before*** *proceeding with the permanent installation process in this chapter.*

Installation Precautions

WARNING

Always remove power to the 6201 before performing wiring installation or changing DIP switch and jumper settings.

To help ensure personal safety and long life of system components, pay special attention to the following installation precautions.

Protective Circuits

The 6201 has built-in protective circuits that work automatically.

- Short Circuit Protection
- Overtemperature Protection
- Regeneration/Power Dump

Because these features work automatically, they can cause unexpected results if you are not aware of their presence. For full specifications about each circuit see Chapter 5, *Hardware Reference.*

Heat & Humidity

Operate the 6201 system at an ambient temperature between 32° and 113°F (0° to 45°C). Keep the relative humidity below 95%.

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

Electrical Connections

Hazardous voltages are present on the 6201's connectors when power is applied. To prevent injuries to personnel and damage to equipment, note the following guidelines:

- Never connect/disconnect the motor from the 6201 when power is applied. If you do, the motor connector may be damaged. Power should never be applied to the 6201 when the motor is not connected.
- Never increase the current setting (using the 6201's DIP switches) to more than 10% greater than the current specified for the motor you are using. Excessive current may cause the motor to overheat and result in a motor failure.
- Verify that there are no wire *whiskers* that can short out the motor connections.
- If the motor turns the wrong direction after you connect the motor wires to the connector and the connector to the 6201, you can change the direction by reversing the leads going to **A+** and **A-** on the motor terminal.
- Never probe the 6201. Never connect anything other than the motor to the motor terminals.

Electrical Noise

Minimize the potential for electrical noise before installing the 6201, rather than attempting to solve such problems after installation. You can prevent electrical noise by observing the following installation precautions:

- Do not route high-voltage wires and low-level signals in the same conduit.
- Ensure that all components are properly grounded.
- Ensure that all wiring is properly shielded.

For more information on electrical noise, refer to *Electrical Noise . . . Sources, Symptoms, and Solutions* in the Engineering Reference Section of the Parker Compumotor/Digiplan catalog.

Airborne Contaminants

Contaminants that may come in contact with the 6201 should be carefully controlled. Particulate contaminants, especially electrically conductive material such as metal shavings, can damage the 6201.

Follow Installation Procedure

To ensure proper installation of the 6201 system, this chapter is organized in logical, sequential steps. *Deviating from this prescribed format may result in system problems.*

① Mount the 6201
② Perform system connections
③ Mount the motors
④ Perform the installation verification (system test)
⑤ Attach the loads
⑥ Tune the system

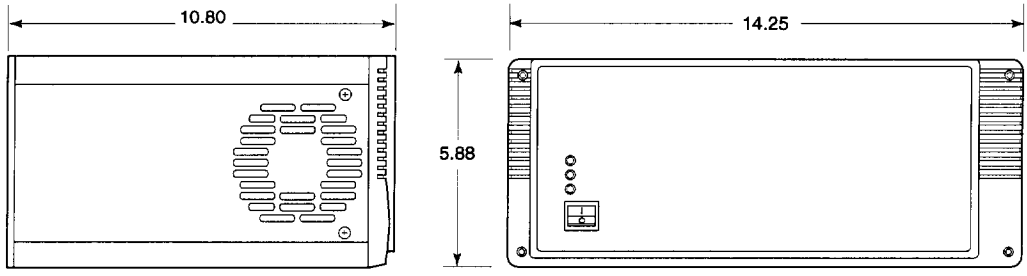
① Mount the 6201

The 6201 should be used in a location that will protect it from atmospheric contaminants such as oil, metal, moisture, and dirt.

You can place the 6201 on a desktop, and operate it without further mounting. Or, you can use optional 19" rack-mount brackets, and mount the 6201 to a rack.

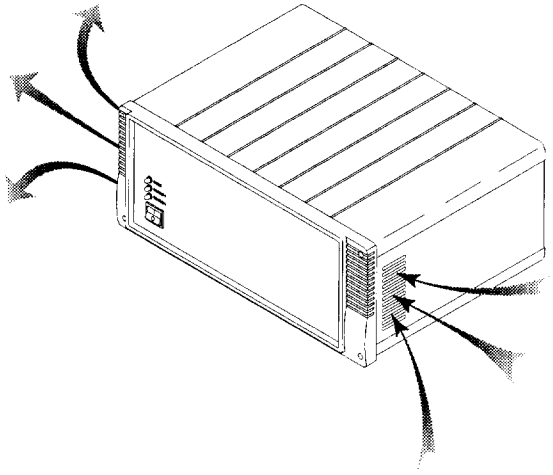
The next drawing shows the 6201's dimensions.

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



Airflow

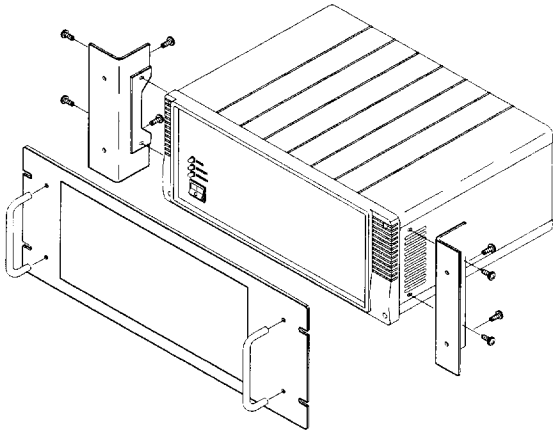
An internal fan forces air through the 6201, as shown in this drawing. If you place the 6201 near other equipment, be sure to maintain at least 2 inches of unrestricted airflow space around the chassis. Do not block the ventilation openings on the ends of the 6201.



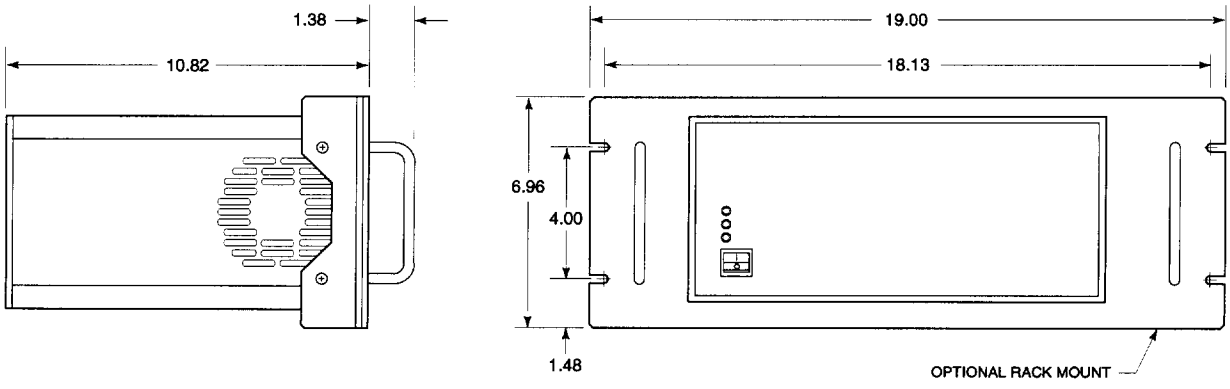
Panel Mounting

You can mount the 6201 to an equipment panel. This drawing shows the 6201 with optional rack mount kit, and illustrates how to attach the brackets, face plate and handles.

The side brackets need to be attached to the unit first, then the face plate and handles are attached to the brakets.



If you mount the 6201 in an enclosure with other equipment, be sure to maintain at least 2 inches of unrestricted airflow space around the chassis. The maximum allowable ambient temperature directly below the 6201 is 113°F (45°C). Do not mount heat producing equipment directly below the 6201.



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

② System Connections

This section describes procedures for the following 6201 system connections:

- End-of-travel and home limits
- Encoders
- Auxiliary +5VDC output
- Pulse cut input (**P-CUT**)
- Programmable inputs and outputs (including auxiliary outputs **OUT-A** and **OUT-B**)
- Output pull-up (**OUT-P**)
- Trigger inputs (**TRG-A** and **TRG-B**)
- RP240 Remote Operator Panel
- Joystick and analog inputs
- Extending cables

Refer to the bench test procedures in Chapter 2 for the following connections:

- Motor
- Power
- RS-232C communications

Refer to Chapter 4 for connection procedures on the following:

- PLC
- Thumbwheels
- RS-232C daisy-chain

NOTE

Refer to Chapter 5, *Hardware Reference*, for system specifications and detailed I/O circuit drawings and signal descriptions.

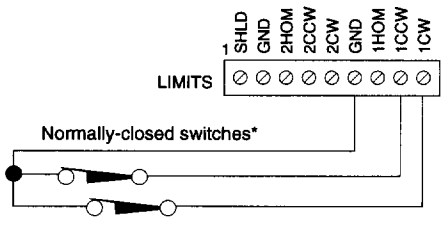
Active High/Active Low Conventions

Many people refer to a voltage level when referencing the state of inputs and outputs. Because current loops are less susceptible to electrical noise disturbances than voltage levels, Compumotor has adopted the convention of current loops in both its hardware and documentation. Therefore, an input/output that is "low" means that no current is flowing and a voltage may be present at the terminal. Conversely, if an input/output is "high," current is flowing and no voltage is present.

End-of-Travel Limit Connectionsconnections:end-of-travel limits

The 6201 provides CCW and CW end-of-travel limit inputs for both axes via the **LIMITS** connector. End-of-travel inputs serve as safety stops that prevent the load from crashing into mechanical stops and damaging equipment or injuring personnel. The drawing below illustrates typical end-of-travel limit switch connections.

End-of-Travel Limits



* LHLVL command changes active level of switches

NOTE

Motion will not occur until you do one of the following:

- Install limit switches
- Disable the limits with the LH command
- Change the active level of the limits with the LHLVL command

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

Use of hardware (and software) end-of-travel limits is discussed in detail in the End-of-Travel Limits *section in Chapter 4.*

Mount normally-closed switches such that the load forces them to open before it reaches the physical travel limit **(leave enough room for the load to stop)**. When the load opens the limit switch, the motor comes to a halt. The actual stopping distance depends on motor speed and the Hard Limit Deceleration (LHAD) setting. The motor will not be able to move in that same direction until you clear the limit (close the switch) **and** execute a move in the opposite direction (or you can disable the limits with the LH command, but this is recommended only if the motor is not coupled to the load).

CAUTION

To avoid stalling, the LHAD setting (deceleration rate) may need to be decreased when moving heavy loads or operating at high speeds.

Use the TLIM or TAS commands to check the status of the limit switches.

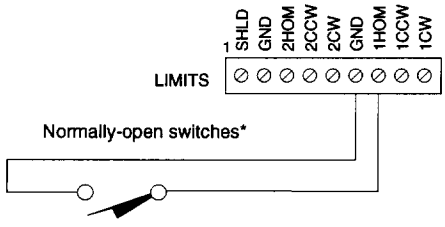
Home Limit Connections

Use the Home input to establish a *home* position or zero position reference point. The home input (TTL compatible) is used for homing the motor in open-loop (ENCØ) and closed-loop (ENC1) operations. If you are using encoder feedback, the encoder's Z channel pulse can be used in conjunction with the home switch to determine the home position. To use the encoder's Z channel, the HOMZ command must be enabled.

Homing is discussed in detail in the Homing *section in Chapter 4.*

The 6201 is shipped configured for use with normally-open home switches. You can, if you wish, reverse the home input polarity (to use normally-closed switches) with the HOMLVL command. The most common way to use the *home* switch is to mount it at a *home reference position*. The drawing below illustrates typical home limit switch connections to the 6201.

Home Limit



* HOMLVL command changes active level of switch

CAUTION

Compumotor cannot guarantee proper homing performance with the home and end-of-travel limit inputs tied together.

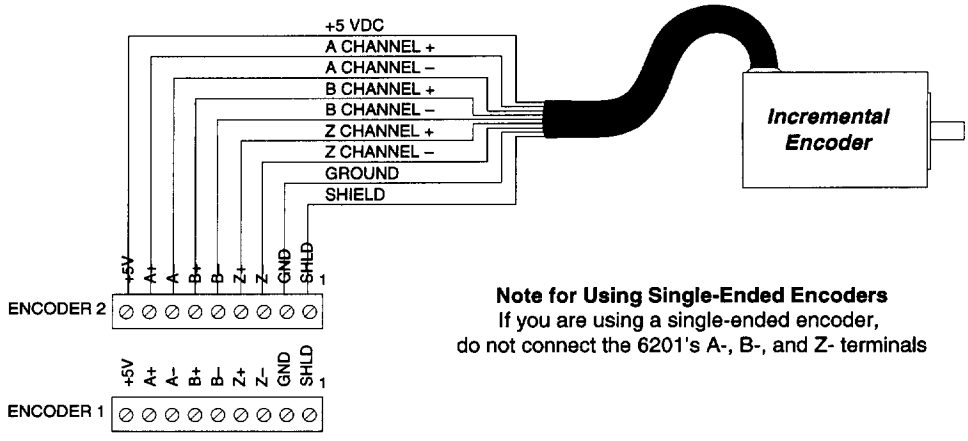
Encoder Connections

The 6201 supports up to two incremental encoders. If you use encoders other than those supplied by Compumotor, pay special attention to the following requirements:

- Use incremental encoders with two-phase quadrature output. An index or *Z channel* output is optional. **Differential outputs are recommended.**
- It must be a 5V encoder to use the 6201's **+5V** output. Otherwise, it must be separately powered, with TTL-compatible or open-collector outputs.
- The decoded quadrature resolution should be less than the motor resolution by a factor of four to take advantage of the 6201's position maintenance capability.

The illustration below shows the wiring techniques that you must use to connect encoders to the 6201. Refer to Chapter 5 for the 6201's encoder input circuit drawing.

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



Note for Using Single-Ended Encoders
If you are using a single-ended encoder, do not connect the 6201's A-, B-, and Z- terminals

Pin Outs

Each axis has a 9-pin removable screw terminal connector for encoder connections. The pin-out description for the **ENCODER** connectors is provided below.

Pin	In/Out	Name	Compumotor E Series Encoder Cable Colors	Description
9	OUT	+5V	Red	+5VDC output to power the encoder
8	IN	A Channel +	Brown	A+ channel quadrature signal from incremental encoder
7	IN	A Channel -	Brown/White	A- channel quadrature signal from incremental encoder
6	IN	B Channel +	Green	B+ channel quadrature signal from incremental encoder
5	IN	B Channel -	Green/White	B- channel quadrature signal from incremental encoder
4	IN	Z Channel +	Orange	Z+ channel quadrature signal from incremental encoder
3	IN	Z Channel -	Orange/White	Z- channel quadrature signal from incremental encoder
2	-----	Ground	Black	Isolated logic ground
1	-----	Shield	Shield	Internally connected to chassis ground (earth)

Auxiliary +5V Output Connection

The 6201 provides +5VDC output on the **AUX**, **ENCODER**, and **RP240** connectors. As much as 1.5A is available. 1.5A is sufficient power for the total load on all the I/O connectors. For example, using two encoders (each drawing 250mA) and one RP240 (drawing 100mA), 900mA would be left for other purposes. The drawing below illustrates example connections for powering the output pull-up.

Pulse Cut (P-CUT) Connection

The **P-CUT** (pulse cut) input is located on the **AUX** connector. The 6201 is shipped with **P-CUT** wired to **GND** (isolated logic ground). This input must be grounded to allow motor motion. A loss of connection to ground on this input will shut off step pulses independent of microprocessor control. If the **P-CUT** input is not grounded when motion is commanded, the error message `WARNING: PULSE CUTOFF ACTIVE` will be displayed.

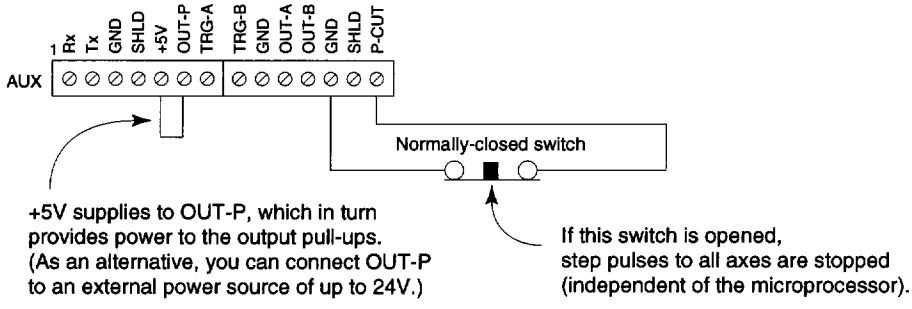
See the illustration below for an example connection using a normally-closed switch. Opening the switch will stop the pulse output instantly. This will also unconditionally stop program execution and, if error bit #9 of the `ERROR` command is enabled, the error program specified by the `ERRORP` command will be initiated. Motor position may be lost when the **P-CUT** input is activated (encoder position will still be accurate). You can check the status of **P-CUT** with the `TINO` or `INO` commands, and the `TER` or `ER` commands.

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

Programmable Inputs & Outputs Connections

The **PROGRAMMABLE INPUTS** connector provides 24 programmable inputs and the **PROGRAMMABLE OUTPUTS** connector provides 24 programmable outputs. Two additional (and functionally identical) programmable outputs, **OUT-A** and **OUT-B**, are available on the **AUX** connector. Two additional *fast* trigger inputs are also available on the **AUX** connector, but due to their functional differences they are discussed later in the *Fast Triggers* section. **All these inputs and outputs are optically isolated and TTL compatible.**

The outputs are pulled up using the **OUT-P** pin on the **AUX** connector (see illustration below). If +5VDC logic is to be used, connect **OUT-P** to pin 5, **+5V**.



Changing inputs from sourcing to sinking

The inputs are internally pulled up to +5V by means of an internal jumper. If you wish to have the inputs sink current instead of source current, you can change the jumper setting (refer to *Optional DIP Switch and Jumper Settings* in Chapter 5 for instructions). For compatibility with equipment operating at 24VDC, the inputs may be pulled up to 24VDC by using an external power supply.

These I/O are typically used with normally-open or normally-closed switches or sensors; however, they can also be used with I/O module racks, PLCs, and thumbwheels (including the Compumotor TM8).

If you are using PLCs or thumbwheels, refer to the connection instructions and application considerations provided in the *Programmable Inputs and Outputs* section of Chapter 4.

Also provided in the *Programmable Inputs and Outputs* section are instructions for defining and controlling programmable inputs and outputs via programs written with the 6000 Series programming language.

Programmable I/O Pin Outs

The following table lists the pin outs on the two 50-pin flat cable headers labeled **PROGRAMMABLE INPUTS** and **PROGRAMMABLE OUTPUTS**. Refer to Chapter 5, *Hardware Reference*, for internal I/O schematics.

PROGRAMMABLE INPUTS Connector			
Pin #	**Function**	**Pin #**	**Function**
49	+5 VDC	23	Input #13
47	Input #1 (LSB)	21	Input #14
45	Input #2	19	Input #15
43	Input #3	17	Input #16
41	Input #4	15	Input #17
39	Input #5	13	Input #18
37	Input #6	11	Input #19
35	Input #7	09	Input #20
33	Input #8	07	Input #21
31	Input #9	05	Input #22
29	Input #10	03	Input #23
27	Input #11	01	Input #24 (MSB)
25	Input #12		

PROGRAMMABLE OUTPUTS Connector			
Pin #	**Function**	**Pin #**	**Function**
49	+5 VDC	23	Output #13
47	Output #1 (LSB)	21	Output #14
45	Output #2	19	Output #15
43	Output #3	17	Output #16
41	Output #4	15	Output #17
39	Output #5	13	Output #18
37	Output #6	11	Output #19
35	Output #7	09	Output #20
33	Output #8	07	Output #21
31	Output #9	05	Output #22
29	Output #10	03	Output #23
27	Output #11	01	Output #24 (MSB)
25	Output #12		

NOTE: All even-numbered pins are connected to logic ground (DC ground).
LSB = least significant bit; MSB = most significant bit

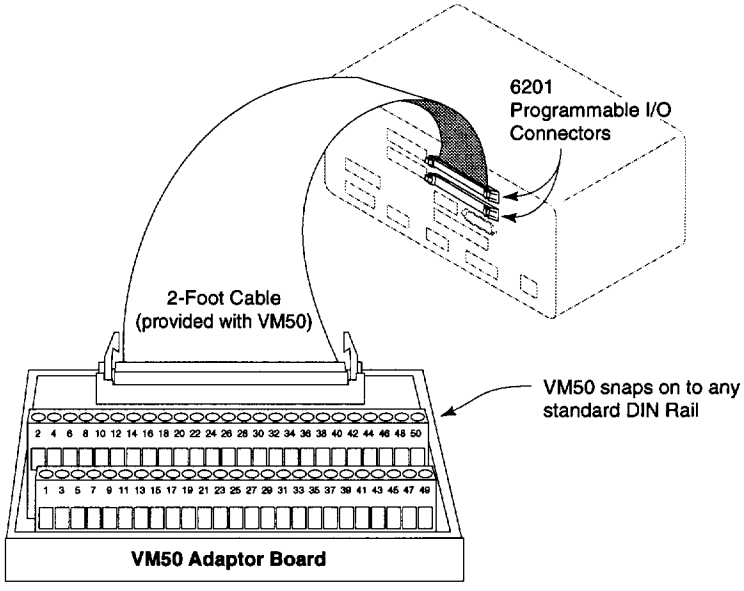
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Optional VM50 Adaptor

If you wish to use screw terminal connections for the 24 programmable I/O, Compumotor offers the VM50 adaptor (p/n VM50). If you wish to use screw terminal connections for both the 24 inputs **and** the 24 outputs, you will need two VM50 adaptors.

The pin numbers on the VM50's screw terminals correspond to the same pin outs on the **PROGRAMMABLE INPUTS** and **PROGRAMMABLE OUTPUTS** connectors. The VM50 simply attaches to the 6201 via the 2-foot, 50-pin ribbon cable that comes with the VM50 (see drawing below).

To order the VM50, contact your distributor or ATC.



Fast Trigger Input Connections

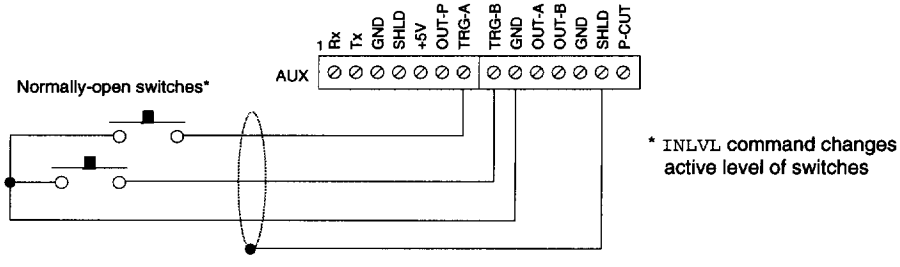
The 6201 provides two fast trigger inputs. Like the programmable inputs described earlier, trigger inputs can be connected to PLC outputs, discrete switches, or electronic sensors, and are monitored under program control. The status of triggers A and B is represented respectively by bits 25 and 26 in the `INFNC`, `INLVL`, `TIN`, `[IN]`, and `ONIN` commands.

☞ *Position Capture and Registration features are described in the* Programmable Inputs and Outputs *section of Chapter 4.*

The difference between the trigger inputs and the regular 24 programmable inputs (discussed earlier) is that the triggers can be programmed with the `INFNCi-H` command to function as *position* (motor & encoder) *capture* and *registration* inputs, latching positions within 50µs.

The trigger switches can be normally-open (default) or normally-closed, depending on the `INLVL` command setting. Using the `WAIT` command, the 6201 can be programmed to wait until one or more inputs switch to a desired state.

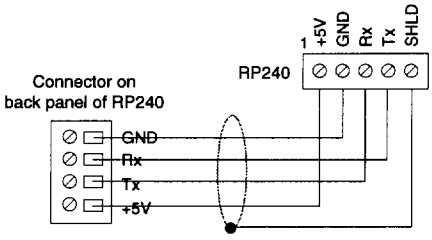
The drawing below illustrates normally-open (not grounded) trigger connections to the **AUX** connector.



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

RP240 Remote Operator Panel Connections

Using a four-wire shielded cable, connect the RP240 to the 6201's **RP240** connector (see below). For cable lengths up to 50 feet, use 20 AWG wire (*cable lengths longer than 50 feet are not recommended*). Refer to the ***RP240 User Guide*** for mounting instructions.



NOTE

In order for the 6201 to recognize the RP240, the RP240 connection must be made prior to powering up or resetting the 6201. If you **connect the RP240 to the 6201 before powering up or resetting the 6201**, the 6201 will recognize the RP240 and send the `*RP24Ø CONNECTED` message to the RS-232C terminal. If the 6201 does not detect the RP240 upon power up or reset, then the following message will be sent to the RS-232C terminal: `*NO REMOTE PANEL`.

Joystick and Analog Input Connections

You can use the three analog inputs on the **JOYSTICK** connector for joystick control, feedrate override of the axes, and/or as a low-resolution analog input (8-bit A/D) for process control.

Refer to the Analog Inputs *section in Chapter 4 for a detailed discussion of joystick control and feedrate override.*

The input range of the analog input is 0V to 2.5V. A joystick with a linear taper 5KΩ potentiometer (pot) with 60° of travel is recommended *(the pot has 300° of travel, but typically only 60° is usable with a joystick)*. The pot should be adjusted so that its resistance is close to 0Ω when the joystick is all the way to one side, and about 1KΩ when the joystick is all the way to the other side. Also, connect a 1KΩ resistor between the analog input and +5V to provide optimum noise suppression (see illustration below).

Joystick Connector Pin Outs

The **JOYSTICK** connector is a 25-pin D connector. The pin-out descriptions are provided in the table below. The 6201's internal analog input circuit diagram is provided in Chapter 5, *Hardware Reference*.

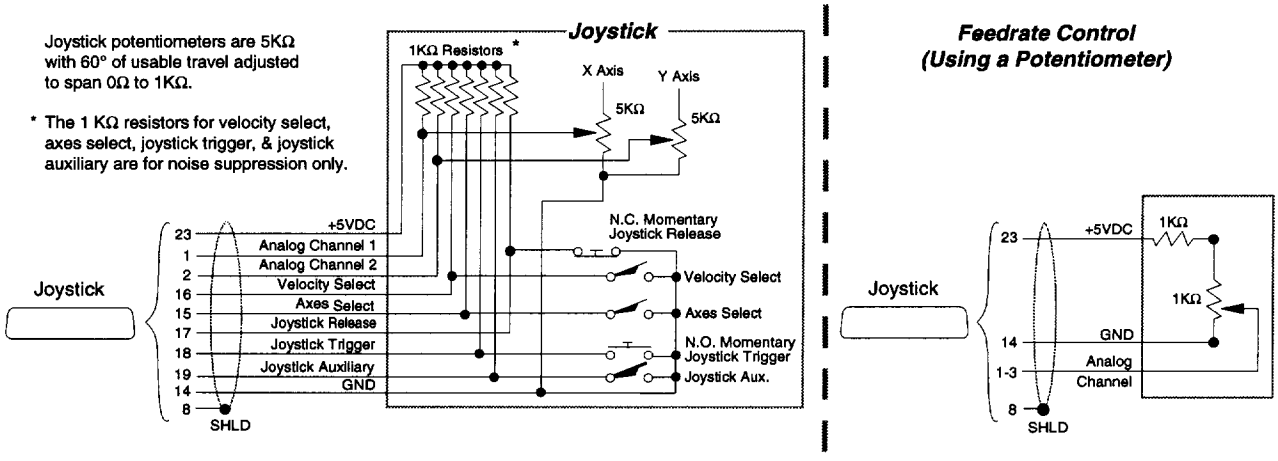
Pin	In/Out	Name	Description
1	IN	Analog Channel 1	Analog input for feedrate control or joystick control of axis (can override with the `ANVO` command)
2	IN	Analog Channel 2	Analog input for feedrate control or joystick control of axis (can override with the `ANVO` command)
3	IN	Analog Channel 3	Analog input for feedrate control or joystick control of axis (can override with the `ANVO` command)
8	—	Shield	Shield
14	—	Ground	Isolated ground
15	IN	Axes Select	If only using one analog input, you can use this input to alternately control axes 1 or 2
16	IN	Velocity Select	Input to select high or low velocity range (as defined with `JOYVH` or `JOYVL` command)
17	IN	Joystick Release	Normally closed. Input to release the 6201 from joystick mode (`JOY`). Same as issuing the `!JOYØØ` command. Program execution will continue with the first statement after the joystick enable (`JOY1`) command.
18	IN	Joystick Trigger	Status of this active-low input can be read by a program (using the `INO` or `TINO` commands) to control program flow, or to enter the 6201 into joystick mode.
19	IN	Joystick Auxiliary	Status of this active-low input can be read by a program (using the `INO` or `TINO` commands) to control program flow, or to teach positions to a program.
23	OUT	+5VDC (out)	+5VDC power output

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

Analog Inputs

You can use the analog inputs for feedrate control of both axes, or for joystick control of the axes. When used for feedrate control, an analog input may be used to control the overall velocity of the two axes from 0% to 100%. When used for joystick control of axes, an analog input can command an axis velocity from full CW to full CCW. The following drawing illustrates typical connection examples for both joystick control and feedrate control.

☞ *Analog inputs can be overridden with the* ANVO *command.*



Axes Select Input

You can specify two configurations (JOYAXH and JOYAXL) that determine which axes are controlled by which channels. The axes select input allows you to select the current configuration. An axes select input *high* references the JOYAXH command. An axes select input *low* references the JOYAXL command.

One possible configuration is as follows: With the axes select input high, analog channel #1 controls axis one and analog channel #2 controls axis two (JOYAXH1,2). With the axes select input low, analog channel #3 controls both axes (JOYAXL3,3).

Velocity Select Input

This input may be used to select either the high or low velocity range as defined with the JOYVH and JOYVL commands, respectively. The high range could be used to quickly move to a location while the low range could be used for accurate positioning. If this input is not connected, the low velocity range (JOYVL) is selected. Refer to the illustration above.

Joystick Release Input

The joystick release input allows you to indicate to the 6201 that you have finished using the joystick and program execution may continue with the next statement. When a program enables joystick control of motion, program execution will stop and then resume when the user is finished with joystick mode (assuming the Continuous Command Execution Mode is disabled with the COMEXCØ command).

The joystick release input has an internal pull-up resistor to +5V. When the joystick release input is not grounded, joystick enable statements (JOY1) will be disabled upon execution. To enable the joystick mode, the joystick release input must be inactive (connected to ground). Refer to the illustration above.

Joystick Trigger Input

The status of this input can be read by a program and may be used to control program flow (see INO and TINO command). Refer to the illustration above.

Joystick Auxiliary Input

The status of this input can be read by a program and may be used to control program flow (see INO and TINO command). Refer to the illustration above.

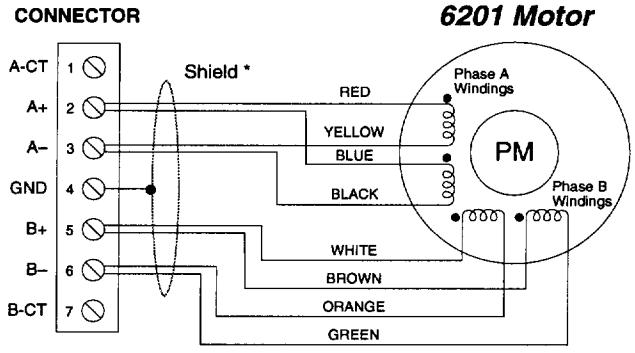
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Extending 6201 System Cables

This section describes options for extending 6201 system motor, encoder, and I/O cables. If you wish to order longer cables, contact Compumotor's Customer Service Department at (800) 722-2282 or contact your local Compumotor Distributor or ATC.

6201-to-Motor Cables

Compumotor 6201 Series motors are supplied with a permanently attached 8-conductor cable that is 10 feet long. The next drawing shows cable connections inside the motor, and connector wiring for parallel motor windings.



* Shield is internally connected to the motor's case

You can extend the motor cable.; use 18AWG or greater diameter wire. However, cables longer than 10 feet may degrade system performance.

To extend the cable, attach the new cable to the connector, or splice it into the original cable. Make sure that the motor windings remain wired in parallel (see the drawing above).

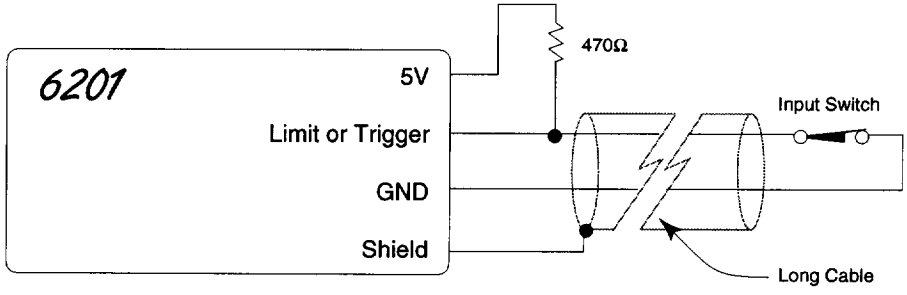
6201-to-Encoder Cables

6201 series motors may be ordered with an optional Compumotor E Series encoder. The encoder is supplied with a permanently attached 10-foot cable. The maximum cable length between the encoder and the 6201 is 100 feet. If you wish to lengthen the encoder cable yourself, use 24 AWG wire. Encoder cables should be shielded with the shield connected to **SHLD** (pin 1 on the **ENCODER** connector).

I/O Cables

To avoid interference from external noise, you must shield all I/O cables, regardless of the length. The maximum length of cables is determined by the environment in which the equipment will be used. For cables longer than 50 feet or in electrically noisy environments, you should follow the guidelines below (refer also to illustration below).

- 22 AWG wire is the minimum recommended wire size.
- Use twisted pair shielded cables and connect the shield to the **SHLD** terminal on the 6201 connector. Leave the other end of the shield disconnected.
- Do not route these signals in the same conduit or wiring trays as high-voltage AC wiring.
- Limit and trigger inputs are internally pulled up to +5VDC and are TTL compatible. In electrically noisy environments or when using long cable lengths, use an external pull-up resistor with a value of 330Ω to 2.2KΩ between the input and +5V. The external resistor will lower the input impedance and will make the input less susceptible to electrical noise.



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

③ Motor Mounting

Rotary stepper motors should be mounted using flange bolts and positioned with the centering flange on the front face. Foot-mount or cradle configurations are not recommended because the torque of the motor is not evenly distributed around the motor case. Any radial load on the motor shaft is multiplied by a much longer lever arm when a foot mount is used rather than a face flange.

WARNING

Improper mounting can compromise system performance and **jeopardize personal safety.**

The motors used with the 6201 can produce very high torque values. These motors can also produce high accelerations. This combination can shear shafts and mounting hardware if the mounting is not adequate. High accelerations can produce shocks and vibrations that require much heavier hardware than would be expected for static loads of the same magnitude. The motor, under certain profiles, can produce low-frequency vibrations in the mounting structure. These vibrations can also cause metal fatigue in structural members if harmonic resonance is induced by the move profiles you are using. A mechanical engineer should check the machine design to ensure that the mounting structure is adequate. **Do not attach the load to the motor yet. Coupling the load to the motor is discussed later in this chapter.**

CAUTION

Consult a Compumotor Applications Engineer (800-358-9070) before you machine the motor shaft. Improper shaft machining can destroy the motor's bearings, and may void the warranty. *Never* disassemble the motor (disassembly will cause a significant loss of torque).

④ Installation Verification

Before you couple the motors or encoders to the loads, perform the installation verification discussed in this section.

WARNING

This installation verification section assumes that you have **not** coupled the motors or the encoders to the loads. **Do not proceed until you are sure the loads are not coupled.**

Step 1 Return to the *Test Procedure* in Chapter 2 to test the motor interface and the RS-232C interface.

Step 2 Use the information in the following table to test the features appropriate to your application. If you receive responses other than those expected, check your system wiring and refer to the command description in the ***6000 Series Software Reference Guide*** for assistance.

NOTE

The following table is based on the assumption that you have **not** changed the active levels of the 6201's inputs and outputs. Verify these settings with the following *status* commands:

Command Entered	Response Should Be
INLVL	*INLVLØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ
HOMLVL	*HOMLVLØØ
LHLVL	*LHLVLØØØØ
OUTLVL	*OUTLVLØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ

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

Connections	Test Procedure	Response Format (left to right)
End-of-travel and Home Limits	**NOTE:** If you are not using end-of-travel limits, issue the Disable Limits (LHØ,Ø) command and ignore the first two bits in each response field. 1. Close the end-of-travel switches and open the home switches. 2. Enter the TLIM command. The response should be *TLIM11Ø_11Ø. 3. Open the end-of-travel switches and close the home switches. 4. Enter the TLIM command. The response should be *TLIMØØ1_ØØ1. 5. Close the CW end-of-travel switch on axis #1 (1CW) and open the home switch on axis #2 (2HOM). 6. Enter the TLIM command. The response should be *TLIM1Ø1_ØØØ.	TLIM response: bit 1= axis 1 CW limit bit 2 = axis 1 CCW limit bit 3 = axis 1 home limit bit 4 = axis 2 CW limit bit 5 = axis 2 CCW limit bit 6 = axis 2 home limit
Encoder Feedback	1. Enter the ENCØ command to enable the motor step mode. Enter the PSETØ,Ø command to set the motor position on both axes to zero. Enter the TPM command to determine the motor position. The response should be *TPM+Ø,+Ø (both motors are at position zero) 2. Enter the ENC1 command to enable the encoder step mode. Enter the PSETØ,Ø command to set the encoder position on both axes to zero. Enter the TPE command to determine the encoder position. The response should be *TPE+Ø,+Ø (both encoders are at position zero) 3. Enter the GO command. If your drive's resolution is 25,000 steps/rev and you have not changed the default distance and resolution settings, the motors will move approximately one revolution in the clockwise direction (as viewed from the flange end). If the encoders are coupled to the motor, the encoders will also experience a one-rev CW move. **Encoder Not Coupled:** If the encoders are not coupled to the motors, manually rotate both encoders one revolution in the clockwise direction. 4. Enter the TPE command to determine the encoder position. The response should be (approximately) *TPE+4ØØØ,+4ØØØ (both encoders are at position +4000). 5. Enter the ENCØ command to enable the motor step mode. Issue the TPM command to determine the motor position. The response should be *TPM+25ØØØ,+25ØØØ (both motors are at position +25000).	TPE response (encoder counts): ±encoder 1, ±encoder 2 TPM response (motor counts): ±motor 1, ±motor 2
Programmable Inputs (incl. triggers)	1. Open the input switches or turn off the device driving the inputs. 2. Enter the TIN command. The response should be *TINØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ. 3. Close the input switches or turn on the device driving the inputs. 4. Enter the TIN command. The response should be *TIN1111_1111_1111_1111_1111_1111_11.	TIN response: bits 1-24 = prog. inputs 1 - 24 bits 25 & 26 = TRG-A & TRG-B
Programmable Outputs	1. Enter the OUTALL1,26,1 command to turn on (sink current on) all outputs. 2. Enter the TOUT command. The response should be *TOUT1111_1111_1111_1111_1111_1111_11. 3. Enter the OUTALL1,26,Ø command to turn off all outputs. 4. Enter the TOUT command. The response should be *TOUTØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ.	TOUT response: bits 1-24 = prog. outputs 1 - 24 bits 25 & 26 = OUT-A & OUT-B

RP240	1. Cycle power to the 6201.
	2. If the RP240 is connected properly, the RP240's status LED should be green and one of the lines on the computer or terminal display should read *RP24Ø CONNECTED.
	If the RP240's status LED is off, check to make sure the +5V connection is secure.
	If the RP240's status LED is green, but the message on the terminal reads *NO REMOTE PANEL, the RP240 Rx and Tx lines are probably switched. Remove power and correct.
	3. Assuming you have not written a program to manipulate the RP240 display, the RP240 screen should display the following:
	`COMPUMOTOR 6201 MOTION CONTROLLER`
	`RUN JOG STATUS DISPLAY ETC`	

Joystick Inputs and Pulse Cut	1. Open the input switches or turn off the device driving the inputs. 2. Enter the TINO command. The response should be *TINOØØØØ_ØØØØ. 3. Close the input switches or turn on the device driving the inputs. 4. Enter the TINO command. The response should be *TINO1111_11ØØ.	TINO response: bit 1= joystick auxiliary bit 2= joystick trigger bit 3 = joystick axes select bit 4 = joystick velocity select bit 5= joystick release bit 6 = pulse cut input bits 7 & 8 are not used

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

⑤ Attaching the Load

This section discusses the main factors involved when attaching the load to the motor. Three types of misalignments 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 the center lines of two shafts intersect at an angle other than zero degrees.
- End Float: A change in the relative distance between the ends of two shafts.

The motor and load should be aligned as accurately as possible. Excessive misalignment may degrade your system's performance. The type of misalignment in your system will affect your choice of coupling.

Couplings

There are three types of shaft couplings: single-flex, double-flex, and rigid. Like a hinge, a single-flex coupling accepts angular misalignment only. A double-flex coupling accepts both angular and parallel misalignments. Both single-flex and double-flex couplings, depending on their design, may or may not accept end-play. A rigid coupling cannot compensate for any misalignment.

Coupling Manufacturers

HELI-CAL
901 West McCoy Lane
P.O. Box 1069
Santa Maria, CA 93456
(805) 928-3851

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

For unusual motor installations contact a Compumotor Applications Engineer for assistance at (800) 358-9070.

⑥ Tuning

Resonance

Resonance exists in all stepper motors and is a function of the motor's mechanical construction. It can cause the motor to stall or run rough at low speeds. Unlike full step controllers, the 6201's microstepping capability allows you to operate a motor smoothly at low speeds. Since the resonant frequency will change when the load is attached to the motor, the 6201 has tuning capability to maintain maximum smoothness.

Motors that will not accelerate past 1 rps may be stalling due to resonance. You can add inertia to the motor shaft, thus changing the resonant frequency, by putting a drill chuck on the shaft. The drill chuck may provide enough inertia to test the motor when it is not loaded. Most mechanical systems have some friction which will provide damping and alleviate resonance problems. In extreme cases, a viscous damper may also be needed.

Mid-Range Instability

All step motors are subject to mid-range instability, also referred to as parametric oscillations. These oscillations may stall the motor at speeds from 6 to 16 rps.

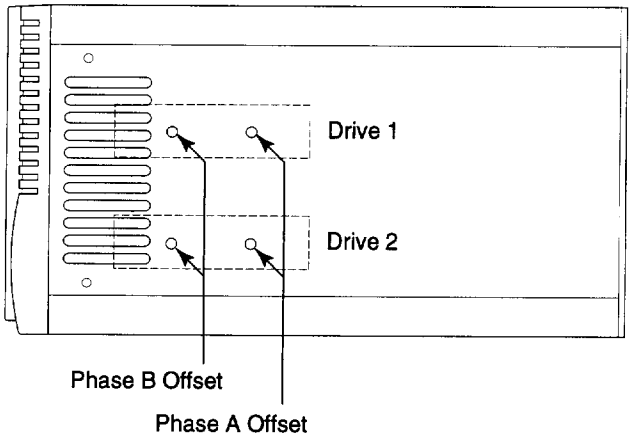
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tuning Procedures

You can tune the 6201's internal drives to minimize resonance and optimize smoothness by adjusting the small single turn potentiometers (pots) on the right side of the unit (see illustration below).

- Phase A Offset: Adjusts DC offset of the phase current for Phase A.
- Phase B Offset: Adjusts DC offset of the phase current or Phase B.

Since tuning is affected by operating current, you may have to adjust these pots during the configuration or installation process. *For best results, the 6201 should be on. The motors should be connected to the load and warmed up for 30 minutes prior to tuning.*



Gauging Motor Resonance

There are several methods that you can use to determine the level of motor resonance in your system.

Tachometer Method

Use an oscilloscope to gauge the output of a tachometer attached to the motor shaft. The tachometer will output a DC voltage, proportional to speed. This voltage will oscillate around an average voltage when the motor is resonating. The amplitude of this oscillation will be at its maximum when you run the motor at its *resonance speed*. The goal of this tuning method is to tune the motor for its lowest oscillation amplitude.

Sounding Board Method

You can practice your tuning skills with an unloaded motor placed on a sounding board or table. When you command a velocity that is near the motor's *resonance speed*, the phenomenon will cause an audible vibration. The goal of this tuning method is to tune the motor for the least amount of vibration.

Stethoscope Method

When you tune your motor under loaded conditions, you can hear the audible vibration caused by the motor's natural frequency by placing the tip of a screw driver against the motor casing and placing the handle of the screw driver close to your ear (as you would a stethoscope). You will also be able to hear the different magnitudes of vibration caused by the motor's natural frequency. The goal of this tuning method is to tune the motor for the least amount of vibration.

Touch Method

After you have had some experience with tuning, you should be able to locate the motor's ***resonance speed*** by placing your fingertips on the motor shaft and adjusting the motor's velocity. Once the ***resonance speed*** is located, you can tune the motor for maximum smoothness in the same way.

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

Tuning the Drive to the Motor

To tune the 6201's internal drives, follow the directions below for each motor:

Step 1 Command the unit so that the motor is running at maximum roughness, as shown below for the 1st speed motor resonance for 6201 series motors.

Resonance values in this table are for unloaded motors.

Motor Size	1st Speed Resonance	2nd Speed Resonance
6201-57-51	1.8 rps	3.6 rps
6201-57-83	1.8 rps	3.6 rps
6201-57-102	1.8 rps	3.6 rps
6201-83-62	1.4 rps	2.8 rps
6201-83-93	1.4 rps	2.8 rps
6201-83-135	1.4 rps	2.8 rps

Step 2 Adjust Offsets A and B for best smoothness. To do this, first adjust one offset for best smoothness, then adjust the other offset for best smoothness. You may need to repeat these adjustments several times.

Step 3 Double the motor speed (2nd speed resonance) until the motor again runs rough.

Step 4 Adjust offsets A and B again for best smoothness.

Step 5 Repeat above steps until no further improvement is noted.

What's Next?

This chapter has provided you with mounting, connection, and test procedures for your 6201 system.

If you intend to use thumbwheels or PLCs, or if you intend to daisy-chain multiple 6201s, refer to the connection instructions and application considerations provided in Chapter 4.

Determine Your Application Requirements First

Applications can vary greatly from one to another. Consequently, the 6201 is equipped with many features to satisfy a wide variety of application requirements—*but not all features are appropriate for every application.* Therefore, you must first determine the necessary motion features you need for your application. Once you have done that, you can proceed to Chapter 4, *Feature Implementation*, to read about the 6201's features and how to implement them in your application. Refer to the ***6000 Series Software Reference Guide*** for programming guidelines and detailed descriptions of each command.

You will develop your application by creating and refining motion programs using the 6000 Series Command Language. You will most likely use one of the two support software tools, *Motion Architect*® or the *6000 DOS Support Disk*,to aid in your programming effort. These software tools are discussed in Chapter 4.

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

C H A P T E R ④

Feature Implementation

The information in this chapter will enable you to understand and implement the 6201's features into your application. For information on programming guidelines and command descriptions, refer to the ***6000 Series Software Reference Guide.***

Before You Proceed With This Chapter

CAUTION

To ensure proper installation and operator safety, you should complete all the installation and test procedures provided in Chapter 2, *Getting Started*, and Chapter 3, *Installation*, before proceeding with any of the motion programming examples in this chapter.

6000 Series Software Reference Guide

Since this chapter often refers to the 6000 Series Command Language employed by the 6201's operating system, keep the ***6000 Series Software Reference Guide*** nearby as a programming reference.

Compumotor Bulletin Board Service

Compumotor offers a bulletin board service (BBS)—free of charge. The BBS allows you to send or receive messages and download the latest revisions of Compumotor software (such as support software, sample programs, and programming tools).

To dial in, you must have at least a 2400 baud modem with your computer. Set the baud rate to 2400, 8 data bits, 1 stop bit, and NO parity; any communications program such as Procomm™, Crosstalk™, or PC-Talk™ should allow you to set these. The BBS number is 707-584-4059.

Once connected, you will be asked some questions about yourself. Take your time in answering them, because you will only have to do this once. When you have completed the personal information, you are free to explore the services of the bulletin board.

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

Basic Motion Control Concepts

If you are unfamiliar with motion control concepts such as motion profiles, mechanical factors, positional accuracy, and repeatability refer to the *Engineering Reference* section of the Parker Compumotor/Digiplan catalog.

Support Software

The 6201 is shipped with two support software tools, *Motion Architect®* and the *6000 DOS Support Disk*. CompuCAM™, CAD-to-Motion software that translates DXF, HP-GL, and G-code files into 6000 motion programs, is available for purchase through your local distributor or ATC.

6000 DOS Support Disk

The *6000 DOS Support Disk* (p/n 95-012266-01) contains a program that provides terminal emulation and program editing capabilities specifically designed for use with any 6000 Series stand-alone product. Also included on the disk are sample 6000 command language programs. For more detailed user information, refer to the ***6000 DOS Support Disk Quick Reference***.

Motion Architect®

Motion Architect is an intuitive Microsoft® Windows™ based programming tool. A brief description of Motion Architect's basic features is provided below. For more detailed user information, refer to the ***Motion Architect User Guide***.

- **System Configurator and Code Generator**: Automatically generate controller code of basic system set-up parameters (I/O definitions, encoder operations, etc.).
- **Program Editor**: Create blocks or lines of 6201 controller code, or copy portions of code from previous files. You can save program editor files for later use in BASIC, C, etc., or in the terminal emulator or test panel.
- **Terminal Emulator**: Communicating directly with the 6201, the terminal emulator allows you to type in and execute controller code and transfer code files to and from the 6201.
- **Test Panel and Program Tester**: You can create your own test panel to run your programs and check the activity of I/O, motion, system status, etc. This can be invaluable during start-ups and when fine tuning machine performance.
- **On-line Context-sensitive Help and Command Reference**: These on-line resources provide help information about Motion Architect, as well as interactive access to the contents of the ***6000 Series Software Reference Guide***.

System Configuration

To ensure proper system operation, configure the following parameters at the beginning of the application program.

- Number of axes in use
- Allocation of user memory space
- Drive fault level

NOTE

Except for memory allocation (`MEMORY`), these configuration parameters are not saved in the 6201's battery-backed RAM. To ensure that these parameters are used every time you power up the 6201, place them in the start-up program (`STARTP`). For information on creating a start-up program, refer to the *Programming Guide* section in the ***6000 Series Software Reference Guide.***

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

Number of Axes

By configuring the number of axes in use, you limit the number of axes you can control. This may be desired if you are only using one of the two axes available. The INDAX command configures the number of axes. INDAX2 (the default setting) allows both command fields to be entered (e.g., A1,1). If you enter INDAX1, instead of entering A1,1 you should enter A1, and all responses from the 6201 will also only show the one field; if you enter the command A the response will be *A1.

Memory Allocation

The 6201 provides 40,000 bytes of memory to divide between user programs and contouring paths. The 6201-M (expanded memory option) provides 150,000 bytes.

Programs defined with the DEF command are stored in the memory allocated for program storage. Paths compiled with the PCOMP command are stored in the memory allocated to contouring paths. This memory allocation can be modified with the MEMORY command.

The default allocation for a standard 6201 is MEMORY214ØØ,186ØØ, which allocates 21,400 bytes for user programs and 18,600 bytes for contouring paths. The default allocation for the 6201-M is MEMORY756ØØ,744ØØ.

For example, if you are not using contouring, you may want to redefine the memory partitions to minimize contouring path storage and maximize user program storage. However, since the smallest memory partition allowed is 1,000 bytes, you must use the MEMORY39ØØØ,1ØØØ command (or MEMORY149ØØØ,1ØØØ command for the 6201-M option).

CAUTION

Issuing a memory allocation command (e.g., MEMORY1ØØØ,39ØØØ) will erase all existing programs and compiled contouring path segments. However, issuing the MEMORY command by itself (to request the status of how the memory is allocated) will not affect existing programs or path segments.

Drive Fault Level

The DRFLVL command sets the drive fault level to active high. The default drvie fault level for the 6201 is DRFLVL11. *Once the drive fault level has been configured, you must enable the drive fault input with the INFEN1 command.*

Scaling

The scaling commands allow you to scale acceleration, deceleration, velocity, and position to values that are appropriate for the application. The SCALE, SCLA, SCLV, SCLD, PSCLA, PSCLV and PSCLD commands are used to implement the scaling features.

Define scaling factors before defining a program or contouring path

To maximize the efficiency of the 6201's microprocessor, the scaling multiplications are performed when the program or contouring path is defined or downloaded. Therefore, you must enable scaling (SCALE) and define the scaling factors (SCLD, SCLA, SCLV, PSCLA, PSCLV, PSCLD) **prior** to defining (DEF), uploading (TPROG), or running (RUN, PRUN) the program or path. This can be accomplished by defining all scaling factors via a terminal emulator just before defining or downloading the program or path; you should also put the scaling factors into the startup (STARTP) program.

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

Default: Scaling is Disabled

The default condition of the 6201 is with scaling disabled (SCALEØ):

- All programmed acceleration and deceleration values are entered in motor revs/sec^2; these values are internally multiplied by the drive resolution (DRES) value to obtain acceleration and deceleration values in motor steps/sec^2 for the motion trajectory calculations. The entered values are always in reference to motor steps, not encoder steps, regardless of the ENC command setting.
- All programmed velocity values are entered in motor revs/sec; these values are internally multiplied by the drive resolution (DRES) value to obtain velocity values in motor steps/sec for the motion trajectory calculations. The entered values are always in reference to motor steps, not encoder steps, regardless of the ENC command setting.
- All non-path distance values (D, PSET, and REG) are entered in motor steps or encoder steps, depending on the ENC command setting. All path (*contouring*) distance values (PARCM, PARCOM, PARCOP, PARCP, PLC, PLIN, PRTOL, and PWC) are entered in motor steps, regardless of the ENC command setting.

Acceleration & Deceleration Scaling (SCLA/PSCLA)

If scaling is enabled (SCALE1), all accel/decel values entered are internally multiplied by the acceleration scaling factor (SCLA) to convert user units/sec^2 to motor steps/sec^2. Acceleration scaling affects the following commands: A, AD, HOMAD, JOGA, JOGAD, JOYA, JOYAD, LHAD, and LSAD.

Path Scaling: If you are using the 6201's linear interpolation feature or the contouring feature, the PA and PAD command values are internally multiplied by the path acceleration scaling factor (PSCLA).

NOTE

All acceleration and deceleration values are always in reference to motor steps, not encoder steps, regardless of the state of the Encoder/Motor Step Mode (ENC) command.

As the acceleration scaling factor (SCLA/PSCLA) changes, the accel/decel command's range and its decimal place also change (see table below). An acceleration value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLA1Ø, the A9.9999 command would be truncated to A9.9.

SCLA/PSCLA Value	Decimal Places	Max. Accel/Decel	Min. Accel/Decel (resolution)
1 - 9	0	$\frac{999.9999 \times DRES}{SCLA}$	$\frac{0.001 \times DRES}{SCLA}$
10 - 99	1		
100 - 999	2		
1000 - 9999	3		
10000 - 99999	4		
100000 - 999999	5		

Velocity Scaling (SCLV/PSCLV)

If scaling is enabled (SCALE1), all velocity values entered are internally multiplied by the velocity scaling factor (SCLV) to convert user units/sec to motor steps/sec. Velocity scaling affects the following commands: V, HOMV, HOMVF, JOGVH, JOGVL, JOYVH, JOYVL, and SSV.

Path Scaling: If you are using the 6201's linear interpolation feature or the contouring feature, the PV command is internally multiplied by the path velocity scaling factor (PSCLV).

NOTE

All velocity values are always in reference to motor steps, not encoder steps, regardless of the state of the Encoder/Motor Step Mode (ENC) command.

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

As the velocity scaling factor (SCLV/PSCLV) changes, the velocity command's range and its decimal place also change (see table below). A velocity value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLV1Ø, the V9.9999 command would be truncated to V9.9.

SCLV/PSCLV Value (steps/unit)	Velocity Resolution (units/sec)	Decimal Places	Max. Velocity Calculation
1 - 9	1	0	$\frac{\left(\frac{8,000,000}{n}\right)}{\text{SCLV}}$
10 - 99	0.1	1	
100 - 999	0.01	2	
1000 - 9999	0.001	3	
10000 - 99999	0.0001	4	n = PULSE x 16; If $n < 5$, then n is set equal to 5. If $n > 5$, then all fractional parts of n are truncated.
100000 - 999999	0.00001	5	

Distance Scaling (SCLD/PSCLD)

If scaling is enabled (SCALE1), the D, PSET, and REG command values entered are internally multiplied by the distance scaling factor (SCLD). Since the SCLD units are in terms of steps/unit, all distances will thus be internally represented in motor or encoder steps, depending on the ENC command setting. For instance, if you enable the encoder step mode (ENC1), set the distance scaling factor to 10000 (SCLD1ØØØØ), and enter a distance of 75 (D75), the actual distance moved will be 750,000 (10000 x 75) encoder steps or counts.

Path Scaling: If you are using the 6201's linear interpolation feature or the contouring feature, the PARCM, PARCOM, PARCOP, PARCP, PLC, PLIN, PRTOL, and PWC command values are internally multiplied by the path distance scaling factor (PSCLD) and are always referenced in motor steps.

As the distance scaling factor (SCLD) changes, the distance command's range and its decimal place also change (see table below). A distance value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLD4ØØØ, the D1Ø5.2776 command would be truncated to D1Ø5.277.

SCLD/PSCLD Value (steps/unit)	Distance Resolution (units)	Distance Range (units)	Decimal Places
1 - 9	1	0 - ±999,999,999	0
10 - 99	0.1	0.0 - ±99,999,999.9	1
100 - 999	0.01	0.00 - ±9,999,999.99	2
1000 - 9999	0.001	0.000 - ±999,999.999	3
10000 - 99999	0.0001	0.0000 - ±99,999.9999	4
100000 - 999999	0.00001	0.00000 - ±9999.99999	5

NOTE **FRACTIONAL STEP TRUNCATION** **NOTE**

If you are operating in the incremental mode (MAØ), when the distance scaling factor (SCLD) and the distance value are multiplied, a fraction of one step may possibly be left over. This fraction is truncated when the distance value is used in the move algorithm. This truncation error can accumulate over a period of time, when performing incremental moves continuously in the same direction. To eliminate this truncation problem, set the distance scale factor (SCLD) to 1, or a multiple of 10.

Scaling Example

A user has a 25,000 step/rev motor/drive attached to a 5-pitch leadscrew that he wants his operator to position in inches (25,000 steps/rev x 5 revs/inch = 125,000 steps/inch). A scale factor of 125,000 could then be assigned to the distance scale factor (SCLD). If the operator entered a move value of 1.000, the 6201 would send out 125,000 pulses, which would correspond to one inch.

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

The commands below define the units that are used for both axes. The units for position are 125000 and 25000 for axes #1 and #2, respectively. The units for axis #1 enable the user to program a 25000 step/rev drive with a 5-pitch lead screw in units of inches. The units for axis #2 allows the user to program a 25000 step/rev drive in revs. The velocity and acceleration units allow the user to program both axes in revs/sec (rps) and revs/sec^2 (rps^2), respectively.

Command	Description
> SCALE1	Enable scaling
> SCLD125ØØØ,25ØØØ	Distance scale factor
> SCLV25ØØØ,25ØØØ	Velocity scale factor (rps)
> SCLA25ØØØ,25ØØØ	Acceleration and deceleration scale factor (rps^2)

End-of-Travel Limits

The 6201 can respond to both hardware and software end-of-travel limits. The 6201 is shipped from the factory with the hardware limits enabled. *If you are not using end-of-travel limits in your application, you must disable these limits either through software or hardware before motion will occur.*

Refer to Chapter 3, Installation, for instructions to wire hardware end-of-travel limit switches.

End-of-travel limits prevent the motor's load from traveling past defined limits. Once a hardware or software limit is reached, the 6201 will decelerate that axis at a rate specified with the LHAD or LSAD command. Typically, software and hardware limits are positioned in such a way that when the software limit is reached the motor will start to decelerate towards the hardware limit. This will allow for a much smoother stop at the hardware limit. Software limits can be referenced using either motor or encoder positioning. Refer to the LH, LS, LHAD, and LSAD commands in the ***6000 Series Software Reference Guide*** for more information.

The example below uses the Distance Scaling (SCLD) command to define software limits in revolutions (assuming 25000 steps/rev resolution). Software limits are defined by the LSCW and LSCCW commands. They are enabled with the LS command. The software limits are referenced from a position of absolute zero. Both software limits may be defined with positive values (axis #2 in example below) or negative values. *Care must be taken when performing incremental moves because the software limits are always defined in absolute terms.* They must be large enough to accommodate the moves, or a new zero point must be defined (using the PSET command) before each move.

NOTE

To ensure proper motion when using soft end-of-travel limits, be sure to set the LSCW value to a greater absolute value than the LSCCW value.

Example

In this example, the hardware limits are enabled on axes #1 and #2. Deceleration rates are specified for both software and hardware limits. If a limit is encountered, the motors will decelerate to a stop.

Command	Description
> SCALE1	Enable scaling
> @SCLD25ØØØ	Distance scale factor
> @SCLA25ØØØ	Acceleration scale factor
> @SCLV25ØØØ	Velocity scale factor
> LH3,3	Enable limits 1 and 2
> LHAD1Ø,1Ø	Hard limit deceleration
> LSAD5,1Ø	Soft limit deceleration
> LSCCWØ,2	Establish CCW soft limit
> LSCW1Ø,2Ø	Establish CW soft limit
> LS3,3	Enable soft limits 1 and 2

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

Homing *(Using the Home Inputs)*

Refer to Chapter 3, Installation, for instructions to wire hardware home limit switches.

The HOM command initiates a sequence of moves that position an axis using the Home input by itself or with the Z channel input from an encoder. The result of any *homing* operation is a repeatable initial starting location. The home inputs to be used, the edge of those inputs, and the final approach direction may all be defined by the user. If the Z channel input is to be used, the HOMZ command must be enabled. The input polarity (normally-open or normally-closed) of the home input or switch is defined with the HOMLVL command.

The velocity for a move to the home position is specified with the HOMV command. The acceleration and deceleration rates are specified with the HOMA and HOMAD commands, respectively. If *backup to home* (HOMBAC) is enabled, the velocity of the final approach toward the home position is specified with the HOMVF command.

Enabling backup to home (HOMBAC) allows you to use two other homing features, HOMEDG and HOMDF. The HOMEDG command allows you to specify the side of the home switch on which to stop. The HOMDF command allows you to specify the final approach direction. If HOMBAC is not enabled, HOMEDG and HOMDF will have no effect on the homing algorithm (see Figures A and B).

Figures A and B show the homing operation when HOMBAC is not enabled. If a limit is encountered during the homing operation, the motion will be reversed and the home switch will be sought in the opposite direction. If a second limit is encountered, the homing operation will be terminated, stopping motion at the second limit.

As soon as the homing operation is successfully completed, the absolute position register is automatically reset to zero. Bit 5 of the TAS and [AS] commands can be used as the check for a successful home.

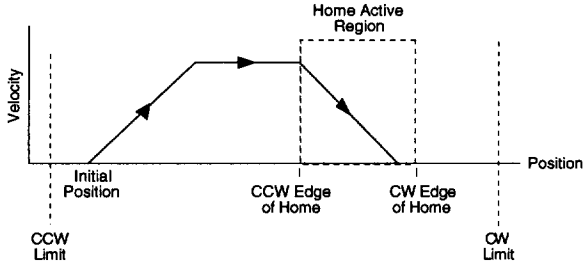


Figure A. Homing in a CW Direction (HOMØ) with backup to home disabled (HOMBACØ)

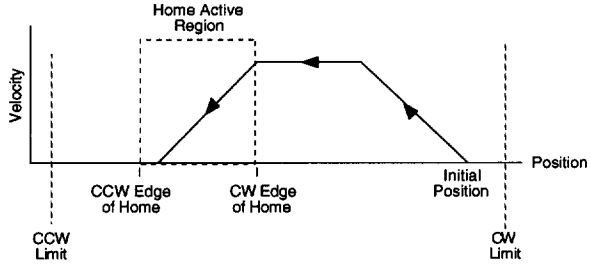


Figure B. Homing in a CCW Direction (HOM1) with backup to home disabled (HOMBACØ)

CW Homing with Backup to Home Enabled

The seven steps below describe a sample homing operation when HOMBAC is enabled (see Figure C). The final approach direction (HOMDF) is CW and the home edge (HOMEDG) is the CW edge.

NOTE

To better illustrate the direction changes in the backup-to-home operation, the illustrations in the remainder of this section show the backup-to-home movements with varied velocities. In reality, the backup-to-home movements are performed at the same velocity (defined with the HOMVF command).

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

1. A CW home move is started with the HOMØ command at the HOMA acceleration. Default HOMA is 2,500,000 steps/sec^2
2. The HOMV velocity is reached (move continues at that velocity until home input goes active).
3. The CCW edge of the home input is detected, this means the home input is active. At this time the move is decelerated at the HOMAD command value. It does not matter if the home input becomes inactive during this deceleration.
4. After stopping, the direction is reversed and a second move with a peak velocity specified by the HOMVF value is started.
5. This move continues until the CCW edge of the home input is reached.
6. Upon reaching the CCW edge, the move is decelerated at the HOMAD command value, the direction is reversed, and another move is started in the CW direction at the HOMVF velocity.
7. As soon as the home input CW edge is reached, this last move is killed (pulses shut-off). The load is at home and the absolute position register is reset to zero.

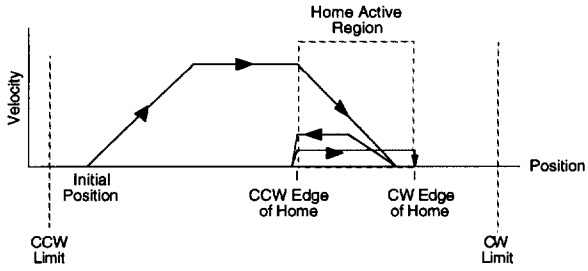


Figure C. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDGØ, HOMDFØ

Figures D through F show the homing operation for different values of HOMDF and HOMEDG, when HOMBAC is enabled.

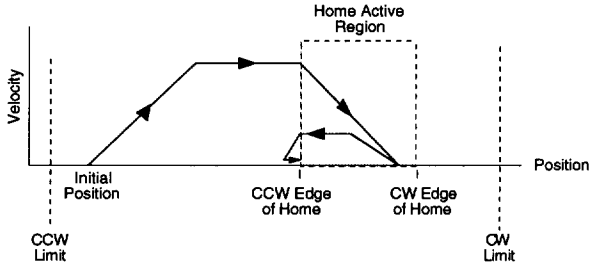


Figure D. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDG1, HOMDFØ

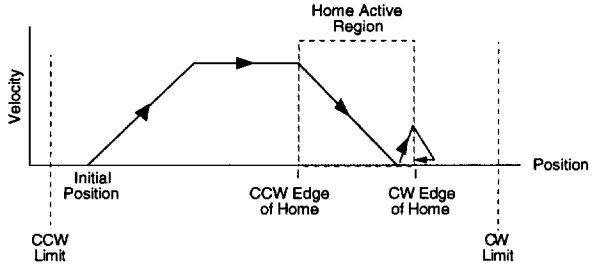


Figure E. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDGØ, HOMDF1

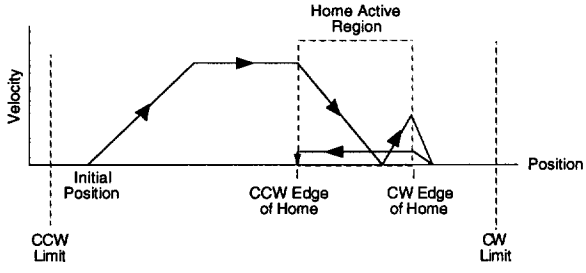


Figure F. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDG1, HOMDF1

CCW Homing with Backup to Home Enabled

Figures G through J show the homing operation for different values of HOMDF and HOMEDG, when HOMBAC is enabled.

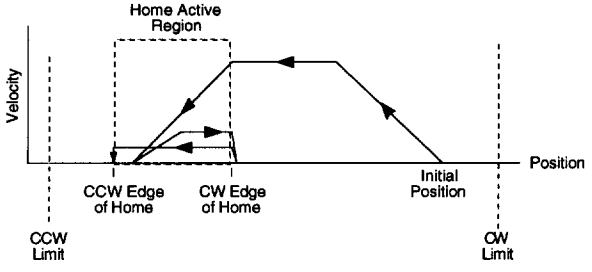


Figure G. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDF1

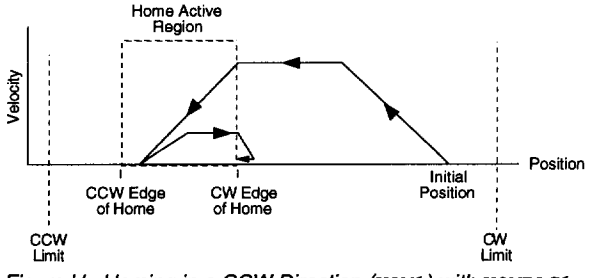


Figure H. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDF1

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

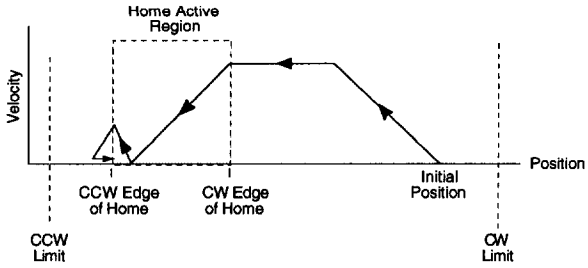


Figure I. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDFØ

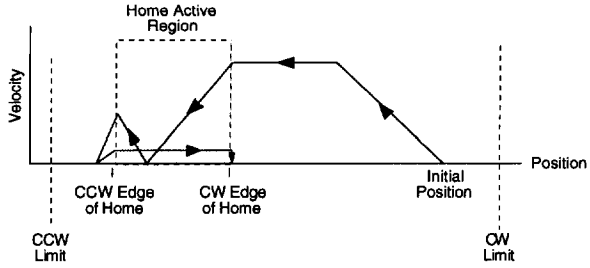


Figure J. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

Move HOME Using The Z-Channel

Figures K through O show the homing operation when homing to an encoder index pulse, or Z channel, is enabled (HOMZ1). The Z-channel will only be recognized after the home input is activated. It is desirable to position the Z channel within the home active region; this reduces the time required to search for the Z channel.

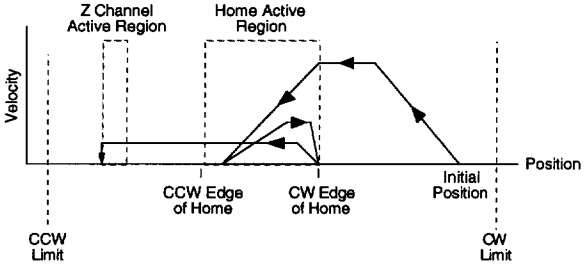


Figure K. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDF1

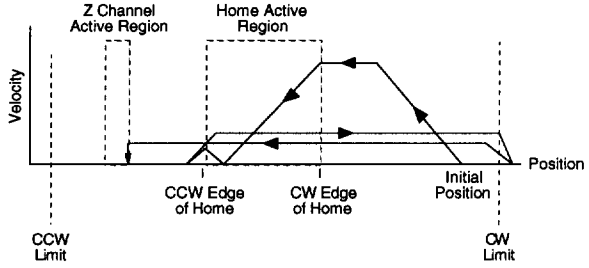


Figure L. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

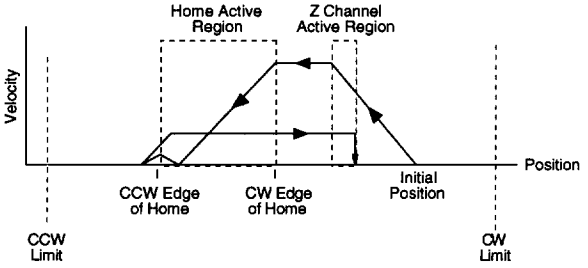


Figure M. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

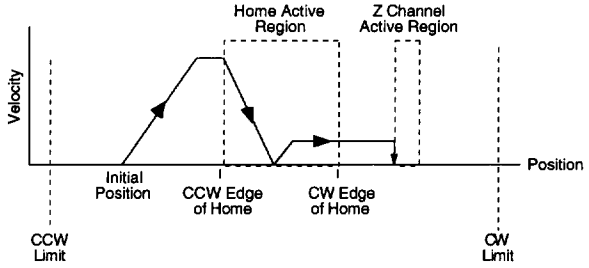


Figure N. Homing in a CW Direction (HOMØ) with HOMBACØ, HOMEDGØ, HOMDFØ

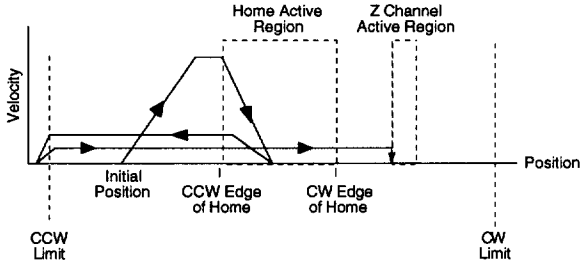


Figure O. Homing in a CW Direction (HOMØ) with HOMBACØ, HOMEDGØ, HOMDF1

Positioning Modes

The 6201 can be programmed to position in either the preset (incremental or absolute) mode or the continuous mode. You should select the mode that will be most convenient for your application. For example, a repetitive cut-to-length application requires incremental positioning. X-Y positioning, on the other hand, is better served in the absolute mode. Continuous mode is useful for applications that require constant movement of the load based on internal conditions or inputs, not distance.

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

Refer also to the Scaling *section above.*

Positioning modes require acceleration, deceleration, velocity, and distance commands (*continuous mode does not require distance*). The following table identifies these commands and their units of measure, and which scaling command affects them.

Parameter	Command	Command Value expressed in: Units	or Scaling Ratio set with this command:
Acceleration	A	revs per second^2 (rps^2)	SCLA
Deceleration	AD	revs per second^2 (rps^2)	SCLA
Velocity	V	revs per second (rps)	SCLV
Distance	D	steps	SCLD

Preset Mode

A *preset* move is a point-to-point move of a specified distance. You can select preset moves by putting the 6201 into normal mode using the Preset Mode (MCØ) command. Preset moves allow you to position the motor in relation to the motor's previous stopped position (*incremental mode*—enabled with the MAØ command) or in relation to a defined zero reference position (*absolute mode*—enabled with the MA1 command). Both absolute and incremental modes can be specified in either motor step mode (ENCØ) or encoder step mode (ENC1).

Incremental Mode Moves

The incremental mode is the 6201's default power-up mode. When using the incremental mode (MAØ), a preset move moves the shaft of the motor the specified distance from its starting position. For example, to move the motor shaft 1.5 revolutions, a preset move with a distance of +37,500 steps (1.5 revs @ 25,000 steps/rev) would be specified. Every time the indexer executes this move, the motor moves 1.5 revs from its resting position. You can specify the direction of the move by using the optional sign (D+375ØØ or D-375ØØ). Whenever you do not specify the direction (e.g., D25ØØØ), the unit defaults to the positive (CW) direction.

Example

Command	Description
> SCALEØ	Disable scaling
> DRES25ØØØ	Set axis 1 drive resolution
> MAØ	Sets axis 1 to Incremental Position Mode
> A2	Sets axis 1 acceleration to 2 rps^2
> V5	Sets axis 1 velocity to 5 rps
> D25ØØØ	Sets axis 1 distance to 25,000 CW steps
> GO1	Initiate motion on axis 1 (motor moves 1 rev in CW direction)
> GO1	Repeats the move
> D-5ØØØØ	Sets axis 1 distance to 50,000 CCW steps (return to original position)
> GO1	Initiate motion on axis 1 (motor moves 2 revs in CCW direction and ends at its original starting position)

Absolute Preset Mode Move

A preset move in the Absolute Mode (MA1) moves the motor the distance that you specify (in motor steps) from the *absolute zero position*. You can set the absolute position to any value with the Set Position (PSET) command. When the Go Home (HOM) command is issued, the absolute position is automatically set to zero after the motor reaches the home position.

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 +12,500, and you instruct the motor to move to position +5,000, the motor will move in the negative (CCW) direction a distance of 7,500 steps to reach the absolute position of +5,000.

The 6201 retains the absolute position, even while the unit is in the incremental mode. You can use the Absolute Position Report (TPM) command to read the absolute position.

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

Example	**Command**	**Description**
	> SCALEØ	Disable scaling
	> DRES25ØØØ	Set axis 1 drive resolution
	> MA1	Sets the 6201 to the absolute positioning mode
	> PSETØ	Sets axis 1 current absolute position to zero
	> A5	Sets axis 1 acceleration to 5 rps^2
	> V3	Sets axis 1 velocity to 3 rps
	> D5ØØØ	Sets axis 1 move to absolute position 5,000
	> GO1	Initiates axis 1 move (motor moves to absolute position 5,000)
	> D1ØØØØ	Sets axis 1 move to absolute position 10,000.
	> GO1	Initiates axis 1 move (Since the motor was already at position 5,000, it moves 5,000 additional steps in the CW direction.)
	> DØ	Sets axis 1 move to absolute position zero.
	> GO1	Initiates axis 1 move (Since the motor is at absolute position 10,000, the motor moves 10,000 steps in the CCW direction.)

Continuous Mode

The Continuous Mode (MC) is useful in the following situations:

- Applications that require constant movement of the load
- Changing the motion profile after a specified distance or after a specified time period (T command) has elapsed
- Synchronize the motor to external events such as trigger input signals

Buffered vs. *Immediate* Commands

You can manipulate the motor movement with either buffered or immediate commands. After you issue the GO command, buffered commands are not executed unless the continuous command execution mode (command) is enabled. Once COMEXC is enabled, buffered commands are executed in the order in which they were programmed.

The command can be specified as *immediate* by placing an exclamation mark (!) in front of the command. When a command is specified as immediate, it is placed at the front of the command queue and is executed immediately.

Examples	**Command**	**Description**
	> COMEXC1	Enable continuous command processing mode
	> MC1	Sets axis 1 mode to continuous
	> A1Ø	Sets axis 1 acceleration to 10 rps^2
	> V1	Sets axis 1 velocity to 1 rps
	> GO1	Initiates axis 1 move (Go)
	> WAIT(1VEL=1)	Wait for motor to reach continuous velocity
	> T5	Time delay of 5 seconds
	> S1	Initiate stop of axis 1 move
	> WAIT(MOV=bØ)	Wait for motion to completely stop on axis 1
	> COMEXCØ	Disable continuous command processing mode

The motor accelerates to 1 rps, continues at that same velocity for 5 seconds, then decelerates to a stop.

While in continuous mode, motion can also be stopped by one of the following:

- The load trips an end-of-travel limit switch
- The load trips a trigger input switch and a registration move occurs
- You issue an immediate Stop (!S) or Kill (!K) command
- The load trips an input switch tied to an input configured as a kill or stop input with the INFNCi-C or INFNCi-D commands, respectively

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

On-The-Fly Changes

You can change velocity and acceleration *on the fly* (while motion is in progress and operating in the continuous positioning mode) by issuing an immediate velocity (!V) and/or acceleration (!A) command followed by an immediate go (!GO). If the continuous command processing mode (COMEXC) is enabled, you can also make *on-the-fly* velocity and acceleration changes by using buffered commands (V and A), followed by a GO command.

> **NOTE**
>
> While the axis is moving, you cannot change the parameters of some commands (such as D and HOM). This rule applies during the COMEXC mode and even if you prefix the command with an immediate command identifier (!). For more information, refer to *Changing Command Parameters During Motion* in the *Programming Guide* section of the ***6000 Series Software Reference Guide***.

Example

Command	Description
> DEF vsteps	Begin definition of program vsteps
- COMEXC1	Enable continuous command processing mode
- MC1	Set axis 1 mode to continuous
- A1Ø	Set axis 1 acceleration to 10 rps^2
- V1	Set axis 1 velocity to 1 rps
- GO1	Initiate axis 1 move (Go)
- WAIT(1VEL=1)	Wait for motor to reach continuous velocity
- T3	Time delay of 3 seconds
- A5Ø	Set axis 1 acceleration to 50 rps^2
- V1Ø	Set axis 1 velocity to 10 rps
- GO1	Initiate acceleration and velocity changes on axis 1
- T5	Time delay of 5 seconds
- S1	Initiate stop of axis 1 move
- WAIT(MOV=bØ)	Wait for motion to completely stop on axis 1
- COMEXCØ	Disable continuous command processing mode
- END	End definition of program vsteps
- RUN VSTEPS	Run program vsteps

Closed-Loop Operation *(Using an Encoder)*

Closed-loop refers to operating with position feedback from an encoder. This section discusses how you can use an incremental encoder to perform closed-loop functions with the 6201.

When using an encoder in an application, you must configure the following (described below):

- Motor & Encoder Resolutions
- Encoder vs. Motor Step Mode
- Position Maintenance Enable & Correction Parameters
- Position Maintenance Deadband
- Stall Detection & Kill-on-Stall
- Stall Backlash Deadband
- Counter

Motor & Encoder Resolutions

You must specify the motor and encoder resolutions used for the 6201. The DRES and ERES commands define the motor and encoder resolutions respectively (4000 steps/rev is a typical encoder resolution value). The power-up default values for motor resolution and encoder resolution are 25,000 and 4,000 respectively.

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

If the 6201's resolution (set with the DRES command – default is 25,000 steps/revs) does not match the drive's resolution, the motor will not move according to the programmed distance and velocity.

To achieve accurate encoder step positioning, the motor resolution should be at least four times greater than the encoder resolution. This allows the motor to successfully find the commanded encoder step position.

Encoder Step Mode

The 6201 can perform moves in either motor steps or encoder steps. In Motor Step Mode (ENCØ), the distance command (D) defines moves in motor steps. Motor Step Mode is the 6201's default setting. In Encoder Step mode (ENC1), the distance command defines moves in encoder steps.

Position Maintenance

The EPM1 command enables the position maintenance function. To enable position maintenance, you must first connect an encoder and set the indexer to the Encoder Step Mode (ENC1).

Enabling position maintenance causes the indexer to *servo* (adjust) the motor until the correct encoder position is achieved. This occurs at the end of a move (if the final position is incorrect) or any time the indexer senses a change in position while the motor is at zero velocity.

The EPMG and EPMV commands define the gain factor and the maximum velocity of a position maintenance correction move. The gain factor is the velocity increment used per encoder step of error. The EPMV command sets the maximum velocity a position maintenance move can achieve. If either of these is too large, position instability may result. If either is too small, position correction may be too slow. *The values are determined by experiment.*

Position Maintenance Deadband

The EPMDB command sets the number of encoder steps of error allowed before a position maintenance correction move is initiated. This is useful for situations in which it is not desirable to have the load position continuously corrected, but correction is required outside the deadband. Positioning under a microscope is an example.

Stall Detection & Kill-on-Stall

The ESTALL1 command allows the 6201 to detect stall conditions. If used with *Kill-on-Stall* enabled (ESK1 command), the move in progress will be aborted upon detecting a stall. If queried with the ER or the AS commands, the user may branch to any other section of program when a stall is detected. Refer to the ENC, ER, and AS command descriptions in the ***6000 Series Software Reference Guide*** for more information.

Stall Detection functions in either motor step or encoder step mode. Kill-on-Stall functions only if the stall detection is enabled (ESTALL1).

WARNING

Disabling the Kill-on-Stall function with the ESKØ command will allow the 6201 to finish the move regardless of a stall detection, even if the load is jammed. **This can potentially damage user equipment and injure personnel.**

Stall Backlash Deadband

Another encoder set-up parameter is the Stall Backlash Deadband (ESDB) command. This command sets the number of encoder steps of error allowed before a stall will be detected. This is useful for situations in which backlash in a system can cause false stall situations.

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

Encoder Set Up Example

The example below illustrates the features discussed in the previous paragraphs. The first statement defines the motor resolutions. The next statement defines the number of encoder steps per encoder revolution. Standard 1000-line encoders are used on both axes that produce 4000 quadrature steps/rev. Stall detect and kill-on-stall are enabled.

If a stall is detected (more than 10 steps out without an encoder step back), the motor's movement is killed. The next group of commands define the deadband width and enable the position maintenance function. The 6201 will attempt to move to the correct encoder position until it is within the specified deadband.

Examples

Command	Description
> DRES25000,25000	Set drive resolution
> ERES4000,4000	Set encoder resolution
> ESK11	Enable kill motion on stall
> ESDB10,10	Set stall deadband
> ESTALL11	Enable stall detection
> EPMG1000,1000	Set position maintenance gain
> EPMV1,1	Set position maintenance velocity
> EPM11	Enable position maintenance
> ENC11	Enable encoder step mode

Counter

Each encoder channel can be redefined as a 16-bit up/down counter. The CNTE command redefines the encoder channels. The direction of the count is specified by the signal on the encoder channel B+ and B- connections. A positive differential signal, when measured between B+ and B-, indicates a positive count direction. A negative differential signal, when measured between B+ and B-, indicates a negative count direction.

The count itself is determined from the signal on A+ and A-. Each count is registered on the positive edge of a transition for a signal measured between A+ and A-. To reset the counter, issue the CNTR command or apply a positive differential signal to Z+ and Z-. The counter will be reset on the rising edge. Allow 50µs for the counter to reset, and allow 50µs after reset for counting to begin.

The value of the counter can be accessed at any time through the hardware registers or by doing a software transfer (TCNT). The hardware registers provide information on encoder position, but when an encoder input is defined as a counter, the information in the register is a count value.

User Interface Options

Below are the three basic user interface options for controlling the 6201:

Stand-alone operation: After defining and storing 6201 programs with a RS-232C terminal, you can operate the 6201 as a stand-alone controller. A program stored in the 6201 may interactively prompt the user for input as part of the program (via programmable I/O, thumbwheels, an RP240, or an RS-232C terminal).

PLC interface: The 6201's programmable I/O may be connected to most PLCs. The PLC typically executes programs, loads data, and manipulates inputs to the 6201. The PLC instructs the 6201 to perform the motion segment of a total machine process.

Host computer operation: A computer may be used to control a motion or machine process. A PC can monitor processes and orchestrate motion by sending motion commands to the 6201 or by executing motion programs already stored in the 6201. This control might come from a BASIC or C program.

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

Of course, you can use any one, or combination, of these options in your application. Some application examples are provided below. The following sections discuss programmable I/O (including thumbwheel and PLC interfacing), the RP240 interface, host computer interfacing, stored programs, and other aspects of the 6201 that allow the user to apply the 6201 in the variety of interface options as noted.

User Interface Option	Application Example
Stand-alone:	
Programmable I/O and Thumbwheel/TM8 interface	**Cut-to-length**: Load the stock into the machine, enter the length of the cut on the thumbwheels, and activate a programmable input switch to initiate the predefined cutting process (axis #1). When the stock is cut, a sensor activates a programmable input to stop the cutting process and the 6201 then initiates a predefined program that indexes the stock forward (axis #2) into position for the next cut.
RP240 remote operator panel interface	**Grinding**: Program the RP240 function keys to select certain part types, and program one function key as GO button. Select the part you want to grind, then put the part in the grinding machine and press the GO function key. The 6201 will then move the machine according to the predefined program assigned to the function key selected.
Joystick interface	**X-Y scanning/calibration**: Enter the joystick mode and use the 2-axis joystick to position an X-Y table under a microscope to arbitrarily scan different parts of the work piece (e.g., semi-conductor wafer). You can record certain locations to be used later in a motion process to drilling, cutting, photographing, etc. the work piece.
PLC Interface	**X-Y point-to-point**: A PLC controls other machine functions including a solenoid-operated liquid dispenser. The 6201 is programmed to move an X-Y table in a switch-back matrix, stopping at two-inch intervals. At every interval, the PLC dispenses the liquid and controls several other machine functions. Then the PLC tells the 6201 to continue the matrix until all receptacles are filled.
Host Computer (PC) Interface	A BASIC program example is provided later in the section labeled *Host Computer Operation*.

Programmable Inputs and Outputs

Refer to Chapter 3 for I/O connection instructions.

There are 26 programmable inputs (includes 2 trigger inputs on the **AUX** connector) and 26 programmable outputs (includes 2 auxiliary outputs on the **AUX** connector).

I/O circuit drawings and specifications are provided in Chapter 5.

All the 6201's inputs and outputs are optically isolated. The 24 inputs and 24 outputs, located on the **PROGRAMMABLE INPUTS** and **PROGRAMMABLE OUTPUTS** connectors, are OPTO-22 compatible for those applications that require interfacing to 120VAC I/O or to I/O with higher current requirements than the 6201 can support. An optional 50-pin screw terminal adaptor (the VM50) is available.

Programmable inputs and outputs are provided to allow the 6201 to detect and respond to the state of switches, thumbwheels, electronic sensors, and outputs of other equipment such as drives and PLCs. Based on the state of the inputs and outputs, read with the `[IN]` and `[OUT]` commands, the 6201 can make program flow decisions and assign values to binary variables for subsequent mathematical operations. These operations and the associated program flow, branching, and variable commands are listed below.

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

Operation based on I/O State	Associated Commands	Discussed later in this manual*
I/O state assigned to a binary variable	[IN], [OUT], VARB	*Variables* section (this chapter)
I/O state used as a basis for comparison in conditional branching & looping statements	[IN], [OUT], IF, ELSE, REPEAT, UNTIL, WAIT, WHILE, NWHILE	*Program Flow Control* section in ***6000 Series Software Reference Guide***
I/O state used as a basis for a program interrupt (GOSUB) conditional statement	ONIN	*Program Interrupts* section in ***6000 Series Software Reference Guide***

* Refer also to the command descriptions in the ***6000 Series Software Reference Guide***

As discussed below, you can program and check the status of each input and output with the INFNC and OUTFNC commands, respectively. To receive a binary report of the state (on or off) of the I/O, use the TIN command (inputs) or TOUT command (outputs).

Using the INLVL and OUTLVL commands, you can define the logic levels of the 26 inputs and 26 outputs as positive or negative.

Output Functions

You can turn the 26 programmable outputs on and off with the Output (OUT or OUTALL) commands, or you can use the Output Function (OUTFNC) command to configure them to activate based on six different situations. These output functions configurations are typically located in the start-up (STARTP) program.

The output functions are assigned with the OUTFNCi-<a>c command. The "i" represents the number of the output (the 24 general-purpose outputs are outputs 1 - 26, and **OUT-A** and **OUT-B** are outputs 25 and 26). The "<a>" represents the number of the axis and is optional for the B, D, and E functions (see list below); when no axis specifier is given, the output will be activated when the condition occurs on any axis. The "c" represents the letter designator of the function (A, B, C, D, E or F). For example, the OUTFNC5-2D command configures output #5 to activate when axis #2 encounters a hard or soft limit.

NOTE

To activate the function of an output with the OUTFNC command, you must enable the output functions with the OUTFEN1 command.

A: Programmable Output (**default function**)
B: Moving/Not Moving
C: Program in Progress
D: At Soft or Hard Limits
E: Stall Indicator
F: Fault Output

Output Status

As shown below, you can use the OUTFNC command to determine the current function and state (on or off) of one or all the outputs. The TOUT command also reports the outputs' state, but in a binary format in which the left-most bit represents output #1 and the right-most bit represents output #26.

Command	Description
> OUTFNC	Query status of all outputs; response indicating default conditions is: *OUTFNC1-A NO FUNCTION OUTPUT - STATUS OFF *OUTFNC2-A NO FUNCTION OUTPUT - STATUS OFF (response continues until all 26 outputs are reported)
> OUTFNC1	Query status of output #1; response indicating default conditions is: *OUTFNC1-A NO FUNCTION OUTPUT - STATUS OFF
> OUTFNC1-C	Change output #1 to function as a *Program in Progress* output
> OUTFNC1	Query status of output #1; response should be now be: *OUTFNC1-C PROGRAM IN PROGRESS - STATUS OFF
> TOUT	Query binary status report of all outputs; response indicating default conditions is: *TOUT0000_0000_0000_0000_0000_0000_00

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

Programmable Output
(OUTFNCi-A)

The default function for the outputs is *Programmable*. As such, the output is used as a standard output, turning it on or off with the OUT or OUTALL commands to affect processes external to the 6201. To view the state of the outputs, use the TOUT command. To use the state of the outputs as a basis for conditional branching or looping statements (IF, REPEAT, WHILE, etc.), use the [OUT] command (refer to the *Conditional Looping and Branching* section in the ***6000 Series Software Reference Guide*** for details).

Moving/ Not Moving
(OUTFNCi-<a>B)

When assigned the *Moving/Not Moving* function, the output will activate when the axis is moving. As soon as the move is completed, the output will change to the opposite state.

Example

The following example defines output 1 and output 2 as *Programmable* outputs and output 3 as a *Moving/Not Moving* output. Before the motor moves 25,000 steps, output 1 turns on and output 2 turns off. These outputs remain in this state until the move is completed, then output 1 turns off and output 2 turns on. While the motor is moving, output 3 remains on.

Command	Description
> PS	Pauses command execution until the 6201 receives a Continue (!C) command
SCALEØ	Disable scaling
MCØ	Sets axis 1 to Normal mode
A1Ø	Sets axis 1 acceleration to 10 rps^2
V5	Sets axis 1 velocity to 5 rps
D25ØØØ	Sets axis 1 distance to 25,000 steps
OUTFEN1	Enable output functions
OUTFNC1-A	Sets output 1 as a programmable output
OUTFNC2-A	Sets output 2 as a programmable output
OUTFNC3-1B	Sets output 3 as a axis 1 Moving/Not Moving output
OUT1Ø	Turns output 1 on and output 2 off
GO1	Initiates axis 1 move
OUTØ1	Turns output 1 off and output 2 on
!C	Initiates command execution to resume

Program in Progress
(OUTFNCi-C)

When assigned the *Program in Progress* function, the output will activate when a program is being executed. After the program is finished, the output's state is reversed.

Limit Encountered
(OUTFNCi-<a>D)

When assigned the *Limit Encountered* function, the output will activate when a hard or soft limit has been encountered.

If a hard or soft limit is encountered, you will not be able to move the motor in that same direction until you clear the limit by changing direction (D) and issuing a GO command. (An alternative is to disable the limits with the LHØ command, but this is recommended only if the motor is not coupled to the load.) The output's state is reversed once the limit condition is cleared.

Stall Indicator
(OUTFNCi-<a>E)

When assigned the *Stall Indicator* function, the output will activate when stall is detected. To detect a stall, you must first connect an encoder, enable the encoder step mode with the ENC1 command, enable the position maintenance function with the EPM1 command, and enable stall detection with the ESTALL1 command. The output's state is reversed once the stall condition is cleared. Refer to the *Closed-Loop Operation* section earlier in this chapter for further discussion on stall detection.

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

Fault Output (OUTFNCi-F)

When assigned the *Fault Output* function, the output will activate when either the user fault input or the drive fault input becomes active. The user fault input is a general-purpose input defined as a user fault input with the INFNCi-F command. The drive fault input is found on the **DRIVE** connector, pin #5; make sure the drive fault active level (DRFLVL) is appropriate for the drive you are using, and be sure to enable the drive fault monitoring function with the INFEN1 command. The output's state is reversed once the fault condition is cleared.

Input Functions

You can configure each programmable input to perform one of twelve different functions . The input functions are assigned with the INFNCi-ac command. The "i" represents the number of the input (the 24 general purpose inputs are inputs 1 through 24, and **TRG-A** & **TRG-B** are inputs 25 & 26). The "a" represents the number of the axis, if required. The "c" represents the letter designator of the function (A through P). For example, the INFNC5-2D command configures output #5 to function as a *stop* input, stopping motion on axis #2 when activated.

> **NOTE**
> To activate the function of an input with the INFNC command, you must first enable the input functions with the INFEN1 command. Because the INFEN1 command enables the drive fault input, you should verify the fault active level (DRFLVL) is set properly.

- A: No Function (default)
- B: BCD Program Select
- C: Kill
- D: Stop
- E: Pause/Continue
- F: User Fault
- H: Trigger Interrupt for Position Capture and Registration **(applies to Trigger inputs only)**
- I: Interrupt to PC-AT
- J: Jog+ (CW)
- K: Jog- (CCW)
- L: Jog Speed Select
- P: Program Select

Input Status

As shown below, you can use the INFNC command to determine the current function and state (on or off) of one or all the inputs. The TIN command also reports the inputs' state, but in a binary format in which the left-most bit represents input #1 and the right-most bit represents input #26.

Example Command	Description
> INFNC	Query status of all inputs; response indicating default conditions is: *INFNC1-A NO FUNCTION INPUT - STATUS OFF *INFNC2-A NO FUNCTION INPUT - STATUS OFF (response continues until all 26 inputs are reported)
> INFNC1	Query status of input #1; response indicating default conditions is: *INFNC1-A NO FUNCTION INPUT - STATUS OFF
> INFNC1-D	Change input #1 to function as a Stop input
> INFNC1	Query status of input #1; response should be now be: *INFNC1-D STOP INPUT - STATUS OFF
> TIN	Query binary status report of all inputs; response indicating default conditions is: *TIN0000_0000_0000_0000_0000_0000_00

Input Debounce Time

Using the Input Debounce Time (INDEB) command, you can change the input debounce time for all 24 general-purpose inputs (one debounce time for all 24), or you can assign a unique debounce time to each of the 2 trigger inputs.

General-Purpose Input Debounce: The input debounce time for the 24 general-purpose inputs is the period of time that the input must be held in a certain state before the 6201 recognizes it. This directly affects the rate at which the inputs can change state and be recognized.

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

Trigger Input Debounce: For trigger inputs, the debounce time is the time required between a trigger's initial active transition and its secondary active transition. This allows rapid recognition of a trigger, but prevents subsequent bouncing of the input from causing a false position capture or registration move.

The INDEB command syntax is INDEB<i>,<i>. The first <i> is the input number and the second <i> is the debounce time **in even increments** of milliseconds (ms). The debounce time range is 1 - 250 ms. The default debounce time is 4 ms for the 24 general-purpose inputs, and 50 ms for the 2 trigger inputs (**TRG-A** & **TRG-B**). If the first <i> is in the range 1 - 24, the specified debounce time is assigned to all 24 general-purpose inputs. If the first <i> is 25 or 26, the specified debounce is assigned only to the specified trigger input.

For example, the INDEB5,6 command assigns a debounce time of 6 ms to all 24 general-purpose inputs. The INDEB26,12 command assigns a debounce time of 12 ms only to input #26, which is trigger B (**TRG-B**).

No Function (INFNCi-A)

When an input is defined as a *No Function* input (default function), the input is used as a standard input. You can then use this input to synchronize or trigger program events. To view the current state of the inputs, use the TIN command. To use the state of the outputs as a basis for conditional branching or looping statements (IF, REPEAT, WHILE, etc.), use the [IN] command (refer to the *Conditional Looping and Branching* section in the ***6000 Series Software Reference Guide*** for details).

Example

Command	Description
> DEF prog1	Begin definition of program prog1
- INFEN1	Enable input functions
- INFNC1-A	No function for input 1
- INFNC2-A	No function for input 2
- INFNC3-D	Input 3 is a stop input
- INFNC4-A	No function for input 4
- A1Ø	Set acceleration
- V1Ø	Set velocity
- D25ØØØ	Set distance
- WAIT(IN=b1XX1)	Wait for input 1 and 4
- GO1	Initiate motion
- IF(IN=bX1)	If input 2
- TPM	Transfer motor position
- NIF	End IF statement
- END	End prog1
> RUN prog1	Initiate program prog1

BCD Program Select (INFNCi-B)

Inputs can be defined as *BCD program select* inputs. This allows you to execute defined programs (DEF command) by activating the program select inputs. Program select inputs are assigned BCD weights. The table below shows the BCD weights of the 6201's inputs when inputs 1- 8 are configured as program select inputs. The inputs are weighted with the least weight on the smallest numbered input.

Input	BCD Weight
Input 1	1
Input 2	2
Input 3	4
Input 4	8
Input 5	10
Input 6	20
Input 7	40
Input 8	80

If inputs 6, 9, 10 and 13 are selected instead of inputs 5, 6, 7 and 8, then the weights would be as follows:

Input #6 = 10
Input #9 = 20
Input #10 = 40
Input #13 = 80

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

Since only 100 programs can be defined, a maximum of 9 inputs are all that are required to select all possible programs.

The program number is determined by the order in which the program was downloaded to the 6201. The program number can be obtained through the TDIR command.

If the inputs are configured as in the above table, program #6 will be executed by activating inputs 2 and 3. Program #29 will be executed by activating inputs 1, 4, and 6.

To execute programs using the program select lines, enable the INSELP command. Once enabled, the 6201 will continuously scan the input lines and execute the program selected by the active program select lines. To disable scanning for program select inputs, enter !INSELPØ or place INSELPØ in a program that can be selected.

Once enabled (INSELP1), the 6201 will run the program number that the active program select inputs and their respective BCD weights represent. After executing and completing the selected program, the 6201 will scan the inputs again. If a program is selected that has not been defined, no program will be executed.

The INSELP command also determines how long the program select input must be maintained before the indexer executes the program. This delay is referred to as *debounce time*. The following examples demonstrate how to select programs via inputs.

Example

Command	Definition
> RESET	Return 6201 to factory defaults
> ERASE	Erase all programs
> DEF prog1	Begin definition of program prog1
- TPM	Transfer position of motor
- END	End program
> DEF prog2	Begin definition of program prog2
- TREV	Transfer software revision
- END	End program
> DEF prog3	Begin definition of program prog3
- TSTAT	Transfer statistics
- END	End program
> INFNC1-B	Input 1 is a BCD program select input
> INFNC2-B	Input 2 is a BCD program select input
> INFEN1	Enable input functions
> INSELP1,5Ø	Enable scanning of inputs

You can now execute programs by making a contact closure from an input to ground to activate the input.

- Activate input 1 to execute program #1
- Activate input 2 to execute program #2
- Activate input 1 & 2 to execute program #3

Kill (INFNCi-C)

An input defined as a *Kill* input will stop motion on all axes with no deceleration ramp. The program currently in progress will also be terminated, and commands currently in the command buffer will be eliminated.

CAUTION

Defining an input to be a kill input will cause motion to stop instantaneously when the input is activated. Since there is no deceleration ramp, there is a possibility that the motor may stall, or the drive may fault out, possibly causing the load to *free wheel*.

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

Stop
(INFNCi-D)

An input defined as a *Stop* input will stop motion on any one or all axes. Deceleration is controlled by the programmed (AD) deceleration ramp. After the Stop input is received, further program execution is dependent upon the COMEXS command setting:

COMEXSØ: Upon receiving a stop input, program execution will be terminated and every command in the buffer will be discarded.

COMEXS1: Upon receiving a stop input, program execution will pause, and all commands following the command currently being executed will remain in the command buffer (*but the move in progress will not be saved*).

You can resume program execution (but not the move in progress) by issuing an immediate Continue (!C) command or by activating a pause/resume input (i.e., a general-purpose input configured as a pause/continue input with the INFNCi-E command—see below). *You cannot resume program execution while the move in progress is decelerating.*

COMEXS2: Upon receiving a stop input, program execution will be terminated, but the INSELP value is retained. This allows external program selection, via inputs defined with the INFNCi-B or INFNCi-aP commands, to continue.

Pause/Continue
(INFNCi-E)

An input defined as a *Pause/Continue* input will affect motion and program execution depending on the COMEXR command setting, as described below. In both cases, when the input is activated, the current command being processed will be allowed to finish executing before the program is paused.

COMEXRØ: Upon receiving a pause input, only program execution will be paused; any motion in progress will continue to its predetermined destination. Releasing the pause input or issuing a !C command will resume program execution.

COMEXR1: Upon receiving a pause input, both motion and program execution will be paused; the motion stop function is used to halt motion. Releasing the pause input or issuing a !C command will resume motion and program execution. *You cannot resume program execution or motion while the move in progress is decelerating.*

User Fault
(INFNCi-F)

An input defined as a *User Fault* input will set error bit 7 (TER, ER), and act as a Kill (K) command. Once this bit has been set, the error program (ERRORP) will be initiated if the specific error condition was enabled (ERRORxxxx xx1). Within the error program, a response to the fault condition can be initiated.

Trigger Interrupt
(INFNCi-aH)

Any trigger input defined as a *Trigger Interrupt* input can be used for two functions—position capture and registration.

Position Capture

Activating either trigger input defined as a *trigger interrupt* input will capture the position of the encoders and motors on both axes.

There is a time delay of up to 50 µs between activating the trigger interrupt input and capturing the position; therefore, the accuracy of the captured position is equal to 50 µs multiplied by the velocity of the axis at the time the input was activated.

☞ *You can change the input debounce time with the INDEB command.*

A trigger interrupt input will be debounced for 50 ms before another interrupt input on that trigger will be recognized. If your application requires a shorter debounce time, you can change it with the INDEB command (refer to the *Input Debounce Time* section discussed earlier). When the trigger interrupt inputs are used with IF, ON, and WAIT statements, it is the **non**-debounced state that is recognized; therefore, rapid transitions, as short as 2 ms, will be noticed by these statements.

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

After the positions are captured, they are stored in their respective registers and are serviced during the next 2-ms update period. The position information is available through the use of the TPCE and TPCM commands to read the captured encoder and motor positions, or through the use of the PCE and PCM commands to assign or compare the captured encoder and motor positions.

System status bits #25 and #26, reported with the TSS and SS commands, are set to 1 when the position has been captured on the respective trigger input (**TRG-A** and **TRG-B**, respectively). As soon as the captured position is transferred (TPCE or TPCM) or assigned/compared (PCE or PCM), the respective system status bit is cleared, but the position information is still available from the register until it is overwritten by a new position capture from the trigger input.

NOTE

If you issue a PSET command, the captured positions (motor and/or encoder) will be offset by the distance specified in the PSET command.

Registration

If registration is enabled (with the RE command), activating a trigger interrupt input will initiate registration move(s) defined with the REG command.

Do this first, before initiating Registration moves:

- Configure one of the trigger inputs (**TRG-A** or **TRG-B**) to function as a trigger interrupt, or *registration*, input; this is done with the INFNCi-H command, where i can be 25 or 26 representing trigger inputs A and B, respectively.
- Issue the INFEN1 command to enable the trigger interrupt/registration function defined with the INFNC command.
- Specify the distance of the registration move with the REG command; **then** you can enable the registration function with the RE command.

The registration move is executed when the trigger interrupt input is activated. There is a time delay of up to 50 µs between activating the trigger interrupt input and capturing the position; however, the accuracy of the registration move distance is equal to ±50 µs multiplied by the velocity of the axis at the time the input was activated.

☞ *You can change the input debounce time with the INDEB command.*

A trigger interrupt input will be debounced for 50 ms before another interrupt input on that trigger will be recognized. If your application requires a shorter debounce time, you can change it with the INDEB command (refer to the *Input Debounce Time* section discussed earlier). When the trigger interrupt inputs are used with IF, ON, and WAIT statements, it is the **non**-debounced state that is recognized; therefore, rapid transitions, as short as 2 ms, will be noticed by these statements.

The registration move distance (REG) is based on the captured position, although the registration move does not start until up to 2 ms after the position is captured. The captured position (motor or encoder) and the positioning mode (motor steps or encoder steps) used for registration depend upon the ENC command setting when the registration move was defined with the REG command; this holds true regardless of the ENC command setting at the time the registration input is activated.

Registration moves will not be executed while the motor is not performing a move, while in the joystick mode (JOY1) or the jog mode (JOG1), or while decelerating due to a stop, kill, soft limit, or hard limit.

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

The registration move may interrupt any preset, continuous, or registration move in progress. When the registration input is activated, the motion profile currently being executed is replaced by the registration profile with its own distance (REG), positioning mode (ENC), acceleration (A), deceleration (AD), and velocity (V) values. (The registration ENC, A, AD, and V values are the ones that were in effect when the REG command was entered.) *NOTE: To prevent position overshoot, the registration distance must be greater than 4 milliseconds multiplied by the incoming velocity.*

The registration move does not alter the rest of the program being executed when registration occurs, nor does it affect commands being executed in the background if the controller is operating in the continuous command execution mode (COMEXC1).

NOTE

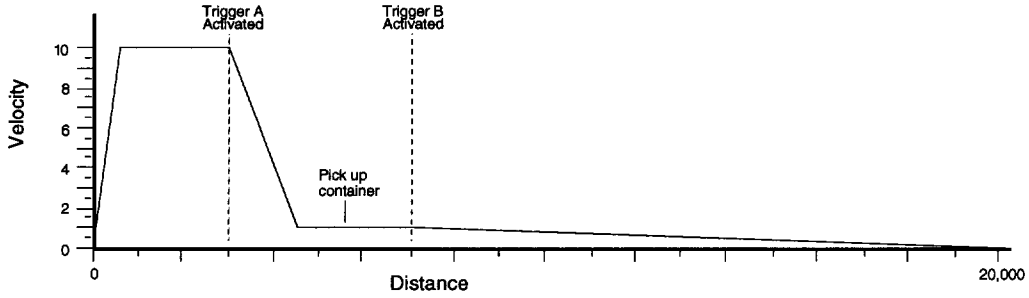
Registration is performed only on the axis or axes with the registration function enabled, and with a non-zero distance specified in the respective axis-designation field of the REG command; the other axes will not be affected.

Each trigger has a distinct move defined for each axis; therefore, with two trigger inputs and two axes available, four different registration moves can be stored.

Registration Example

In this example, two-tiered registration is achieved (see illustration below). While axes 1 is executing it's 50,000-unit move, trigger input A is activated and executes registration move A to slow the load's movement. An open container of volatile liquid is then placed on the conveyor belts. After picking up the liquid and while registration move A is still in progress, trigger input B is activated and executes registration move B to slow the load to gentle stop.

Command	Description
> INFNC25-H	Set input #25 (trigger A) as a trigger interrupt input
> INFNC26-H	Set input #26 (trigger B) as a trigger interrupt input
> INFEN1	Enable programmable input functions defined with the INFNC command
> ENCØx	Set positioning mode to motor step mode
> A2Ø	Set acceleration on axis 1 to 20 units/sec^2
> AD4Ø	Set deceleration on axis 1 to 40 units/sec^2
> V1	Set velocity on axis 1 to 1 unit/sec
> REGA4ØØØ	Set trigger A's registration distance on axis 1 to 4000 units (registration A move will use the ENC, A, AD, & V values above)
> ENCØx	Set positioning mode to motor step mode
> A5	Set acceleration on axis 1 to 5 units/sec^2
> AD2	Set deceleration on axis 1 to 2 units/sec^2
> V.5	Set velocity on axis 1 to 0.5 units/sec
> REGB13ØØØ	Set trigger B's registration distance on axis 1 to 13,000 units (registration B move will use the ENC, A, AD, & V values above)
> RE1Ø	Enable registration on axis 1 only
> A5Ø	Set acceleration to 50 units/sec^2 on axis 1
> AD5Ø	Set deceleration to 50 units/sec^2 on axis 1
> V1Ø	Set velocity to 10 unit/sec on axis 1
> D5ØØØØ	Set distance to 50000 units on axis 1
> GO1Ø	Initiate motion on axis 1



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

Jogging the Motor (INFNCi-aJ) (INFNCi-aK) (INFNCi-aL)

In some applications, you may want to move the motor manually. The 6201 allows you to do this using the jogging functions which are set up with the jogging commands and the input functions commands.

You must define the jogging velocity with the Jog Velocity High (JOGVH) and Jog Velocity Low (JOGVL) commands. The acceleration and deceleration of the JOG move can be configured using JOGA, and JOGAD respectively (refer also to the ***6000 Series Software Reference Guide***).

You can define three different inputs for jogging: CW Jog input (INFNCi-aJ), CCW Jog Input (INFNCi-aK), and Jog Speed Select High/Low (INFNCi-aL). You must also enable the jogging feature with the JOG command.

Once you set up these parameters, you can attach a switch to the jog inputs that you defined and perform jogging. The following example shows how you can define a program to set up jogging.

	Command	**Description**
Step 1	> DEF prog1	Begin definition of program prog1
	- LHØ	Disables the limits (*not needed if you have limit switches installed*)
	- SCALEØ	Disable scaling
	- JOGA25	Set jog acceleration to 25 rps^2
	- JOGAD25	Set jog deceleration to 25 rps^2
	- JOGVL.5	Sets low-speed jog velocity to 0.5 rps
	- JOGVH5	Sets high-speed jog velocity to 5 rps
	- INFEN1	Enable input functions
	- INFNC1-1J	Sets input 1 as a CW jog input
	- INFNC2-1K	Sets input 2 as a CCW jog input
	- INFNC3-1L	Sets input 3 as a speed-select input
	- JOG1	Enables Jog function for axis 1
	- END	End program definition

Step 2 Activate input 1 to move the motor in the CW direction at 0.5 rps (until input 1 is released).

Step 3 Activate input 2 to move the motor in the CCW direction at 0.5 rps (until input 2 is released).

Step 4 Activate input 3 to switch to high-speed jogging.

Step 5 Repeat steps 2 and 3 to perform high-speed jogging.

One-to-One Program Select (INFNCi-aP)

Inputs can be defined as *One-to-One Program Select* inputs (INFNCi-aP). This allows programs defined by the DEF command to be executed by activating an input. Different from BCD Program Select inputs, One-to-One Program Select inputs correspond directly to a specific program number. The program number is determined by the order in which the program was downloaded to the 6201. The program number can be obtained through the TDIR command.

To execute programs using the program select lines, enable the INSELP command for one-to-one program selection. Once enabled, the 6201 will continuously scan the input lines and execute the program selected by the active program select line. To disable scanning of the program select lines, enter !INSELPØ, or place INSELPØ in a program that can be selected.

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

Command	Description
> RESET	Return 6201 to factory defaults
> ERASE	Erase all programs
> DEF proga	Begin definition of program proga
- TPM	Transfer position of motor
- END	End program
> DEF progb	Begin definition of program progb
- TREV	Transfer software revision
- END	End program
> DEF progc	Begin definition of program progc
- TSTAT	Transfer statistics
- END	End program
> INFNC1-1P	Input 1 will select proga
> INFNC2-2P	Input 2 will select progb
> INFNC3-3P	Input 3 will select progc
> INFEN1	Enable input functions
> INSELP2,5Ø	Enable scanning of inputs

You can now execute programs by making a contact closure from an input to ground to activate the input.

- Activate input 1 to execute program #1 (proga)
- Activate input 2 to execute program #2 (progb)
- Activate input 3 to execute program #3 (progc)

Thumbwheel Interface

You can connect the 6201's programmable I/O to a bank of thumbwheel switches to allow operator selection of motion or machine control parameters.

The 6201 allows two methods for thumbwheel use. One method uses Compumotor's TM8 thumbwheel module. The other allows you to wire your own thumbwheels.

The TM8 requires a multiplexed BCD input scheme to read thumbwheel data. Therefore, a decode circuit must be used for thumbwheels. Compumotor recommends that you purchase Compumotor's TM8 module if you desire to use a thumbwheel interface. The TM8 contains the decode logic; therefore, only wiring is needed.

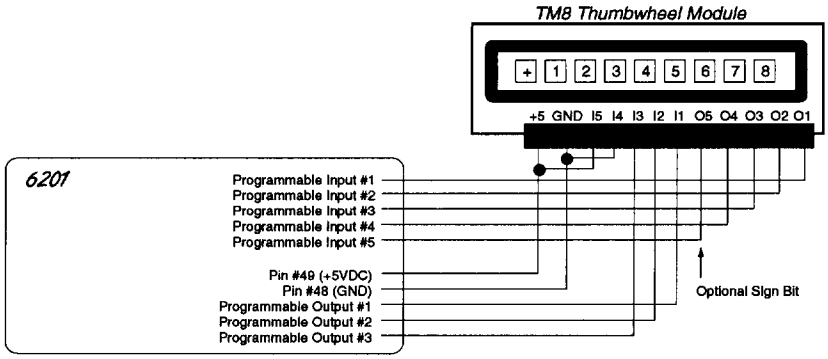
The 6201 commands that allow for thumbwheel data entry are:

Command	Description
INSTW	Establish thumbwheel data inputs (TM8)
OUTTW	Establish thumbwheel data outputs (TM8)
TW	Read thumbwheels or PLC inputs
INPLC	Establish PLC data inputs (Other thumbwheel module)
OUTPLC	Establish PLC data outputs (Other thumbwheel module)

Using the TM8 Module

To use Compumotor's TM8 Module, follow the procedures below.

Step 1 Wire your TM8 module to the 6201 as shown below.



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

Step 2 Configure your 6201 as follows:

Command	Description
> OUTTW1,1-3,Ø,1Ø	Configure thumbwheel output set 1 as follows: outputs 1-3 are strobe outputs, 10 ms strobe time per digit read. **The minimum strobe time recommended for the TM8 module is 10 ms.**
> INSTW1,1-4,5	Configure thumbwheel input set 1 as follows: inputs 1-4 are data inputs, input 5 is a sign input.
> INLVLØØØØØ	Inputs 1-5 configured active low

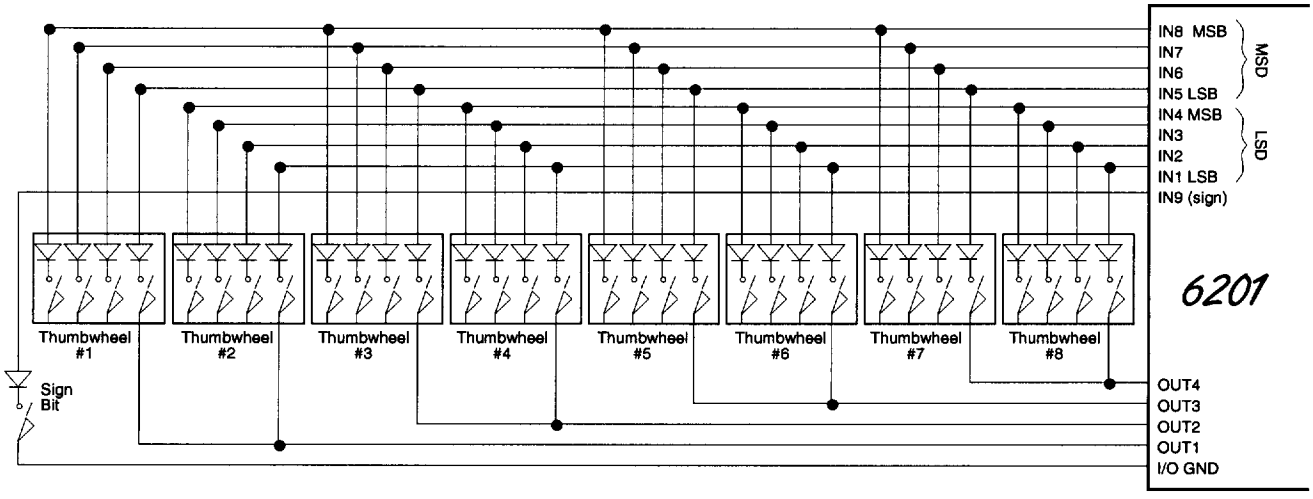
Step 3 Set the thumbwheel digits on your TM8 module to **+12345678**. To verify that you have wired your TM8 module(s) correctly and configured your 6201 I/O properly, enter the following commands:

Command	Description
> VAR1=TW1	Request distance data from all 8 thumbwheel digits
> VAR1	Displays the variable—*VAR1=+Ø.12345678. If you do not receive the response shown, return to step 1 and retry.

Using your own Thumbwheel Module

As an alternative to Compumotor's TM8 Module, you can use your own thumbwheels. The 6201's programming language allows direct input of BCD thumbwheel data via the programmable inputs. Use the following steps to set up and read the thumbwheel interface. Refer to the ***6000 Series Software Reference Guide*** for descriptions of the commands used below.

Step 1 Wire your thumbwheels according to the following schematic.



Step 2 Set up the inputs and outputs for operation with thumbwheels. The data valid input will be an input which the operator holds active to let the 6201 read the thumbwheels. This input is not necessary; however, it is often used when interfacing with PLCs.

Command	Description
> OUTPLC1,1-4,Ø,12	Configure PLC output set 1 as follows: outputs 1-4 are strobe outputs, no output enable bit, 12 ms strobe time per digit read.
> INPLC1,1-8,9	Configure PLC input set 1 as follows: inputs 1-8 are data inputs, input 9 is a sign input, no data valid input.
> INLVLØØØØØØØØØ	Inputs 1-9 configured active low

Step 3 The thumbwheels are read sequentially by outputs, which strobe in two digits at a time. The sign bit is optional. Set the thumbwheels to **+12345678** and type in the following commands:

Command	Description
> VAR1=TW5	Request distance data from all 8 thumbwheel digits
> VAR1	Displays the variable—*VAR1=+12345678. If you do not receive the response shown, return to step 1 and retry.

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

PLC Interface

The 6201's optically-isolated programmable I/O may be connected to most PLCs with discrete inputs and outputs.

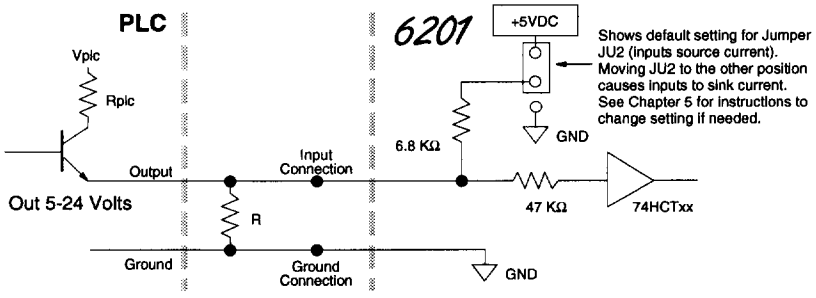
The PLC should be able to sink at least 1mA of current on its outputs. For +5VDC operation, the programmable outputs may be pulled up to +5VDC using the programmable output pull-up (**OUT-P**) on the **AUX** connector; the programmable inputs are internally pulled to +5VDC (default) or to ground, depending on the setting of jumper JU2.

Chapter 5 provides instructions to change jumper JU2.

For voltages up to 24VDC, an external power supply may be used to bias the inputs and outputs (refer to Chapter 5 for specifications).

For higher current or voltages above 24VDC, you must use external signal conditioning such as OPTO-22 compatible I/O signal conditioning racks. Contact your local Compumotor distributor for information on these products.

Certain PLCs have open-emitter outputs. To wire this type of output to an input on the 6201, an external resistor must be wired between the input connection and ground (see illustration below). This provides a path for current to flow from the PLC when the output is active.



Typical value for R = 450Ω (assuming Rpic = 0)
Note: The value of R may vary depending on the value of Rpic and Vpic

Analog Inputs

The 6201 has three 8-bit analog input channels (CH1 - CH3). The analog inputs are configured as three discrete single-ended inputs, with an input range of 0.0V to 2.5V. These inputs can be used to control an axis with a joystick, or to scale velocity during feedrate override. The voltage value on the analog inputs can be read using the ANV or TANV commands. *Refer to Chapter 3 for connection procedures.*

JS6000 Joystick

The 6201 is compatible with the Daedal JS6000 joystick. To order the JS6000, contact Daedal at (800) 245-6903 or contact your local Automation Technology Center.

Joystick Control

Joystick control can be achieved by simply connecting a joystick potentiometer to one of the analog inputs. Joystick operation is enabled with the JOY1 command.

Travel limitations in potentiometers and voltage drops along the cables may make it impossible to achieve the full 0.0V to 2.5V range at the joystick input. Therefore, you must configure the 6201 to optimize the joystick's usable voltage range. This configuration will affect the *velocity resolution*. The velocity resolution is determined by the following equation:

$$\frac{\text{maximum velocity set with the JOYVH or the JOYVL command}}{\text{voltage range between the joystick's no-velocity region (}\textit{center deadband}\text{) and its maximum-velocity region (}\textit{end deadband}\text{)}}$$

To establish the velocity resolution, you must define the full-scale velocity and the usable voltage.

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

Define Full-Scale Velocity

You must define the full-scale velocity for your application with the JOYVH and JOYVL commands. Both commands establish the maximum velocity that can be obtained by deflecting the potentiometer fully CW or fully CCW. The JOYVH command establishes the *high* velocity range (selected if the joystick select input is high—sinking current). The JOYVL command establishes the *low* velocity range (selected if the joystick select input is low—not sinking current).

The JOYAXL and JOYAXH commands define which analog channels are to be used with which joystick axes when the joystick select input is low or high, respectively.

Define Usable Voltage

Use the commands described in the following table to establish the joystick's usable velocity range.

The analog-to-digital converter is an 8-bit converter with a voltage range of 0.0V to 2.5V. With 8 bits to represent this range, there are 256 distinct voltage levels from 0.0V to 2.5V. 1 bit represents 2.5/256 or 0.00976 volts/bit.

Command	Name	Purpose
JOYEDB	End Deadband	This command defines voltage levels (shy of the 0.0V and 2.5V endpoints) at which maximum velocity occurs. Specifying an end deadband effectively decreases the voltage range of the analog input to compensate for joysticks that cannot reach the 0.0V and 2.5V endpoints.
JOYCTR (or JOYZ)	Center Voltage*	This command defines the voltage level for the center of the analog input range (the point at which zero velocity will result). As an alternative, you can use the JOYZ command, which reads the current voltage on the joystick input and considers it the center voltage. You can check the center voltage by typing in JOYCTR[cr].**
JOYCDB	Center Deadband	This command defines the voltage range on each side of the center voltage in which no motion will occur (allows for minor drift or variation in the joystick center position without causing motion).

* Because the center voltage can be set to a value other than the exact center of the potentiometer's voltage range, and because there could be two different velocity resolutions, the CW velocity resolution may be different than CCW velocity resolution.

** Because of finite voltage increments the 6201 will not report back exactly what you specified with the JOYCTR command.

Joystick Control Inputs

The table below lists the 6201's four joystick control inputs and their active levels and what the active levels affect.

Joystick Input Bit	Active Level	Effect of Active Level
Axis select	Ø 1	Selects JOYAXL Selects JOYAXH
Velocity select	Ø 1	Selects JOYVL Selects JOYVH
Joystick release	Ø 1	Exit Joystick Mode (equiv. to !JOYØØ command). To use the joystick again, issue the JOY11 command. Stay in Joystick Mode
Joystick trigger (general purpose)	Ø or 1	Interpreted by user program (status is reported with the TINO and INO commands)
Joystick auxiliary (general purpose)	Ø or 1	Interpreted by user program (status is reported with the TINO and INO commands)

Typical Applications

A typical joystick application is two-axis, in which a high velocity range is required to move to a region, then a low velocity range is required for a fine search. After the search is completed it is necessary to record the motor positions, then move to the next region. The joystick trigger input can be used to indicate that the position should be read. The joystick release is used to exit the joystick mode and continue with the motion program.

Joystick Set Up Example

The following table describes the requirements of the application described above, and how the 6201 is configured to satisfy those requirements. The resulting joystick voltage configuration is illustrated below. Given: one analog input channel is used for each axis.

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

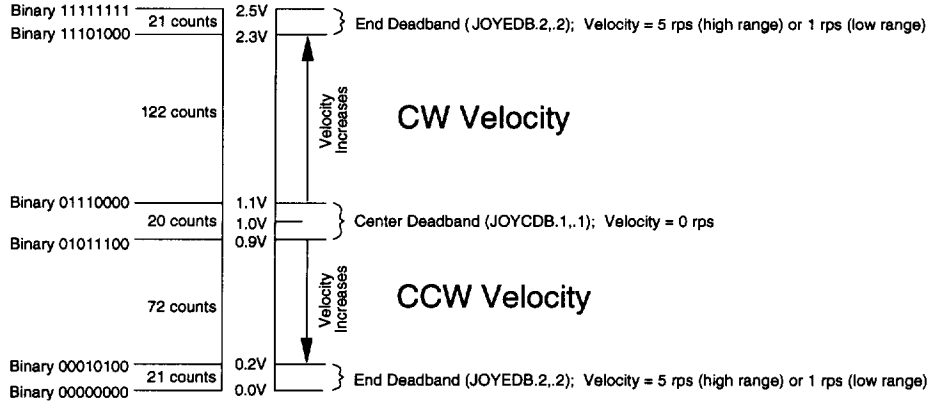
Requirement	Configuration
Set max. high-range velocity to 5 rps (on both axes)	Type in the `JOYVH5,5` command
Set max. low-range velocity to 1 rps (on both axes)	Type in the `JOYVL1,1` command
No velocity when voltage is at 1.0V	Set center voltage with `JOYCTR1,1,` command, or set voltage level at both analog inputs to 1.0V and type in `JOYZ11`
Joystick cannot reliably rest at 1.0V, but can rest within ±0.1V of 1.0V	Set center deadband of 0.1V with `JOYCDB.1,.1` command (0.1V is the system default)
Joystick can only produce maximum of 2.3V and minimum of 0.2V	Set end deadband to get max. velocity at 2.3V or 0.2V with the `JOYEDB.2,.2` command.
	Voltage range: CW = 1.1V to 2.3V (1.2V total) CCW = 0.9V to 0.2V (0.7V total)
	Voltage resolution: see below

The high-range velocity resolutions (at 5 rps max.) is calculated as follows:

CW: $\frac{5 \text{ rps}}{\text{voltage range of 1.2V (122 counts)}} = 0.041$ rps/count; CCW: $\frac{5 \text{ rps}}{\text{voltage range of 0.7V (72 counts)}} = 0.069$ rps/count

The low-range velocity resolutions (at 1 rps max.) is calculated as follows:

CW: $\frac{1 \text{ rps}}{\text{voltage range of 1.2V (122 counts)}} = 0.008$ rps/count; CCW: $\frac{1 \text{ rps}}{\text{voltage range of 0.7V (72 counts)}} = 0.014$ rps/count



Analog Voltage Override

Before you actually wire the analog inputs, you can simulate their activation in software by using the `ANVO` command. For instance, `ANVO1.2,1.6,1.8` overrides the hardware analog input channels—1.2V on channel 1, 1.6V on channel 2, and 1.8V on channel 3. The `ANVO` values are used in any command or function that references the analog input channels, but only those channels for which `ANVOEN` is set to 1 (e.g., Given `ANVOENØ11`, the `ANVO` values 1.6V and 1.8V are referenced for analog channels 2 and 3 only.).

Feedrate Override

Feedrate override is used to synchronously scale all phases of motion (except distance) on both axes. The amount of scaling is expressed in terms of percentage from 0 to 100. The percentage of feedrate can be controlled by an analog voltage or by the `FRPER` command.

When feedrate override is enabled, the frequency at which the 6201 motion algorithm updates the velocity output varies according to the feedrate percentage specified. Without feedrate override, the motion algorithm is updated every 2 ms. At 100%, the velocity output is updated every 2 ms. At 50%, the velocity output is updated every 4 ms.

Using Feedrate Override While Contouring

When you enter or exit the feedrate override mode with the `FR` command, you will have to recompile (`PCOMP`) any previously compiled contouring paths.

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

Hardware Feedrate Override (FR1)

To adjust feedrate using an analog voltage, enable feedrate control using the FR1 command. The lower the voltage, the lower the feedrate percentage. The higher the voltage, the higher the feedrate percentage. The end-deadband (JOYEDB) for the analog input channel is in effect during feedrate control, while the center-deadband has no effect. If the end-deadband is zero, ØVDC commands 0% feedrate, and 2.5VDC commands 100% feedrate. The velocity update (2 ms at 100%) can be calculated as follows:

$$\frac{2 * (2.5 - 2 * \text{JOYEDB})}{\text{Analog Voltage} - \text{JOYEDB}}$$

Command	Description
FRA5ØØØØ	Set the feedrate acceleration to 50000 percent/sec^2
FRH1	Control the axes with analog input 1 when the axes select input is active (Joystick connector pin 15 connected to **GND**)
FRL2	Control the axes with analog input 2 when the axes select input is not active (Joystick connector pin 15 not connected to **GND**)
FR1	Enable feedrate override
A1ØØ	Set axis 1 acceleration to 100 units/sec^2
V1Ø	Set axis 1 velocity to 10 units/sec
D1ØØØØØ	Set axis 1 distance to 100000 units
GO1	Initiate motion on axis 1. If at any time during the move the voltage on analog input #1 changes, the velocity of this move will change correspondingly.

Software Feedrate Override (FR2)

To adjust the feedrate using the FRPER command, enable feedrate control using the FR2 command. The feedrate percentage can then be specified directly. The command FRPER5Ø would set the feedrate to 50%, while FRPER1ØØ would set the feedrate to 100%. The velocity update (2 ms at 100%) can be calculated with the following equation:

$$\frac{2 * 100}{\text{FRPER}}$$

The example below demonstrates using the software command FRPER to control axis 1 when feedrate override is enabled. Within the example, inputs 1, 2, and 3 are used to control the feedrate override percentage. When input #3 is activated, the original move velocity is doubled. When input #2 is activated, the move slows down to half its original velocity. Input #1 returns the move to its original velocity. Input #4 stops the move and continues command processing with the command after the NWHILE command.

Command	Description
> DEF temp	Begin definition of program temp
- FRA5ØØØØ	Set the feedrate acceleration to 50000 percent/sec^2
- FRPER5Ø	Feedrate override percentage equals 50%
- FR2	Enable feedrate override
- MC1	Enable continuous mode moves
- A1ØØ	Set acceleration
- V4	Set velocity
- D+	Set direction to CW
- COMEXC1	Enable continuous command processing mode
- GO1	Initiate motion on axis 1
- WHILE(IN=bXXXØ)	Repeat commands between WHILE and NWHILE until input #4 becomes active
- IF(IN=b1)	If input #1 is active, change the feedrate percentage to 50%
- FRPER5Ø	Set feedrate percentage to 50%
- NIF	End IF command
- IF(IN=bX1)	If input #2 is active, change the feedrate percentage to 25%
- FRPER25	Set feedrate percentage to 25%
- NIF	End IF command
- IF(IN=bXX1)	If input #3 is active, change the feedrate percentage to 100%
- FRPER1ØØ	Set feedrate percentage to 100%
- NIF	End IF command
- NWHILE	End WHILE command
- S1	Stop motion on axis 1
- COMEXCØ	Disable continuous command processing mode
- END	End definition of program temp
> RUN temp	Initiate program temp

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

RP240 Remote Operator Panel Interface

The 6201 is directly compatible with the Compumotor RP240 Remote Operator Panel. This section describes how to use the 6201 with the RP240. RP240 connections are demonstrated in Chapter 3, *Installation.*

NOTE

Refer to the *Model RP240 User Guide* (p/n 88-012156-01), shipped with every RP240, for user information on the following:

- *Hardware Specifications*
- *Environmental Considerations*
- *Mounting Guidelines*
- *Troubleshooting*

Operator Interface Features

The RP240 is used as the 6201's remote *operator interface*, not a program entry terminal. As an operator interface, the RP240 offers the following features:

- Displays text and variables
- 8 LEDs can be used as programmable status lights
- Operator data entry of variables: read data from RP240 into variables and command value substitutions (see table in Appendix B of S/W guide)

Typically the user creates a program in the 6201 to control the RP240 display and RP240 LEDs. The program can read data and make variable assignments via the RP240's keypad and function keys.

The 6000 Series software commands for the RP240 are listed below:

```
DCLEAR .....Clear The RP240 Display
DJOG .....Enter RP240 Jog Mode
DLED .....Turn RP240 Leds On/Off
DPASS .....Change RP240 Password
DPCUR .....Position The Cursor On The RP240 Display
[DREAD] .....Read RP240 Data
[DREADF] .....Read RP240 Function Key
DREADI .....RP240 Data Read Immediate Mode
DVAR .....Display Variable On RP240
DWRITE"" .....Display Text On The RP240 Display
```

The example below demonstrates the majority of the 6000 Series commands for the RP240.

Example

Command	**Description**
`> DEF panel1`	Define program `panel1`
`- REPEAT`	Start of repeat loop
`-  DCLEARØ`	Clear display
`-  DWRITE"SELECT A FUNCTION KEY"`	Display text `"SELECT A FUNCTION KEY"`
`-  DPCUR2,2`	Move cursor to line 2 column 2
`-  DWRITE"DIST"`	Display text `"DIST"`
`-  DPCUR2,9`	Move cursor to line 2 column 9
`-  DWRITE"GO"`	Display text `"GO"`
`-  DPCUR2,35`	Move cursor to line 2 column 35
`-  DWRITE"EXIT"`	Display text `"EXIT"`
`-  VAR1 = DREADF`	Input a function key
`-  IF (VAR1=1)`	If function key #1 hit
`-  GOSUB panel2`	GOSUB program `panel2`
`-  ELSE`	Else
`-  IF (VAR1=2)`	If function key #2 hit
`-  DLED1`	Turn on LED #1
`-  GO1`	Start motion on axis #1
`-  DLEDØ`	Turn off LED #1
`-  NIF`	End of `IF (VAR1=2)`
`-  NIF`	End of `IF (VAR1=1)`
`- UNTIL (VAR1=6)`	Repeat until `VAR1=6` (function key 6)
`- DCLEARØ`	Clear display
`- DWRITE"LAST FUNCTION KEY = F"`	Display text `"LAST FUNCTION KEY = F"`
`- DVAR1,1,Ø,Ø`	Display variable 1
`- END`	End of `panel1`
`>`	
`> DEF panel2`	Define prog `panel2`
`-  DCLEARØ`	Clear display
`-  DWRITE"ENTER DISTANCE"`	Display text `"ENTER DISTANCE"`
`-  D(DREAD)`	Enter distance number from RP240
`- END`	End of `panel2`

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

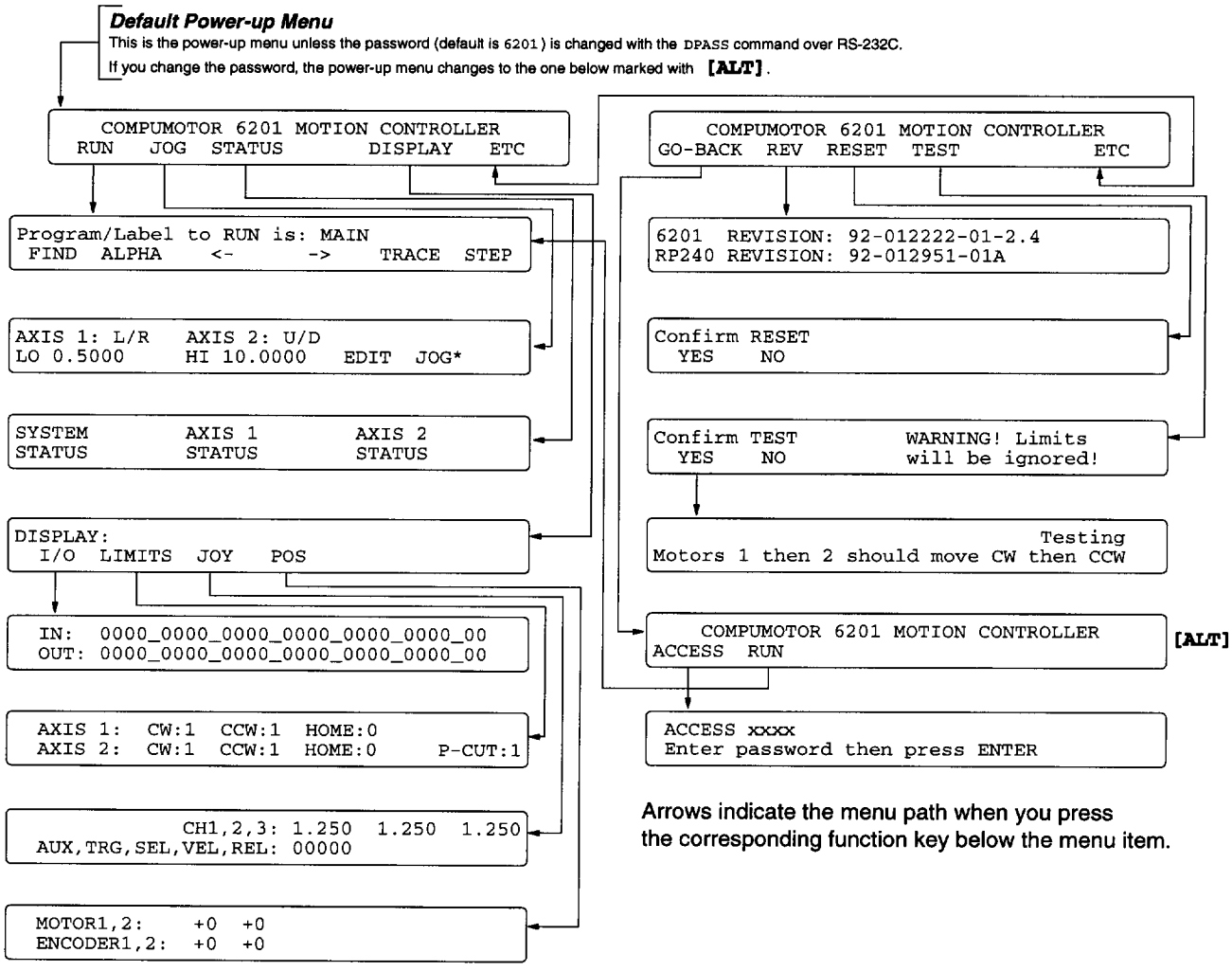
Using the Default Mode

In addition to the 6201/RP240 operator interface features, there are some other built-in features that are described here.

On power-up, the 6201 will automatically default to a mode in which it controls the RP240 with the menu-driven functions listed below. To disable this menu, a power-up user program (STARTP) must contain the DCLEARØ command.

- Run a stored program (RUN, STOP, PAUSE and CONTINUE functions)
- Jogging
- Display status of I/O and analog inputs and position (TIN, TOUT, TLIM, TANV, TINO, TPM, TPE values can be displayed on the display)
- Display revision levels of the RP240 and the 6201 software
- Run the TEST program
- RESET the 6201

The flow chart below illustrates the RP240's menu structure in the default operating mode (when no 6201 user program is controlling the RP240). Press the ***Menu Recall*** key to back up to the previous screen. Each menu item is described below.



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

Menu Screen	Description
COMPUMOTOR 6201 MOTION CONTROLLER RUN JOG STATUS DISPLAY ETC	**Default menu** (first half): This is the default menu. Function Key — Description RUN — Go to the RUN menu JOG — Go to the JOG menu (enter RP240 jog mode) STATUS — Go to the STATUS menu DISPLAY — Go to the DISPLAY menu ETC — Go to the second half of the default menu
COMPUMOTOR 6201 MOTION CONTROLLER GO-BACK REV RESET TEST ETC	**Default menu** (second half): Function Key — Description GO-BACK — Go back to the ACCESS menu REV — Display revision levels RESET — RESET the 6201 TEST — Execute the TEST command ETC — Go to the first half of the default menu
SYSTEM STATUS AXIS 1 STATUS AXIS 2 STATUS	**Status Menu:** You can select a system status or axis status display. You can then scroll through the status bits while getting the bit descriptions. Function Key — Description SYSTEM STATUS — System status bits (TSS) AXIS 1 STATUS — Axis 1 Status bits (TAS) AXIS 2 STATUS — Axis 2 Status bits (TAS)
Program/Label to RUN is: MAIN FIND ALPHA <- -> TRACE STEP	**Run menu**: You can select or edit a program name to be RUN. Paths cannot be RUN, you must use PRUN (but PRUN can be placed in a program that can be RUN). You can enable RP240 trace mode and/or step mode. By pressing ENTER, the program name shown will be searched for and run. When a program is RUN and TRACE is selected, the RP240 display will trace all program commands as they are executed. This is different from the TRACE command in that the trace output goes to the RP240 display, not to a terminal via the RS-232C port. When a program is RUN and STEP is selected, step mode has been entered. This is similar to the STEP command, but when selected from the RUN menu, step mode also allows single stepping by pressing the ENTER key. Both RP240 trace mode and step mode are exited when program execution is terminated. Function Key — Description FIND — Find program names to run ALPHA — Allows entry of alpha characters <- — Backspace for editing <- — Forward space for editing TRACE — Enable RP240 trace mode STEP — Enable step mode
AXIS 1: L/R AXIS 2: U/D LO 0.5000 HI 10.0000 EDIT JOG*	**Jog menu**: You can jog individual axes by pressing the RP240 arrow keys. Pressing an arrow key will start motion and releasing the arrow key will stop motion (using the jog acceleration and deceleration values specified by JOGA and JOGAD). The left and right arrow keys correspond to axis #1 CCW and CW motion. The up and down arrows keys are for axis #2. You may select either the jog low velocity or the jog high velocity by pressing the appropriate LO/HI function key. You may edit the jog velocity by pressing the EDIT function key, then selecting which velocity you want to edit. Once a cursor is placed under the desired velocity, you can change the number by using the numeric keypad and pressing ENTER when done. To jog with the new velocity, first press the JOG function key to enable the arrow keys again. Function Key — Description LO/HI — Select either jog low velocity or jog high velocity EDIT — Enable edit of jog velocities JOG — Enable jog arrow keys
DISPLAY: I/O LIMITS JOY POS	**Display menu**: You can select several possible displays. Once a particular display has been selected, the 6201 will continually update the information to the RP240's display until the display has been exited. Function Key — Description I/O — Display 24 inputs and 24 outputs LIMITS — Display CW, CCW, Home Enable, & P-CUT inputs JOY — Display the 3 analog channel voltages, and the associated joystick connector inputs POS — Display motor and encoder counts for both axes
COMPUMOTOR 6201 MOTION CONTROLLER ACCESS RUN	**Access menu**: If you press the GO-BACK function key at the default menu (second half), the 6201/RP240 will go back one additional level to the access menu. The access menu allows entry of the user definable RP240 password (DPASS). At this access menu level, only the run menu is allowed if the correct password has not been entered. The default password is 6201. If the password is modified with the DPASS command, the access menu then becomes the new default menu (the password must then be entered to get to the original default menu). Function Key — Description ACCESS — Allows entry of the RP240 password RUN — Go to the RUN menu

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

`6201 REVISION: 92-012222-01-2.4` `RP240 REVISION: 92-012951-01A`	**Rev display:** If you press the REV function key at the default menu (second half), the 6201/RP240 will display the current 6201 and RP240 software revision levels.
`Confirm RESET` `YES NO`	**Reset menu:** Allows you to reset the 6201 (same as entering the RESET command). Function Key — Description YES — Perform reset of 6201 NO — Return to default menu (second half)
`Confirm TEST WARNING! Limits` `YES NO will be ignored!`	**Test menu:** Allows you to test the 6201 (same as entering the TEST command). Function Key — Description YES — Execute TEST sequence NO — Return to default menu (second half)

Host Computer Operation

Another choice for a user interface is to use a host computer and execute a motion program using the RS-232C serial interface. A host computer may be used to run a motion program interactively from a BASIC or C program (high-level program controls the 6201 and acts as a user interface). A BASIC program example is provided below.

```
10  '       6000 Series Serial Communication BASIC Routine
12  '                          6000.BAS
14  '
16  ' ************************************************************
18  '
20  ' This program will set the communications parameters for the
22  ' serial port on a PC in order to communicate with a 6000 series
24  ' stand-alone product.
26  '
28  ' ************************************************************
30  '
100 '*** open com port 1 at 9600 baud, no parity, 8 data bits, 1 stop bit
110 '*** activate Request to Send (RS), suppress Clear to Send (CS), suppress
120 '*** DATA set ready (DS), and suppress Carrier Detect (CD) ***
130 OPEN "COM1:9600,N,8,1,RS,CS,DS,CD" FOR RANDOM AS #1
140 '
150 '*** initialize variables ***
160 MOVE$ = ""          ' *** commands to be sent to the product ***
170 RESPONSE$ = ""      ' *** response from the product ***
180 POSITION$ = ""      ' *** motor position reported ***
190 SETUP$ = ""         ' *** setup commands ***
200 '
210 '*** format the screen and wait for the user to start the program ***
220 CLS : LOCATE 12, 20
230 PRINT "Press any key to start the program"
240 '
250 '*** wait for the user to press a key ***
260 PRESS$ = INKEY$
270 IF PRESS$ = "" THEN 260
280 CLS
290 '
300 '*** set a pre-defined move to make ***
310 SETUP$ = "ECHO1:ERRLVL0:LH0,0:"
320 MOVE$ = "A100,100:V2,2:D50000,50000:GO11:TPM:"
330 '
340 '
400 '*** send the commands to the product ***
410 PRINT #1, SETUP$
420 PRINT #1, MOVE$
430 '
500 '*** read the response from the TPM command ***
510 '    *** the indexer will send a leading "+" or "-" in response to the TPM command to
520 '    *** indicate which direction the motor traveled. ***
530 WHILE (RESPONSE$ <> "+" AND RESPONSE$ <> "-") ' *** this loop waits for the "+"
540   RESPONSE$ = INPUT$(1, #1)                    ' *** or "-" characters to be returned
550 WEND                                           ' *** before reading the position ***
560 '
570 WHILE (RESPONSE$ <> CHR$(13))          ' *** this loop reads one character at a time
580   POSITION$ = POSITION$ + RESPONSE$    ' *** from the serial buffer until a carriage
590   RESPONSE$ = INPUT$(1, #1)            ' *** return is encountered ***
600 WEND
610 '
620 '*** print the response to the screen ***
630 LOCATE 12, 20: PRINT "Motor position is " + POSITION$
640 '
650 'END
```


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

Variables

The 6201 has 3 types of variables (numeric, binary, and string). There are 150 numeric variables, numbered 1 - 150. There are 25 binary and string variables, numbered 1 - 25. Each type of variable is designated with a different command. The VAR command designates a numeric variable, the VARB command designates a binary variable, and the VARS command designates a string variable.

Variables do not share the same memory (i.e., VAR1, VARB1, and VARS1 can all exist at the same time and operate separately).

Numeric variables are used to store numeric values with a range of -999,999,999.ØØØØØØØØ to 999,999,999.99999999. Mathematical, trigonometric, and boolean operations are performed using numeric variables.

Binary variables can be used to store 32-bit binary or hexadecimal values. Binary variables can also store I/O, system, axis, or error status (e.g., the VARB2=IN.12 command assigns input bit 12 to binary variable 2). Bitwise operations are performed using binary variables.

String variables are used to store message strings of 20 characters or less. These message strings can be predefined error messages, user messages, etc.

NOTE

The programming examples in the remainder of this section make use of the colon (:) command delimiter to allow entering more than one command per line.

Converting Between Binary and Numeric Variables

Using the Variable Type Conversion (VCVT) operator, you can convert numeric values to binary values, and vice versa. The operation is a signed operation as the binary value is interpreted as a two's complement number. Any *don't care*s (x) in a binary value is interpreted as a zero (Ø).

If the mathematical statement's result is a numeric value, then VCVT converts binary values to numeric values. If the statement's result is a binary value, then VCVT converts numeric values to binary values.

Numeric to Binary

Example	Description/Response
> VAR1=-5	Set numeric variable value = -5
> VARB1=VCVT(VAR1)	Convert the numeric value to a binary value
> VARB1	*VARB1=11Ø1_1111_1111_1111_1111_1111_1111_1111

Binary to Numeric

Example	Description/Response
> VARB1=bØØ1Ø_Ø11Ø_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ	Set binary variable = +100.0
> VAR1=VCVT(VARB1)	Convert the binary value to a numeric value
> VAR1	*VAR1=+1ØØ.Ø

Performing Operations with Numeric Variables

Some Math Operations Reduce Precision

The following math operations reduce the precision of the return value: Division and Trigonometric functions yield 5 decimal places; Square Root yields 3 decimal places; and Inverse Trigonometric functions yield 2 decimal places.

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

Mathematical Operations

The following examples demonstrate how the 6201 can perform math operations with its numeric variables.

Addition (+)

Example	Response
> VAR1=5+5+5+5+5+5+5 : VAR1	VAR1=35.Ø
> VAR23=1ØØØ.565	
> VAR11=VAR1+VAR23 : VAR11	*VAR11=+1Ø35.565
> VAR1=VAR1+5 : VAR1	*VAR1=+4Ø.Ø

Subtraction (-)

Example	Response
> VAR3=2Ø-1Ø	
> VAR2Ø=15.5	
> VAR3=VAR3-VAR2Ø : VAR3	*VAR3=-5.5

Multiplication (*)

Example	Response
> VAR3=1Ø	
> VAR3=VAR3*2Ø : VAR3	*VAR3=+2ØØ.Ø

Division (/)

Example	Response
> VAR3=1Ø	
> VAR2Ø=15.5 : VAR2Ø	*+15.5
> VAR3=VAR3/VAR2Ø : VAR3	*+Ø.64516
> VAR3Ø=75 : VAR3Ø	*+75.Ø
> VAR19=VAR3Ø/VAR3 : VAR19	*+116.25Ø23

Square Root

Example	Response
> VAR3=75	
> VAR2Ø=25	
> VAR3=SQRT(VAR3) : VAR3	*+8.66Ø
> VAR2Ø=SQRT(VAR2Ø)+SQRT(9)	
> VAR2Ø	*+8.Ø

Trigonometric Operations

The following examples demonstrate how the 6201 can perform trigonometric operations with its numeric variables.

Sine

Example	Response
> RADIANØ	
> VAR1=SIN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=SIN(3Ø) : VAR1	*VAR1=+Ø.5
> VAR1=SIN(45) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=SIN(6Ø) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=SIN(9Ø) : VAR1	*VAR1=+1.Ø
> RADIAN1	
> VAR1=SIN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=SIN(PI/6) : VAR1	*VAR1=+Ø.5
> VAR1=SIN(PI/4) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=SIN(PI/3) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=SIN(PI/2) : VAR1	*VAR1=+1.Ø

Cosine

Example	Response
> RADIANØ	
> VAR1=COS(Ø) : VAR1	*VAR1=+1.Ø
> VAR1=COS(3Ø) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=COS(45) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=COS(6Ø) : VAR1	*VAR1=+Ø.5
> VAR1=COS(9Ø) : VAR1	*VAR1=+Ø.Ø
> RADIAN1	
> VAR1=COS(Ø) : VAR1	*VAR1=+1.Ø
> VAR1=COS(PI/6) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=COS(PI/4) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=COS(PI/3) : VAR1	*VAR1=+Ø.5
> VAR1=COS(PI/2) : VAR1	*VAR1=+Ø.Ø

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

Tangent

```
Example                          Response
> RADIANØ
> VAR1=TAN(Ø) : VAR1             *VAR1=+Ø.Ø
> VAR1=TAN(3Ø) : VAR1            *VAR1=+Ø.57735
> VAR1=TAN(45) : VAR1            *VAR1=+1.Ø
> VAR1=TAN(6Ø) : VAR1            *VAR1=+1.732Ø5
> RADIAN1
> VAR1=TAN(Ø) : VAR1             *VAR1=+Ø.Ø
> VAR1=TAN(PI/6) : VAR1          *VAR1=+Ø.57735
> VAR1=TAN(PI/4) : VAR1          *VAR1=+1.Ø
> VAR1=TAN(PI/3) : VAR1          *VAR1=+1.732Ø5
```

Inverse Tangent (Arc Tangent)

```
Example                          Response
> RADIANØ
> VAR1=SQRT(2)
> VAR1=ATAN(VAR1/2) : VAR1       *VAR1=+35.26
> VAR1=ATAN(.57735) : VAR1       *VAR1=+3Ø.Ø
```

Boolean Operations

The following examples demonstrate how the 6201 can perform boolean operations with its numeric variables. Refer to the ***6000 Series Software Reference Guide*** for more information on boolean operations.

Boolean And (&)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 & VAR2 : VAR3        *VAR3=+Ø.Ø
```

Boolean Or (|)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 | VAR2 : VAR3        *VAR3=+1.Ø
```

Boolean Exclusive Or (^)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 ^ VAR2 : VAR3        *VAR3=+1.Ø
```

Boolean Not (~)

```
Example                          Response
> VAR1=5
> VAR3=~(VAR1) : VAR3            *VAR3=+Ø.Ø
> VAR1=-1
> VAR3=~(VAR1) : VAR3            *VAR3=+1.Ø
```

Performing Operations with Binary Variables

The 6201 has the ability to perform bitwise functions with its binary variables. The following examples illustrate the bit manipulation capabilities of the 6201.

Bitwise And (&)

```
Example   > VARB1=b11Ø1 : VARB1
Response  *VARB1=11Ø1_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example   > VARB1=VARB1 & bXXX1 11Ø1 : VARB1
Response  *VARB1=XXØ1_XXØX_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example   > VARB1=hØØ32 FDA1 & h1234 43E9 : VARB1
Response  *VARB1=ØØØØ_ØØØØ_11ØØ_ØØØØ_ØØ1Ø_1ØØØ_Ø1Ø1_1ØØØ
```

Bitwise Or (|)

```
Example   > VARB1=h32FD : VARB1
Response  *VARB1=11ØØ_Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ_ØØØØ

Example   > VARB1=VARB1 | bXXX1 11Ø1 : VARB1
Response  *VARB1=11X1_11Ø1_1111_1X11_XXXX_XXXX_XXXX_XXXX

Example   > VARB1=hØØ32 FDA1 | h1234 43E9 : VARB1
Response  *VARB1=1ØØØ_Ø1ØØ_11ØØ_Ø11Ø_1111_1111_Ø111_1ØØ1
```

Bitwise Exclusive Or (^)

```
Example   > VARB1=h32FD ^ bXXX1 11Ø1 : VARB1
Response  *VARB1=XXX1_1ØØ1_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example   > VARB1=hØØ32 FDA1 ^ h1234 43E9 : VARB1
Response  *VARB1=1ØØØ_Ø1ØØ_ØØØØ_Ø11Ø_11Ø1_Ø111_ØØ1Ø_ØØØ1
```

Bitwise Not (~)

```
Example   > VARB1=~(h32FD) : VARB1
Response  *VARB1=ØØ11_1Ø11_ØØØØ_Ø1ØØ_1111_1111_1111_1111

Example   > VARB1=~(b1Ø1Ø XX11 Ø1Ø1) : VARB1
Response  *VARB1=Ø1Ø1_XXØØ_1Ø1Ø_XXXX_XXXX_XXXX_XXXX_XXXX
```


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

Shift Left to Right (>>)	**Example**	`> VARB1=h32FD >> h4 : VARB1`
	Response	`*VARB1=ØØØØ_11ØØ_Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ`
	Example	`> VARB1=b1Ø1Ø XX11 Ø1Ø1 >> b11 : VARB1`
	Response	`*VARB1=ØØØ1_Ø1ØX_X11Ø_1Ø1X_XXXX_XXXX_XXXX_XXXX`
Shift Right to Left (<<)	**Example**	`> VARB1=h32FD << h4 : VARB1`
	Response	`*VARB1=Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ`
	Example	`> VARB1=b1Ø1Ø XX11 Ø1Ø1 << b11 : VARB1`
	Response	`*VARB1=ØXX1_1Ø1Ø_1XXX_XXXX_XXXX_XXXX_XXXX_XØØØ`

X-Y Linear Interpolation

The 6201 allows you to perform *X-Y linear interpolation*, the process of moving two orthogonal (right angle) linear axes to achieve linear (straight line) motion. The task is to derive appropriate move parameters to move from a current location to a new location, where each position is specified by a set of *Cartesian coordinates*. Both axes must start, accelerate, decelerate, and stop in a synchronized manner.

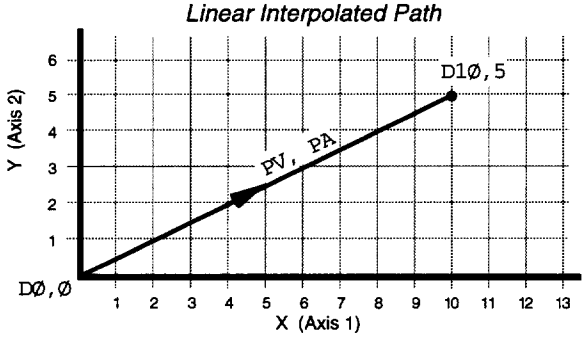
The Initiate Linear Interpolated Motion (GOL) command initiates linear interpolation moves based on the parameters set with the D, PA, PAD, and PV commands. You simply enter the desired path acceleration (PA), the path deceleration (PAD), and the path velocity (PV) to arrive at the point in space (*end point*) specified with the distance (D) command; the 6201 internally calculates each axis' actual move profiles to achieve a straight-line path with these parameters.

You can scale the acceleration, velocity, and distance with the PSCLA, PSCLV, and SCLD commands, respectively (see example below).

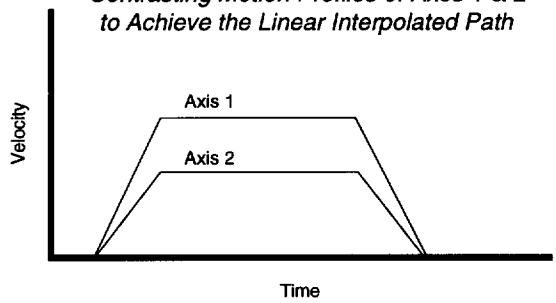
The GOL command starts motion on either or both axes. If the GOL command is issued without any arguments, motion will be started on both axes.

Example **Command**	**Description**
`> SCALE1`	Enable scaling
`> PSCLA25ØØØ`	Set path acceleration scale factor to 25000 steps/unit
`> PSCLV25ØØØ`	Set path velocity scale factor to 25000 steps/unit
`> @SCLD1ØØØØ`	Set distance scale factor to 10000 step/unit on all axes
`> PA25`	Set the path acceleration to 25 units/sec^2
`> PAD2Ø`	Set the path deceleration to 20 units/sec^2
`> PV2`	Set the path velocity to 2 units/sec
`> D1Ø,5`	Set the distance to 10 & 5 units on axes 1 & 2, respectively
`> GOL11`	Initiate linear interpolated motion on all axes (**see figure below**) (a GOL command could have been issued instead of a GOL11 command)

Linear Interpolated Path



Contrasting Motion Profiles of Axes 1 & 2 to Achieve the Linear Interpolated Path



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

Contouring (Circular Interpolation)

The 6201 allows you to define and execute two-dimensional motion paths. A *path* refers to the path traveled by the load in an X-Y plane, and must be defined before any motion takes place along that path. The X and Y axes can be specified as either of the 6201's two axes.

A *path* consists of one or more line or arc segments whose end-points are specified in terms of X and Y positions. The end-point position specifications may be made using either absolute or incremental programming. The segments may be lines or arcs, both of which are described in greater detail in the following sections. Each path segment is determined by the end-point coordinates, and in the case of arcs, by the direction and radius or center. It is possible to accelerate, decelerate or travel at constant velocity (feedrate) during any type of segment, even between segments. For each segment, the user may also specify an output pattern which can be applied to the programmable outputs at the beginning of that segment.

All paths are continuous paths, (i.e., the motion will not stop between path segments, but must stop at the end of a path). It is not possible to define a path that stops motion within the path definition and then continues that path. To achieve this result, two individual paths must be defined and executed. A path may, however, be stopped and resumed by issuing the Immediate Stop (`!S`) and Continue (`!C`) commands while the path is executing. In this case, motion will be decelerated and resumed along the path without loss of position. If one axis is stopped due to any other reason, the other axis will stop abruptly, and motion may not be resumed. Causes for motion being stopped may include encountering an end of travel limit, issuing a Kill (`!K`) command, or detecting a stall.

Path Definition

Path definitions are created using the `DEF` command. The 6201 compiles a completed path definition (`PCOMP`) and stores the compiled path data in a form that can be used for subsequent execution. The compiled paths are stored in the memory space allocated by the `MEMORY` command.

The default memory allocation for a standard 6201 is `MEMORY21ØØ,18ØØ`, which allocates 21,400 bytes for user programs and 18,600 bytes for contouring path segments. The default allocation for the 6201-M is `MEMORY75ØØ,74ØØ`. The maximum allocation for contouring paths is 39,000 bytes for the standard 6201 and 149,000 bytes for the 6201-M.

CAUTION

Issuing a memory allocation command (e.g., `MEMORY1ØØØ,39ØØØ`) will erase all existing programs and compiled contouring path segments. However, issuing the `MEMORY` command by itself (i.e., `MEMORY`—to request the status of how the memory is allocated) will not affect existing programs or path segments.

Up to 75 individual paths may be defined and compiled, (up to 300 paths for 6201-M users) as long as each path has at least one segment, and the sum of all the segments of all the paths does not exceed the 6201's memory limitation. All 75 (or 300) path definitions may be compiled and ready to execute at any time. Paths defined using 6201 commands are specified with a path name. Once a path definition is compiled, it may be executed repeatedly without being re-compiled.

Deleting (`DEL`) an existing path name will automatically delete the existing path compilation with that name. The `PUCOMP` command only deletes the path compilation, not the path program.

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

In the following example, storage space is made available for the definition of path WID3 by first deleting the compiled version of paths WID1 and WID2. The DEF statement begins the definition of the path WID3.

Example

Command	Description
> PUCOMP WID1	Remove compilation of WID1
> PUCOMP WID2	Remove compilation of WID2
> DEF WID3	Begin definition of WID3

Participating Axes

You can change the X-Y plane to an Y-X plane by using the PAXES command.

A path definition default is PAXES1,2. For any path that uses axes 2 as the X axis and axis 1 as the Y axis, the path definition must start with PAXES2,1.

NOTE

The mechanical resolution of both axes must be identical; scaling cannot compensate for mechanical variances. In addition, both axes (identified with PAXES command) must have the same PULSE and DRES settings. If you change the PULSE setting, you will need to recompile (PCOMP) any previously compiled paths.

Example

Command	Description
> DEF DRAW1	Begin definition of DRAW1
- PAXES2,1	Set contouring axes

Path Acceleration, Deceleration, and Velocity

A path may be composed of many segments, each with their own motion parameters. The path velocity, acceleration, and deceleration specifications currently in effect at the time a segment is defined will apply to that segment. This allows construction of a path that moves at one velocity for a section of the path, then moves at a different velocity for another section.

In most cases, it will be desirable to maintain a constant velocity throughout the path, but it is easy to define a path in which each segment has its own velocity. For example, this may be useful when a tool needs to slow down to round a corner, or to allow the rate of glue application to be controlled by the path speed. Acceleration and deceleration may also be specified separately. The short example below illustrates the specification of velocity, acceleration, and deceleration in that order.

Example

Command	Description
> SCALE1	Enable scaling
> PSCLA25ØØØ	Scale path acceleration by 25,000
> PSCLV2ØØØØ	Scale path velocity by 20,000
> PVØ.5	Path velocity 10,000 steps/sec
> PA16	Path acceleration 400,000 steps/sec^2
> PAD28	Path deceleration 700,000 steps/sec^2

Segment End-point Coordinates

All end-point position specifications are in units of motors steps, regardless of the current state of the ENC command. The end-point position specifications of lines and arcs may be either absolute or incremental. The 6201 stores the end-point data for all of its compiled segments internally as incremental, relative to the start of the segment. But in order to ease the programming task, absolute coordinates and multiple coordinate systems may be used.

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

When incremental coordinates are used to specify an end-point, the X and Y end-point values represent the distances from the X and Y start point of the segment being specified. Center specifications of an arc are always incremental (i.e., relative to the start of that arc segment). When absolute coordinates are used to specify an end-point, the X and Y end-point values represent that segment's position in the specified coordinate system. Incremental and absolute programming are specified with the `PAB` command. Incremental programming is the default state at the beginning of a path definition.

Coordinate systems allow the assignment of an arbitrary X-Y position as a reference position for subsequent absolute end-point specifications. The 6201 allows the use of two coordinate systems for use with absolute coordinate programming. These are called the *Work* coordinate system and the *Local* coordinate system. These are specified with the `PWC` and `PLC` commands. Neither coordinate system needs to represent the actual absolute position of the axes when the path actually executes.

The Work and Local coordinate systems are provided to allow absolute end-point definition of a segment without needing to know the actual position of each axis when the segment is executed. If no `PWC` command precedes the first segment command when a path definition begins, the 6201 will place the start of the first segment at location (0,0) in the Work coordinate system. By using the `PWC xpos,ypos` command, the programmer defines subsequent absolute end-points to refer to the Work coordinate system, and also locates that coordinate system such that the starting position of the next segment is at (xpos, ypos) of the Work coordinate system.

The Local coordinate system is provided so that if a section of a path is to appear in multiple locations along the path, the segments that compose that section can be programmed in absolute coordinates. By using the `PLC xpos,ypos` command, the programmer defines subsequent absolute end-points to refer to the Local coordinate system, and also locates that coordinate system such that the starting position of the next segment is at (xpos, ypos) of the Local coordinate system.

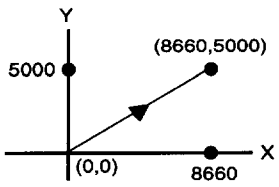
A single path definition may include both absolute and incremental programming, and be required to switch between Work and Local coordinates several times. At any point along a path definition, coordinates may be switched from absolute to incremental, or from incremental to absolute. When switching to absolute, all subsequent end-point specifications are assumed to be absolute with respect to the coordinate system in effect at that time. This remains true until the reference system is switched to incremental, or to a new absolute reference.

When switching from Work coordinates to Local coordinates, the Local X and Y start positions of the following segment must be specified with the `PLC` command. When starting a path definition with Work coordinates, or when switching to Work coordinates, the starting position of the next segment may either be specified or assumed. The 6201 toggles between the Work coordinate system and the Local coordinate system with the `PL` command.

Ease of programming results from the ability to switch between absolute and incremental, and to re-define the coordinate systems between sections of a path. This allows individual sections of path definition to have Local coordinate systems, yet still be integrated into the complete path.

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

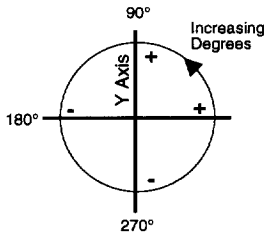
Line Segments



Lines are the simpler of the two path segment types. The placement, length, and orientation of the line is completely specified by the end-point of the line segment and the end-point of the previous segment. As described above, end-points can be specified with absolute or incremental coordinates. The example below is specified with incremental coordinates and results in a line segment 10,000 steps in length, at 30 degrees in the X-Y plane.

Command	Description
> PLIN866Ø,5ØØØ	Line segment to (8660,5000) — see illustration at left

Arc Segments

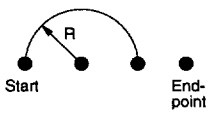


Arcs are more complex to specify than lines, because there are four possible ways to get from the start point to the end point. The radius of an arc may either be specified directly or implied by the center specification. In the 6201, all path descriptions refer to the X-Y plane. The general convention describing the X-Y plane, as viewed from a drawing, is as follows. The X axis is shown as the left-right axis, with left being negative and right being positive. The Y axis is the up-down axis with down being negative and up being positive. Angles start at zero and increase in the CCW direction of rotation. A line segment, or the radius of an arc is at zero degrees if the incremental end-point has a positive X component and zero Y component. The angle is 90 degrees if the end-point has a positive Y component and zero X component.

Radius Tolerance Specifications

All arcs have an associated radius. In the 6201, the radius may either be specified explicitly, or implied by a center specification. In both cases, it is possible that the radius may not be consistent with the specified end-point of the arc. This could be a result of improper specification, user calculation error, or of round-off error in the internal arithmetic of the 6201. For this reason, the 6201 allows the specification of a radius tolerance (PRTOL). The radius tolerance is specified in the same units as the radius and X and Y data. The radius tolerance has a factory default of ± one step, which is just enough to overcome round-off errors. The radius tolerance may be specified at any point along the path definition, and may be changed between one arc and the next. Each arc definition will be compared to the most recently specified radius tolerance. The radius tolerance should be about the same as the dimension tolerances of the finished product. The following paragraphs explain how the radius tolerance is used for the two types of arc specifications, and gives syntax examples for the radius tolerance specification.

Radius Specified Arcs

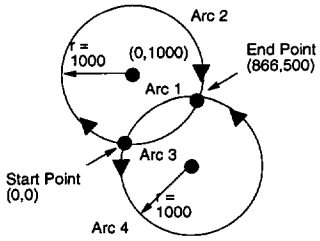


Specification of an arc using the radius method requires knowledge of the start point, the end point, and the sign and magnitude of the radius. The 6201 knows the start point to be either the start of the path, or the end of the previous segment. The end point and radius are provided by the user's program. It is possible to specify an impossible arc by specifying an end point that is more than twice the radius away from the start point (see drawing at left). In this case, the 6201 will automatically extend the radius to reach the end-point, provided that the automatic radius change does not exceed the user specified radius tolerance. If the required radius extension exceeds the radius tolerance, the 6201 will respond with an execution error, and no arc will be generated.

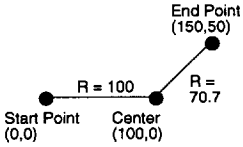
The following illustration shows the four possible ways to move from the start point to the end point using an arc of radius 1000. Arc 1 and 2 both travel in the CW direction, arc 3 and 4 both travel in the CCW direction. Arc 1 and 3 are both less than 180 degrees. An arc of 180 degrees or less is specified with a positive radius. Arc 2 and 4 are both greater than 180 degrees. An arc of more than 180 degrees is specified with a negative radius. The example below shows the radius tolerance specification and the specifications of arcs 1, 2, 3, and 4 respectively.

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

Example	**Command**	**Description**
	> SCALE1	Enable scaling
	- PSCLD1ØØ	Path position scaler
	> DEF arcs	Begin definition of path arcs
	- PRTOL5	5 steps of radius tolerance
	- PARCP866,5ØØ,1ØØØ	Arc 1, CW < 180 degrees
	- PARCP866,5ØØ,-1ØØØ	Arc 2, CW > 180 degrees
	- PARCM866,5ØØ,1ØØØ	Arc 3, CCW < 180 degrees
	- PARCM866,5ØØ,-1ØØØ	Arc 4, CCW > 180 degrees
	- END	End path definition
	> PCOMP arcs	Compile path arcs
	> PRUN arcs	Initiate path arcs (see drawing at right)



Center Specified Arcs



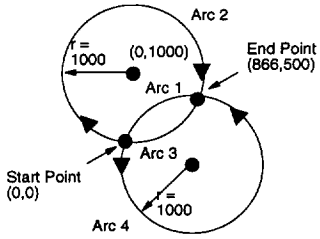
Specification of an arc using the center method requires knowledge of the start point, the end point, and the center point of the arc. The X coordinate of the center is referred to with the letter I, and the Y coordinate of the center is referred to with the letter J. When an arc is specified with the center, another potential problem arises.

It is possible to specify the center of an arc such that the radius implied by the start point does not equal the radius implied by the end point (see illustration on left). In this case, the 6201 will re-locate the center so that the resulting arc has a uniform radius and the starting and ending angles come as close as possible to those implied by the user's center specification. This automatic center relocation will take place only if the start point and end point radius difference does not exceed the user specified radius tolerance. If the radius tolerance is exceeded, an execute error will result, and the arc will not be included in the path.

While automatic center relocation will ensure a continuous path, it may result in an abrupt change in path direction. This happens because a new location for the center results in a new tangent direction for an arc about that center.

The example below shows the specifications of arcs 1, 2, 3, and 4 for the drawing on the right. In the 6201 commands, the order of the data is X, Y, I, J from left to left.

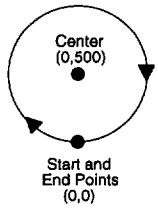
Example	**Command**	**Description**
	> SCALE1	Enable scaling
	- PSCLD1ØØ	Path position scaler
	> DEF arcs2	Begin definition of path arcs2
	- PARCOP866,5ØØ,866,-5ØØ	Arc 1, CW < 180 degrees
	- PARCOP866,5ØØ,Ø,1ØØØ	Arc 2, CW > 180 degrees
	- PARCOM866,5ØØ,Ø,1ØØØ	Arc 3, CCW < 180 degrees
	- PARCOM866,5ØØ,866,-5ØØ	Arc 4, CCW > 180 degrees
	- END	End path definition
	> PCOMP arcs2	Compile path arcs2
	> PRUN arcs2	Initiate path arcs2 (see drawing at right)



Circles

A circle is a special case of an arc whose end-point is the same as the starting point. Because these two points are the same, it is impossible to determine the location of the circle's center from a radius specification. For this reason, an arc that is a complete circle must be specified using the arc center specification method. An arc with identical starting and ending points specified with the radius method will be ignored. The circle shown below is specified with the example below.

Example	**Command**	**Description**
	> DEF circle	Begin definition of path circle
	- PARCOPØ,Ø,Ø,5ØØ	Circle with center at (0,500)
	- END	End path definition
	> PCOMP circle	Compile path circle
	> PRUN circle	Initiate path circle



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

Segment Boundary

So far, all the examples given have shown isolated line or arc segments. Most paths will consist of many segments put together. The point at which the segments are connected is called a *segment boundary* in this text. The 6201 automatically ensures that the path is continuous, in that segments are placed end-to-end.

The path velocity may either be constant or change from segment to segment, according to user specification. Velocity changes use the specified acceleration and deceleration and may take place even across segment boundaries.

The programmer should ensure that direction of travel is also continuous across segment boundaries (see Figure A). If the direction change is abrupt (as shown in Figure B) the X and Y axes will suffer abrupt acceleration or deceleration. The 6201 ensures that there will be no abrupt direction change within a segment, but the programmer is responsible for ensuring that the direction is continuous across segment boundaries. At low speeds, some motor and mechanical configurations will tolerate such abrupt changes, and the 6201 will accept such a program; however, it is generally good practice to design paths with smooth direction changes. This may be done by designing a path using arcs to round corners.

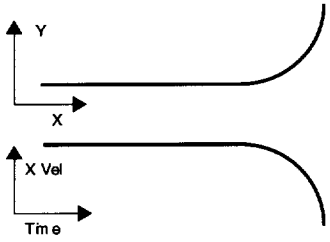


Figure A (Segment Example)

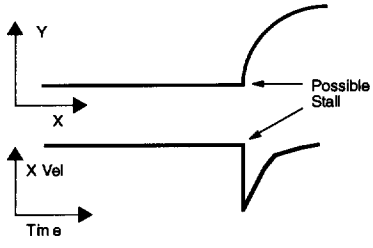


Figure B (Stall Example)

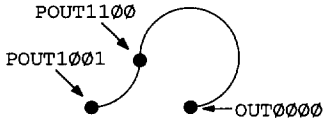
Outputs Along the Path

For each segment, the user may also specify an output pattern (POUT), which is to be applied to the programmable outputs at the beginning of that segment and remains throughout that segment. These segment defined programmable output patterns are stored as part of the compiled path definition. These outputs will change state at some time in the range of 1.5 ms before the beginning of the segment to 0.5 ms after the beginning of the segment. The programmable outputs may not be controlled more precisely than this, because the 6201 updates its record of path position every 2 ms. These are the most precise outputs that the 6201 can provide.

The path segment defined programmable outputs are provided so that plotting applications may raise and lower the pen, laser cutters may turn the laser on and off, glue applicators may be turned on and off, all at prescribed positions along the path. The output specification is stated before the segment definition, which holds that output state. In the example below, programmable outputs 2 and 4 are changed during the path segments.

Example

Command	**Description**
> DEF prog1	Begin definition of path program prog1
- POUT1ØØ1	Output pattern during first arc
- PARCM5,5,5	Specify incremental X-Y endpoint position and radius arc <180° for 1/4 circle CCW arc
- POUT11ØØ	Output pattern during second arc
- PARCM5,-5,-5	Specify incremental X-Y endpoint position and radius arc >180° for 3/4 circle CW arc
- END	End definition of path program prog1
> PCOMP prog1	Compile path program prog1
> PRUN prog1	Execute path program prog1
> OUTØØØØ	Turn off programmable outputs 1-4



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

Paths Built Using 6000 Commands

When using the 6000 Series commands to define a given path, the commands that specify all of the path definitions must be contained in a named block defining that path. Each path definition block has a unique name that is used to distinguish one path from another. Because the path definition is stored as a program, many different paths may be stored, each defined with a unique name. A path definition block begins with a DEF command which contains its name, and ends with an END command.

The 6201 offers a command to compile (PCOMP) a named path definition block, and a separate command to execute (PRUN) a named path. Once a named path is compiled, it may be executed repeatedly without delay.

Compiling the Path

A PCOMP command will cause the 6201 to find the named path definition block and compile the path described by those commands, even if that pathname had been previously compiled. The use of variables (VAR) as parameters in path definition statements allow the same basic path to be re-defined with slightly different sizes and shapes. They may also be used to conditionally include or omit sections of the path.

The 6201 was designed to allow compile-time determination of path parameters. There may be cases when the 6201 should prompt the operator or host computer for the value to be used for path velocity or segment end-points. Alternatively, these values may be read with the READ or DAT commands, allowing multiple calls of a single subroutine to define similar path sections with different data values. Commands that retrieve this data would be placed within the path definition, and would only prompt for the information when the path is compiled—an example is PV(READ1).

Example

Command	Description
`> VARS1="PATH VELOCITY ? "`	Create message string
`> DEF path1`	Begin definition of path1
`- PAXES1,2`	Set path X & Y axes
`- PAB1`	Absolute path mode
`- PA1ØØ`	Path acceleration
`- PV(READ1)`	Path velocity, to be read in when compiling
`- PLIN25ØØØ,25ØØØ`	Move in a line
`- PLIN(VAR2),(VAR3)`	Move in a line, to be read in when compiling
`- END`	End definition of path1
`> PCOMP path1`	Compile path1

Executing the Path

A PRUN command will cause the 6201 to find the named path definition block and execute the path described by those commands, if that pathname has already been compiled (PCOMP).

The use of variables as parameters in the path definition statement is a method of allowing segment parameters to take new values each time the path is compiled. When the path is executing, the values of the variables do not affect the path parameters. If a change in a variable value is intended to affect the path parameters, that path must be re-compiled. The PRUN command performs the equivalent of a GOSUB to the named path definition block.

Possible Programming Errors

It is possible to create a situation in which the segment statements are interrupted. This could occur if an enabled ON condition becomes true. If an enabled ON condition (ONCOND) becomes true while running a compiled path, the branch to the ONP program will result. Motion from the path that was being executed will continue at the last segment velocity until it is stopped. Within the ONP program, a Stop command should be issued for all axes to stop the path from executing.

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

Programming Examples

Figure A and Figure B show two simple paths that illustrate most of the 6201 segment types. For both figures, axis 1 is X and axis 2 is Y.

Figure A specifies the end-points with absolute coordinates. The default Work coordinate system with start point of (0,0) is used, so no PLØ statement is needed.

Figure B specifies the end-points with incremental coordinates. The state of the programmable outputs needs to be different for Handles than for Knobs. No other 6201 actions take place during these paths.

Example	**Command**	**Description**
	> SCALE1	Enable scaling
	> PSCLA25ØØØ	Path acceleration scaler
	> PSCLV25ØØØ	Path velocity scaler
	> PSCLD25ØØØ	Path position scaler
	> DEF HANDLE	Begin HANDLE path definition
	- PAXES1,2	Set X axis as axis 1, Y axis as axis 2
	- PAB1	Use absolute coordinates
	- POUT11ØØ	Programmable pattern for next segments
	- PARCOM1Ø,1Ø,Ø,1Ø	CCW quarter circle
	- PLIN1Ø,2Ø	Vertical LINE segment
	- PARCP2Ø,1Ø,-1Ø	CW 3/4 circle
	- PARCM2Ø,Ø,5	CCW half circle
	- END	End of HANDLE path definition
	> DEF KNOB	Begin KNOB path definition
	- PAXES1,2	Set X axis to be axis 1, Y axis to be axis 2
	- PABØ	Use incremental coordinates
	- POUTØØ11	Programmable pattern for next segments
	- PLIN3Ø,Ø	Long LINE into circular knob
	- PARCOMØ,Ø,Ø,1Ø	CCW circle for the knob
	- PLIN1Ø,Ø	Short LINE out of knob
	- END	End of KNOB path definition

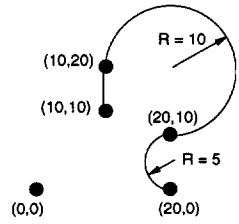


Figure A (HANDLE Example)

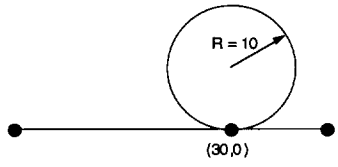
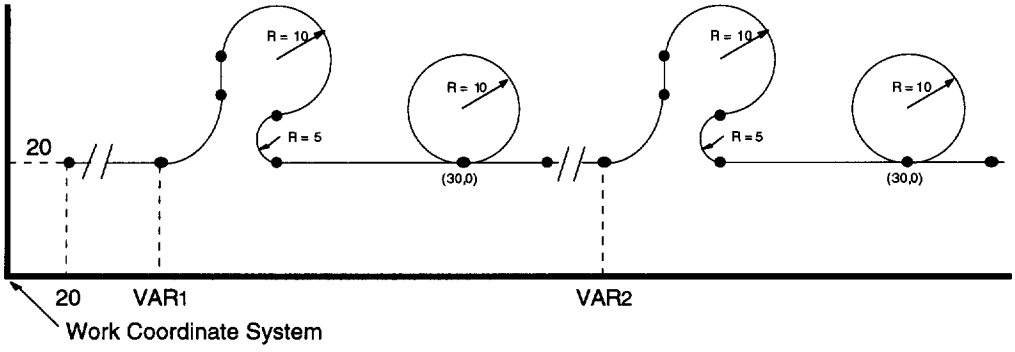


Figure B (KNOB Example)

Our third path consists of two pairs of the first two. Each pair is placed at variable locations within the Work coordinate system and the two pairs are connected with a Line segment. The Line leading into the first pair starts at (20,20) in the Work coordinate system. The first pair starts at (VAR1,2Ø) and the second pair starts at (VAR2,2Ø) in the Work coordinate system. HANDLE is defined using the Local coordinate system. Even though HANDLE is defined in absolute coordinates and appears in two different places along the path in Parts, the statements describing it appear only once, in a path definition using local coordinates.



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

Example	**Command**	**Description**
	> SCALE1	Enable scaling
	> PSCLA1ØØØØ	Path acceleration scaler
	> PSCLV1ØØØØ	Path velocity scaler
	> PSCLD1ØØØØ	Path position scaler
	> DEF PARTS	Begin PARTS path definition
	- PAXES1,2	Set X axis to be axis 1, Y axis to be axis 2
	- PAB1	Use absolute coordinates
	- PWC2Ø,2Ø	Establish WORK coordinates
	- PLØ	Enable WORK coordinates
	- PLIN(VAR1),2Ø	LINE to (VAR1,20) — Q1
	- PLCØ,Ø	Specify LOCAL coordinate system
	- PL1	Enable LOCAL coordinate system
	- PAB1	Use absolute coordinates
	- POUT11ØØ	Programmable pattern for next segments
	- PARCOM1Ø,1Ø,Ø,1Ø	CCW quarter circle
	- PLIN1Ø,2Ø	Vertical LINE segment
	- PARCP2Ø,1Ø,-1Ø	CW 3/4 circle
	- PARCM2Ø,Ø,5	CCW half circle
	- PABØ	Use incremental coordinates
	- POUTØØ11	Programmable pattern for next segments
	- PLIN3Ø,Ø	Long LINE into circular knob
	- PARCOMØ,Ø,Ø,1Ø	CCW circle for the knob
	- PLIN1Ø,Ø	Short LINE out of knob
	- PAB1	Use absolute coordinates
	- PLØ	Return to WORK coordinates
	- PLIN(VAR2),2Ø	LINE to (VAR2,20) — Q2
	- PLCØ,Ø	Specify LOCAL coordinate system
	- PL1	Enable LOCAL coordinate system
	- PAB1	Use absolute coordinates
	- POUT11ØØ	Programmable pattern for next segments
	- PARCOM1Ø,1Ø,Ø,1Ø	CCW quarter circle
	- PLIN1Ø,2Ø	Vertical LINE segment
	- PARCP2Ø,1Ø,-1Ø	CW 3/4 circle
	- PARCM2Ø,Ø,5	CCW half circle
	- PABØ	Use incremental coordinates
	- POUTØØ11	Programmable pattern for next segments
	- PLIN3Ø,Ø	Long LINE into circular knob
	- PARCOMØ,Ø,Ø,10	CCW circle for the knob
	- PLIN1Ø,Ø	Short LINE out of knob
	- END	End of PARTS path definition
	> PCOMP PARTS	Compile PARTS path definition

Teach Mode

The Teach Mode is simply a method of storing (*teaching*) variable data and later using the stored data as a source for motion program parameters. The variable data can be any value that can be stored in a numeric (VAR) variable (e.g., position, acceleration, velocity, etc). The variable data is stored into a *data program*, which is an array of *data elements* that have a specific address from which to write and read the variable data. Data programs do not contain 6000 Series commands.

The information below describes the principles of using the data program in a teach mode application. Following that is a teach mode application example in which the joystick is used to teach position data to be used in a motion program.

Teach Mode Basics

The basic process of using a data program for teach mode applications is as follows:

1. Initialize a data program.
2. Teach (store/write) variable data into the data program.
3. Read the data elements from the data program into a motion program.

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

Initialize a Data Program

This is accomplished with the DATSIZ command. The DATSIZ command syntax is DATSIZi<,i>. The first integer (i) represents the number of the data program (1 - 50). You can create up to 50 separate data programs. The data program is automatically given a specific program name (DATPi). The second integer represents the total number of data elements (up to 6,500) you want in the data program. Upon issuing the DATSIZ command, the data program is created with all the data elements initialized with a value of zero.

The data program has a tabular structure, where the data elements are stored 4 to a line. Each line of data elements is called a *data statement*. Each element is numbered in sequential order from left to right (1 - 4) and top to bottom (1 - 4, 5 - 8, 9 - 12, etc.). You can use the TPROG DATPi command ("i" represents the number of the data program) to display all the data elements of the data program.

For example, if you issue the DATSIZ1,13 command, data program #1 (called DATP1) is created with 13 data elements initialized to zero. The response to the TPROG DATP1 command is depicted below. Each line (*data statement*) begins with DATA=, and each data element is separated with a comma.

```
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

Each data statement, comprising four data elements, uses 39 bytes of memory. The memory for each data statement is subtracted from the memory allocated for user programs (see MEMORY command).

Teach the Data to the Data Program

The data that you wish to write to the data elements in the data program must first be placed into numeric variables (VAR). Once the data is stored into numeric variables, the data elements in the data program can be edited by using the Data Pointer (DATPTR) command to move the data pointer to that element, and then using the Data Teach (DATTCH) command to write the datum from the numeric variable into the element.

When the DATSIZ command is issued, the internal data pointer is automatically positioned to data element #1. Using the default settings for the DATPTR command, the numeric variable data is written to the data elements in sequential order, incrementing one by one. When the last data element in the data program is written, the data pointer is automatically set to data element #1 and a warning message (*WARNING: POINTER HAS WRAPPED AROUND TO DATA POINT 1) is displayed. The warning message does not interrupt program execution.

The DATPTR command syntax is DATPTRi,i,i. The first integer (i) represents the data program number (1 through 50). The second integer represents the number of the data element to point to (1 through 6500). The third integer represents the number of data elements by which the pointer will increment after writing each data element from the DATTCH command, or after recalling a data element with the DAT command.

The DATTCH command syntax is DATTCHi<,i,i,i>. Each integer (i) represents the number of a numeric variable. The value of the numeric variable will be stored into the data element(s) of the currently active data program (i.e., the program last specified with the last DATSIZ or DATPTR command). As indicated by the number of integers in the syntax, the maximum number of variable values that can be stored in the data program per DATTCH command is 4. Each successive value from the DATTCH command is stored to the data program according to the pattern established by the third integer of the DATPTR command.

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

As an example, suppose data program #1 is configured to hold 13 data elements (DATSIZ1,13), the data pointer is configured to start at data element #1 and increment 1 data element after every value stored from the DATTCH command (DATPTR1,1,1), and the values of numeric variables #1 through #3 are already assigned (VAR1=2, VAR2=4, VAR3=8). If you then enter the DATTCH1,2,3 command, the values of VAR1 through VAR3 will be assigned respectively to the first three data elements in the data program, leaving the pointer pointing to data element #4. The response to the TPROG DATP1 command would be as follows (the text is highlighted to illustrate the final location of the data pointer after the DATTCH1,2,3 command is executed):

```
*DATA=2.Ø,4.Ø,8.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

If you had set the DATPTR command to increment 2 data elements after every value from the DATTCH command (DATPTR1,1,2), the data program would be filled differently and the data pointer would end up pointing to data element #7:

```
*DATA=2.Ø,Ø.Ø,4.Ø,Ø.Ø
*DATA=8.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

Recall the Data from the Data Program

After storing (*teaching*) your variables to the data program, you can use the DATPTR command to point to the data elements and the DATi ("i" = data program number) data assignment command to read the stored variables to your motion program. *You cannot recall more than one data element at a time; therefore, if you want to recall the data in a one-by-one sequence, the third integer of the DATPTR command must be a 1 (this is the default setting).*

Summary of Related 6000 Series Commands

NOTE: A detailed description of each command is provided in the **6000 Series Software Reference Guide**.

DATSIZ.....Establishes the number of data elements a specific data program is to contain. A new DATPi program name is automatically generated according to the number of the data program (i = 1 through 50). The memory required for the data program is subtracted from the memory allocated for user programs (see MEMORY command).

DATPTR.....Moves the data pointer to a specific data element in any data program. This command also establishes the number of data elements by which the pointer increments after writing each data element from the DATTCH command and after recalling each data element with the DAT command.

DATTCH.....Stores the variable data into the data program specified with the last DATSIZ or DATPTR command. After the data is stored, the data pointer is incremented the number of times entered in the third integer of the DATPTR command. *The data must first be assigned to a numeric variable before it can be taught to the data program.*

TDPTR.....Responds with a 3-integer status report (i,i,i): First integer is the number of the active data program (the program # specified with the last DATSIZ or DATPTR command); Second integer is the location number of the data element to which the data pointer is currently pointing; Third integer is the increment set with the last DATPTR command.

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

[DPTR] ....From the currently active data program, uses the number of the data pointer's location in a numeric variable assignment operation or a conditional statement operation.

[DATPi] ..The name of the data program created after issuing the DATSIZ command. The integer (i) represents the number of the data program. Data programs can be deleted just like any other user program (e.g., DEL DATP1).

[DATi] ....From the data program specified with i, assigns the numeric value of the data element (currently pointed to by the data pointer) to a specified variable parameter in a 6000 series command (e.g., D(DAT3),(DAT3)).

Teach Mode Application Example

In this example, 2 axes of the 6201 are used to move a 2-axis stage. This example illustrates a common method of teaching a path by using the joystick to move the load into position, teach the position (triggered by the Joystick Release input), then move to the next position. Five positions will be taught from each axis (2 axes at one trigger), for a total of 10 data elements in the data program. After all 10 positions are taught to the data program, the 6201 will automatically move both axes to a home position, move to each position that was taught, and then return to the home position.

For the sake of brevity, this example is limited to teaching 10 position data points; however, in a typical application, many more points would be taught. Also, it is assumed that end-of-travel and home limits are wired and a homing move has been programmed.

What follows is a suggested method of programming the 6201 for this application. To accomplish the teach mode application, a program called MAIN is created, comprising three subroutines: SETUP (to set up for teach mode), TEACH (to teach the positions), and DOPATH (to implement a motion program based on the positions taught).

The joystick operation in this example is based on setting the Joystick Axes Select input (pin #15 on the Joystick connector) to high to select analog input channels #1 and #2 (pins #1 and #2) for joystick use, and using the Joystick Release input (pin #17) to trigger the position teach operation.

Step 1 **Initialize a Data Program.**

> DEL DATP1	Delete data program #1 (DATP1) in preparation for creating a new data program #1
> DATSIZ1,1Ø	Create data program #1 (named DATP1) with an allocation of 10 data elements. Each element is initialized to zero.

Step 2 **Define the SETUP Subroutine.** Note that the SETUP subroutine need only run once.

> DEF SETUP	Begin definition of the subroutine called SETUP
- JOYVH3,3	Set the high velocity speed to 3 rps
- JOYVL.2,.2	Set the low velocity to 0.2 rps
- JOYAXH1,2	When axes select input is set high, apply analog input 1 to axis 1 and apply analog input 2 to axis 2
- VAR1=Ø	Set variable #1 equal to zero
- VAR2=Ø	Set variable #2 equal to zero
- DRIVE11	Enable the drives for both axes
- MA11	Enable the absolute positioning mode for both axes
- END	End definition of the subroutine called SETUP

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

Step 3 **Define the TEACH Subroutine.**

> DEF TEACH	Begin definition of the subroutine called TEACH
- HOM11	Home both axes (absolute position counter is set to zero after homing move)
- DATPTR1,1,1	Select data program #1 (DATP1) as the current active data program, and move the data pointer to the first data element. After each DATTCH value is stored to DATP1, increment the data pointer by 1 data element.
- REPEAT	Set up a repeat/until loop
- JOY11	Enable joystick mode on both axes. At this point, you can start moving the axes into position with the joystick. While using the joystick, command processing is stopped here until you activate the joystick release input. Activating the joystick release input disables the joystick mode and allows the subsequent commands to be executed (assign the motor positions to the variables and then store the positions in the data program).
- VAR1=1PM	Set variable #1 equal to the position of motor 1
- VAR2=2PM	Set variable #2 equal to the position of motor 2
- DATTCH1,2	Store variable #1 and variable #2 into consecutive data elements. (The first time through the repeat/until loop, variable #1 is stored into data element #1 and variable #2 is stored into data element #2. The data pointer is automatically incremented once after each data element and ends up pointing to the third data element in anticipation of the next DATTCH command.)
- WAIT(INO.5=b1)	Wait for the joystick release input to be de-activated
- UNTIL(DPTR=1)	Repeat the loop until the data pointer wraps around to data element #1 (data program full)
- END	End definition of the subroutine called TEACH

Step 4 **Define the DOPATH Subroutine.**

> DEF DOPATH	Begin definition of the subroutine called DOPATH
- HOM11	Move both axes to the home position (absolute counters set to zero)
- A5Ø,5Ø	Set up the acceleration
- V3,3	Set up the velocity
- DATPTR1,1,1	Select data program #1 (DATP1) as the current active data program, and set the data pointer to the first data element. Increment the data pointer one element after every data assignment with the DAT command. *If you wanted to move only axis 1 down the taught path, you would set the increment (third integer) to a 2, thus accessing only the axis 1 stored positions.*
- REPEAT	Set up a repeat/until loop
- D(DAT1),(DAT1)	The position of axis 1 and axis 2 are recalled into the distance command
- GO11	Move to the position
- T.5	Wait for 0.5 seconds
- UNTIL(DPTR=1)	Repeat the loop until the data pointer wraps around to data element #1 (all data elements have been read)
- HOM11	Move both axes back to the home position
- END	End definition of the subroutine called DOPATH

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

Step 5 **Define the `MAIN` Program (Include `SETUP`, `TEACH`, and `DOPATH`).**

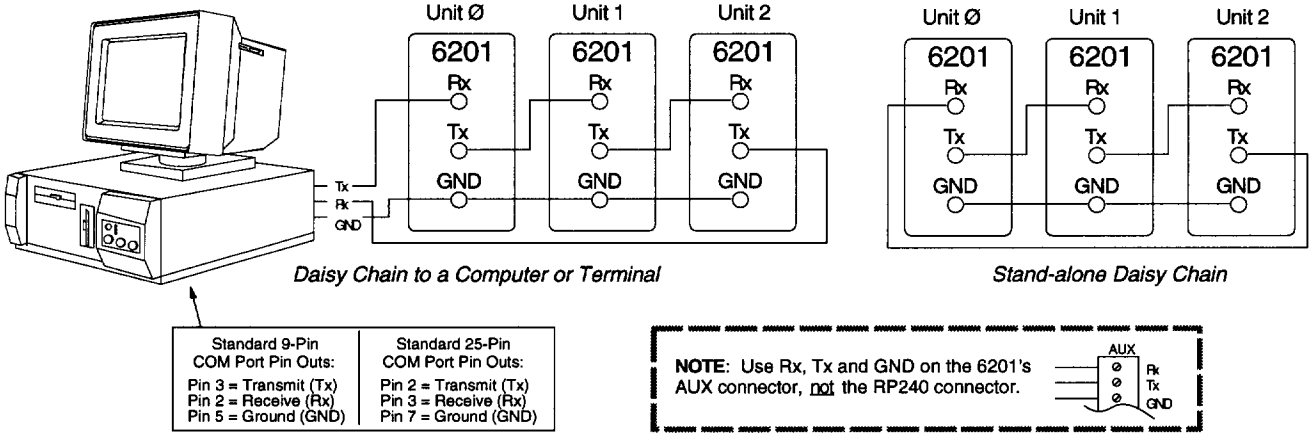
```
> DEF MAIN        Begin definition of the program called MAIN
- SETUP           Execute the subroutine called SETUP
- TEACH           Execute the subroutine called TEACH
- DOPATH          Execute the subroutine called DOPATH
- END             End definition of the program called MAIN
```

Step 6 **Run the `MAIN` Program and Teach the Positions with the Joystick.**

1. Enter the `MAIN` command to execute the teach mode program and set the joystick's *axis select* input to high.
2. Use the joystick to move to the position to be taught.
3. Once in position, activate the *joystick release* input to teach the positions. Two positions (one for each axis) are taught each time you activate the joystick release input.
4. Repeat steps 2 and 3 for the remaining four teach locations. After triggering the joystick release input the fifth time, the 6201 will home the axes, repeat the path that was taught, and then return both axes to the home position.

RS-232C Daisy-Chaining

Up to ninety-nine 6201s may be daisy-chained. There are two methods of daisy-chaining: one uses a computer or terminal as the controller in the chain; the other uses a 6201 as the master controller. The figure below illustrates examples of both daisy-chain types for three 6201 Two-Axis Motion Controllers. *Be sure to use the* **Rx**, **Tx** *and* **GND** *on the* **AUX** *connector, not the* **RP240** *connector.*



Follow these steps to implement daisy-chaining:

Step 1 To enable and disable communications on a particular 6201 unit in the chain, you must establish a unique device address using the unit's address DIP switches or the Daisy-chain Address (`ADDR`) command.

DIP switches: Instructions for accessing and changing these DIP switch settings are provided in the *Optional DIP Switch Settings* section in Chapter 5. Device addresses set with the DIP switches range from 0 to 7.

`ADDR` command: The `ADDR` command automatically configures unit addresses for daisy chaining by disregarding the DIP switch setting. This command allows up to 99 units on a daisy chain to be uniquely addressed.

Sending `ADDRi` to the first unit in the daisy chain sets its address to be `(i)`. The first unit in turn transmits `ADDR(i + 1)` to the next unit to

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

set its address to (i + 1). This continues down the daisy chain until the last unit of (n) daisy-chained units has its address to (i + n).

Setting ADDR to Ø re-enables the unit's daisy-chain address configured on its internal DIP switch.

Note that a 6201 with the default device address of zero (0) will send the initial power-up start messages. Example:

```
*PARKER COMPUMOTOR 62Ø1 MOTION CONTROLLER
*NO REMOTE PANEL
*CONTOURING OPTION INSTALLED
*ADVANCED FOLLOWING FEATURES NOT INSTALLED
```

Step 2 Connect the daisy-chain with a terminal as the master (as shown in the diagram above).

It is necessary to have the error level set to 1 for all units on the daisy-chain (ERRLVL1). When the error level is not set to 1, the 6201 sends ERROK or ERRBAD prompts after each command, which makes daisy-chaining impossible. Send the ERRLVL1 command from the master terminal as many times as there are units on the chain:

Command	Description
ERRLVL1	Set error level to 1

After this has been accomplished a carriage return sent from the terminal will not cause any 6201 to send a prompt. Verify this. Instructions below show how to set the error level to 1 automatically on power-up by using the 6201's power-up start program (highly recommended).

After the error level for all units has been set to ERRLVL1, to send a 6000 series command to all units on the daisy-chain, simply enter that command from the master terminal.

Command	Description
OUT1111	Turn on outputs #1 - #4 on all units
A5Ø,5Ø	Set acceleration to 50 rps^2 for all axes (all units, both axes)

To send a 6000 series command to one particular unit on the daisy-chain, prefix the command with the appropriate unit's device address and a underline:

Command	Description
2_OUT1	Turn on output #1 on unit #2
4_OUTØ	Turn off output #1 on unit #4

To receive data from a 6201, you **must** prefix the command with the appropriate unit's device address and a underline:

Command	Description
1_A	Request acceleration information from unit #1
*A5Ø,5Ø	Response from unit #1

Use the (E) command to enable/disable RS-232C communications for an individual unit. If all 6201 units on the daisy chain are enabled, commands without a device address identifier will be executed by all units. Because of the daisy-chain's serial nature, the commands will be executed approximately 1 ms per character later on each successive unit in the chain (assuming 9600 baud).

Command	Description
3_EØ	Disable RS-232C on unit #3
VAR1=1	Set variable #1 to 1 on all other units
3_E1	Enable RS-232C on unit #3
3_VAR1=5	Set variable #1 to 5 on unit #3

Verify communication to all units by using the techniques described above.

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

Step 3 Now that communication is established programming of the units can begin (alternately, units can be programmed individually by connecting the master terminal to one unit at a time). To allow daisy-chaining between multiple 6201s, the ERRLVL1 command must be used to prevent units from sending error messages and command prompts. In every daisy-chained unit the ERRLVL1 command should be placed in the program that is defined as the STARTP program:

Command	Description
DEF chain	Begin definition of program chain
ERRLVL1	Set error level to 1
GOTO main	Go to program main
END	End definition of program chain
STARTP chain	Designates program chain as the power-up program

To define program main for unit #0:

Command	Description
Ø_DEF main	Begin definition of program main on unit #0
Ø_GO	Start motion
Ø_END	End definition of program main on unit #0

Step 4 After all programming is completed program execution may be controlled by either a master terminal (diagram above), or by a master 6201 (diagram above).

Daisy-Chaining from a Computer or Terminal

Controlling the daisy-chain from a master computer or terminal follows the examples above:

Command	Description
Ø_RUN main	Run program main on unit #0
1_RUN main	Run program main on unit #1
2_GO1	Start motion on unit #2 axis #1
3_2A	Get A command response from unit #3 axis #2

Daisy-Chaining from a Master 6201

Controlling the daisy-chain from a master 6201 (the first unit on the daisy-chain) requires stored programs in the master 6201 which can control program and command execution on the slave 6201s. The example below demonstrates the use of the WRITE command to send commands to other units on the daisy-chain.

NOTE: The last unit on the daisy-chain must have RS-232C echo disabled (ECHOØ command).

Master 6201's main program:

Command	Description
DEF main	Program main
L	Indefinite loop
WHILE (IN.1 = bØ)	Wait for input #1 to go active
NWHILE	
GOL	Initiate linear interpolated move
WHILE (IN.1 = b1)	Wait for input #1 to go inactive
NWHILE	
WRITE"2_D25ØØ,4ØØØ"	Send message "2_D25ØØ,4ØØØ" down the daisy chain
WRITE"2_ACK"	Send message "2_ACK" down the daisy chain
LN	End of loop
END	End of program main

6201 unit #2 ack program:

Command	Description
DEF ack	Program ack
GO11	Start motion on both axes
END	End of program ack

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

Daisy-Chaining and RP240s

RP240s cannot be placed in the 6201 daisy chain; RP240s can only be connected to the designated RP240 port on a 6201. It is possible to use only one RP240 with a 6201 daisy-chain to input data for multiple units on the chain. The example below (for the 6201 master with an RP240 connected) reads data from the RP240 into variables #1 (*data1*) & #2 (*data2*), then sends the messages `3_Ddata1,data2`<CR> and `3_GO`<CR>.

Command	**Description**
`L`	Indefinite loop
`VAR1=DREAD`	Read RP240 data into variable #1
`VAR2=DREAD`	Read RP240 data into variable #2
`EOT0,Ø,Ø,Ø`	Turn off <CR>
`WRITE"3_D"`	Send message `"3_D"` down the daisy chain
`WRVAR1`	Send variable #1 data down the daisy chain
`WRITE","`	Send message `","` down the daisy chain
`EOT13,Ø,Ø,Ø`	Turn on <CR>
`WRVAR2`	Send variable #2 data down the daisy chain
`WRITE"3_GO"`	Send message `"3_GO"` down the daisy chain
`LN`	End of loop

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

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

C H A P T E R ⑤

Hardware Reference

Use this chapter as a quick-reference tool for 6201 system specifications (general specifications, I/O circuit drawings and pin outs, DIP switch and jumper settings, and motor specifications).

General Specifications

The following table contains general specifications for the 6201. I/O pin outs and circuit drawings and optional DIP switch settings are provided later in this chapter.

Parameter	Specification
Power	
AC Input	90–132VAC and 180–265VAC, 50/60Hz
Output Voltage	5VDC Isolated
Output Voltage at Motor Connector	75VDC maximum
Output Current at Motor Connector	7.1A maximum
Output Power at Motor Shaft	200 W Continuous, 300 W Peak (total shaft power from both motors combined)
Environmental	
Operating Temperature	32°F to 113°F (0°C to 45°C)
Storage Temperature	-22°F to 185°F (-30°C to 85°C)
Humidity	0% to 95% non-condensing
Performance	
Position Range	±419,430,000 steps per move (±2,147,483,648 absolute range)
Velocity Range	1 to 1,600,000 steps/sec
Acceleration Range	1 to 24,999,975 steps/sec^2
Stepping Accuracy	±0 steps from preset total
Velocity Accuracy	±0.02% of maximum rate
Velocity Repeatability	±0.02% of set rate
Motion Algorithm Update Rate	2 ms
Accuracy	±5 arcminutes (unloaded, bidirectional) with 6201 Series motors
Repeatability	±5 arcseconds typical (unloaded, bidirectional)
Hysteresis	Less than 2 arcminutes – 0.0334° (unloaded, bidirectional)
Calculation to determine contouring deviation from an arc (due to straight-line approximation to a curve)	Error in steps = $\left[\frac{[V_p(0.001\ \text{sec})^2]}{2r}\right]$ Where: V_p = steps/sec, r = radius in steps
RS-232C Interface	
Connections	3-wire (Rx, Tx and GND) connection to the AUX connector
Maximum number of daisy-chained 6201s	Up to 99 units
Address settings	Selectable (see *Optional DIP Switch & Jumper Settings*)
Communication Parameters	9600 baud (auto-baud option—see *Optional DIP Switch & Jumper Settings*), 8 data bits, 1 stop bit, no parity bit, full duplex

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

Specifications Table *(continued)*

Parameter	Specification
Inputs (see also *I/O Pin Outs & Circuit Drawings*)	
Home, CW/CCW Limits, Pulse Cutoff, Joystick Trigger, Joystick Release, Axes Select, Joystick Velocity	Optically isolated; TTL-compatible*; internal 6.8 KΩ pull-ups to 5V; voltage range is 0–24V.
Incremental Encoder	Optically isolated; Differential comparator accepts two-phase quadrature encoders with differential (recommended) or single-ended outputs (+5VDC TTL-compatible*). Maximum frequency = 1.6 MHz. Minimum time between transitions = 625 ns.
24 Programmable	Optically isolated, TTL-compatible* with internal 6.8 KΩ pull-up sourcing current (or you can change jumper JU2 to sink current). Voltage range = 0V–24V. 50-pin plug is compatible with OPTO-22™ signal conditioning equipment. Controllable with the 6000 Series programming language.
Triggers (*fast* triggers [TRG-A and TRG-B] on AUX connector)	Optically isolated, TTL-compatible* with internal 6.8 KΩ pull-up to +5VDC. Controllable with the 6000 Series programming language.
Analog (Joystick)	Voltage range = 0 - 2.5VDC, 8-bit A/D converter. **Input voltage must not exceed 5V.**
Outputs (see also *I/O Pin Outs & Circuit Drawings*)	
26 Programmable (includes OUT-A and OUT-B on AUX connector)	Optically isolated, TTL-compatible*, open collector output. Can be pulled up by connecting **OUT-P** to **+5V** on AUX connector, or to user-supplied voltage of up to 24V. Max. voltage in OFF state (not sinking current) = 24V, max. current in ON state (sinking) = 30mA. 50-pin plug is compatible with OPTO-22™ signal conditioning equipment. Controllable with the 6000 Series programming language.
Protective Circuits	
Short circuit	75VDC output shuts down if there is a short circuit in cables or motors. Shutdown is a LATCHED condition. Fix problem and cycle power to restart.
Overtemperature	6201 shuts down if its heatsink reaches a temperature of 60°C (140°F). Shutdown is a LATCHED condition. To restart, cool the 6201 below 30°C (86°F) and cycle power.
Power Dump	Dissipates excess regenerated energy. Threshold Voltage: 85VDC ± 3VDC Average Power Dissipation Rate: 8 Watts Peak Power Dissipation Rate: 722.5 Watts

* TTL-compatible voltage levels: Low ≤ 0.4V, High ≥ 2.4V

I/O Pin Outs & Circuit Drawings

This section, organized by connector, provides pin outs and circuit drawings for all 6201 inputs and outputs. **All inputs and outputs are optically isolated.**

Connector Part Numbers

The following table lists all input and output connectors on the 6201, whether or not a mating connector is supplied with the 6201, and the Compumotor part number for mating connectors. You can order connectors from Compumotor's Customer Service Department (800) 722-2282.

Connector Name	Connector Type	Is Mating Connector Supplied with 6201	Compumotor Part # for Mating Connector
LIMITS	9-pin screw terminal	Yes	43-008755-01
RP240	5-pin screw terminal	Yes	43-005561-01
AUX	Two 7-pin screw terminal	Yes	43-011995-01
Motor 1 Motor 2	7-pin screw terminal	No*	43-010262-01
Encoder 1 Encoder 2	9-pin screw terminal	No**	43-008755-01
Joystick	25-pin D terminal	No	None
Programmable Inputs Programmable Outputs	50-pin flat cable header	No***	43-000506-01

* Compumotor 6201 Series motors are supplied with motor connectors attached to the motor cable.
** Compumotor 6201 Series motors with encoder option are supplied with encoder connectors attached to the encoder cable.
*** Compumotor's VM50 adapter has a 50-pin connector attached to its 2-foot, 50-pin ribbon cable

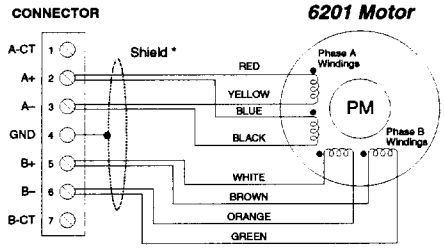
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Motor Connectors

The following table lists the pin outs for the 6201's two 7-pin screw terminal **MOTOR** connectors. Motor cable connections inside 6201 Series motors and connector wiring for parallel windings are shown below.

Pin	Name	Compumotor 6201 Series Motor Cable Color Code	Description
1	A-CT	No Connection	Phase A Center Tap
2	A+	Red & Blue	Phase A +
3	A-	Black & Yellow	Phase A –
4	GND	Shield	Ground
5	B+	White & Brown	Phase B +
6	B-	Green & Orange	Phase B –
7	B-CT	No Connection	Phase B Center Tap

6201 Series Motor Wiring



* Shield is internally connected to the motor's case

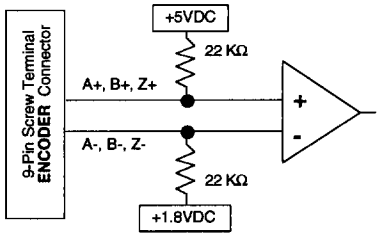
Encoder Connectors (For Use With Incremental Encoders Only)

The following table lists the pin outs for the 6201's two 9-pin screw terminal **ENCODER** connectors. The internal encoder input circuit is shown below.

Pin	In/Out	Name	Compumotor E Series Encoder Cable Colors	Description
9	OUT	+5V	Red	+5VDC output to power the encoder
8	IN	A Ch. +	Brown	A+ channel quadrature signal
7	IN	A Ch. -	Brown/White	A- channel quadrature signal
6	IN	B Ch. +	Green	B+ channel quadrature signal
5	IN	B Ch. -	Green/White	B- channel quadrature signal
4	IN	Z Ch. +	Orange	Z+ channel quadrature signal
3	IN	Z Ch. -	Orange/White	Z- channel quadrature signal
2	-----	Ground	Black	Isolated logic ground
1	-----	Shield	Shield	Internally connected to chassis ground (earth)

Schematic

Internal Encoder Input Circuit



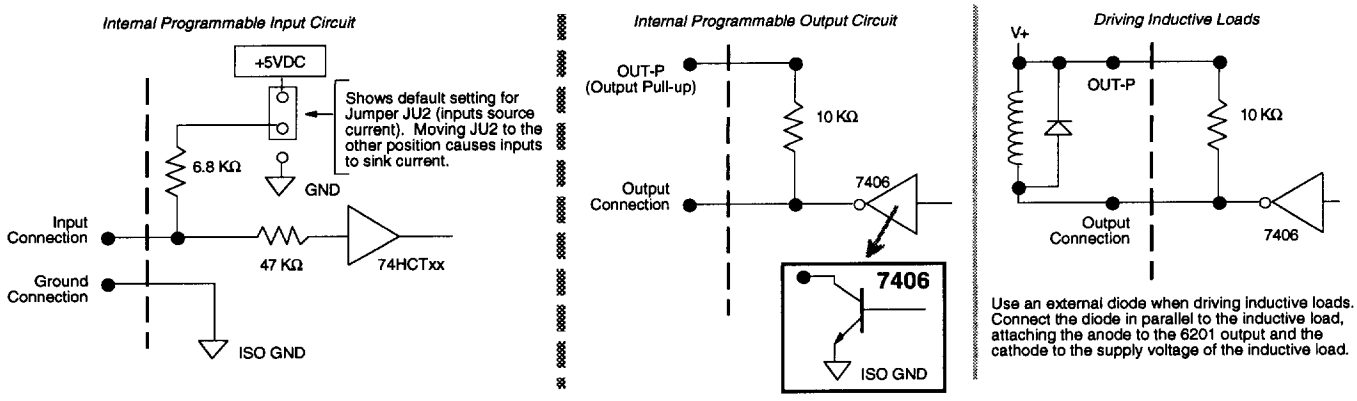
Programmable I/O Connectors

The following table lists the pin outs for the 6201's two 50-pin programmable I/O connectors. The internal input and output circuits are illustrated below.

Pin	Output Connector	Input Connector	Pin	Output Connector	Input Connector
49	+5VDC	+5VDC	23	Output #13	Input #13
47	Output #1 (LSB)	Input #1 (LSB)	21	Output #14	Input #14
45	Output #2	Input #2	19	Output #15	Input #15
43	Output #3	Input #3	17	Output #16	Input #16
41	Output #4	Input #4	15	Output #17	Input #17
39	Output #5	Input #5	13	Output #18	Input #18
37	Output #6	Input #6	11	Output #19	Input #19
35	Output #7	Input #7	09	Output #20	Input #20
33	Output #8	Input #8	07	Output #21	Input #21
31	Output #9	Input #9	05	Output #22	Input #22
29	Output #10	Input #10	03	Output #23	Input #23
27	Output #11	Input #11	01	Output #24 (MSB)	Input #24 (MSB)
25	Output #12	Input #12			

All even-numbered pins are connected to logic ground.

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



Auxiliary (**AUX**) Connector

The following table lists the pin outs for the 6201's auxiliary (**AUX**) 14-pin screw terminal.

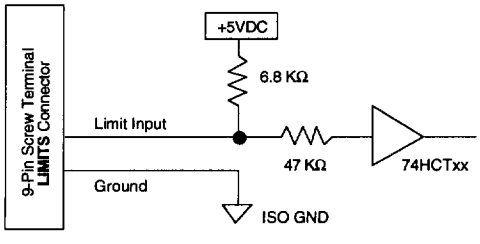
Pin	In/Out	Name	Description
1	IN	Rx	Receive input for RS-232C interface
2	OUT	Tx	Transmit output for RS-232C interface
3	------	GND	Isolated ground for RS-232C interface
4	------	SHLD	*Shield*—Internally connected to chassis ground (earth)
5	OUT	+5V	Connect to **OUT-P** to power the 26 programmable outputs. 1.5A limit (applies to total load on all of the I/O connectors) — e.g., if 2 encoders are drawing at total of 500mA, then 1.0A is left for other purposes.
6	IN	OUT-P	Internally connected to pull-up resistors for the 24 programmable outputs. Connecting this input to the +5V pin (this is already done at the factory) makes the outputs TTL compatible. Connection to other voltages (max. = 24V) allows for compatibility with other signal levels.
7	IN	TRG-A	Fast trigger input A: Like programmable inputs, but can function as encoder capture input that latches positions within 50 µs (see `INFNC` command). Internal circuit is identical to the limit input (see limit input circuit drawing below)
8	IN	TRG-B	Same function as trigger input A above
9	------	GND	Isolated ground
10	OUT	OUT-A	Auxiliary programmable output A: Function and circuit is identical to the other 24 programmable outputs (see programmable I/O circuit drawing above)
11	OUT	OUT-B	Same function as auxiliary programmable output A above
12	------	GND	Isolated ground
13	------	SHLD	*Shield*—Internally connected to chassis ground (earth)
14	IN	P-CUT	Normally grounded. When un-grounded, step pulses are inhibited independent of the microprocessor, *but the 6201 may lose track of the motor's position (encoder position will still be accurate).* Internal circuit is identical to limit circuit (see limit circuit drawing below).

Limits (**LIMITS**) Connector

The following table contains the pin outs for the 6201's limit (**LIMITS**) 9-pin screw terminal. The internal limit input circuit is illustrated below.

Pin	In/Out	Name	Description
1	—	SHLD	*Shield*—Internally connected to chassis ground (earth)
2	—	GND	Isolated ground
3	IN	2HOM	Home limit input for axis 2
4	IN	2CCW	Counter-clockwise limit input for axis 2
5	IN	2CW	Clockwise limit input for axis 2
6	—	GND	Isolated ground
7	IN	1HOM	Home limit input for axis 1
8	IN	1CCW	Counter-clockwise limit input for axis 1
9	IN	1CW	Clockwise limit input for axis 1

Internal Limit Input Circut

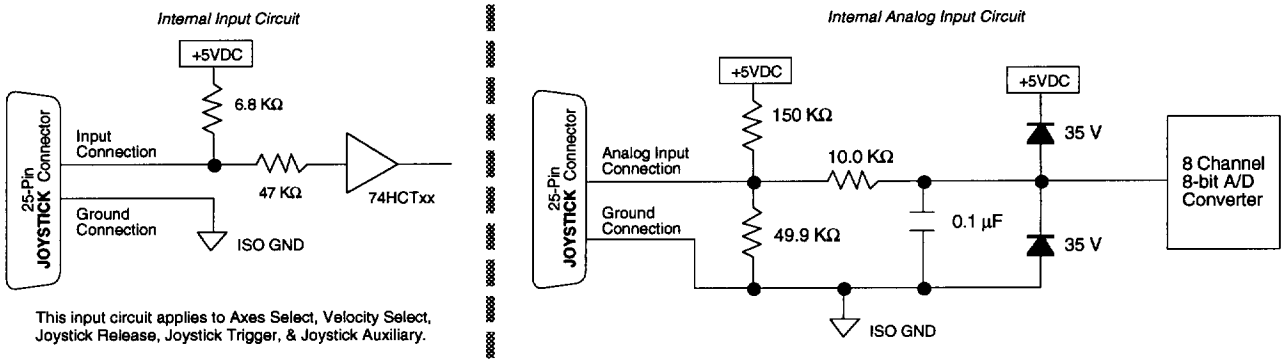


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Joystick Connector

Pin outs for the 6201's Joystick 25-pin D connector are listed below. The following illustration shows the internal input circuits.

Pin	In/Out	Name	Description
1	IN	Analog Channel 1	Analog input for feedrate control or joystick control of axis. Input voltage must not exceed 5V.
2	IN	Analog Channel 2	Analog input for feedrate control or joystick control of axis. Input voltage must not exceed 5V.
3	IN	Analog Channel 3	Analog input for feedrate control or joystick control of axis. Input voltage must not exceed 5V.
8	—	Shield	Shield
14	—	Ground	Isolated Ground
15	IN	Axes Select	If using only one analog channel, you can use this input to alternately control axes 1 or 2
16	IN	Velocity Select	Input to select high or low velocity range (as defined with JOYVH or JOYVL command)
17	IN	Joystick Release	When low (grounded), joystick mode can be enabled. When high (not grounded), program execution will continue with the first command after the joystick enable (JOY) statement.
18	IN	Joystick Trigger	Status of this active-low input can be read by a program (using the INO or TINO commands) to control program flow, or to enter the 6201 into joystick mode.
19	IN	Joystick Auxiliary	Status of this active-low input can be read by a program (using the INO or TINO commands) to control program flow.
23	OUT	+5VDC (out)	+5VDC power output



RP240 Connector

Pin outs for the **RP240** 5-pin screw terminal connector are listed below

Pin	In/Out	Name	Description
5	—	Shield	Shield
4	OUT	Tx	Transmit output to RP240's Rx input
3	IN	Rx	Receive input from RP240's Tx output
2	—	Ground	Ground
1	OUT	+5VDC (out)	+5VDC power output

Optional DIP Switch & Jumper Settings

The 6201 is equipped with a four-position DIP switch you can use to select the device address (necessary only for daisy-chaining multiple 6201s with one RS-232C circuit), and to use the auto baud rate feature.

The 6201 is also equipped with a jumper (JU2) you can use if you need to change the setting for the programmable inputs from sourcing current (default) to sinking current.

Accessing the DIP Switches and Jumpers

Use the following procedure to access the DIP switches and jumper inside the 6201.

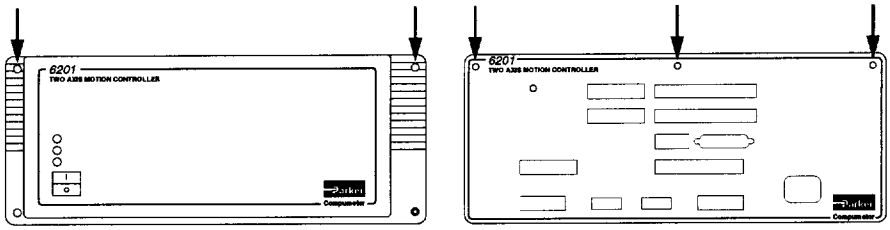
CAUTION

While handling the 6201 printed circuit assembly (PCA), be sure to observe proper grounding techniques to prevent electro-static discharge (ESD).

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

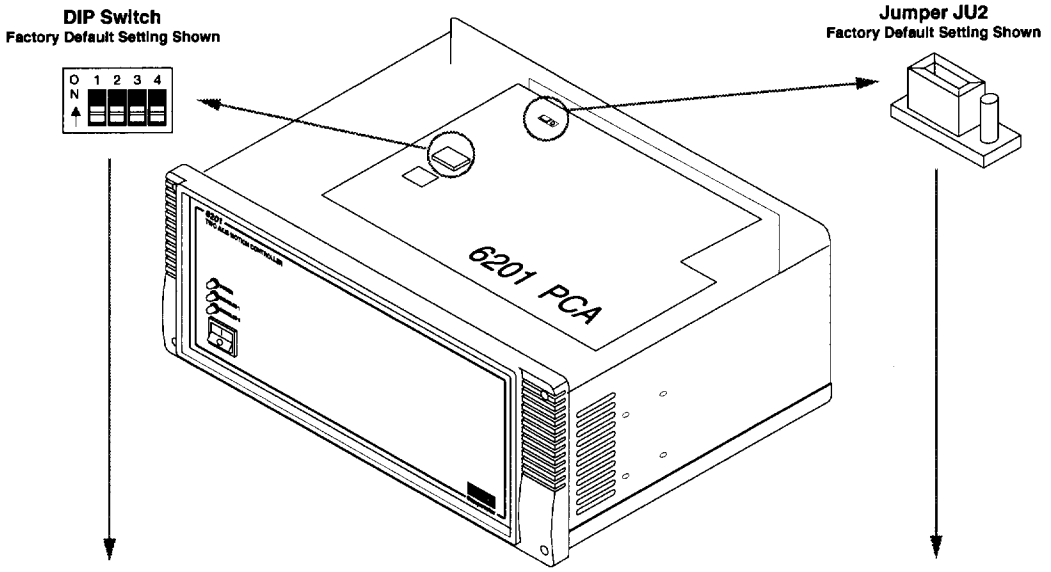
Step 1 Remove power before removing the 6201's enclosure.

Step 2 Using a Phillips screwdriver, remove the two screws on the front panel and the three screws on the back panel that hold the 6201's top cover.



Step 3 Gently lift the top cover from the 6201.

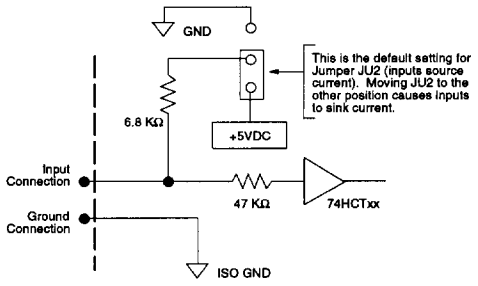
The illustration below shows the DIP switches and Jumper JU2 and lists the optional settings.



Switch #1	Switch #2	Switch #3	Device Address	
OFF	OFF	OFF	Ø	(default)
ON	OFF	OFF	1	
OFF	ON	OFF	2	
ON	ON	OFF	3	
OFF	OFF	ON	4	
ON	OFF	ON	5	
OFF	ON	ON	6	
ON	ON	ON	7	

* Device address is checked upon power up or reset.

Internal Programmable Input Circuit



Switch #4 ON = Auto Baud Enabled
Switch #4 OFF = Auto Baud Disabled (default)

Following these steps to implement the Auto Baud feature:

1. Change Switch #4 to the ON position.
2. Connect the terminal to the 6201's RS-232C serial port on the AUX connector.
3. Power up the terminal.
4. Cycle power to the 6201 and immediately press the space bar several times.
5. The 6201 should send a message to the terminal confirming the baud rate (i.e., `*9600`). You should now be communicating to the 6201 via the terminal. If no baud rate messages is received, verify steps 1 through 3 and repeat step 4.
6. Change Switch #4 to the OFF position.
7. Cycle power to the 6201. You should be communicating to the 6201 via the terminal at the previously determined rate.

NOTE: If Auto Baud is enabled, the 6201 performs its auto baud routine every time it is powered up or reset. The 6201 is only capable of matching 1200, 2400, 4800, 9600 and 19200 baud. Once the baud rate has been determined, the 6201 stores that baud rate in non-volatile memory; therefore, Switch #4 should be set to the OFF position after the baud rate has been determined.

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

Motor Specifications

This section contains information about Compumotor 6201 Series motors – rotor inertia, speed/torque curves, and motor dimensions . For information about using non–Compumotor motors, see the following section.

Rotor Inertia

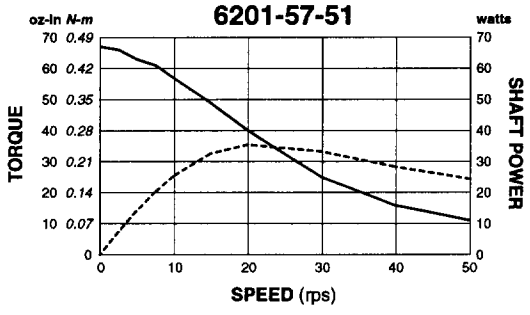
The table below lists rotor inertia for each 6201 Series motor.

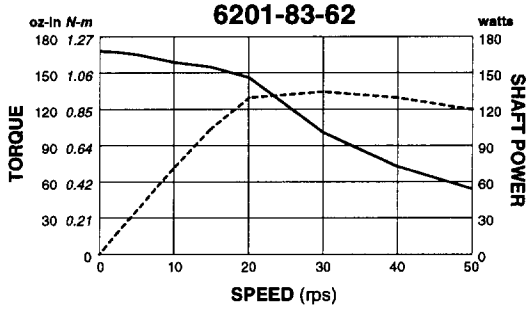
	Rotor Inertia oz-in^2	Rotor Inertia Kg-cm^2
SIZE 23		
6201-57-51-MO	0.48	0.088
6201-57-83-MO	1.28	0.234
6201-57-102-MO	1.75	0.320
SIZE 34		
6201-83-62-MO	3.50	0.64
6201-83-93-MO	6.70	1.23
6201-83-135-MO	10.24	1.87

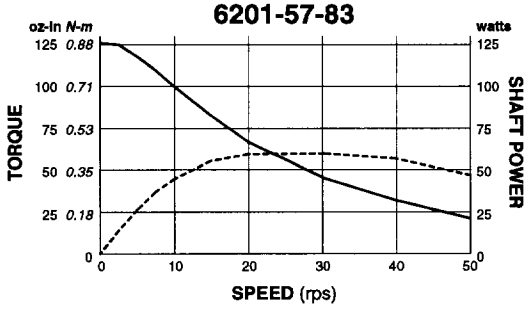
Speed/Torque Curves

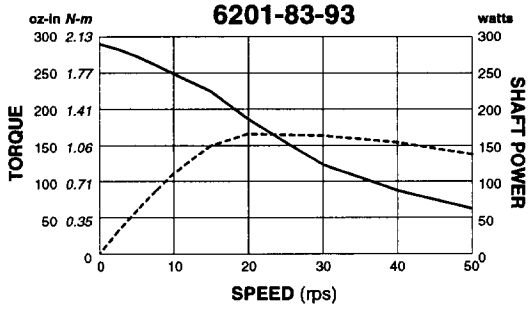
This section provides speed/torque (performance) curves for each 6201 Series motor.

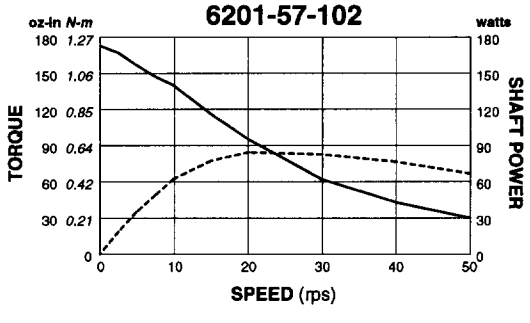
—— Torque
------ Shaft Power

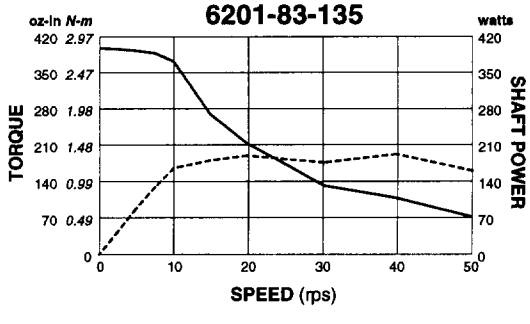












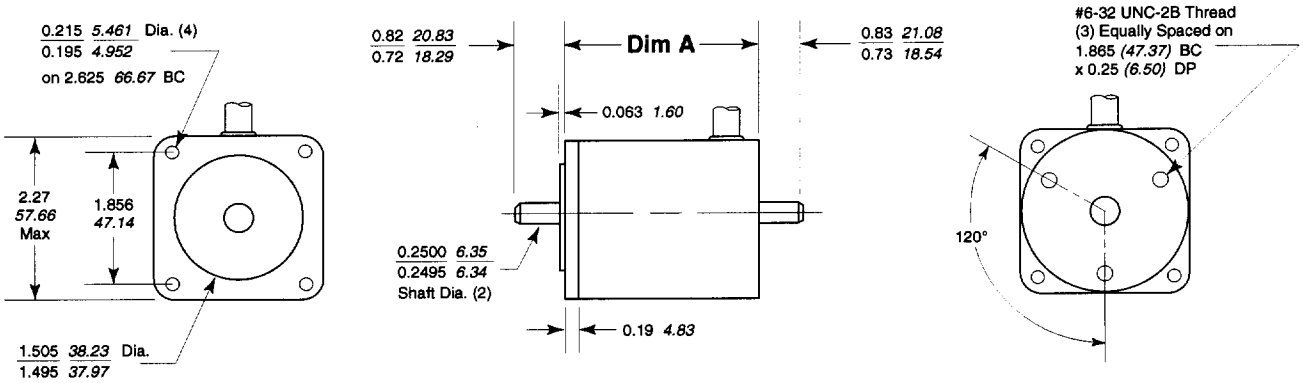
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Motor Dimensions

This section provides dimensions for 6201 Series motors.

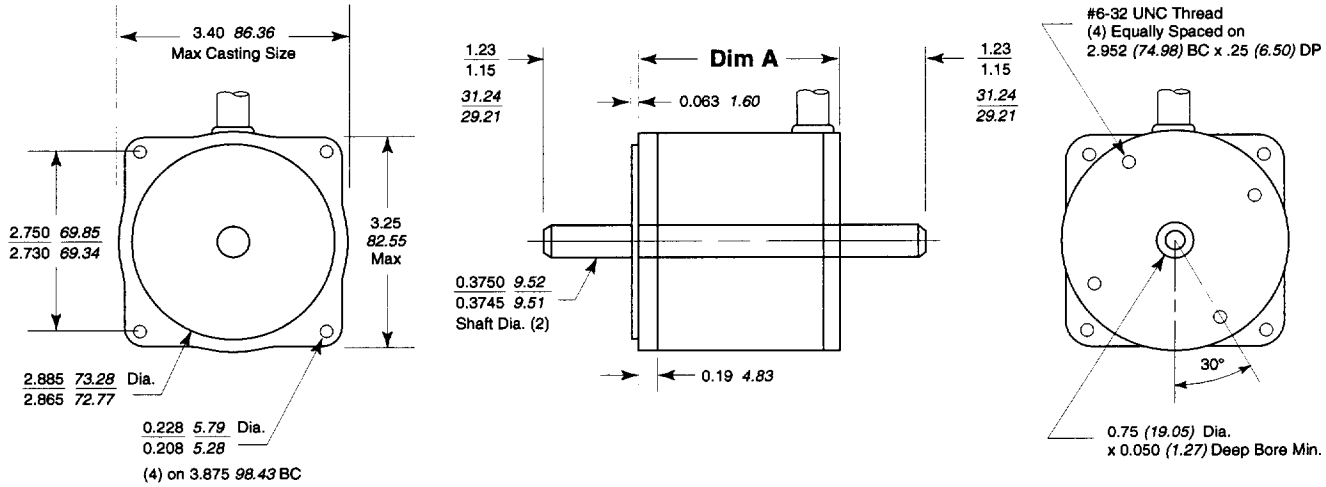
23 Frame Motors	
Model Size	**Dim A**
6201-57-51	2.0 *50.8*
6201-57-83	3.1 *78.7*
6201-57-102	4.0 *101.6*

NOTE:
Dimensions are in
inches *millimeters*



34 Frame Motors	
Model Size	**Dim A**
6201-83-62	2.5 *63.5*
6201-83-93	3.7 *94.0*
6201-83-135	5.2 *132.1*

NOTE:
Dimensions are in
inches *millimeters*



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

Using Non-Compumotor Motors

We recommend that you use Compumotor 6201 Series motors with the 6201. If you use a non-Compumotor motor, it must meet the following requirements:

- For best performance, motor inductance should be between 1 mH and 10 mH, but motors with inductance ratings as low as 0.5 mH may be used. *The motor inductance cannot drop below 0.5 mH.*
- A minimum of 500VDC high-pot insulation rating from phase-to-phase and phase-to-ground.
- The motor must not have riveted rotors or stators.
- Do not use solid rotor motors.
- Test all motors carefully. Verify that the motor temperature in your application is within the system limitations. *The motor manufacturer's maximum allowable motor case temperature must not be exceeded.* You should test the motor over a 2-to-3 hour period. Motors tend to have a long thermal time constant, but can still overheat, which results in motor damage.

CAUTION

Consult a Compumotor Applications Engineer if you have any questions regarding the use of a non-Compumotor motor. Call (800) 358-9070.

Wiring Configurations

Refer to the manufacturer's motor specification document to determine the motor's wiring configuration. You can also determine the wiring configuration with an ohmmeter using the procedures below (*4-Lead Motor, 6-Lead Motor, 8-Lead Motor*). Once you determine the correct motor wiring configuration, use the terminal connection diagram, shown at the end of this section, that applies to your configuration.

4-Lead Motor

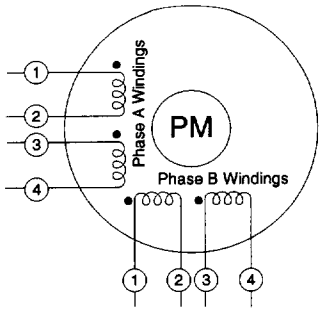
1. Label one motor lead **A+**.
2. Connect one lead of an ohmmeter to the **A+** lead and touch the other lead of the ohmmeter to the three remaining motor leads until you find the lead that creates continuity. Label this lead **A-**.
3. Label the two remaining leads **B+** and **B-**. *Verify that there is continuity between the* **B+** *and* **B-** *leads.*
4. Proceed to the *Terminal Connections* section below.

6-Lead Motor

1. Determine, with an ohmmeter, which three of the six motor leads are common (one phase).
2. Label each one of these three motor leads **A**.
3. Using the ohmmeter, verify that the remaining three leads are common.
4. Label the remaining three leads **B**.
5. Set the ohmmeter range to approximately the 100 ohm scale.
6. Connect the ohmmeter's negative lead to one of the motor leads labeled **A**. Alternately measure the resistance to the two remaining motor leads also labeled **A**. The resistance measurements will reflect one of the following two scenarios.
   Scenario #1 — The resistance measurements to the two remaining motor leads are virtually identical. Label the two remaining motor leads **A+**and **A–**. Label the motor lead connected to the negative lead of the ohmmeter **A-CT** (this is the center tap lead for Phase A of the motor).
   Scenario #2 — The resistance measurement to the second of the three motor leads measures 50% of the resistance measurement to the third of the three motor leads. Label the second motor lead **A-CT** (this is the center tap lead for Phase A of the motor). Label the third motor lead **A–** Label the motor lead connected to the ohmmeter **A+.**
7. Repeat the procedure as outlined in step 6 for the three leads labeled **B** (**B–CT** is the center tap lead for Phase B of the motor).
8. Connect the **A-CT** motor lead to the **A-CT** pin on the **MOTOR** connector. Connect the **B-CT** motor lead to the **B-CT** pin on the **MOTOR** connector.
9. Proceed to the *Terminal Connections* section below.

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

8-Lead Motor



Because of the complexity involved in phasing an 8-lead motor, you must refer to the manufacturer's motor specification document. You can configure the 8-lead motor in parallel or series. Using the manufacturer's specifications, label the motor leads as shown in the next drawing.

Parallel Configuration Use the following procedure for parallel configurations.

1. Connect motor leads A1 & A3 together and relabel this common point **A+**.
2. Connect motor leads A2 & A4 together and relabel this common point **A–**.
3. Connect motor leads B1 & B3 together and relabel this common point **B+**.
4. Connect motor leads B2 & B4 together and relabel this common point **B–**.
5. Proceed to the *Terminal Connections* section below.

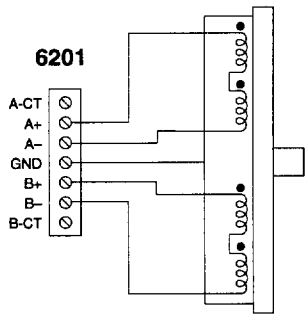
Series Configuration Use the following procedure for series configurations.

1. Connect A2 & A3 to **A-CT**. Also connect B2 & B3 to **B-CT** (**A-CT**. and **B-CT** are located on the 6201's 7-pin **MOTOR** connector).
2. Relabel the A1 lead **A+**.
3. Relabel the A4 lead **A-**.
4. Relabel the B1 lead **B+**.
5. Relabel the B4 lead **B-**.
6. Proceed to the *Terminal Connections* section below.

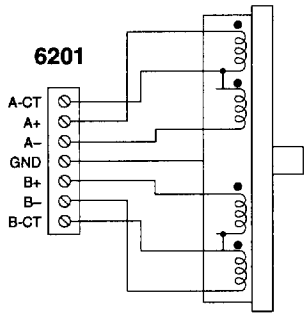
Terminal Connections

After you determine the motor's wiring configuration, connect the motor leads to the 7-pin **MOTOR** connector according to the following figure.

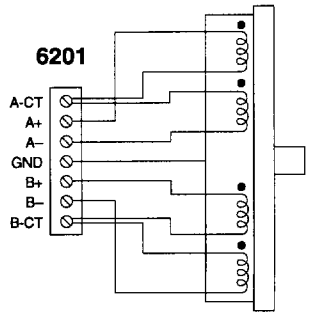
4-Lead Motor



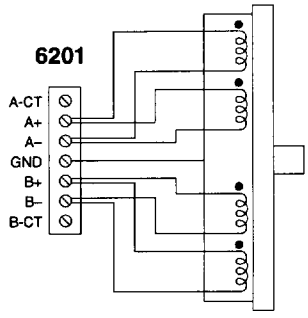
6-Lead Motor



8-Lead Motor Series



8-Lead Motor Parallel



WARNING

Do not connect or disconnect the motor with the power on. Doing so will damage the contacts of the motor connector and may cause personal injury.

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

Setting Motor Current – Non-Compumotor Motors

To set motor current for a non-Compumotor motor, refer to the formulas below that correspond to your motor (4-lead, 6-lead, 8-lead) and use the motor current settings shown in the table below to set the motor's current.

Current	Switch #1	Switch #2	Switch #3	Switch #4	Switch #5	Switch #6
2.5A	**ON**	off	off	off	off	off
3.1A	off	**ON**	off	off	off	off
3.5A	off	off	**ON**	off	off	off
4.4A	off	off	off	**ON**	off	off
4.5A	**ON**	off	**ON**	off	off	off
4.8A	off	**ON**	**ON**	off	off	off
5.0A	**ON**	off	off	**ON**	off	off
5.2A	off	**ON**	off	**ON**	off	off
5.3A	off	off	**ON**	**ON**	off	off
5.6A	off	off	off	off	**ON**	off
5.7A	**ON**	off	**ON**	**ON**	off	off
5.8A	off	**ON**	**ON**	**ON**	off	off
5.9A	**ON**	off	off	off	**ON**	off
6.0A	**ON**	**ON**	**ON**	**ON**	off	off
6.2A	off	**ON**	**ON**	off	**ON**	off
6.3A	**ON**	off	off	**ON**	**ON**	off
6.4A	off	off	**ON**	**ON**	**ON**	off
6.5A	off	**ON**	**ON**	**ON**	**ON**	off
6.6A	**ON**	**ON**	**ON**	**ON**	**ON**	off
7.0A	off	off	off	off	off	**ON**
7.1A	**ON**	**ON**	**ON**	**ON**	off	**ON**

☞ **Never increase current more that 10% above the specified rating.**

4-Lead Motors

If you use a 4-lead motor, the manufacturer's current specification will translate directly to the values shown in the motor current settings table.

6-Lead Motors

If you use a 6-lead motor, and the manufacturer specifies the motor current as a bipolar rating, you can use the motor current settings table directly (no conversion) to set motor current.

If the manufacturer specifies the motor current as a unipolar rating, use the following formula to convert the unipolar current rating to the correct bipolar rating:

Unipolar Current * 0.707 = Bipolar Current

After you make the conversion, use the motor current settings table to set the motor current.

8-Lead Motors

Manufacturers generally use either a unipolar rating or a bipolar rating for motor current in 8-lead motors.

Unipolar Rating: If the manufacturer specifies the motor current as a unipolar rating:

- Use the following formula to convert the unipolar current rating to the correct bipolar rating:

  Unipolar Current * 0.707 = Bipolar Current

- If you wire the motor in **series**, use the motor current table and the converted value to set the motor current.
- If you wire the motor in **parallel**, you must **double** the converted value and use the motor current table to set the motor current.

Bipolar Rating: If the manufacturer specifies the motor current as a bipolar series rating:

- If you wire the motor in **series**, use the motor current settings table directly.
- If you wire the motor in **parallel**, you must double the manufacturer's rating and then use the motor current table to set the motor current.

If you have any questions about setting motor current, call Compumotor's Applications Engineering Department at (800) 358-9070.

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

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

C H A P T E R 6

Troubleshooting

The information in this chapter will enable you to isolate and resolve system hardware and software problems.

Troubleshooting Basics

If 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 have accomplished 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 additional units available, you may want to exchange them with existing components in your system to help identify the source of the problem.

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 isolate and 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.

If you are having difficulty isolating a problem be sure to document all occurrences of the problem along with as much specific information, such as time of occurrence, 6201 status, and anything else that was happening when the problem occurred.

Once you have isolated a 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 at (800) 358-9070.

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

Reducing Electrical Noise

For detailed information on reducing electrical noise, refer to the *Engineering Reference* section of the Parker Compumotor/Digiplan catalog.

Error Messages and Debug Tools

A list of all possible error messages, and their causes, is provided in the ***6000 Series Software Reference Guide***. For instructions on using the 6201's program debug tools (Trace mode, Single-Step mode, I/O activation, bad command detection, etc.) refer to *Program Debug Tools* in the *Programming Guide* section of the ***6000 Series Software Reference Guide***.

Test Program

A test program is available to verify the 6201 is sending pulses to the drives and to verify the drives/motors are functioning properly. The test program can be initiated by issuing the TEST command over the RS-232C interface (see Test Procedure in Chapter 2), or by accessing the RP240 TEST menu (see *RP240 Front Panel Interface* in Chapter 4).

WARNING

If you use the TEST program, the end-of-travel limits will be ignored. If necessary, disconnect the load to ensure the test moves do not damage your equipment or injure personnel.

Common Problems & Solutions

The following table presents some guidelines to help you isolate problems with your motion control system. Some common symptoms are listed along with a list of possible causes and remedies.

- Look for the symptom that most closely resembles what you are experiencing.
- Look through the list of possible causes so that you better understand what may be preventing proper operation.
- Start from the top of the list of remedies and use the suggested procedures to isolate the problem.
- Refer to other sections of the manual for more information on 6201 set up, system connections, and feature implementation. You may also need to refer to the ***6000 Series Software Reference Guide***.

Problem	Cause	Solution
Encoder counts are missing	1. Improper wiring 2. Encoder slipping 3. Encoder too hot 4. Electrical noise 5. Encoder frequency too high	1. Check wiring 2. Check and tighten encoder coupling 3. Reduce encoder temperature with heatsink, thermal insulator, etc. 4a. Shield wiring 4b. Use encoder with differential outputs 5. Peak encoder frequency must be below 1.6MHz post-quadrature. Peak frequency must account for velocity ripple
Erratic operation	1. Electrical Noise 2. Improper shielding 3. Improper wiring	1. Reduce electrical noise or move the 6201 away from noise source 2. Refer to the Technical Reference section of the Compumotor product catalog. 3. Check wiring for opens, shorts, and mis-wired connections
Motor creeps at slow velocity in encoder mode	1. Encoder direction opposite of motor direction 2. Encoder connected to wrong axis	1. Switch encoder connections CHA+ & CHA- with CHB+ & CHB- 2. Check encoder wiring
Motor does not move in joystick mode	1. Joystick Release input not grounded 2. Improper wiring	1. Ground Joystick Release input 2. Check wiring for opens, shorts, and mis-wired connections

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

Problems, Causes & Solutions (*continued*)

Problem	Cause	Solution
No motion	1. Status LED off or red 2. Limits engaged 3. Step pulse too narrow for drive to recognize 4. Drive fault level incorrect 5. Improper wiring 6. P-CUT (Pulse cut-off) not grounded 7. Load is jammed 8. No torque from motor	1. See status LED problems below 2. Move load off of limits or disable limits with `LHØ,Ø` 3. Set pulse width to drive specifications using `PULSEx,x` 4. Set drive fault level using `DRFLVL11` 5. Check limit connections 6. Connect the P-CUT terminal to GND 7. Remove power and clear jam 8. See problem: *No Torque*
No torque	1. Improper wiring 2. No power to 6201 3. Drive failed 4. Drive faulted 5. Drive shutdown	1. Check system wiring 2. Check power to connection 3. Consult factory 4. Consult factory 5. Enable drive with `DRIVE11`
Program execution: power-up program (`STARTP`) does not run	1. P-CUT not grounded 2. `STARTP` not defined	1. Connect the P-CUT terminal to ground and `RESET` 2. Check response to `STARTP` command. Define `STARTP` program and `RESET` unit.
Program execution: stops at the `INFEN1` command	1. `INFEN1` enables drive fault monitoring, but the drive fault level (`DRFLVL`) command is set incorrectly for the drive being used.	1. Issue the correct `DRFLVL` command for your drive (refer to the `DRFLVL` command)
Program execution: the first time a program is run, the move distances are incorrect. Upon downloading the program the second time, move distances are correct.	1. Scaling parameters were not issued when the program was downloaded; or scaling parameters have been changed since the program was defined	1. Issue and the scaling parameters (`SCALE1`, `SCLA`, `SCLD`, `SCLV`, `PSCLA`, `PSCLD`, `PSCLV`) before saving any programs
Programmable inputs not working	1. Improper wiring	1. Check wiring for opens, shorts, and mis-wired connections
Programmable outputs not working	1. Output connected such that it must source current (pull to positive voltage) 2. OUT-P not connected to +5V or other positive voltage source 3. Improper wiring	1. Outputs are open-collector and can only sink current – change wiring. 2. Connect OUT-P to +5V supplied or other voltage in system 3. Check wiring for opens, shorts, and mis-wired connections
RS-232C communication does not function	1. Improper RS-232C Interface or communication parameters 2. RS-232C disabled 3. In daisy chain, unit may not be set to proper address	1. See *RS-232C Troubleshooting* section below 2. Enable RS-232C with the `E` command (all units if daisy-chained) 3. Verify DIP switch settings (see *Optional DIP Switch & Jumper Settings* in Chapter 5)
STATUS LED on back panel is off	1. No AC power	1. Check AC power and connections
STATUS LED on back panel is red	1. Internal Board Monitor Alarm (BMA) has detected an non-recoverable fault	1. Enter the `RESET` command or cycle power
POWER LED on front panel is off	1. No AC power	1. Check AC power and connections
DISABLED LED on front panel is on	1. The motor is not connected 2. A motor winding is open 3. 6201 has detected a short circuit in the motor wiring 4. 6201 is overheating 5. Drive shutdown 6. The 6201 may have internal damage	1. Connect the motor 2. Measure winding continuity. Check the series connections for an 8-leaded motor 3. Check for miswiring—carefully check the motor wires for loose strands shorting the windings 4. Verify that the 6201's ambient temperature does not exceed 45°C (113°F). Cool 6201 below 30°C (86°F), cycle power to restart. 5. Enable drive with `DRIVE11` 6. Return the 6201 to Compumotor for servicing
Trigger inputs not working	1. Improper wiring	1. Check wiring for opens, shorts, and mis-wired connections
Wrong direction	1. Phase of Step motor reversed	1. Switch PHA+ with PHA- connection from 6201 to motor
Wrong speed or distance	1. Wrong resolution setting 2. Improper wiring	1. Check and set resolution on 6201 with `DRES25ØØØ, 25ØØØ` & `ERES4ØØØ, 4ØØØ` (encoder resolution may be different if using non-Compumotor encoder) 2. Check wiring

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

Problems, Causes & Solutions (*continued*)

Problem	Cause	Solution
Motor moves erratically at low speeds	1. Motor current is set incorrectly	1. Check the current select DIP switches and verify that the current is set correctly
Motor stalls at high speeds	1. Velocity is too high 2. Motor current is not set correctly 3. Motor is undersized for application	1. The 6201's internal drives will operate with a maximum pulse rate of 2 MHz or 50 rps, whichever comes first—decrease the velocity 2. Check the current select DIP switches and verify that the current is set correctly 3. Verify that the motor is sized correctly for your application
Motor stalls during acceleration	1. Motor current is not set correctly 2. Acceleration is set too high 3. Insufficient rotor inertia 4. Motor is undersized for application	1. Check the current select DIP switches and verify that the current is set correctly 2. Decrease the acceleration 3. Add inertia to the motor shaft 4. Verify that the motor is sized correctly for your application
Motor (unloaded) stalls at nominal speed	1. Insufficient rotor inertia 2. Mid-frequency resonance	1. Add inertia to the motor shaft 2. Add a damper to the shaft

RS-232C Troubleshooting

If you are having problems communicating with the 6201, try the following procedure to troubleshoot the communications interface.

Step 1 Power-up your computer or terminal *and then* power-up the 6201.

Step 2 The serial port of your computer/terminal may require hardware handshaking. If so, you must disable handshaking with your terminal emulator software package. You can also disable hardware handshaking by connecting the computer's/terminal's RTS to CTS (usually pins 4 and 5) and DSR to DTR (usually pins 6 to 20).

Step 3 Verify that the computer/terminal and 6201 are configured to the same baud rate, number of data bits, number of stop bits, and parity. If your terminal is not capable of 9600 baud, you can use the 6201's *auto-baud* function to automatically set the 6201's baud rate equal to the terminal's baud rate. Refer to the *Optional DIP Switch & Jumper Settings* section in Chapter 5 for instructions.

Step 4 Check to make sure you are using DC common or signal ground as your reference, *not* earth ground.

Step 5 Cable lengths for RS-232C should not exceed 50 feet. As with any control signal, be sure to shield the cable to earth ground at one end only.

Step 6 Press the return key several times. The cursor should move down 1 or 2 lines each time you press the return key. If your terminal displays garbled characters, check the terminal's protocol set-up. The baud rate setting probably does not match the 6201's setting. See step 3 above.

Step 7 If the cursor does not move after pressing the space bar:

a. Disconnect the RS-232C cable from the 6201.

b. Connect the RS-232C cable's Rx and Tx lines together at the end that connects to the 6201.

c. Press the space bar. If the cursor does not move, either the computer (or terminal) or the cable is defective.

Step 8 Once you are able to make the cursor move, enter some characters. These characters should appear on the computer or terminal display. If each character appears twice, your host is set to half-duplex; set it to full-duplex.

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

Returning the System

If you must return your 6201 system to affect repairs or upgrades, use the following steps:

Step 1 Get the serial number and the model number of the defective unit, and a purchase order number to cover repair costs in the event the unit is determined by the manufacturers to be out of warranty.

Step 2 Before you return the unit, have someone from your organization with a technical understanding of the 6201 system and its application include answers to the following questions:

- What is the extent of the failure/reason for return?
- How long did it operate?
- Did any other items fail at the same time?
- What was happening when the unit failed (e.g., installing the unit, cycling power, starting other equipment, etc.)?
- How was the product configured (in detail)?
- What, if any, cables were modified and how?
- With what equipment is the unit interfaced?
- What was the application?
- What was the system environment (temperature, enclosure, spacing, unit orientation, contaminants, etc.)?
- What upgrades, if any, are required (hardware, software, user guide)?

Step 3 In the USA, call Parker Compumotor for a Return Material Authorization (RMA) number. Returned products cannot be accepted without an RMA number. The phone number for Parker Compumotor Applications Department is (800) 358-9070.

Ship the unit to: Parker Hannifin Corporation
Compumotor Division
5500 Business Park Drive, Suite D
Rohnert Park, CA 94928
Attn: RMA # xxxxxxx

Step 4 In the UK, call Parker Digiplan for a GRA (Goods Returned Authorization) number. Returned products cannot be accepted without a GRA number. The phone number for Parker Digiplan Repair Department is 0202-690911. The phone number for Parker Digiplan Service/Applications Department is 0202-699000.

Ship the unit to: Parker Digiplan Ltd.,
21, Balena Close,
Poole,
Dorset,
England.
BH17 7DX

Step 5 Elsewhere: Contact the distributor who supplied the equipment.

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

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

Appendix A: Alphabetical Command List

Command Name	Command Description
[<cr>]	Carriage Return
[<lf>]	Line Feed
[:]	Colon
!	Immediate Command Identifier
@	Global Command Identifier
;	Begin Comment
$	Label Deceleration
#	Step Through a Program
'	Enter Data (Single quote)
[.]	Bit Select
["]	Begin and End String
[\ ]	ASCII Character Designator
[=]	Assignment or Equivalence
[>]	Greater Than
[>=]	Greater Than or Equal
[<]	Less Than
[<=]	Less Than or Equal
[<>]	Not Equal
[( )]	Operation Priority Level
[+]	Addition
[-]	Subtraction
[*]	Multiplication
[/]	Division
[&]	Boolean And
[|]	Boolean Or
[^]	Boolean Exclusive Or
[~( )]	Boolean Not
[>>]	Shift from Right to Left
[<<]	Shift from Left to Right
A	Acceleration
[A]	Acceleration Assignment
AD	Deceleration
[AD]	Deceleration Assignment
ADDR	Daisy-Chain Address
[AND]	And
[ANV]	Analog Input Value
[AS]	Axis Status Value
[ATAN( )]	Inverse Tangent
ANVO	Analog Voltage Override
ANVOEN	Analog Voltage Override Enable
[b]	Binary Identifier
BP	Set a Program Break Point
BREAK	Terminate Program Execution
C	Continue
[CNT]	Counter Value
CNTE	Hardware Up/Down Counter Input
CNTR	Hardware Up/Down Counter Reset
COMEXC	Enable Continuous Command Mode
COMEXL	Continue Command Execution on Limit
COMEXP	Continue Command Execution on In Position
COMEXS	Continue Command Execution on Stop
[COS( )]	Cosine
D	Distance
[D]	Distance Assignment
[DAT]	Data Assignment
DATA	Data Statement

Command Name	Command Description
[DATP]	Data Program
DATPTR	Set Data Pointer
DATRST	Reset Data Pointer
DATSIZ	Data Program Size
DATTCH	Data Teach
DCLEAR	Clear RP240 Display
DEF	Define a Program/Subroutine
DEL	Delete a Program/Subroutine
DJOG	Enable RP240 Jog Mode
DLED	Turn RP240 LEDs On/Off
DPASS	Set RP240 Password
DPCUR	Position Cursor on RP240 Display
[DPTR]	Data Pointer Assignments
[DREAD]	Read Numeric Keypad on RP240
[DREADF]	Read Function Key on RP240
DREADI	RP240 Data Read Immediate Mode
DRES	Drive Resolution
DRFLVL	Drive Fault Level
DRIVE	Drive Shutdown
DVAR	Display Variable on RP240 Display
DWRITE" "	Write Text to the RP240 Display
E	RS-232C Enable
ECHO	Echo Enable
ELSE	Else Condition of IF Statement
EMOVDB	Encoder Move Deadband Enable
ENC	Encoder / Motor Step Mode
END	Program/Subroutine End
EOL	End of Line Terminating Characters
EOT	End of Transmission Characters
EPM	Position Maintenance Mode Enable
EPMDB	Position Maintenance Deadband
EPMG	Position Maintenance Gain Factor
EPMV	Position Maintenance Maximum Velocity
[ER]	Error Value
ERASE	Erase all Programs/Subroutines
ERES	Encoder Resolution
ERRBAD	Bad Prompt
ERRDEF	Program Definition Prompt
ERRLVL	Error Detection Level
ERROK	Good Prompt
ERROR	Error Program Enable
ERRORP	Error Program
ESDB	Encoder Stall Backlash Deadband
ESK	Kill on Stall Enable
ESTALL	Stall Detect Enable
FR	Feedrate Override Enable
FRA	Feedrate Override Acceleration
FRH	Feedrate Override Analog Input to use when Channel Select High
FRL	Feedrate Override Analog Input to use when Channel Select Low
FRPER	Feedrate Override Percentage
GO	Initiate Motion
GOL	Initiate Linear Interpolated Motion
GOSUB	Execute a Subroutine with Return
GOTO	Execute a Subroutine without Return

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

Command Name	Command Description
[h]	Hexadecimal Identifier
HALT	Terminate Program Execution
HELP	Applications Help
HOM	Go Home
HOMA	Home Acceleration
HOMAD	Home Deceleration
HOMBAC	Home Backup Enable
HOMDF	Home Direction Final
HOMEDG	Home Reference Edge
HOMLVL	Home Active Level
HOMV	Home Velocity
HOMVF	Home Velocity Final
HOMZ	Home to Z-channel Enable
IF( )	If Statement
[IN]	Input Status
INDAX	Participating Axes
INDEB	Input Debounce Time
INDUSE	Enable/Disable User Status
INDUST	User Status
INEN	Input Enable
INFEN	Input Function Enable/Disable
INFNC	Input Function
INLVL	Input Active Level
[INO]	Other Input Status
INPLC	Establish PLC Data Inputs
INSELP	Select Program Enable
INSTW	Establish Thumbwheel Data Inputs
JOG	Jog Mode Enable
JOGA	Jog Acceleration
JOGAD	Jog Deceleration
JOGVH	Jog Velocity High
JOGVL	Jog Velocity Low
JOY	Joystick Mode Enable
JOYA	Joystick Acceleration
JOYAD	Joystick Deceleration
JOYAXH	Joystick Analog Input High
JOYAXL	Joystick Analog Input Low
JOYCDB	Joystick Center Deadband
JOYCTR	Joystick Center
JOYEDB	Joystick End Deadband
JOYVH	Joystick Velocity High
JOYVL	Joystick Velocity Low
JOYZ	Joystick Zero
JUMP	Jump to a Program or Label (do not return)
K	Kill Motion
<ctrl>K	Immediate Kill
L	Loop
LH	Hard Limit Enable
LHAD	Hard Limit Deceleration
LHLVL	Hard Limit Active Level
[LIM]	Limit Status
LN	End Loop
LS	Soft Limit Enable
LSAD	Soft Limit Deceleration
LSCCW	Soft Limit CCW Range
LSCW	Soft Limit CW Range
LX	Terminate Loop
MA	Absolute / Incremental Mode Enable
MC	Preset / Continuous Mode Enable
MEMORY	Configure Memory
[MOV]	Axis Moving Status

Command Name	Command Description
NIF	End IF Statement
[NOT]	Not
NWHILE	End WHILE Statement
ONCOND	On Condition Enable
ONIN	On an Input Condition Gosub
ONP	On Program
ONUS	On a User Status Condition Gosub
ONVARA	On Variable 1 Condition Gosub
ONVARB	On Variable 2 Condition Gosub
[OR]	Or
OUT	Output State
[OUT]	Output Status
OUTALL	Multiple Output State
OUTEN	Output Enable
OUTFEN	Output Function Enable
OUTFNC	Output Function
OUTLVL	Output Active Level
OUTPLC	Establish PLC Strobe Data Outputs
OUTTW	Establish Thumbwheel Strobe Data Outputs
PA	Path Acceleration
PAB	Path Absolute
PAD	Path Deceleration
PARCM	Radius Specified CCW Arc
PARCOM	Origin Specified CCW Arc
PARCOP	Origin Specified CW Arc
PARCP	Radius Specified CW Arc
PAXES	Set Contouring Axes
[PCE]	Position of Captured Encoder
PCOMP	Path Compile
[PE]	Position of Encoder
[PI]	Pi (π)
PL	Define Path Local Mode
PLC	Define Path Local Coordinates
PLIN	Move in a Line
[PM]	Position of Motor
POUT	Path Outputs
PRTOL	Path Radius Tolerance
PRUN	Run a Path
PS	Pause Program Execution
PSCLA	Path Acceleration Scale Factor
PSCLD	Path Distance Scale Factor
PSCLV	Path Velocity Scale Factor
PSET	Establish Absolute Position
PUCOMP	Path Uncompile
PULSE	Set Pulse Width
PV	Path Velocity
PWC	Path Work Coordinate
RADIAN	Radian Enable
[READ]	Read a Value from PC
REPEAT	Repeat Statement
RESET	Reset 6200
RUN	Execute a Program/Subroutine
S	Stop Motion
SCALE	Enable/Disable Scale Factors
SCLA	Accel / Decel Scale Factor
SCLD	Distance Scale Factor
SCLV	Velocity Scale Factor
[SIN( )]	Sine
[SQRT( )]	Square Root
[SS]	System Status
SSV	Start / Stop Velocity
STARTP	Set Power-Up Program
STEP	Program Step Mode Enable

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

Command Name	Command Description
T	Time Delay
[TAN( )]	Tangent
TANV	Transfer Analog Input Value
TAS	Transfer Axis Status
TCMDER	Transfer Command Error
TCNT	Transfer Counter
TDIR	Transfer Directory
TDPTR	Transfer Data Pointer Status
TER	Transfer Error Status
TEST	Motion Test Sequence
TEX	Transfer Program Execution Status
[TIM]	Current Timer Value
TIMST	Start Timer
TIMSTP	Stop Timer
TIN	Transfer Input Status
TINO	Transfer Other Input Status
TLABEL	Transfer Labels
TLIM	Transfer Limit Status
TMEM	Transfer Memory Usage
TOUT	Transfer Output State
TPCE	Transfer Position of Captured Encoder
TPE	Transfer Position of Encoder
TPER	Transfer Position Error
TPM	Transfer Position of Motor
TPROG	Transfer Program
TRACE	Program Trace Mode Enable
TRANS	Translation Mode Enable
TREV	Transfer Revision Level
TSS	Transfer System Status
TSTAT	Transfer Indexer Status
TTIM	Transfer Time
TUS	Transfer User Status
TVEL	Transfer Present Velocity
[TW]	Thumbwheel Data Read

Command Name	Command Description
UNTIL( )	Until Part of REPEAT Statement
[US]	User Status
V	Velocity
[V]	Velocity Assignment
VAR	Variable
VARB	Binary Variable
VARS	String Variable
VCVT( )	Variable Type Conversion
[VEL]	Current Velocity
WAIT( )	Wait for a Specific Condition
WHILE( )	While a Condition is True
WRITE" "	Transmit a String to the PC
WRVAR	Transmit a Variable
WRVARB	Transmit a Binary Variable
WRVARS	Transmit a String Variable

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

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

Index

4-lead motor wiring 91
6-lead motor wiring 91
6000 DOS Support Disk 1, 26, 28
6000 Series Software Reference Guide ii, 27
6201
  description 1
  features 2
  general system specifications 83
  ship kit 5
8-lead motor wiring 92

A

absolute position
  absolute positioning mode 36
  absolute zero position 36
  report 36
  reset to zero after homing 33
AC input voltage 8, 83
acceleration
  change on the fly 38
  range 83
  scaling 30
accuracy 28, 83
active high/active low conventions 14
adaptor
  VM50 18
address 79
  daisy-chain 79
  DIP switch settings 88
airborne contaminants 12
alignment 24
allocating memory 29
ambient temperature 11
analog inputs 19, 20, 53
  overriding 55
application examples 41
application requirements 26
arc segments 65, 68
assumptions
  skills required to use the 6200 i
auto-baud 8, 98
AUX connector 6, 86
auxiliary programmable outputs 2, 17
AWG-motor wires 21
axes
  number of 29
axes select input 20

B

backlash deadband 39
baud rate 8, 87, 98
BBS – bulletin board 27
BCD program select input 45
BCD weights 45
bench test 6
binary variables 61, 63
bipolar current 93
bitwise operations (and, or, not, etc.) 63
block diagram
  system hardware 2
Boolean operations 63
branching 2
buffered commands 37
bulletin board service (BBS) 27

C

cables
  color code
    encoder 16
    motor 7
  custom
    drive 21
    encoder 21
    I/O 21
  extending 21
  RS-232C 98
  shielding 21
  wire gauge 21
CAD-to-Motion software 28
capture encoder position 47, 48
capture motor position 47
CCW end-of-travel limits 14, 32, 86
CCW jog input (INFNCi-aK) 50
CCW rotation ii
center joystick position 54
center specified arcs 69
circles 69
circuit drawings 84
  encoder input 85
  joystick/analog input 87
  limit inputs 86
  programmable I/O 86
  trigger input 86
clockwise rotation ii
closed-loop operation 38
color code
  encoder 16
  motor cable 7
command
  after stop 47
  buffer 37, 47
    control execution of 38, 47
  list,alphabetical 101
communication
  daisy-chaining 79
  parameters 8
compiling the path 71
CompuCAM™ 1, 28
computer-to-terminal conversion 7
conditional branching 2
conduit 12, 21
configuration
  address 79
  DIP switch settings 88
  indexer 28
  inputs 44
  jogging 50
  jumper 87
  outputs 42
  system ii
  thumbwheel 52
connections 19
  AC power cable 8
  cable extensions 21
  connector part numbers 84
  daisy-chain 78
  electronic I/O devices 53
  encoder 15, 85
  end-of-travel limits 14
  home limits 15
  installation process overview ii
  joystick 19, 87
  motor 7
  PLC 53
  prewired 6
  programmable I/O 17
  RP240 19, 87
  RS-232C 7, 78
  testing 22, 23
  thumbwheels 51
  TM8 thumbwheel module 51
  triggers 18
  VM50 adaptor 18
connector part numbers 84
contaminants 12
continue (!C) 47
continue execution on
  pause/resume (COMEXR) 47
  stop (COMEXS) 47
continue input 47
continuous command execution mode 20, 38
continuous positioning mode 35, 37
contouring 65
  accel, decel, velocity 66
  affected by drive resolution 66
  affected by pulse width 28
  arcs 68, 69
    deviation 83
  circles 69
  compiling a path 71
  defining a path 65
  distance scaling 31
  effected by pulse width 66
  endpoints 67
  executing a path 71
  lines 68
  local coordinates 67
  memory allocation 29
  outputs along path 70
  participating axes 66
  programming errors 71
  segment boundary 70
  stall 70
  velocity scaling 30
  work coordinates 67
conventions
  active high/active low 14
  direction ii
counter 40
counter-clockwise rotation ii
couplings 24
Crosstalk™ 7

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

current
bipolar rating 93
unipolar rating 93
current selection
6201 series motors 6
non-Compumotor motors 93
CW end-of-travel limits 14, 32, 86
CW jog input (INFNCi-aJ) 50
CW rotation ii

D

daisy-chain 78, 87
including RP240 81
data bits 8
data elements 73
data program 73
deadband
joystick 54
position maintenance 39
stall 39
debounce time for inputs 44
program select input 46
debug tools 59, 96
deceleration
scaling 30
default connections 6
define usable voltage 54
detecting a stall 39, 43
device address (see address)
dimensions 12
6201 12
6201 series motors 90
DIP switch settings
address settings 78, 87
baud rate 87
motor current
6201 motors 6
non-Compumotor motors 93
direction conventions ii
discrete switches 18
distance
fractional step truncation 31
registration 49
scaling (SCLD) 32
DOS Support Disk 28
drive
connections 84
custom cabling 21
fault input 44
fault level 29
offsets 25
resolution 30, 39
effect on contouring 66

E

earth ground connection 8, 12
electrical codes ii
electrical connections 12
electrical noise 12, 21, 96
electro-static discharge (ESD) 87
electronic sensors 18
electronics concepts i
encoder
cables 21
compatibility 15
Compumotor encoder cable colors 16
connections 15, 85
testing 23
counter 40
custom cabling 21
differential outputs 15
encoder step mode 39
feedback 38
pin outs 16
position capture 47
resolution 39
set-up example 40
single-ended outputs 15
Z channel 15
end point
contouring 66
linear interpolation 64
end-of-travel limits 14, 32, 37, 86
status 59
testing 23
error level 1 on power up 79
error messages 96
error program 16
ESD 87
expanded memory 29
extending cables 21

F

factory settings
default connections 6
DIP switches 87
fast trigger inputs (see trigger inputs)
fault output 44
features 2
feedrate override 19, 55
five VDC output 16
full duplex 8
full-scale velocity 54

G

gauge – motor wires 21
gauging motor resonance 25
general specifications 83
GRA (goods returned authorization 99
grounding 8, 12

H

hard limits (see end-of-travel limits)
hardware feedrate override 56
hardware specifications 83
heat 11
helpful resources (publications) ii
homing 33
home limit input 23, 86
connection 15
polarity 15
status 59
reference position 15
host computer operation 40, 60
humidity 11

I

I/O cabling 21
IBM operations guide ii
immediate commands 37
immediate stop 37
incremental encoders (see encoder)
incremental positioning mode 35
inertia – 6201 motors 89
input voltage 83
inputs
analog 19, 53, 55, 87
overriding 55
drive 84
encoder 21, 85
end-of-travel limits 14, 32, 86
home limits 15, 33, 86
jogging 50
joystick 19, 53, 87
kill 37, 46
no function 45
one-to-one program select 50
pause/continue 47
PLC 53
program select 46, 50
programmable 17, 41, 85
changing from sinking to sourcing 88
changing from sourcing to sinking current 87
debounce time 45
function assignments 44
pin outs 85
polarity 42
status 59
pulse cut 16, 86
specifications 84
status 44
stop 37, 47
thumbwheel 51
triggers 86
debounce time 44
interrupt/registration 47
user fault 47
installation
encoder connections 15
joystick connections 19
limit connections 15
mounting 12
PLC connections 53
precautions 11
procedure 12
programmable I/O connections 17
RP240 connections 19
test/verification 22
thumbwheels 51
trigger connections 18
installation process overview ii
interface
host computer 60
options 40
RP240 57
interpolation
circular/contouring 65
linear 64

J

jogging 50
input 50
RP240 59
speed select high/low (INFNCi-aL) 50
velocity
high (JOGVH) 50
low (JOGVL) 50
status 59

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

joystick
auxiliary input 20
center deadband 54
center voltage 54
connections 20, 87
testing 23
control 53
inputs 19
status 59
interface 41
release input 20
select input 54
teachmode example 76
trigger input 20
velocity resolution 54
JS6000 joystick 53
jumper settings 87

K

kill input 37, 46
kill-on-stall 39

L

LEDs 9, 97
limits
end-of-travel 32
connections 14, 86
encountered 43
used as basis to activate output 43
home 33
connections 15
status 59
line segments 65, 68
linear interpolation 64
distance scaling 31
end point 64
velocity scaling 30
local coordinate system 67

M

master/slave daisy-chain 78, 80
mathematical operations 62
mechanical factors 28
memory
allocation 29
contouring 65
expanded 29
menus
RP240 58
mid-range instability 24
misalignment 24
Motion Architect® 1, 26, 28
motion control concepts i, 28
motion profiles 28
motor 24
cables 21
connections 7, 85
couplings 24
current selection
6201 series motors 6
non-Compumotor motors 93
dimensions 90
motor step mode 39
mounting 22
position capture 47
resolution 39
rotor inertia 89
speed/torque curves 89
using non-Compumotor motors 91
mounting 12
6201 12
motor 22
moving/not moving 43

N

National Electrical Code Handbook ii
no function input 45
noise
electrical 12, 21, 96
suppression
limits & triggers 21
on analog inputs 19
non-Compumotor motors 91
normal mode 36
number of axes 29
numeric variables 61

O

offsets 25
on-the-fly changes 38
one-to-one program select input 50
operating system 27
operations with binary variables 63
operator panel 19, 57
optional DIP switch & jumper settings 87
oscillation 25
oscilloscope 25
outputs 17, 53, 86
+5V connection 16
configuration 42
contouring path 70
fault output 44
limit encountered 43
moving/not moving 43
power 83
program in progress 43
programmable 17, 41, 53, 85
function assignments 42
pin outs 85
polarity 42
specifications 84
status 59
pull-up 17, 53
stall indicator 43
status 42
voltage 83
override 55
overtemperature protection 84

P-Q

P-CUT 16
status 59
panel – operator panel 57
parametric oscillations 24
parity 8
part numbers 5, 84
partitioning memory 29
password
RP240 59
paths (see contouring)
pause 47
pause/continute input 47
PC-Talk™ 7
peripheral system components ii
phase A offset 25
phase B offset 25
pin outs 16, 84
6201 motor cables 21, 85
auxiliary connector 86
encoder 16, 85
joystick 19, 87
limits 86
programmable I/O 17, 85
RP240 87
triggers 86
PLC 17
interface 40, 41, 53
outputs 18
point-to-point move 36
polarity
limits
end-of-travel 14
home 15
programmable inputs & outputs 42
trigger inputs 18
position
absolute 36
reset to zero after homing 33
capture 2
encoder 47, 48
for registration 48
motor 47, 48
feedback 38
home 15
incremental 36
maintenance 39
positioning modes 35
range 83
status 59
potentiometer 19
joystick 54
power cable connections 8
power dump protection 84
power, output 83
power-up 9
user program (STARTP) 58
pre-wired connections 6
precautions
installation 11
mounting 12
preset positioning mode 36
problems & solutions 96
Procomm™ 7
program editing 28
in Motion Architect 28
programmable I/O 17, 41
application example 41
binary variable 42
change from sourcing to sinking current 87
conditional branching & looping 42
configuration 42, 44
inputs 53, 85
outputs 43, 53, 85
pin outs 85
polarity 42
program interrupt 42
screw terminal connections 18
testing 23

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

programming
6000 DOS support disk 28
contouring errors 71
contouring examples 72
debugging 96
via RP240 59
edit programs via RP240 59
error messages 96
memory allocation 29
Motion Architect® 28
problems 97
program in progress 43
program select
BDC 45
debounce time 46
one-to-one 50
run programs via RP240 59
skills i
protective circuits 84
pulse cut 16, 86
status 59
testing 23
pulse width
effect on contouring 66
effects max. velocity 31

R

rack mounting 13
radius specified arcs 68
radius tolerance specifications 68
reading inputs and outputs 41
thumbwheel data 51
reference publications ii
registers
hardware 40
registration 37, 48
related publications ii
relative humidity 11
remote operator panel
connections 19
interface 57
repeatability 28, 83
reset 97
check for address & auto baud 87
via RP240 60
resolution
drive 39
encoder 39
motor 39
resonance speed 25
return material authorizaiton (RMA) 99
return procedure 99
RMA number 99
rotor inertia 89
RP240 57, 96
application example 41
connections 19, 87
testing 23
in daisy chain 81
menu structure 58
password 58
RS-232C communication 8
connections 7
daisy-chaining 78
disable handshaking 98
troubleshooting 98

S

safety 11
safety stops 14
scaling 29, 36
segment boundary 70
set-up test procedure 9
shielding 12, 21
shift
left to right (>>) 64
right to left (<<) 64
shipment
inspection 5
ship kit 5
short circuit protection 84
single-ended encoders 15
single-step mode 2, 59
sinking to sourcing current 88
smoothness 25
software feedrate override 56
sourcing to sinking current 88
specifications 83
6201 83
motor 89
speed/torque curves 89
stall backlash deadband 39
stall detection 39, 43
stand-alone operation 40, 41
status
axis 59, 61
error 61
inputs 44, 59, 61
joystick inputs 20, 59
LED 83, 97
limits 59
outputs 42, 59, 61
position 59
system 59, 61
step & direction 84
stop
effect on program execution 47
input 37, 47
stop bits 8
streaming mode
affected by pulse width 28
string variables 61
support software 28
system configuration 28
system specifications 83

T

teach mode 73
temperature
operating 11
overtemperature limit 84
terminal connections 8
terminal emulation 28
6000 DOS support disk 7, 28
Motion Architect® 7, 28
test
bench test procedure 9
program 60, 96
system installation 22
thumbwheels 17
application example 41
connections 52
TM8 module 51
toll free 800 phone number 99
torque/speed curves 89
trace mode 2, 59
travel limit 15
trigger inputs 18, 37
connections 18
interrupt function 47
position capture 48
trigonometric operations 62
troubleshooting
common problems & solutions 96
diagnostic LEDs 96
error messages 96
methods 95
RS-232C problems 98
test program 96
TTL-compatible voltage levels 84
tuning 26
motor resonance 26
potentiometers 25
procedure 25

U

unipolar current 93
user fault input 44, 47
user interface options 40
user programs (see programming)

V

variables
binary 61
conversion between binary & numeric 61
numeric 61
teach mode 73
string 61
velocity
change on the fly 38
full scale 54
joystick velocity select input 20
range 83
resolution 53
scaling 30
verification – installation test 22
viscous damper 24
VM50 adaptor 18
voltage – define usable 54
voltage, output 83

W-Z

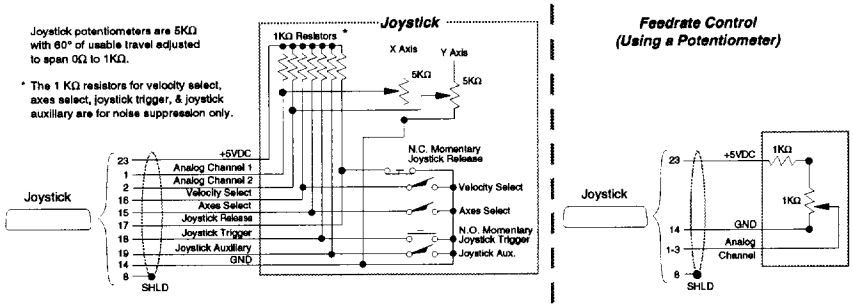
wire gauge 21
work coordinate system 67, 72
X-Y linear interpolation 64
XON/XOFF 8
Z channel output 15

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

Connections

JOYSTICK

Connections – see page 19
Joystick Operations – see pages 53-56
Internal Circuit – see page 87



Axes Select	Use to alternately control axis 1 or 2.
Velocity Select	Input to select high or low velocity range (JOYVH or JOYVL).
Joystick Release	Input to release the 6201 from joystick mode (JOY).
Joystick Trigger	Active-low input to enter the joystick mode. Monitor with INO command to control program flow.
Joystick Auxiliary	Active-low input to teach positions to a program. Monitor with INO command to control program flow.

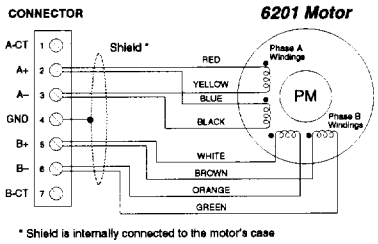
AUX

Pre-Wired Connections – see page 6
RS-232C Connections and Set-up – see page 8
Trigger (TRG-A & TRG-B) Connections – see page 18

Pin	Name	Description
1	Rx	Receive input for RS-232C interface.
2	Tx	Transmit output for RS-232C interface.
3	GND	Isolated ground for RS-232C interface.
4	SHLD	Shield – chassis ground (earth).
5	+5V	Connect to OUT-P to power the 26 programmable outputs. 1.5A limit.
6	OUT-P	Connect to +5V pin to make the 24 programmable outputs TTL compatible. Connect to other voltages (max. = 24V) for compatibility with other signal levels.
7	TRG-A	Trigger input A: general-purpose and position capture functions (INFNC).
8	TRG-B	Same function as trigger input A above.
9	GND	Isolated ground.
10	OUT-A	Function and circuit identical to other 24 programmable outputs.
11	OUT-B	Same function as auxiliary programmable output A above.
12	GND	Isolated ground.
13	SHLD	Shield – chassis ground (earth).
14	P-CUT	When un-grounded, step pulses stop.

MOTOR 1 & 2

Compumotor Motor Connections – see pages 6-7
Non-Compumotor Motor Connections – see page 93
6201-to-Motor Cables – see page 21
Torque Speed Curves – see page 89



* Shield is internally connected to the motor's case

RP240

Connections – see page 19
Operations and Menus – see pages 57-60

PROGRAMMABLE INPUTS & OUTPUTS

Connections – see page 17
Programming – see pages 41-51
Thumbwheel Connections/Operation – see page 51
PLC Connection/Operation – see page 53
Internal Circuit – see page 86

Pin	Output Connector	Input Connector
49	+5VDC	+5VDC
47	Output #1 (LSB)	Input #1 (LSB)
45	Output #2	Input #2
43	Output #3	Input #3
41	Output #4	Input #4
39	Output #5	Input #5
37	Output #6	Input #6
35	Output #7	Input #7
33	Output #8	Input #8
31	Output #9	Input #9
29	Output #10	Input #10
27	Output #11	Input #11
25	Output #12	Input #12
23	Output #13	Input #13
21	Output #14	Input #14
19	Output #15	Input #15
17	Output #16	Input #16
15	Output #17	Input #17
13	Output #18	Input #18
11	Output #19	Input #19
9	Output #20	Input #20
7	Output #21	Input #21
5	Output #22	Input #22
3	Output #23	Input #23
1	Output #24 (MSB)	Input #24 (MSB)

All even numbered pins are connected to logic ground.

Inputs are internally pulled up to +5V. To change from sourcing to sinking, change internal jumper JU2 *(see pages 87-88 for details)*

LIMITS

Connections – see pages 14-15
End-of-Travel Limit Operations – see pages 14 & 32
Homing Operations – see pages 14 & 33-35
Internal Circuit – see page 86

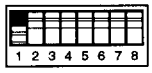
ENCODER 1 & 2

Connections – see page 15
Internal Encoder Circuit – see page 85
6201-to-Encoder Cables – see page 21
Encoder-based Operations – see pages 38-40

Motor Current

Non-Compumotor Motor Current Selection – see page 93

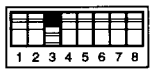
6201-57-51



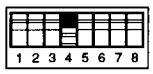
6201-57-83



6201-57-102



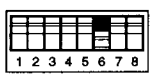
6201-83-62



6201-83-93



6201-83-135



I/O Specifications

All I/O are optically isolated.

RS-232C SET-UP	9600 baud rate, 8 data bits, 1 stop bit, no parity bit, full duplex.
INPUTS	
Home and CW/CCW Limits, Pulse Cutoff, Joystick Inputs	TTL-compatible*; internal 6.8 KΩ pull-ups to 5V; voltage range is 0-24V.
Incremental Encoder	Use two-phase quadrature encoders with differential (recommended) or single-ended outputs (+5VDC TTL-compatible*). Maximum frequency = 1.6 MHz. Min. between transitions = 625 ns.
24 Programmable	TTL-compatible* with internal 6.8 KΩ pull-up sourcing current (or change internal jumper JU2 to sink current). Voltage range = 0V–24V. Plug compatible with OPTO-22™ equipment.
Triggers (TRG-A & TRG-B)	Optically isolated, TTL-compatible* with internal 6.8 KΩ pull-up to +5VDC.
Analog (Joystick CH 1 & 2)	Voltage range = 0–2.5VDC, 8-bit A/D converter. **Input voltage must not exceed 5V.**
OUTPUTS	
26 Programmable (includes OUT-A & OUT-B)	TTL-Compatible* open collector output. Can be pulled up by connecting OUT-P to +5V or to user-supplied voltage of up to 24V. Max. voltage in OFF state (not sinking current) = 24V; max. current in ON state (sinking) = 30mA. Plug compatible with OPTO-22™ equipment.

** TTL-compatible voltage levels: Low 0.4V, High 2.4V.*

Troubleshooting Tips

Detailed Troubleshooting – see pages 95-98

- **P-CUT** terminal must be grounded to **GND** terminal to allow motion; check status with TINO command.
- TAS status command reports most problem situations (see description in ***6000 Series Software Reference Guide***).
- Programmable input (INFNC) functions and drive fault detection will not be operable until you enable the input functions with the INFEN1 command.
- To help prevent electrical noise, make sure all connections are shielded at one end only.
- Error messages – see *Programming Guide* section in ***6000 Series Software Reference Guide***.
- RS-232C communication problems – see page 98.
- Tuning the internal drives – see page 25.
- Scaling feature described on pages 29-32.
- LEDs:
  - STATUS: **OFF** — No AC power
  - STATUS: **RED** — Fault (must cycle power)
  - POWER: **OFF** — No AC power
  - DISABLED: **ON** — Bad motor connection
  - — Motor short circuit
  - — Overheating
  - — Shutdown (issue DRIVE11 to enable)
  - — Internal damage

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