ControlLogix™
Motion Module

(Cat.No. 1756-M02AE)

Setup and Configuration Manual
Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted publication, in whole or part, without written permission of Rockwell Automation, is prohibited.

Throughout this manual we use notes to make you aware of safety considerations:

**ATTENTION**

Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

**IMPORTANT**

Identifies information that is critical for successful application and understanding of the product.
If this product has the CE mark it is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

**EMC Directive**

This product is tested to meet the Council Directive 89/336/EC Electromagnetic Compatibility (EMC) by applying the following standards, in whole or in part, documented in a technical construction file:

- EN 50081-2 EMC — Generic Emission Standard, Part 2 — Industrial Environment
- EN 50082-2 EMC — Generic Immunity Standard, Part 2 — Industrial Environment

This product is intended for use in an industrial environment.

**Low Voltage Directive**

This product is tested to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 - Equipment Requirements and Tests. For specific information required by EN 61131-2, see the appropriate sections in this publication, as well as the Allen-Bradley publication Industrial Automation Wiring and Grounding Guidelines For Noise Immunity, publication 1770-4.1.

This equipment is classified as open equipment and must be mounted in an enclosure during operation to provide safety protection.
# Table of Contents

## Preface

### About This Manual
- Using This Manual
- Who Should Use This Manual
- The Purpose of This Manual
- Related Documentation
- Rockwell Automation Support
  - Local Product Support
  - Technical Product Assistance
- On the Web

## Chapter 1

### The ControlLogix Motion Control System
- ControlLogix Motion Control
- Components of the ControlLogix Motion System
  - The Logix5550 Controller
  - The Analog/Encoder Servo Module (1756-MO2AE)
  - The 8 Axis SERCOS interface Module (1756-MO8SE)
- RSLogix5000 Programming Software
- Developing a Motion Control Application Program
  - Application Program Development
  - The MOTION_INSTRUCTION Tag
  - Understanding Motion Status and Configuration Parameters
- Modifying Motion Configuration Parameters
- Handling Motion Faults

## Chapter 2

### Adding and Configuring Your 1756-M02AE Motion Module
- Adding the 1756-M02AE Module
  - New Module
    - Type
    - Major Revision
    - Type (list box)
    - Description (list box)
    - Show:
    - Select All
    - Clear All
  - Editing Your Motion Module Settings
Chapter 3

Configuring the 1756-M08SE Module

Adding the 1756-M08SE ......................... 3-1
1756-M08SE 8 Axis Motion Module Overview .. 3-4
1756-M08SE 8 Axis Properties .................. 3-4
### Table of Contents

- General Tab ........................................ 3-4
- Type ........................................ 3-5
- Vendor ........................................... 3-5
- Name ........................................ 3-5
- Description ....................................... 3-5
- Slot ........................................ 3-5
- Revision ........................................ 3-6
- Electronic Keying. .................................. 3-6
- SERCOS Interface Tab .................................. 3-7
  - Data Rate ........................................ 3-8
  - Cycle Time ....................................... 3-8
  - Transmit Power .................................. 3-8
- SERCOS Interface Info Tab .......................... 3-8
  - Ring Comm. Phase. ................................ 3-9
  - Fault Type ..................................... 3-9
  - Refresh ........................................ 3-9

### Chapter 4 Editing Your Controller Properties

- Controller Properties .............................. 4-1
  - General Tab Overview. .......................... 4-1
  - Vendor ......................................... 4-1
  - Type ........................................... 4-1
  - Name ........................................ 4-2
  - Description .................................... 4-2
  - Chassis Type .................................. 4-2
  - Slot ........................................ 4-2
  - Revision ..................................... 4-2
  - Change Type .................................. 4-3
- Change Controller Type Dialog Box Overview .... 4-3
  - Select a processor to change to ................. 4-3
- Serial Port Tab Overview ........................... 4-4
  - Mode ......................................... 4-4
  - Baud Rate ..................................... 4-4
  - Data Bits ....................................... 4-4
  - Parity ........................................ 4-4
  - Stop Bits ..................................... 4-5
  - Control Line ................................... 4-5
  - Continuous Carrier............................... 4-5
  - RTS Send Delay ................................ 4-5
  - RTS Off Delay ................................ 4-5
- System Protocol Tab Overview ....................... 4-5
  - Common Parameters .............................. 4-6
  - Protocol ..................................... 4-6
Station Address ............................................. 4-6
Error Detection ............................................. 4-7
Enable Duplicate Detection ............................... 4-7
ACK Timeout .................................................. 4-7
DF1 Point to Point Parameters ............................ 4-7
DF1 Inquiries ............................................... 4-7
NAK Receive Limit ......................................... 4-7
DF1 Slave Parameters ...................................... 4-7
Transmit Retries ............................................ 4-7
Slave Poll Timeout ......................................... 4-7
DF1 Master Parameters .................................... 4-8
Transmit Retries ............................................ 4-8
Reply Message Wait ........................................ 4-8
Polling Mode ................................................ 4-8
Master Transmit ............................................ 4-8
Normal Poll Node Tag ...................................... 4-8
Normal Poll Group Size .................................... 4-9
Priority Poll Node Tag ..................................... 4-9
Active Station Tag ......................................... 4-9
User Protocol Tab Overview ............................... 4-9
Protocol ...................................................... 4-10
Buffer Size .................................................. 4-10
Termination Character 1 and 2 ........................... 4-10
Append Character 1 and 2 .................................. 4-11
XON/OFF ..................................................... 4-11
Echo Mode ................................................... 4-11
Delete Mode .................................................. 4-12
Major Faults Tab Overview ............................... 4-12
Number of Major Faults Since Last Cleared ............ 4-12
Recent Faults ............................................... 4-13
Clear Majors ............................................... 4-13
Minor Faults Tab Overview ............................... 4-13
Number of Minor Faults Since Last Cleared ............ 4-13
Recent Faults ............................................... 4-14
Clear Minors ............................................... 4-14
Fault Bits ..................................................... 4-14
Date/Time Tab Overview .................................. 4-14
Date .......................................................... 4-15
Time .......................................................... 4-15
Set ........................................................... 4-15
Coordinated System Time master ....................... 4-15
Status ......................................................... 4-15
Advanced Tab Overview .................................. 4-16
Memory Used ............................................... 4-16
Memory Unused ............................................. 4-16
Memory Total ............................................... 4-16
Controller Fault Handler ........................................ 4-17
Power-Up Handler ............................................. 4-17
System Overhead Time Slice ................................. 4-17
File Tab Overview ............................................ 4-17
Name ............................................................... 4-18
Path ............................................................... 4-18
Created ........................................................... 4-18
Edited ............................................................. 4-18

Chapter 5

Naming & Configuring Your Motion Axis

Naming an Axis .................................................. 5-1
Entering Tag Information ...................................... 5-2
Common Parameters ......................................... 5-3
Name ............................................................... 5-3
Description ....................................................... 5-3
Tag Type ........................................................... 5-3
Data Type .......................................................... 5-3
Configuring a Motion Axis .................................... 5-4
General Tab ....................................................... 5-5
Axis Configuration ............................................. 5-5
Assigned Motion Group ....................................... 5-5
Ellipsis (... ) button .............................................. 5-6
New Group button .............................................. 5-6
Module ............................................................. 5-6
Module Type ....................................................... 5-6
Channel ............................................................. 5-6
New Group Button .............................................. 5-7
Assigned Motion Group ....................................... 5-8
Axes Assigned ................................................... 5-8
Coarse Rate ......................................................... 5-8
Servo Update Period .......................................... 5-8
General Fault Type ............................................. 5-8
Units Tab .......................................................... 5-12
Position Units ..................................................... 5-13
Average Velocity Timebase .................................... 5-13
Conversion Tab .................................................. 5-14
Positioning Mode ................................................ 5-14
Conversion Constant .......................................... 5-15
Position Unwind .................................................. 5-15
Servo Tab .......................................................... 5-16
External Drive Configuration ............................... 5-17
Enable Drive Fault Input ..................................... 5-17
Drive Fault Input ............................................... 5-17
Real Time Axis Information ........................................ 5-17
Homing Tab ......................................................... 5-18
  Mode ......................................................... 5-19
  Position ...................................................... 5-19
  Offset ......................................................... 5-19
  Sequence ..................................................... 5-19
  Limit Switch .................................................. 5-20
  Direction .................................................... 5-20
  Speed ......................................................... 5-21
  Return Speed .................................................. 5-21
  Homing Configurations ........................................ 5-22
Homing Tab ......................................................... 5-22
  Home Position ............................................... 5-22
  Mode ......................................................... 5-23
  Sequence ..................................................... 5-23
  Homing Direction ............................................ 5-24
  Homing Speed .................................................. 5-24
  Home Return Speed ............................................ 5-24
Hookup Tab ......................................................... 5-26
  Test Increment ............................................... 5-26
  Feedback Polarity ............................................ 5-27
  Output Polarity ............................................... 5-27
  Test Marker .................................................. 5-28
  Test Feedback ............................................... 5-28
  Test Output & Feedback ...................................... 5-28
Tune Tab .......................................................... 5-29
  Tuning Travel Limit .......................................... 5-29
  Tuning Speed ................................................ 5-29
  Tuning Direction ............................................. 5-29
  Damping Factor ............................................... 5-30
  Tune .......................................................... 5-30
  Start Tuning .................................................. 5-30
  Tune Bandwidth Dialog ....................................... 5-30
  Bandwidth .................................................... 5-30
Dynamics Tab ..................................................... 5-32
  Maximum Velocity ............................................. 5-33
  Maximum Acceleration ....................................... 5-33
  Maximum Deceleration ....................................... 5-34
  Program Stop Action ......................................... 5-34
  Manual Tune .................................................. 5-35
Gains Tab .......................................................... 5-35
  Velocity Feedforward ....................................... 5-36
  Acceleration Feedforward ................................... 5-36
  Proportional (Position) Gain ................................ 5-37
  Integral (Position) Gain ..................................... 5-38
  Proportional (Velocity) Gain ................................. 5-38
Integral (Velocity) Gain ........................................ 5-39
Integrator Hold .................................................. 5-39
Manual Tune ...................................................... 5-39
Output Tab ......................................................... 5-40
Velocity Scaling ................................................... 5-41
Torque Scaling .................................................... 5-42
Enable Filter Bandwidth ........................................ 5-42
Filter Bandwidth .................................................. 5-42
Manual Tune ...................................................... 5-43
Limits Tab .......................................................... 5-43
Soft Travel Limits ................................................ 5-44
Maximum Positive ............................................... 5-44
Maximum Negative .............................................. 5-45
Position Error Tolerance ........................................ 5-45
Position Lock Tolerance ......................................... 5-45
Output Limit ...................................................... 5-45
Manual Tune ...................................................... 5-46
Offset Tab .......................................................... 5-46
Friction Compensation ......................................... 5-47
Velocity Offset .................................................... 5-48
Torque Offset ..................................................... 5-48
Output Offset ..................................................... 5-48
Manual Tune ...................................................... 5-49
Fault Actions Tab ................................................ 5-49
Drive Fault ....................................................... 5-51
Feedback Noise .................................................... 5-51
Feedback Loss ..................................................... 5-51
Position Error ..................................................... 5-51
Soft Overtravel .................................................... 5-51
Tag Tab .............................................................. 5-52
Name ............................................................... 5-52
Description ......................................................... 5-52
Tag Type ............................................................ 5-52
Data Type .......................................................... 5-53
Scope ............................................................... 5-53
Style ............................................................... 5-53
Produce this tag for up to ..................................... 5-53
Base Tag ............................................................ 5-53
Manual Tune ...................................................... 5-53
Assigning Additional Motion Axes ............................ 5-54
Developing a Motion Application Program .................. 5-54
Understanding a Programming Example ..................... 5-55
Chapter 6

Configuring a 1394C-SJ T05/10/22-D Digital Servo Drive

1394C-SJT05/10/22-D Digital Servo Drive Overview .......................... 6-3
  General Tab ................................................................. 6-4
    Type ................................................................. 6-5
    Vendor ............................................................... 6-5
    Name ................................................................. 6-5
    Description ......................................................... 6-5
    Base Node. ......................................................... 6-5
    Revision ............................................................ 6-5
    Electronic Keying. .................................................. 6-6
  Connection Tab ....................................................... 6-7
    Requested packet Interval ....................................... 6-7
    Inhibit Module checkbox ........................................ 6-8
    Major Fault Controller Connection Fails checkbox .......... 6-9
    Module Fault ....................................................... 6-9
  Associated Axes Tab .................................................. 6-10
    Node X0 ............................................................. 6-10
    Node X1 ............................................................. 6-10
    Node X2 ............................................................. 6-10
    Node X3 ............................................................. 6-11
    New Axis button ................................................... 6-11
  Power Tab ............................................................. 6-11
    Bus Regulator ID ................................................... 6-11
  Module Info tab ....................................................... 6-12
    Identification ....................................................... 6-13
    Product Name ....................................................... 6-13
    Major/Minor Fault Status ....................................... 6-13
    Internal State Status ............................................ 6-14
    (16#xxxx) unknown ................................................ 6-14
    Configured .......................................................... 6-14
    Owned ............................................................... 6-14
    Module Identity .................................................... 6-15
    Refresh ............................................................. 6-15
    Reset Module ...................................................... 6-15

Chapter 7

Motion Instructions

Motion State Instructions .................................................. 7-1
Motion Move Instructions .................................................. 7-2
Motion Group Instructions .................................................. 7-3
Motion Event Instructions .................................................. 7-3
Understanding Motion Configuration Instructions ................. 7-4
Chapter 8  Troubleshooting

1756-M02AE Module Status Using the OK Indicator . . . . . . 8-1
1756-M02AE Module Status Using the FDBK Indicator . . . . 8-2
1756-M02AE Module Status Using the DRIVE Indicator . . . . 8-2
1756-M08SE SERCOS Communication Phase Status
  Using the CP Indicator . . . . . . . . . . . . . . . . . . . . . . . . . . 8-3
1756-M08SE Module Status Using the OK Indicator . . . . . . 8-4
1756-M08SE SERCOS Ring Status . . . . . . . . . . . . . . . . . . . . 8-4

Appendix A  Specifications and Performance

1756-M02AE Motion Module Specifications . . . . . . . . . . . . A-1
Understanding Coarse Update Period Calculations . . . . . . . A-3
  Understanding Action Timing . . . . . . . . . . . . . . . . . . . . A-3
  Using the Sample Calculations Worksheet . . . . . . . . . . A-5
  Sample Calculation . . . . . . . . . . . . . . . . . . . . . . . . . . A-6
Output Cam Timing For 1756 Controller . . . . . . . . . . . . A-8
  For the 1756-L50 Controller . . . . . . . . . . . . . . . . . . . . A-9
  For the 1756-L53 Controller . . . . . . . . . . . . . . . . . . . . A-9

Appendix B  Loop and Interconnect Diagrams

Understanding Block Diagrams . . . . . . . . . . . . . . . . . . B-1
  1756-M02AE Module With a Torque Servo Drive . . . . . . B-2
  1756-M02AE Module With a Velocity Servo Drive . . . . . B-3
Understanding Wiring Diagrams . . . . . . . . . . . . . . . . . . . B-4
  Wiring to a Servo Module RTB . . . . . . . . . . . . . . . . . . . . B-4
  Wiring to an Ultra 100 Series Drive . . . . . . . . . . . . . . . B-5
  Wiring to an Ultra 200 Series Drive . . . . . . . . . . . . . . B-6
  1398-CFLAExx Cable Diagram . . . . . . . . . . . . . . . . . . . . B-7
  Pinouts for 1398-CFLAExx Cable . . . . . . . . . . . . . . . . . . . B-7
  Wiring to a 1394 Servo Drive (in Torque Mode only) . . B-8
  The 1394-CFLAExx Cable Wiring Diagram . . . . . . . . . . B-9
  Pinouts for the 1394-CFLAE . . . . . . . . . . . . . . . . . . . . . B-9
Wiring Registration Sensors . . . . . . . . . . . . . . . . . . . . B-10
  24V Registration Sensor . . . . . . . . . . . . . . . . . . . . . . B-10
  5V Registration Sensor . . . . . . . . . . . . . . . . . . . . . . B-10
Wiring the Home Limit Switch Input . . . . . . . . . . . . . B-11
  Wiring the OK Contacts . . . . . . . . . . . . . . . . . . . . . . B-11
Appendix C  The Motion Control Structures

The AXIS Structure ............................................. C-1
   Servo Configuration Update Status Bits attributes ........ C-6
The MOTION_GROUP Structure ................................. C-7
The MOTION_INSTRUCTION Structure ....................... C-9
   Error codes (.ERR) ........................................... C-12
   Message status (.STATUS) ................................. C-13
   Execution status (.STATE) ................................. C-13
   Profile Segment (.SEGMENT) .............................. C-13
CAM Structure .................................................. C-13
CAM_PROFILE Structure ....................................... C-14

Appendix D  The Motion Attributes

Motion Instance Variables ..................................... D-1

Appendix E  Instruction Timing

Understanding Immediate Type Instructions ............... E-1
Understanding Message Type Instructions ................. E-2
Understanding Process Type Instructions ................. E-3

Appendix F  Fault Handling

Handling Motion Faults ........................................ F-1
Understanding Errors ........................................... F-1
Understanding Minor/Major Faults ......................... F-1

Index

Index ..................................................................... I-1
Using This Manual

This preface describes how to use this manual.

Who Should Use This Manual

To use this manual, you should be able to program and operate the Allen-Bradley Logix5550™ controller to efficiently use your motion control modules.

If you need more information about programming and operating the Logix5550 controller, refer to the Logix5550 Controller User Manual, publication number 1756-6.5.12.

The Purpose of This Manual

This manual describes how to configure and troubleshoot your ControlLogix motion module.

The following table shows the contents of each section in this manual:

<table>
<thead>
<tr>
<th>Section</th>
<th>Contains</th>
</tr>
</thead>
</table>
| Chapter 1
The ControlLogix Motion Control System | Information about the ControlLogix motion control system. |
| Chapter 2
Adding and Configuring Your 1756-M 02AE Motion Module | How to add and configure your 1756-M 02AE motion module using the RSLogix™ 5000 programming software. |
| Chapter 3
Adding and Configuring Your 1756-M 08SE Motion Module | How to add and configure your 1756-M 08SE motion module using the RSLogix™ 5000 programming software. |
| Chapter 4
Controller Properties | Explains how to edit the controller properties. |
| Chapter 5
Adding and Configuring Motion Axes | How to name and configure a motion axis using the RSLogix™ 5000 programming software. |
| Chapter 6
Configuring a 1394C-SJ T05/10/22-D Digital Servo Drive | How to add and configure a 1394C Digital Servo Drive. |
| Chapter 7
The Motion Instructions | Information about the 32 motion instructions provided in the RSLogix 5000 programming software. |
| Chapter 8
Troubleshooting | Information about troubleshooting your ControlLogix motion control system. |
Related Documentation

The following table lists related ControlLogix documentation:

<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Publication Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1756-IN047</td>
<td>Analog Encoder (AE) Servo Module Installation Instructions provides instructions for installing, wiring, and troubleshooting your 1756-M 02AE servo module.</td>
</tr>
<tr>
<td>1756-UM001</td>
<td>Logix5550 Controller User Manual provides information for using your Logix550 controller and its components.</td>
</tr>
<tr>
<td>1756-RM003</td>
<td>Logix5550 Controller Instruction Set Reference Manual provides descriptions of all the instructions supported by the RSLogix 5000 programming software.</td>
</tr>
<tr>
<td>1756-RM007</td>
<td>Motion Instructions Reference Set Manual provides descriptions of all of the motion instructions used in the RSLogix 5000 software.</td>
</tr>
<tr>
<td>1756-IN572</td>
<td>8 Axis SERCOS interface Module Installation Instructions provides instructions for installing, wiring, and troubleshooting your 1756-M 08SE SERCOS module.</td>
</tr>
</tbody>
</table>

For more information on the documentation, refer to the Allen-Bradley Publication Index, publication number SD499.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors, and 260 authorized systems integrators located throughout the United States. In addition,
Rockwell Automation representatives are located in every major country in the world.

**Local Product Support**

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

**Technical Product Assistance**

If you need to contact Rockwell Automation for technical assistance, please review the information in this manual. If the problem persists, call your local Rockwell Automation representative.

The Rockwell Automation Technical Support number is:

1-603-443-5419

**On the Web**

For information about Allen-Bradley, visit the following World Wide Web site:

http://www.ab.com/
Chapter 1

The ControlLogix Motion Control System

This chapter describes the ControlLogix motion control system.

ControlLogix Motion Control

The Logix5550 controller, 1756-M02AE servo module, 1756-M08SE SERCOS interface module, and RSLogix5000 programming software provide integrated motion control support.

- The Logix5550 controller contains a high-speed motion task, which executes ladder motion commands and generates position and velocity profile information. The controller sends this profile information to one or more 1756-M02AE servo modules. You can use several Logix5550 controllers in each chassis. Each controller can control up to 16 1756-M02AE servo modules.

- The 1756-M02AE servo module connects to a servo drive and closes a high-speed position and velocity loop. Each Logix5550 controller can support up to 16 1756-M02AE servo modules. Each 1756-M02AE module can control up to two axes.

- The 1756-M08SE SERCOS interface module serves as the interface between one ControlLogix processor and 1 to 8 axes operating in either position or velocity mode. The module has a programmable ring Cycle Period of 1ms or 2ms depending on the number of axes and a ring Data Rate of 4 Mbaud.

- RSLogix5000 programming software provides complete axis configuration and motion programming support.
Identifying the Components of the ControlLogix Motion System

The Logix5550 Controller

The Logix5550 controller is the main component in the ControlLogix system. It supports sequential and motion functions, and it performs all of the motion command execution and motion trajectory planner functions. You can use one or more Logix5550 controllers in each chassis, and each controller can control up to 16 motion modules.

The Logix5550 controller provides the following motion support:

- Thirty motion instructions
- A high-speed motion task, which manages motion functions and generates move profiles
- The ability to control up to 16 Analog/Encoder servo modules for a total of 32 axes

The Analog/Encoder Servo Module (1756-MO2AE)

The Analog/Encoder servo module provides an analog/quadrature encoder servo drive interface. The servo module receives configuration and move information from the Logix5550 controller and manages motor position and velocity.

The servo module supports:

- Connection capability for up to two drives
- ±10V analog outputs
- Quadrature encoder inputs
- Home limit switch inputs
- Drive fault inputs
- Drive enable outputs
- 5V or 24V position registration inputs
- 200 μs position and velocity loop updates
The 8 Axis SERCOS interface Module (1756-M08SE)

The 8 Axis SERCOS interface module (1756-M08SE) serves as a link between the ControlLogix platform and intelligent drives. The communication link between the module and the drive(s) is via IEC 1491 SErial REal-time COmmunication System (SERCOS) using fiber optic medium.

The SERCOS interface module supports:

- reliable high speed data transmission
- excellent noise immunity
- elimination of interconnect wiring
- ASA messages converted to SERCOS formatted messages

RSLogix5000 Programming Software

The RSLogix5000 programming software provides complete programming and commissioning support for the ControlLogix system. RSLogix5000 is the only programming software needed to fully configure and program ControlLogix motion control systems.

RSLogix5000 software provides the following motion support:

- Wizards for servo axis configuration including drive hookup diagnostics and auto tuning
- Ladder-based application programming including support for 30 motion commands

Developing a Motion Control Application Program

This section provides an introduction to concepts used in developing application programs for motion control. These concepts include:

- Application program development
- The MOTION_INSTRUCTION tag
- Motion status and configuration parameters
- Modifying motion configuration parameters
- Handling motion faults
Application Program Development

Developing a motion control application program involves the following:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the master coordinated system time</td>
<td>Sets one controller as the master controller. Once you complete this step, you can synchronize all the motion modules and Logix5550 controllers in your chassis.</td>
</tr>
<tr>
<td>Name and Configure an axis</td>
<td>Adds an axis to your application program</td>
</tr>
<tr>
<td>Develop a motion application program</td>
<td>Create a program for your motion control application</td>
</tr>
<tr>
<td>Add a motion module</td>
<td>Adds a motion module to your application program</td>
</tr>
<tr>
<td>Assign additional servo modules and axes</td>
<td>Adds additional modules and axes to your application program</td>
</tr>
<tr>
<td>Run hookup diagnostics and auto tuning</td>
<td>Completes hookup diagnostics and auto tuning for each axis</td>
</tr>
</tbody>
</table>

The MOTION_INSTRUCTION Tag

The controller uses the MOTION_INSTRUCTION tag (structure) to store status information during the execution of motion instructions. Every motion instruction has a motion control parameter that requires a MOTION_INSTRUCTION tag to store status information.

WARNING

Tags used for the motion control parameter of instructions should only be used once. Re-use of the motion control parameter in other instructions can cause unintended operation of the control variables.
For more information about the MOTION_INSTRUCTION tag, refer to Appendix C - The Motion Control Structures.

**Understanding Motion Status and Configuration Parameters**

You can read motion status and configuration parameters in your ladder logic program using two methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Example</th>
<th>For more information</th>
</tr>
</thead>
</table>
| Directly accessing the AXIS and MOTION_GROUP structures | • Axis faults  
• Motion status  
• Servo status | Refer to Appendix C - The Motion Control Structures          |
| Using the GSV instruction          | • Actual position  
• Command position  
• Actual velocity | Refer to the Input/Output Instructions chapter of the Logix5550 Controller Instruction Set Reference Manual, publication 1756-RM003B |

**Modifying Motion Configuration Parameters**

In your ladder logic program, you can modify motion configuration parameters using the SSV instruction. For example, you can change position loop gain, velocity loop gain, and current limits within your program.

For more information about the SSV instruction, refer to the Logix5550 Controller Instruction Set Reference Manual, publication 1756-RM003.

**Handling Motion Faults**

Two types of motion faults exist.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| Motion Instruction Errors        | • Do not impact controller operation  
• Should be corrected to optimize execution time and ensure program accuracy | A Motion Axis Move (MAM) instruction with a parameter out of range |
<table>
<thead>
<tr>
<th>Minor/Major Faults</th>
<th>Causes</th>
<th>The application exceeded the PositionErrorTolerance value</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Caused by a problem with the servo loop</td>
<td>• Can shutdown the controller if you do not correct the fault condition</td>
<td></td>
</tr>
</tbody>
</table>

For more information about handling faults, see Handling Controller Faults in the Logix5550 Controller User Manual, publication 1756-UM001 and Appendix F Fault Handling in this manual.
Chapter 2

Adding and Configuring Your 1756-M02AE Motion Module

This chapter describes how to add, configure, and edit your 1756-M02AE motion module for use in your motion control application.

This chapter describes each of the tasks for adding and configuring a motion module.

Adding the 1756-M02AE Module

To use your motion module in a control system, you must add your motion module to the application program.

To add a motion module:

1. Right-click the I/O Configuration folder.
2. Select **New Module**. The Select Module Type window appears.

3. Click on the Clear All button to clear the dialog window then click on Motion to list the available Motion Controllers.
New Module

Use this dialog to select and create a new module. To access this dialog, position the cursor on the I/O Configuration Folder in the Controller Organizer, and right click the mouse button. The context sensitive menu appears, from which you can select a New Module.

Type

The Type field displays the catalog number of the module highlighted in the Type list box. You can either type in a module catalog number in this field to quickly select/find the module you want to create or you can scroll through the list of modules in the Type list box.

Major Revision

Select the major revision number of the physical module that you think want to reside in the chassis.

The major revision is used to indicate the revision of the interface to the module.

Type (list box)

This box lists the installed module catalog numbers based on the selected check boxes.

Description (list box)

This portion of the list box contains descriptions of the modules.

Show:

Displays check boxes, which support filtering on particular types of modules.

<table>
<thead>
<tr>
<th>Check this box:</th>
<th>If you want to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital</td>
<td>display digital modules supported by the software</td>
</tr>
<tr>
<td>Analog</td>
<td>display analog modules supported by the software</td>
</tr>
<tr>
<td>Communication</td>
<td>display communication modules supported by the software</td>
</tr>
<tr>
<td>Motion</td>
<td>display motion modules supported by the software</td>
</tr>
<tr>
<td>Controller</td>
<td>display controller modules supported by the software</td>
</tr>
<tr>
<td>Vendor</td>
<td>display a particular vendor's module profiles that are installed on the system.</td>
</tr>
</tbody>
</table>
Check this box:  If you want to:
Other  display modules that do not fit under the rest of the check box categories.

Select All

Click on this button to display all modules in the list box; all the check boxes in the Show field are checked.

Clear All

Click on this button to clear all check boxes in the Show field.

4. In the Type field, select **1756-M02AE 2 Axis Analog/Encoder Servo**.

5. Select **OK**. The New Module window appears.

6. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a name for the servo module. The name can:</td>
</tr>
<tr>
<td></td>
<td>• have a maximum of 40 characters</td>
</tr>
<tr>
<td></td>
<td>• contain letters, numbers and underscores (_)</td>
</tr>
<tr>
<td>Slot</td>
<td>Enter the number of the chassis slot that contains your module.</td>
</tr>
<tr>
<td>Description</td>
<td>Type a description for your motion module.</td>
</tr>
<tr>
<td></td>
<td>This field is optional.</td>
</tr>
</tbody>
</table>
Further explanations of the fields in this dialog are detailed below.

### Editing Your Motion Module Settings

The following section provides explanations of the Motion Module Properties screens. Use these screens to edit the properties of the module when changes need to be made. You can access the Module Properties screen by highlighting the motion module and right clicking the mouse. Select **Properties** from the displayed pop-up menu screen as shown in the following figure.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic keying</td>
<td>Select the electronic keying level.</td>
</tr>
<tr>
<td>To</td>
<td><strong>Select</strong></td>
</tr>
<tr>
<td>Match the vendor, catalog number, and major revision attributes of the physical module and the software configured module</td>
<td>Compatible module</td>
</tr>
<tr>
<td>Disable the electronic keying protection mode</td>
<td>Disable keying</td>
</tr>
<tr>
<td>Match the vendor, catalog number, major revision, and minor revision attributes of the physical module and the software configured module</td>
<td>Exact match</td>
</tr>
</tbody>
</table>
General Tab

Use this tab to create/view module properties for 1756-M02AE motion module. This dialog provides you with the means to view the type, description, vendor, and the name of the parent module. You can also enter the name and a description for the module. Other fields and buttons on this dialog let you set the slot location of the module, review information for both channels, go to the New Tag dialog to create an axis to associate with one of the channels, select the minor revision number and select an electronic keying option. You can also view the status the controller has about the module but, only when online.

Type

Displays the type and description of the module being created (read only).

Vendor

Displays the vendor of the module being created (read only).

Name

Enter the name of the module.
The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

**Description**

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

**Slot**

Enter the slot number where the module resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button will spin from 0 to 3). If you enter a slot number that is out of this range, you will receive an error message when you go to apply your changes.

The slot number cannot be changed when online.

**Revision**

Select the minor revision number of your module.

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

**Electronic Keying**

Select one of these keying options for your module during initial module configuration:

- **Exact Match** - all of the parameters described below must match or RSLogix will reject the inserted module.

- **Compatible Module**
  - the Module Types, Catalog Number, and Major Revision must match
  - the Minor Revision of the physical module must be equal to or greater than the one specified in the software or RSLogix 5000 will reject the inserted module.

- **Disable Keying** - RSLogix 5000 will not employ keying at all.
When you insert a module into a slot in a ControlLogix chassis, RSLogix 5000 compares the following information for the inserted module to that of the configured slot:

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

**Connection Tab**

The Connection Tab is used to define controller to module behavior: This is where you select a requested packet interval, choose to inhibit the module, configure the controller so loss of the connection to this module causes a major fault, and view module faults.

The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.
**Requested Packet Interval**

This does not apply to motion module.

**Inhibit Module checkbox**

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

**TIP**

Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module.

**ATTENTION**

Inhibiting the module causes the connection to the module to be broken and may result in loss of data.

When you check this box and go online, the icon representing this module in the controller organizer displays the Attention Icon.

<table>
<thead>
<tr>
<th>If you are:</th>
<th>Check this checkbox to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>offline</td>
<td>put a place holder for a module you are configuring</td>
</tr>
<tr>
<td>online</td>
<td>stop communication to a module</td>
</tr>
</tbody>
</table>

If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs go to the last configured Program mode state.

If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is 'Inhibited' and not 'Faulted'.

If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module.

If you are a listener (have chosen a “Listen Only” Communications Format), you can not re-configure the module.

If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.
**Major Fault on Controller if Connection Fails checkbox**

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

**Module Fault**

Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- **Connection Request Error** - The controller is attempting to make a connection to the module and has received an error. The connection was not made.

- **Service Request Error** - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.

- **Module Configuration Invalid** - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).

- **Electronic Keying Mismatch** - Electronic Keying is enabled and some part of the keying information differs between the software and the module.

**Associated Axes Tab**

This tab lets you assign axis tags to specific channels of the servo module. Use this tab to configure the selected 1756-M02AE motion modules by:

- setting the selected 1756-M02AE motion module's Servo Update Period
- associating axis tags, of the type AXIS_SERVO, with channels 0 and 1
Servo Update Period

Selects the periodic rate at which the 1756-M02AE module closes the servo loop for the axis, in microseconds (µs).

Channel 0

Represents Channel 0 on the servo module. This field allows you to associate an AXIS_SERVO tag with channel 0. This field transitions to a read-only state while online. Click on the button to the right of this field to open the Axis Properties dialog for the associated axis.

Channel 1

Represents Channel 1 on the servo module. This field allows you to associate an AXIS_SERVO tag with channel 1. This field transitions to a read-only state while online. Click on the button to the right of this field to open the Axis Properties dialog for the associated axis.

New Axis button

Click on this button to navigate to the New Tag dialog to create an AXIS_SERVO tag to associate with one of the channels.
Module Info Tab

The Module Info tab contains information about the selected module, however, you can click on:

- Refresh – to display new data from the module.

- Reset Module – to return the module to its power-up state by emulating the cycling of power. By doing this, you also clear all faults.

The Module Info Tab displays module and status information about the module. It also allows you to reset a module to its power-up state. The information on this tab is not displayed if you are offline or currently creating a module.

Use this tab to determine the identity of the module.

The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.
Identification

Displays the module’s:

- Vendor
- Product Type
- Product Code
- Revision Number
- Serial Number
- Product Name

The name displayed in the Product Name field is read from the module. This name displays the series of the module. If the module is a 1756-L1 module, this field displays the catalog number of the memory expansion board (this selection applies to any controller catalog number even if additional memory cards are added: 1756-L1M1, 1756-L1M2).
**Major/Minor Fault Status**

If you are configuring a:

- **digital module**
  - EEPROM fault
  - Backplane fault
  - None

- **analog module**
  - Comm. Lost with owner
  - Channel fault
  - None

- **any other module**
  - None
  - Unrecoverable
  - Recoverable

**Internal State Status**

This field displays the module’s current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault
- Run mode
- Program mode
- (16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

**Configured**

This field displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is

reset or power is cycled, even if the owner drops connection to the module.

**Owned**

This field displays a yes or no value indicating whether an owner controller is currently connected to the module.

**Module Identity**

<table>
<thead>
<tr>
<th>Displays:</th>
<th>If the module in the physical slot:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree:</td>
</tr>
<tr>
<td></td>
<td>• Vendor</td>
</tr>
<tr>
<td></td>
<td>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</td>
</tr>
<tr>
<td></td>
<td>• Major Revision</td>
</tr>
<tr>
<td>Mismatch</td>
<td>does not agree with what is specified on the General Tab</td>
</tr>
</tbody>
</table>

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

**Refresh**

Click on this button to refresh the tab with new data from the module.

**Reset Module**

Click on this button to return a module to its power-up state by emulating the cycling of power.

Resetting a module causes all connections to or through the module to be closed, and this may result in loss of control.

| IMPORTANT | The following modules return an error if a reset is attempted: 1756-L1 ControlLogix5550 Programmable Controller; 1336T AC Vector Drive; 1395 Digital DC Drive. |

A controller cannot be reset.
**Backplane Tab**

The Backplane tab on the Module Properties window is displayed for informational purposes. You can use this tab to review diagnostic information about the module's communications over the backplane and the chassis in which it is located, clear a fault, and set the transmit retry limit.

Information on this tab is displayed only if you are online.

If you selected a Listen-Only communication format when you created the module, this tab is not available.

The data on this tab comes directly from the module.

**ControlBus Status**

This box either displays OK or one of the following errors:

- Receiver disabled
- Multicast addresses disabled
- RA/GA miscompare

To clear the module’s backplane fault, click the Clear Fault button.

**ControlBus Parameters**

This box contains the following fields and button.
Multicast CRC Error Threshold

This value is the point where it enters a fault state because of Cyclic Redundancy Check (CRC) errors.

Transmit Retry Limit

Not applicable to motion module.

Set Limit Button

You must click on the Reset Limit button to make the new Transmit Retry Limit effective. If you do not and then click either the OK or the Apply button, this limit is not set.

Receive Error Counters

This box displays the number of receiving errors that occurred in the following categories:

- Bad CRC – errors that occurred on received frames (messages)
- Bus time-out – when the receiver timed out
- CRC error – multicast receive errors

Transmit Error Counters

This box displays the number of transmitting errors that occurred in the following categories:

- Bad CRC – errors that occurred on transmitted frames
- Bus Time-out – when the transmitter bus timed out

Refresh

Click on the Refresh button to refresh the tab. When you refresh the tab:

- if you’re using: digital, analog, or motion modules
  then: counters are cleared
- another module
  the tab is refreshed but the counters are not cleared
Assigning Additional Motion Modules

You can assign additional modules by repeating the preceding sections. You can assign up to 16 1756-M02AE modules to each Logix5550 controller. Each module uses a maximum of two axes.

Running Hookup Diagnostics and Auto Tuning

Once you have added and configured your motion module and axes, you can run hookup diagnostics and auto tuning. To run diagnostics and tuning, you must download a program and go online.

**TIP**

If you have configured Virtual axes or Consumed axes, this step does not need to be performed. Hookup and Tuning only need to be performed for Servo and Position-only axes.

To run diagnostics and tuning:

1. Double-click **Main Routine**.

2. If

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Main Routine window only shows rung 0 and the end rung</td>
<td>• Select rung 0</td>
</tr>
<tr>
<td>The Main Routine window shows ladder logic rungs</td>
<td>• Delete rung 0</td>
</tr>
<tr>
<td></td>
<td>• Go to step 3</td>
</tr>
<tr>
<td></td>
<td>Go to step 3</td>
</tr>
</tbody>
</table>

3. Make sure the keyswitch is in the REM position.

4. From the Communications menu, select **Download**.
A window similar to the following appears.

![Download window]

5. Select **Download**.

6. Under the I/O Configuration folder, right-click the 1756-M02AE module you want to use.
7. Select **Properties**. The Module Properties window appears.

![Module Properties Window](image1)

8. Next to the Channel field of your axis, select the button. The Axis Properties window appears.

![Axis Properties Window](image2)
9. Select the **Hookup** tab. The following window appears.

![Hookup Window](image)

10. Select **Start Motor/Encoder Test**. The following window appears.

![RSLogix 5000 Window](image)

11. Select **Yes**. The following window appears.

![Online Command Window](image)

12. Watch the motor to see which way it turns.
13. Select **OK**. The following window appears.

![Image of RSLogix 5000 window asking if the axis moved in the positive direction]

The following window appears.

![Image of RSLogix 5000 window showing command status]

14. If the axis

| Moved in the positive direction | Select Yes |
| Moved in the negative direction | Select No |

15. Select **OK**. The following window appears.

![Image of RSLogix 5000 window indicating successful apply test]

17. Select **Start Marker Test**. The following window appears.

18. Select **Yes**. The following window appears.
19. Slowly rotate the motor axis until the following window appears.

![Online Command - Marker Test](image)


![Axis Properties - Ax_X](image)

21. Select the **Tune Servo** tab. The following window appears.
22. Select **Start Tuning**. The following window appears.

![Start Tuning Window](image1)

23. Select **Yes**. The following window appears.

![Tune Servo Window](image2)

24. Select **OK**. The Tune Bandwidth window appears.

![Tune Bandwidth Window](image3)

25. **If:**
   - You do not want to change the bandwidth
   - You want to change the bandwidth

   **Then:**
   - Go to step 26.
   - In the Bandwidth field, type the position servo bandwidth, which is the unity bandwidth used to calculate gains.
   - Go to step 26.
3. Select **OK**. The following window appears.

![Online Command - Apply Tune](image1.png)

4. Select **OK**. The following window appears.

![RSLogix 5000](image2.png)

5. Select **OK**. The Axis Properties window appears.

![Axis Properties - Axis X](image3.png)
6. Select the **Gains** tab. The following window appears.

![Gains Tab](image)

The window shows new values for the position loop, velocity loop, and output compensation.

7. 

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
</table>
| You want to change the position loop, velocity loop, and servo output values | 1. Type the new values in the appropriate fields.  
2. Go to step 31. |
| You do not want to change the position loop, velocity, and servo output values | Go to step 31. |

8. Select the **Dynamics** tab. The following window appears.

![Dynamics Tab](image)
This window shows new values for maximum velocity, error tolerance, maximum acceleration, and maximum deceleration.

9. **If** You want to change the dynamics values **Then**
   - 3. Type the new values in the appropriate fields.
   - 4. Go to step 33.

   **If** You do not want to change the dynamics values **Then**
   - Go to step 33.

10. **Select OK.** The Axis Properties window closes.
Configuring the 1756-M08SE Module

Adding the 1756-M08SE

This chapter reviews the necessary steps for configuring the 1756-M08SE motion module. Much of this information is the same as for adding and configuring the 1756-M02AE as discussed in the previous chapter.

To configure a 1756-M08SE motion module:

1. In the Controller Organizer, select the I/O Configuration branch.

2. In the File menu, select New Component then Module...
3. The Select Module Type screen displays. Select Clear All. Select Motion. The list displays only available motion modules.

4. Select 1756-M08SE.

5. Press the OK button to close the Select Module Type dialog. The Module Properties wizard opens.
6. Fill in the required parameters for each page, then click the Next> button. See the section titled 1756-M08SE 8 Axis Motion Module Overview in this chapter for more information on the fields in these screens.

7. When you complete the last page, click the Finish> button. The 1756-M08SE motion module appears in the I/O Configuration branch of the Controller Organizer.
The 1756-M08SE 8 Axis SERCOS motion module has been added. It uses the standard RSLogix 5000 module properties tabbed dialog box, with the following exceptions:

- The General tab references the 1756-M08SE 8 Axis motion module.
- The SERCOS Interface tab is for configuring SERCOS communication settings for the 1756-M08SE 8 Axis motion module.
- The SERCOS Interface Info tab is used to monitor the status of the SERCOS communication ring.

For help on the Connection, Module Info or Backplane tabs of this dialog box, refer to the chapter on Naming and Configuring Your 1756-M02AE Module in this manual.

**General Tab**

Use this tab to create/view module properties for the 1756-M08SE 8 Axis SERCOS motion module.
On this tab, you can:

- view the type and description of the module being created
- view the vendor of the module being created
- enter the name of the module
- enter a description for the module
- select the slot number of the module on the network
- select the minor revision number of your module
- select Exact Match, Compatible Module, or Disable Keying

**Type**

Displays the type and description of the module being created (read only).

**Vendor**

Displays the vendor of the module being created (read only).

**Name**

Enter the name of the module. The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

**Description**

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

**Slot**

Enter the slot number where the module resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button spins from 0 to 3). If you enter a slot number that is out of this range, you receive an error message when you apply your changes.

The slot number cannot be changed when online.
Revision

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

The major revision is used to indicate the revision of the interface to the module. The minor revision is used to indicate the firmware revision.

Select the minor revision number of your module.

Electronic Keying

Select one of these keying options for your module during initial module configuration:

**Exact Match** - all of the parameters must match or the inserted module rejects the connection.

**Compatible Module** - the Module Types, Catalog Number, and Major Revision must match. The Minor Revision of the physical module must be equal to or greater than the one specified in the software or the inserted module rejects the connection.

**Disable Keying** – Logix5550 does not employ keying at all.

---

**WARNING**

Changing the RPI and Electronic Keying selections may cause the connection to the module to be broken and may result in a loss of data.

Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.
When you insert a module into a slot in a ControlLogix chassis, RSLogix5000 compares the following information for the inserted module to that of the configured slot:

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

**SERCOS Interface Tab**

The SERCOS interface Tab is where you set the specific Data Rate, Cycle Time, and Transmit Power for the named 1756-M08SE SERCOS interface module.

Use the SERCOS Interface Tab to set and display the:

- SERCOS baud rate
- update rate for the SERCOS ring
- fiber optic transmit power range for the SERCOS ring
The SERCOS ring consists of the drives and axes connected to the 1756-M08SE motion controller.

**TIP**

The settings on this tab are specific to the 1756-M08SE motion controller.

**Data Rate**

Select the baud rate for the SERCOS ring. (For this release, this value is set to 4 MB and is Read Only.)

**Cycle Time**

Select the update rate for the SERCOS ring: 1 ms or 2 ms.

**Transmit Power**

Select the optic transmit power range for the SERCOS ring: High or Low.

### 1756-M08SE 8 Axis Properties SERCOS Interface Info Tab

The SERCOS interface Tab is for monitoring the SERCOS ring of the selected 1756-M08SE while it is on-line. A REFRESH button is available to access the current values.
Use this tab to monitor the following:

**Ring Comm. Phase**

Displays the communications phase of the SERCOS ring:

- 0: Ring Integrity
- 1: Polling
- 2: Identity
- 3: Configuration
- 4: Cyclic communication

**Fault Type**

Displays the current fault type, if any, on the SERCOS ring. Values include:

- No Fault
- Open Ring
- Not communicating
- Not responding
- Timing error
- Duplicate node
- Excess nodes on Ring
- Invalid data rate
- Invalid cycle time

**Refresh**

Click this button to update this page.

**Note**: this information does not refresh automatically.
Chapter 4

Editing Your Controller Properties

This chapter describes the fields on each of the dialogs for the Controller Properties.

Controller Properties

General Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The General tab displays the controller name and description, as well as the physical properties of the controller.

Vendor

Displays the name of the controller's manufacturer.

Type

The catalog number and description of the controller. When online, this field includes the catalog number of the memory card (if any).
**Name**

The name of the controller. When you create a project, this is the same as the name of the project file. When you change the name of the controller, however, the name of the project file does not change. If you want to keep the two the same, then you must rename the file using Windows Explorer or a similar file management tool.

**Description**

Enter a description for the controller here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software ignores any extra characters.

**Chassis Type**

Select a supported chassis type from the pull-down list. Each entry in the list consists of the catalog number of the chassis, as well as a brief description.

The chassis type cannot be changed when online.

**Slot**

Enter the chassis slot number in which the controller resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button will spin from 0 to 3). If you enter a slot number that is out of this range, you will receive an error message when you go to apply your changes.

The slot number cannot be changed when online.

**Revision**

Displays the major and minor revision of the controller. The minor revision is available only when you are online.
Change Type

Click on this button to access the Change Processor Type dialog, from which you can change your controller to another controller within the same platform.

Change Controller Type Dialog Box Overview

Use this dialog to change your controller to another controller within the same platform (e.g. changing from a 1794-L34/A FlexLogix 5434 controller to a 1794-L33/A FlexLogix 5433 controller).

![Change Controller Type Dialog](image)

**IMPORTANT** At this time, RSLogix 5000 does not support changing to a controller from within another platform (e.g., changing from a FlexLogix controller type to a ControlLogix controller type).

Select a processor to change to

Choose the controller you wish to change to from the pull-down menu. The list of available controllers includes all controller types within the same platform as the current processor, with the exception of the current processor itself.
Serial Port Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Serial Port tab allows you to view and configure the controller's serial port.

Mode

The type of protocol you will be using. Choose from System or User (default).

Baud Rate

The baud rate assigned to the serial port on the Logix5550. Choose from 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 (default).

Data Bits

The actual number of bits of data per character. Choose from 7 (ASCII only) or 8 (default).

Parity

The parity for the link. Choose from Even, Odd (ASCII only) or None (default).
Stop Bits

The actual number of stop bits per character. Choose from 2 (ASCII only) or 1 (default).

Control Line

Choose the type of handshaking you wish to use during communications. The choices available to you will vary, depending on the protocol you have selected:

<table>
<thead>
<tr>
<th>For this mode:</th>
<th>And this protocol:</th>
<th>Choose from:</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>ASCII</td>
<td>No Handshake (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full Duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Half Duplex</td>
</tr>
<tr>
<td>System</td>
<td>Point-to-Point</td>
<td>No Handshake (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full Duplex</td>
</tr>
<tr>
<td>Slave</td>
<td></td>
<td>No Handshake (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Half Duplex</td>
</tr>
<tr>
<td>Master</td>
<td></td>
<td>No Handshake (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full Duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Half Duplex</td>
</tr>
</tbody>
</table>

Continuous Carrier

Check this box if you wish to use Half Duplex communication with continuous carrier. This checkbox is disabled if you have chosen something other than Half Duplex communication, or if you have chosen Master as your protocol. By default, this option is unchecked when enabled.

RTS Send Delay

Enter the time (in ms) to delay transmitting the first character of a message after turning on the RTS line. The default value is 0.

RTS Off Delay

Enter the time (in ms) to delay turning off the RTS line after the last character has been transmitted. The default value is 0.

System Protocol Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The System Protocol tab allows you to configure the
controller’s serial port for DF1 Point to Point, DF1 Master, or DF1 Slave. The parameters present on this tab are dependent upon the protocol you select.

![Controller Properties](image)

**IMPORTANT** Note: If you wish to configure your system for ASCII, click on the User Protocol tab.

The parameters present on this tab are dependent upon the protocol you select.

**Common Parameters**

**Protocol**

Choose the DF1 Driver mode from the pull-down menu. Choose from DF1 Point to Point (default), DF1 Slave, or DF1 Master.

**Station Address**

Enter the current station link address of the com port to which the DF1 object is now associated. Valid values are from 0 to 254; the default value is 0.
Error Detection

Click on one of the radio buttons to specify the error detection scheme used for all messages.

- BCC - the processor sends and accepts messages that end with a BCC byte.
- CRC - the processor sends and accepts messages with a 2-byte CRC.

Enable Duplicate Detection

Check this box to enable duplicate message detection, which causes the object to ignore all duplicate messages. This option is disabled by default.

ACK Timeout

Enter the time the object waits for an acknowledgment to a message transmission. Valid values are from 0 to 65535, in 20 ms increments; the default value is 50 ms.

DF1 Point to Point Parameters

DF1 Inquiries

Enter the number of inquiries you want the processor to send after an ACK Timeout. Valid values are from 0 to 255; the default value is 3.

NAK Receive Limit

Enter the number of NAKs the processor can receive in response to a message before stopping the transmission. Valid values are from 0 to 255; the default value is 3.

DF1 Slave Parameters

Transmit Retries

Enter the number of attempted transmits without getting an acknowledgment before a message is deemed undeliverable. Valid values are from 0 to 255; the default value is 3.

Slave Poll Timeout

Enter the amount of time that the master will wait for an acknowledgment to a message sent to the slave.
DF1 Master Parameters

Transmit Retries

Enter the number of attempted transmits without getting an acknowledgment before a message is deemed undeliverable. Valid values are from 0 to 255; the default value is 3.

Reply Message Wait

Enter the time (in ms) that the master will wait after receiving an acknowledgment to a master-initiated message before polling the slave for a reply. Specify this time in 20 ms increments; the default value is 50 (i.e., 50*20 ms, or 1000 ms, or 1 second).

Polling Mode

Choose a polling mode from the pull-down menu. Choose from:

- Message Based - Do not allow the slave to initiate messages.
- Message Based - Allow the slave to initiate messages.
- Standard - Multiple message transfers per node scan.
- Standard - Single message transfer per node scan.

The default mode is Message Based, allowing a slave to initiate messages.

Master Transmit

Choose the master message transmit that designates when to send any DF1 master message. Choose from:

- Between Station Polls - The master transmits a message before the next station.
- In Poll Sequence - The master transmits messages only when the station number is encountered in the poll list.

The default is Between Station Polls.

Normal Poll Node Tag

Choose the tag name of the structure that contains the normal poll node list. Use the Tag Browser to select the appropriate tag name.
The default tag is <none>.

**Normal Poll Group Size**

Enter the total number of active stations polled from the poll node list. Valid values are 0 to 255; the default value is 0.

**Priority Poll Node Tag**

Choose the tag name of the structure to store the priority poll node list. Use the Tag Browser to select the appropriate tag name.

The default tag is <none>.

**Active Station Tag**

Choose the tag name of the structure to store the status (active/non-active) of each node. Use the Tag Browser to select the appropriate tag name.

The default tag is <none>.

**User Protocol Tab Overview**

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The User Protocol tab allows you to configure the controller's serial port for the ASCII protocol.
**Protocol**

Choose the ASCII protocol. In future releases, the Unicode protocol will also be supported.

**Buffer Size**

Enter the maximum size (in bytes) of the data array that you are planning on sending and receiving. Valid values are from 1 to 65536; the default size is 82.

When the controller sends out the data, if it detects an array that is larger than this buffer size, a minor fault occurs and the extra data is truncated. When the controller receives data, if it detects data that is larger than the size of the buffer, the extra characters are dropped.

**Termination Character 1 and 2**

Enter the characters that will be used to define the end of a line. Valid hex range values are from 0 to 255. The default value for Termination Character 1 is $0D$, and the default value for Termination Character 2 is $FF$.

The ARL and ABL instructions use these characters to signal the end of a line. If you do not wish to use these characters, you can either avoid the use of these instructions, or you can define Termination Character
1 as $FF, where $FF tells the controller not to use any definable termination characters when using the ARL or ABL instructions.

If you want to use only one character to signal the end of a line, use Termination Character 1, and define Termination Character 2 as $FF.

**Append Character 1 and 2**

Enter the characters that are appended to the end of a line. Valid hex range values are from 0 to 255. The default value for Append Character 1 is $0D, and the default value for Append Character 2 is $0A.

The AWA instruction appends the specified characters to the end of the messages it sends out. If you do not wish to use these characters, you can either avoid the use of the AWA instruction, or you can define Append Character 1 as $FF, where $FF tells the controller not to append characters when using the AWA instruction.

If you want to append only one character, define Termination Character 1 as the desired character, and define Termination Character 2 as $FF.

**XON/OFF**

Check this option to throttle incoming data.

For example, when this option is checked and the receive buffer gets to be 80% full, an XOFF (0x13) character is transmitted to tell the sending device to stop sending. When the buffer has been processed so that it is less than 80% full, the XON (0x11) character is sent to tell the device to resume sending.

This option is disabled when the Control Line option is configured for Half Duplex.

**Echo Mode**

Check this option to cause any data received in the ASCII port to be sent right back out to the device that sent it. For example, you could use this option with a dumb terminal that is unable to display what it sends, but can display what is echoed to it.

This option is disabled when the Control Line option is configured for Half Duplex.
Delete Mode

The character received just before the delete character sequence (0x7F) is removed by the serial port driver before it is given to the ladder logic. Choose from:

- Ignore - The delete character sequence is treated the same as any other character that is read in.

- CRT or Printer - The preceding character in the string buffer is removed before being given to the ladder logic. The only difference between CRT and Printer modes is the type of device sending the string to the controller. If Echo mode is disabled, CRT and Printer do exactly the same thing.

Major Faults Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Major Faults tab displays information on the major faults that have occurred in the controller.

Number of Major Faults Since Last Cleared

Displays the number of major fault events that have been reported since the log was last cleared.
Recent Faults

Displays a description of the last three major faults that have occurred. These faults are stored in reverse chronological order. When offline, this field contains the stored contents of the last online session.

Clear Majors

Click on this button to clear the Major Fault log.

Minor Faults Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Minor Faults tab displays information on the minor faults that have occurred in the controller.

Number of Minor Faults Since Last Cleared

Displays the number of minor fault events that have been reported since the log was last cleared.
**Recent Faults**

Displays a description of the last eight minor faults that have occurred. These faults are stored in reverse chronological order. When offline, this field contains the stored contents of the last online session.

**Clear Minors**

Click on this button to clear the Minor Fault log.

**Fault Bits**

Lists the minor fault bits that have a specific fault type assigned to them. If the bit is set, the checkbox is set.

When offline, these checkboxes are disabled, but display the contents of the last online session.

**Date/Time Tab Overview**

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Date/Time tab allows you to view and edit the controller’s wall clock time and the coordinated system time status.
Date

The wall clock date, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

This parameter is read-only. When offline, this parameter is empty.

Time

The wall clock time, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

This parameter is read-only. When offline, this parameter is empty.

Set

Click on this button to bring up the Set Date/Time dialog, from which you can set the date and/or time.

This button is disabled when offline.

Make this controller the Coordinated System Time master

Click on this checkbox to select this controller as the CST master. This does not mean that this controller IS the master, it means that you intend for this controller to be the master. If another controller is already the CST master, “duplicate master detected” appears in the status field.

If you are using a Servo card on this controller, set the CST Master.

Status

Indicates the state of the coordinated system time. There are four status fields, with a circular indicator to the left of each. This indicator is blue if the corresponding status condition is true; otherwise, it will be clear. The status conditions are:

- Is the master - you checked the “Make this controller the master” box, and this controller is the CST master.

- Synchronized with a master - this controller is not the master; its time is being synchronized by a master.

- Duplicate master detected - you checked the “Make this controller the master” box, but there is already a CST master.

- Timer hardware faulted - there is a hardware fault.
All of the circular indicators are clear when you are offline.

**Advanced Tab Overview**

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Advanced tab allows you to view and edit advanced controller properties.

![Controller Properties Dialog](image)

**Memory Used**

The amount of memory used in the controller. When offline, this parameter is empty.

**Memory Unused**

The amount of memory available in the controller. When offline, this parameter is empty.

**Memory Total**

The total amount of memory in the controller (used plus unused). If a memory daughter card is present, this total includes that memory. When offline, this parameter is empty.
Controller Fault Handler

Choose the program that will run as the result of a system fault from the pull-down menu. The list contains all of the unscheduled programs.

Power-Up Handler

Choose the program the processor executes when it powers up in Run mode after a power-down in Run mode. The list contains all of the unscheduled programs.

System Overhead Time Slice

Enter or select the percentage of time the controller spends running its system task, relative to running user tasks.

File Tab Overview

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The File tab displays information about the project file.

The fields on this tab cannot be edited. To change the file name or path, you must use the Save As command.
Name

The name of the project file

Path

The drive and directory of the project file.

Created

The creation date and time of the project file, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

Edited

The date and time that the project file was last edited, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.
Naming & Configuring Your Motion Axis

This chapter describes how to name, configure, and edit your axis properties.

**Naming an Axis**

Naming an axis adds it to your application. To name an axis:

1. Go to the **File** pull-down menu, select **New Component**, and then select **Tag**.
2. The New Tag window appears.

Entering Tag Information

A tag allows you to allocate and reference data stored in the controller. A tag can be a simple, single element, or an array, or a structure. There are three types of tags that you can create:

- A base tag allows you to create your own internal data storage.

- An alias tag allows you to assign your own name to an existing tag, structure tag member, or bit.

- A consumed tag allows you to retrieve data from a tag in another controller.

You must set up only one consumed tag to get data from the same producing tag in another controller.

**WARNING**

Setting up more than one consumed tag results in unpredictable controller to controller behavior.

Use this dialog to create new tags. The parameters that appear on this dialog depend upon the type of tag you are creating.
You can create base tags and alias tags while the controller is online or offline, as long as the new tag is verified. You can only create consumed tags while the controller is offline.

**Common Parameters**

The following parameters appear on the New Tag dialog whether you are creating a base tag, alias tag, or consumed tag.

**Name**

Enter the name of the tag you want to create.

**Description**

Enter a description of the tag.

**Tag Type**

Check the type of tag you are creating:

- **Base** - refers to a normal tag (selected by default)

- **Alias** - refers to a tag, which references another tag with the same definition. Special parameters appear on the New Tag dialog that allow you to identify to which base tag the alias refers.

- **Consumed** (only available when the controller is offline) - refers to a tag that is produced by another controller whose data you want to use in this controller. Special parameters appear on the New Tag dialog that allow you to identify from where the consumed tag is to come.

**Data Type**

In the Data Type field you can either enter the type of tag you want to create directly or click on the dotted button to go to the Select Data Type dialog. From this dialog you can select the appropriate data type. Message, Axis, CAM, CAM PROFILES, and PID data types require additional configuration; the Configure button becomes enabled for these data types.

Axis, Motion Group, and Message data types can only be used in a controller scope. Also, these three data types do not support arrays.

If you are creating an alias tag, the data type definition comes from the base tag, so this field is read-only.
3. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a name for the servo axis. The name can: have a maximum of 40 characters contain letters, numbers and underscores (_).</td>
</tr>
<tr>
<td>Description</td>
<td>Type a description for your motion axis. This field is optional.</td>
</tr>
<tr>
<td>Data type</td>
<td>AXIS</td>
</tr>
<tr>
<td>Scope</td>
<td>Select the scope of the axis variable. To use the axis Within the entire program</td>
</tr>
</tbody>
</table>

**Configuring a Motion Axis**

Once you have named your axis in the New Tag window, you must then configure it. You make your configuring options in the Axis Properties screen. These have a series of Tabs that access a specific dialog for configuring the axis. Make the appropriate entries for each of the fields. An asterisk appears on the Tab to indicate changes have been made but not implemented. Press the Apply button at the bottom of each dialog to implement your selections.

**TIP**

When you configure your axis, some fields may be unavailable (greyed-out) because of choices you made in the New Tag window.
1. In the New Tag window, select **Configure**. The Axis Properties General window appears.

![Axis Properties - serovaxis1](image)

**General Tab**

Use the General tab to associate a module to the axis and establish the intended use of the axis. When RSLogix 5000 software is online and the controller transitions to hard run, or the servo loop is on (active), then all the attributes on this tab transition to a read only state. When any attribute transitions to a read only state, then any pending attribute changes are reverted.

**Axis Configuration**

Selects and displays the intended use of the axis:

- Feedback Only: if the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters.

- Servo: if the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

**Assigned Motion Group**

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group.
sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Un-grouped Axes sub-branch of the Motions Groups branch.

Ellipsis (... ) button

Opens the Motion Group Properties dialog box for the Assigned Motion Group, where you can edit the properties of the Assigned Motion Group. If no Motion Group is assigned to this axis, this button is disabled.

New Group button

Opens the New Tag dialog box, where you can create a new Motion Group tag. This button is enabled only if no Motion Group tag has been created.

Module

Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.

Module Type

This read-only field displays the type of motion module, if any, with which the axis is associated. An axis of the AXIS SERVO data type can be associated only with 1756-M02AE motion modules. Displays <none> if the axis is not associated with any motion module.

Channel

Selects and displays the 1756-M02AE motion module channel - either 0 or 1 - to which the axis is assigned. Disabled when the axis is not associated with any motion module.

2. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select the type of axis you are using,</td>
</tr>
<tr>
<td></td>
<td><strong>To use your axis for</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Select</strong></td>
</tr>
<tr>
<td></td>
<td>Full servo operation</td>
</tr>
<tr>
<td></td>
<td>Servo</td>
</tr>
<tr>
<td></td>
<td>Monitoring position</td>
</tr>
<tr>
<td></td>
<td>Position-only</td>
</tr>
<tr>
<td>Field</td>
<td>Entry</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Positioning mode</td>
<td>Select the positioning mode for you module.</td>
</tr>
<tr>
<td><strong>To enable</strong></td>
<td>Select</td>
</tr>
<tr>
<td>A maximum linear excursion of one billion encoder counts</td>
<td>Linear</td>
</tr>
<tr>
<td>The rotary unwind option of the axis</td>
<td>Rotary</td>
</tr>
</tbody>
</table>

3. **If**  
   If you have already created a motion group for this axis  
   Then go to step 13.  
   If you want to create a new motion group  
   Then go to step 4.

4. Press **Apply** then select the Group tab to access the Axis Properties Group dialog.

**New Group Button**

Use the Group tab to associate the axis to a group and modify the group attributes. When RSLogix 5000 software is online, all of the attributes on this tab transition to a read-only state. When an attribute
transitions to a read-only state, all pending attribute changes are reverted.

**Assigned Motion Group**

Selects and displays the group with which the current axis is associated.

**Axes Assigned**

This box lists all of the axes associated with the assigned motion group.

**Coarse Rate**

Selects the periodic rate at which the motion task executes to compute the servo commanded position, velocity, and accelerations to be sent to the 1756-M02AE modules when executing motion instructions. If the coarse rate is too small, the Logix5550 processor may not have time to execute non-motion related ladder logic. As a general rule, one millisecond per axis is needed by the motion task for the Logix5550 processor to have reasonable execution time. As a result, the configuration sets the lower limit on the coarse rate based on the number of axes in the group. See Appendix A in this manual for more information about calculating the Coarse Update rate.

**Servo Update Period**

In this field enter the periodic rate at which the 1756-M02AE module closes the servo loop for the axis. Currently it is fixed at 200µs.

**General Fault Type**

Use this field to select the general fault type mechanism for the motion group. The available selections are:

- **Non Major Fault** - Any faults detected by the motion group do not cause the processor to fault. The application programmer needs to handle the fault in the program.

- **Major Fault** - Any faults detected by the motion group cause the processor OK light to go blinking red and the fault routine to be invoked. If the fault routine handles the fault and clears it, then the OK light goes green. If the fault routine does not clear the fault, then the OK light becomes solid red and the processor stops executing the program.

5. Select the **New Group** button if no groups are defined and you
need to enter a group. The New Tag window appears.

6. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a name for the motion group. The name can:</td>
</tr>
<tr>
<td></td>
<td>• have a maximum of 40 characters</td>
</tr>
<tr>
<td></td>
<td>• contain letters, numbers and underscores (_)</td>
</tr>
<tr>
<td>Description</td>
<td>Type a description for your motion group. This field is optional.</td>
</tr>
<tr>
<td>Data type</td>
<td>MOTION_GROUP</td>
</tr>
<tr>
<td>Scope</td>
<td>Select the scope of the axis variable.</td>
</tr>
<tr>
<td></td>
<td>To use the axis</td>
</tr>
<tr>
<td></td>
<td>Select</td>
</tr>
<tr>
<td></td>
<td>Within the entire program</td>
</tr>
<tr>
<td></td>
<td>Controller</td>
</tr>
</tbody>
</table>

7. Select Configure. The Axis Properties Axis Assignment window appears.
8. From the Unassigned field, select your axis.

9. Select Add.

10. Press Apply to implement the assignment. Press the Group Attributes tab to access the dialog.

11. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse rate</td>
<td>Type the coarse update rate for the motion group.</td>
</tr>
</tbody>
</table>
12. Press **Apply** to set your entries and press OK to return to the Axis Properties Group window.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servo update period</td>
<td>Select the update period for your motion group.</td>
</tr>
<tr>
<td></td>
<td><strong>For</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Select</strong></td>
</tr>
<tr>
<td></td>
<td>200 µs update rate</td>
</tr>
<tr>
<td>General fault type</td>
<td>Select the type of fault for group faults.</td>
</tr>
<tr>
<td></td>
<td><strong>To classify group faults</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Select</strong></td>
</tr>
<tr>
<td></td>
<td>As minor faults</td>
</tr>
<tr>
<td></td>
<td>Non major fault</td>
</tr>
<tr>
<td></td>
<td>As major faults</td>
</tr>
<tr>
<td></td>
<td>Major fault</td>
</tr>
</tbody>
</table>

13. In the Assigned Motion Group field, select your motion group.

14. In the Coarse Rate field, type the coarse update rate based on the number of axes in your application. For more information about coarse update rates, refer to Appendix A - Specifications and Performance.

15. Press Apply to implement entries. Press the Units tab to move to
that dialog.

![Axis Properties dialog](image)

**Units Tab**

Use this tab to determine the units you will use to define your motion axis.

When RSLogix 5000 software is online and the controller transitions to hard run, or the servo loop is on (i.e., active), then all the attributes on this tab transition to a read only state. When any attribute transitions to a read only state, then any pending attribute changes are reverted.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only...
mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Position Units

User-defined engineering units (rather than feedback counts) used for labeling all motion-related values (e.g., position, velocity, etc.) These position units can be different for each axis.

**Note:** Position Units should be chosen for maximum ease of use in your application. For example, linear axes might use position units of Inches, Meters, or mm whereas rotary axes might use units of Revs or Degrees.

Average Velocity Timebase

Specifies the time (in seconds) to be used for calculating the average velocity of the axis. This value is computed by taking the total distance the axis travels in the amount of time specified, and dividing this value by the timebase.

The average velocity timebase value should be large enough to filter out the small changes in velocity that would result in a “noisy” velocity value, but small enough to track significant changes in axis velocity. A value of 0.25 to 0.50 seconds should work well for most applications.
**Conversion Tab**

Use this tab to view the Positioning Mode and configure the Feedback Resolution for an axis, of the tag types AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

![Conversion Tab](image)

**Positioning Mode**

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a
producing axis in a networked Logix processor. The values that can be passed are:

- Linear - provides a maximum total linear travel of 1 billion feedback counts. With this mode, the unwind feature is disabled and you can limit the linear travel distance traveled by the axis by specifying the positive and negative travel limits for the axis.

- Rotary - enables the rotary unwind capability of the axis. This feature provides infinite position range by unwinding the axis position whenever the axis moves through a complete unwind distance. The number of encoder counts per unwind of the axis is specified by the Position Unwind parameter.

Conversion Constant

Type the number of feedback counts per position unit. This conversion - or “K” - constant allows axis position to be displayed, and motion to be programmed, in the position units set in the Units tab. The conversion constant is used to convert axis position units into feedback counts and vice versa.

**Note:** For axes of the type AXIS_SERVO_DRIVE, the label will indicate the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.

Position Unwind

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a producing axis in a networked Logix processor. For a Rotary axis, this value represents the distance (in feedback counts) used to perform automatic electronic unwind. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind distance from both the actual and command position, every time the axis travels the unwind distance.

**Note:** For axes of the type AXIS_SERVO_DRIVE:

- when you save an edited Conversion Constant or a Drive Resolution value, a message box appears, asking you if you want the controller to automatically recalculate certain attribute settings. (Refer to Conversion Constant and Drive Resolution Attributes.)

- the label will indicate the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.
Servo Tab

For an axis of the tag type AXIS_SERVO, configured for Servo operation in the General tab of this dialog box, use this tab to:

- configure an external drive
- configure the drive fault input
- select up to two axis attributes whose status can be monitored

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.
External Drive Configuration

Select the drive type for the servo loop:

- Velocity - disables the servo module’s internal digital velocity loop.
- Torque - the servo module’s internal digital velocity loop is active, which is the required configuration for interfacing the servo axis to a torque loop servo drive.

Loop Configuration

Select the configuration of the servo loop. For this release, only Position Servo is available.

Enable Drive Fault Input

Check this box if you wish to enable the Drive Fault detection. When the drive fault is detected, appropriate action is taken based on the Drive Fault Action specified in the Fault Actions tab of this dialog box.

Drive Fault Input

Specifies the usual state of the drive fault input – Normally:

- Open
- Closed

Real Time Axis Information

Attribute 1/Attribute 2

Select up to two axis attributes whose status will be transmitted – along with the actual position data – to the Logix processor. The values of the selected attributes can be accessed via the standard GSV or Get Attribute List service.

**Note:** The servo status data update time is precisely the coarse update period.

- If a GSV is done to one of these servo status attributes without having selected this attribute via the Drive Info Select attribute, the attribute value will be static and will not reflect the true value in the servo module.
- For an axis of the tag type AXIS SERVO, configured for Servo operation in the General tab of this dialog box, use this tab to:
Homing Tab

Use this tab to configure the attributes related to homing an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.
Mode

Select the homing mode:

- **Active**: In this mode, the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile.

- **Passive**: In this mode, homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.

- **Absolute**: (AXIS_SERVO_DRIVE only) In this mode, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. The only valid Home Sequence for an absolute Homing Mode is immediate.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position will be set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position.

If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

Offset

Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value will be zero.

Sequence

Select the event that will cause the Home Position to be set:
**Sequence Type:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td>Sets the Home Position to the present actual position, without motion.</td>
</tr>
<tr>
<td>Switch</td>
<td>Sets the Home Position when axis motion encounters a home limit switch.</td>
</tr>
<tr>
<td>Marker</td>
<td>Sets the Home Position when axis encounters an encoder marker.</td>
</tr>
<tr>
<td>Switch-Marker</td>
<td>Sets the Home Position when axis first encounters a home limit switch, then encounters an encoder marker.</td>
</tr>
</tbody>
</table>

**Note:** See the section “Homing Configurations,” below, for a detailed description of each combination of homing mode, sequence and direction.

**Limit Switch**

If a limit switch is used, indicate the normal state of that switch (i.e., before being engaged by the axis during the homing sequence):

- Normally Open
- Normally Closed

**Direction**

For active homing sequences, except for the Immediate Sequence type, select the desired homing direction:

<table>
<thead>
<tr>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Uni-directional</td>
<td>The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).</td>
</tr>
</tbody>
</table>
Speed

Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.

Return Speed

The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.
Homing Configurations

The following examples of Active and Passive homing assume that the initial motion, if any, is in a positive axial direction. Click on an individual homing configuration for more information.

- Active Homing Configurations
  - Active Immediate Home
  - Active Bi-directional Home with Switch
  - Active Bi-directional Home with Marker
  - Active Bi-directional Home with Switch then Marker
  - Active Uni-directional Home with Switch
  - Active Uni-directional Home with Marker
  - Active Uni-directional Home with Switch then Marker
- Passive Homing Configurations
  - Passive Immediate Home
  - Passive Home with Switch
  - Passive Home with Marker
  - Passive Home with Switch then Marker

Homing Tab

Use this tab to configure the attributes related to homing the axis. When RSLogix 5000 software is online and the controller transitions to hard run or the servo loop is on (active), then all the attributes on this tab transition to a read only state. When any attribute transitions to a read only state, then any pending attribute changes are reverted.

Home Position

Specify the desired absolute position for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left at this position.

If the positioning mode for the axis is Linear, then the home position should be within the travel limits, if enabled. If the positioning mode is Rotary, then the home position should be less than the unwind distance in position units.
Mode

Select the homing mode. Choose from:

- **Active** - In this mode, the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile.

- **Passive** - In this mode, homing redefines the absolute position of the axis on the next occurrence of the encoder marker. Passive homing is most commonly used to calibrate a Position Only axis type to its encoder marker, although it can be used on Servo axis types as well. Passive homing is identical to active homing to an encoder marker sequence type except that no motion is commanded. That is, the controller simply waits for the marker to occur.

Sequence

When the homing mode is Active, select the homing sequence type to specify whether a home limit switch and/or the encoder marker is to be used. Active homing sequences, with the exception of the Immediate type, use trapezoidal jog velocity profiles to move the axis, while monitoring for the homing event to occur. Choose from:

- **Immediate** - When this sequence is performed, the controller immediately assigns the home position to the current axis position. This homing sequence produces no axis motion.

- **Switch** - This sequence is useful for multi-turn rotary and linear applications, where there are multiple encoder markers over full axis travel, or when an encoder marker is not available. When using this sequence, the axis moves in the specified homing direction, at the specified homing speed, until the home limit switch is detected. The axis then decelerates to a stop and moves in the opposite direction at the specified home return speed until the home limit switch is cleared. The home position is assigned to the axis position at the moment the limit switch is cleared, and the axis decelerates to a stop. The axis then moves back to the home position at the specified home return speed. The motions for this sequence can be shown as follows:

The accuracy of this homing sequence depends on the time uncertainty in detecting the home limit switch transitions. The position uncertainty of the home position is the product of the
maximum time for the control to detect the home limit switch and the specified home return speed.

- **Marker** - This sequence is useful for single-turn rotary and linear encoder applications since these have only one encoder marker for full axis travel. Using this sequence, the axis moves in the specified home direction at the specified homing speed until the marker is detected. The home position is assigned to the corresponding axis position.

The accuracy of this homing sequence depends on the time delay in detecting the marker transition. The position uncertainty of the home position is the product of the maximum delay for the control to detect the marker pulse and the specified home speed. The axis behavior for this sequence can be shown as follows:

- **Switch-Marker** - This is the most precise homing sequence. Using this sequence, the axis moves in the specified home direction, at the specified home speed, until the home limit switch is detected. The axis then decelerates to a stop and moves in the opposite direction at the specified home return speed until the home limit switch is cleared. After clearing the home limit switch, the axis continues in the same direction at the home return speed until the first encoder marker is detected. The home position is assigned to the axis position at the moment the marker is detected, and the axis then decelerates to a stop. The axis then moves back to the home position at the home return speed. The axis behavior for this sequence can be shown as follows:

**Homing Direction**

Select the homing direction: either Positive or Negative. This determines the initial direction of the homing sequence initiated by a Motion Axis Home (MAH) instruction. Positive indicates a forward direction; Negative indicates a reverse direction.

**Homing Speed**

The speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.

**Home Return Speed**

The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.
16. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home position</td>
<td>Type the absolute position for the axis after a homing sequence completes.</td>
</tr>
<tr>
<td>Mode</td>
<td>Select the type of homing to use.</td>
</tr>
<tr>
<td></td>
<td>To</td>
</tr>
<tr>
<td></td>
<td>• Use a homing sequence</td>
</tr>
<tr>
<td></td>
<td>• Use the trapezoidal velocity profile.</td>
</tr>
<tr>
<td></td>
<td>Redefine the current absolute position on the next occurrence of the encoder marker</td>
</tr>
<tr>
<td>Sequence</td>
<td>Select the type of active homing sequence to use.</td>
</tr>
<tr>
<td></td>
<td>To</td>
</tr>
<tr>
<td></td>
<td>Allow the controller to immediately assign the home position to the current axis position</td>
</tr>
<tr>
<td></td>
<td>Move the axis until it detects the home limit switch</td>
</tr>
<tr>
<td></td>
<td>• Move the axis until it detects the home limit switch</td>
</tr>
<tr>
<td></td>
<td>• Use the most precise active homing sequence.</td>
</tr>
<tr>
<td></td>
<td>Move the axis until it detects the encoder marker</td>
</tr>
<tr>
<td>Limit switch</td>
<td>Select the default setting of the home switch.</td>
</tr>
<tr>
<td></td>
<td>To set the home switch</td>
</tr>
<tr>
<td></td>
<td>To open</td>
</tr>
<tr>
<td></td>
<td>To closed</td>
</tr>
<tr>
<td>Homing direction</td>
<td>Select the initial direction of the homing motion.</td>
</tr>
<tr>
<td></td>
<td>To use</td>
</tr>
<tr>
<td></td>
<td>A positive direction</td>
</tr>
<tr>
<td></td>
<td>A negative direction</td>
</tr>
<tr>
<td>Homing speed</td>
<td>Type the initial speed of the jog profile used in an active homing sequence.</td>
</tr>
<tr>
<td>Home return speed</td>
<td>Type the return speed of the jog profile used in an active homing sequence.</td>
</tr>
</tbody>
</table>
17. Press Apply to implement entries. Press the Overtravel tab to access its dialog.

**Hookup Tab**

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS(servo).

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

![Hookup Tab Image]

**Test Increment**

Specifies the amount of distance traversed by the axis when executing the Output & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.
Feedback Polarity

The polarity of the encoder feedback, this field is automatically set by executing either the Feedback Test or the Output & Feedback Test:

- Positive
- Negative

**Note:** When properly configured, this setting insures that axis Actual Position value increases when the axis is moved in the user defined positive direction. This bit can be configured automatically using the MRHD and MAHD motion instructions.

---

**WARNING**

Modifying automatically input polarity values by running the Feedback or Output & Feedback Tests can cause a runaway condition.

---

Output Polarity

The polarity of the servo output to the drive, this field is automatically set by executing the Output & Feedback Test:

- Positive
- Negative

**Note:** When properly configured, this setting and the Feedback Polarity setting insure that, when the axis servo loop is closed, it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.

---

**WARNING**

Modifying automatically input polarity values by running the Feedback or Output & Feedback Tests can cause a runaway condition.
Test Marker...

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback...

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Output & Feedback

Runs the Output & Feedback Test, which checks and, if necessary, reconfigures both the polarity of encoder feedback (the Feedback Polarity setting) and the polarity of the servo output to the drive (the Output Polarity setting), for an axis configured for Servo operation in the General tab of this dialog box.

**Note:** Executing any test operation automatically saves all changes to axis properties.
Tune Tab

Use this tab to configure and initiate axis tuning sequence for the axis. When RSLogix 5000 software is online and the controller transitions to hard run, or the servo loop is on (active), then all the attributes on this tab transition to a read only state. When any attribute transitions to a read only state, then any pending attribute changes are reverted.

Tuning Travel Limit

Specifies a limit to the excursion of the axis during the tune test. If the servo module determines that the axis is not able to complete the tuning process before exceeding the tuning travel limit, it terminates the tuning profile and reports that this limit was exceeded.

Tuning Speed

Determines the maximum speed for the tune process. This value should be set to the desired maximum operating speed of the motor (in engineering units) prior to running the tune test.

Tuning Direction

The direction of the tuning motion profile. Negative indicates the reverse direction; positive indicates the forward direction.
**Damping Factor**

Specifies the dynamic response of the servo axis. The default is set to 0.8. When gains are tuned using a small damping factor, a step response test performed on the axis may generate uncontrolled oscillation. The gains generated using a larger damping factor would produce a system step response that has no overshoot and is stable, but may be sluggish in response to changes.

**Tune**

Specifies the gains to be determined by the tune process.

**Start Tuning**

Click on this button to begin the tuning test. If the tuning process completes successfully, then it determines the selected gains and all the attributes on the Dynamics tab.

---

**WARNING**

This test may cause axis motion with the controller in program mode.

---

**Tune Bandwidth Dialog**

Use this dialog to modify the bandwidth determined by the tune process.

**Bandwidth**

The tune bandwidth is determined by the tune process. The value for the Position Servo Bandwidth represents the unity gain. The unity gain bandwidth is the frequency beyond which the position servo is unable to provide any significant position disturbance correction. Computing gains based on this maximum results in dynamic response in keeping with the current value of the Damping Factor.

---

**IMPORTANT**

Exceeding this value could result in an unstable system.
18. Make entries in the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuning travel limit</td>
<td>Type the limit of axis motion during the auto tuning.</td>
</tr>
<tr>
<td>Tuning speed</td>
<td>Type the maximum speed initiated during auto tuning.</td>
</tr>
<tr>
<td>Tuning direction</td>
<td>Select the direction of the tuning motion profile.</td>
</tr>
<tr>
<td></td>
<td>To</td>
</tr>
<tr>
<td></td>
<td>Use the positive direction</td>
</tr>
<tr>
<td></td>
<td>Use the negative direction</td>
</tr>
<tr>
<td>Damping factor</td>
<td>Type the value to calculate the maximum position servo bandwidth.</td>
</tr>
<tr>
<td>Tune</td>
<td>Select the values you want to calculate during tuning.</td>
</tr>
</tbody>
</table>
19. Press Apply to implement entries. Select the Gains tab to enter values for gains.

**Dynamics Tab**

Use this tab to view or edit the dynamics related parameters for an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE configured for
Servo operations in the General tab of this dialog box, or AXIS_VIRTUAL.

**IMPORTANT** The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

**Note:** The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

**Maximum Velocity**

The steady-state speed of the axis, it is initially set to Tuning Speed by the tuning process. This value is typically set to about 90% of the maximum speed rating of the motor. This provides sufficient "head-room" for the axis to operate at all times within the speed limitations of the motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

**Maximum Acceleration**

The maximum acceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning acceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum acceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the acceleration limits of the drive and motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.
Maximum Deceleration

The maximum deceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning deceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum deceleration rate of the axis. This provides sufficient “head-room” for the axis to operate at all times within the deceleration limits of the drive and motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

Program Stop Action

Select how a specific axis will stop when the processor undergoes a mode change, or when an explicit Motion Group Programmed Stop (MGPS) instruction is executed:

- **Fast Disable**: The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained until the axis motion has stopped at which time the axis is disabled (i.e., Drive Enable is disabled, and Servo Action is disabled).

- **Fast Shutdown**: The axis is decelerated to a stop using the current configured value for maximum deceleration. Once the axis motion is stopped, the axis is placed in the shutdown state (i.e., Drive Enable is disabled, Servo Action is disabled, and the OK contact is opened). To recover from this state, a reset instruction must be executed.

- **Fast Stop**: The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained after the axis motion has stopped. This mode is useful for gravity or loaded systems, where servo control is needed at all times.

- **Hard Disable**: The axis is immediately disabled (i.e. Drive Enable is disabled, Servo Action is disabled, but the OK contact is left closed). Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop.

- **Hard Shutdown**: The axis is immediately placed in the shutdown state. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from this state, a reset instruction must be executed.
Manual Tune...

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Gains Tab

Use this tab to perform the following offline functions:

- adjust, or “tweak” gain values that have been automatically set by the tuning process (in the Tune tab of this dialog)
- manually configure gains for the velocity and position loops

for an axis of the type AXIS_SERVO, which has been configured for Servo operations (set in the General tab of this dialog box), with Position Loop Configuration (set in the Servo tab of this dialog box).
The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral and feed-forward gains around an optional digitally synthesized inner velocity loop, as described in the following diagram:

![Diagram of the motion control loop]

**Important**  
The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

**Note:** The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

**Velocity Feedforward**

Velocity Feedforward Gain scales the current Command Velocity by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

**Acceleration Feedforward**

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the
Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

**Note:** Acceleration Feedforward Gain is not applicable for applications employing velocity loop servo drives. Such systems would require the acceleration feedforward functionality to be located in the drive itself.

**Proportional (Position) Gain**

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the appropriate output scaling factor (either the Velocity Scaling factor or Torque Scaling factor) in the Output tab of this dialog. Your selection of External Drive Configuration type – either Torque or Velocity – in the Servo tab of this dialog will determine which scaling factor you must configure before manually setting gains.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

\[
\text{Pos P Gain} = 16.667 \times \text{Desired Loop Gain (IPM/mil)}
\]

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

\[
\text{Pos P Gain} = \frac{\text{Bandwidth (Hertz)}}{6.28}
\]

The typical value for the Position Proportional Gain is ~100 Sec⁻¹.
**Integral (Position) Gain**

The Integral (i.e., summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis’ drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tuning tab of this dialog), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Output Scaling factor for the axis must be established for the drive system. Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

\[
\text{Pos I Gain} = 0.025 \times 0.001 \text{ Sec/mSec} \times (\text{Pos P Gain})^2
\]

Assuming a Pos P Gain value of 100 Sec-1 this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

**Proportional (Velocity) Gain**

*Note:* This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

The typical value for the Velocity Proportional Gain is ~250 mSec-1.
Integral (Velocity) Gain

**Note:** This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other, but not both. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

Integrator Hold

If the Integrator Hold parameter is set to:

- **Enabled,** the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.

- **Disabled,** all active position or velocity integrators are always enabled.

Manual Tune...

Click on this button to access the Gains tab of the Manual Tune dialog for online editing.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.
Output Tab

Use this dialog for offline configuration of:

- scaling values, which are used to generate gains, and
- the servo’s low-pass digital output filter

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.
IMPORTANT  The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Scaling

The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity servo drive. This has the effect of “normalizing” the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by servo’s automatic tuning procedure but these values can be calculated, if necessary, using the following guidelines.

If the axis is configured for a velocity external servo drive (in the Servo tab of this dialog), the software velocity loop in the servo module is disabled. In this case the Velocity Scaling value can be calculated by the following formula:

\[
\text{Velocity Scaling} = 100\% / \text{(Speed @ 100\%)}
\]

For example, if this axis is using position units of motor revolutions (revs), and the servo drive is scaled such that with an input of 100% (e.g. 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Velocity Scaling attribute value would be calculated as:

\[
\text{Velocity Scaling} = 100\% / \text{(83.3 RPS)} = 1.2\% / \text{Revs Per Second}
\]
**Torque Scaling**

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of “normalizing” the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller’s automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines:

\[
\text{Torque Scaling} = \frac{100\% \text{ Rated Torque}}{\text{Acceleration @ 100\% Rated Torque}}
\]

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec², the Torque Scaling attribute value would be calculated as shown below:

\[
\text{Torque Scaling} = \frac{100\% \text{ Rated}}{(3000 \text{ RPS}^2)} = 0.0333\% \text{ Rated/Revs Per Second}^2
\]

**Note:** If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains will also not reflect the true performance of the system.

**Enable Filter Bandwidth**

Select this to enable the servo’s low-pass digital output filter. De-select this to disable this filter.

**Note:** During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

**Filter Bandwidth**

With Enable Filter Bandwidth selected, this value sets the bandwidth, in Hertz, of the servo’s low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting will be filtered-out, and not sent to the drive.

If the Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value
usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

**Manual Tune...**

Click on this button to access the Gains tab of the Manual Tune dialog for online editing.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

**Limits Tab**

Use this tab to make the following offline configurations:

- enable and set maximum positive and negative software travel limits, and
- configure both Position Error Tolerance and Position Lock Tolerance, and
- set the servo drive’s Output Limit
for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.

---

**IMPORTANT** The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

**Note:** The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

**Soft Travel Limits**

Enables software overtravel checking for an axis when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the Soft Overtravel setting (in the Fault Actions tab of this dialog). Software overtravel limits are disabled during the tuning process.

**Maximum Positive**

Type the maximum positive position to be used for software overtravel checking, in position units.

**Note:** The Maximum Positive limit must always be greater than the Maximum Negative limit.
**Maximum Negative**

Type the maximum negative position to be used for software overtravel checking, in position units.

*Note:* The Maximum Negative limit must always be less than the Maximum Positive limit.

**Position Error Tolerance**

Specifies how much position error the servo will tolerate before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting Position Error Tolerance to 0.75 position units means that a position error fault will be generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units, as shown here:

*Note:* This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

**Position Lock Tolerance**

Specifies the maximum position error the servo module will accept in order to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units, as shown here:

**Output Limit**

Provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, both with and without servo output limiting, is shown below.

The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque loop mode. The percentage of the drive’s maximum current that the servo controller will ever command is equal to the specified servo output limit. For
example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps.

The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.

**Manual Tune...**

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

**Offset Tab**
Use this tab to make offline adjustments to the following Servo Output values:

- Friction Compensation
- Velocity Offset
- Torque Offset
- Output Offset

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.

**IMPORTANT** The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

**Note:** The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

**Friction Compensation**

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction - called “sticktion” - that, even with a significant position error, the axis refuses to budge. Friction Compensation is used to break “
sticktion” in the presence of a non-zero position error. This is done by adding, or subtracting, a percentage output level), called Friction Compensation to the Servo Output value.

The Friction Compensation value should be just less than the value that would break the “sticktion”

A larger value will cause the axis to “dither”, i.e. move rapidly back and forth about the commanded position.

**Velocity Offset**

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second. Because the position servo loop output value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.

**Torque Offset**

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output. Because velocity servo loop output is updated synchronously every Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.

**Output Offset**

Corrects the problem of axis “drift”, by adding a fixed voltage value (not to exceed ±10 Volts) to the Servo Output value. Input a value to achieve near zero drive velocity when the uncompensated Servo Output value is zero.

When interfacing an external Servo Drive - especially for velocity servo drives, it is necessary to compensate for the effect of drive offset. Cumulative offsets of the servo module’s DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to “drift”. If the drift is excessive, it can play havoc on the Hookup Diagnostic and Tuning procedures, as well as result in a steady-state non-zero position error when the servo loop is closed.
**Manual Tune...**

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

**Fault Actions Tab**

![Fault Actions Tab](image)

Use this tab to specify the actions that will be taken in response to the following faults:

- Drive Fault
- Feedback Noise Fault
- Feedback Loss Fault
- Position Error Fault
- Soft Overtravel Fault

for an axis of the type AXIS_SERVO.
When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- **Shutdown** - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated. Furthermore, this fault action opens the OK contact associated with the servo module which can be used to open the E-stop string to the drive power supply. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.

- **Disable Drive** - If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated.

- **Stop Motion** - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.

- **Status Only** - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.
Drive Fault

Specifies the fault action to be taken when a drive fault condition is detected, for an axis with the Drive Fault Input enabled (in the Servo tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown and Disable Drive.

Feedback Noise

Specifies the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Feedback Loss

Specifies the fault action to be taken when feedback loss condition is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Position Error

Specifies the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Soft Overtravel

Specifies the fault action to be taken when a software overtravel error occurs, for an axis with Soft Travel Limits enabled and configured (in the Limits tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

WARNING
Selecting the wrong fault action for your application can cause a dangerous condition. Keep clear of moving machinery.
Tag Tab

Use this tab to modify the name and description of the axis. When you are online, all of the parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.

Name

Displays the name of the current tag. You can rename this tag, if you wish.

Description

Displays the description of the current tag, if any is available. You can edit this description, if you wish.

Tag Type

Indicates the type of the current tag. This type may be:

- Base
- Alias
- Consumed
**Data Type**

Displays the data type associated with the current tag.

**Scope**

Displays the scope of the current tag. The scope is either controller scope, or program scope, based on one of the existing programs in the controller.

**Style**

Displays the default style in which to display the value of the tag. Note that style is only applicable to an atomic tag; a structure tag does not have a display style.

**Produce this tag for up to**

A checked box indicates that this tag is available to remote controllers through controller-to-controller messaging. If this box is checked, the system displays the maximum number of consumers (i.e., connections) allowed for this tag.

The default number of consumers is 2.

**Base Tag**

If this tag is an alias, this field displays the name of the underlying tag on which this alias was based. The base tag actually defines the memory where the data element is stored.

**Manual Tune**

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

**Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.
Assigning Additional Motion Axes

You can assign additional axes by repeating the preceding sections. To name and assign another axis, refer to the Naming an Axis section.

You can assign up to 16 1756-M02AE modules to each Logix5550 controller. Each module uses a maximum of two axes.

Developing a Motion Application Program

To write a motion application program, you can insert motion instructions directly into the ladder diagram application program. The motion instruction set consists of five groups of motion instructions:

- Motion state instructions
- Motion move instructions
- Motion group instructions
- Motion event instructions
- Motion configuration instructions

For more information about these instructions, refer to the Logix5550 Controller Motion Instruction Set Reference Manual – publication 1756-6.4.3.
Understanding a Programming Example

The following figure shows several rungs of a motion control application program.

**Rung 0:** Enables the Feed and Cut axes when you press the servo_on button.

**Rung 1:** Jogs the Feed axis in the positive direction when you press the jog_plus button.

**Rung 2:** Jogs the Feed axis in the reverse direction when you press the jog_minus button.

**Rung 3:** Stops the Feed axis when you release with the jog_plus button or the jog_minus button.

For more information about instructions and creating application programs, refer to the Logix5550 Controller Instruction Set Reference Manual, publication 1756-RM003 and the Logix5550 Motion Instruction Reference Set Manual, publication 1756-RM007.
Chapter 6

Configuring a 1394C-SJT T05/10/22-D Digital Servo Drive

To configure a 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1756-M08SE motion module.

2. In the File menu, select New Component then Module.

3. In the Select Module Type dialog, select the desired drive module: 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D.
4. Press the OK button to close the Select Module Type dialog. The Module Properties wizard opens.

5. Fill in the required parameters for each page, then click the Next> button.

6. When you complete the last page, click the Finish> button. A new drive module displays beneath the selected 1756-M08SE motion module.
The 1756-M08SE 8 Axis Servo Link motion module can be connected to any of three drives:

- 1394C-SJT05-D 5 KW digital servo drive
- 1394C-SJT10-D 10 KW digital servo drive
- 1394C-SJT22-D 22 KW digital servo drive.

Each drive can be associated with up to 4 axes of the AXIS_SERVO_DRIVE tag type. The 1756-M08SE 8 Axis Servo Link module can support up to 8 axes (using varying combinations of from 2 to 8 drives).

The module for a 1394C-SJT05/10/22-D drive has 5 tabs:

- General tab
- Connection tab
- Axes Association
- Power tab
- Module Info tab.
General Tab

Use this tab to enter the module properties for a 1394C-SJT05-D 5 KW, 1394C-SJT10-D 10 KW or 1394C-SJT22-D 22 KW digital servo drive module.

To create any one of the 1394C-SJT modules, the parent module must be a 1756-M08SE 8 Axis Servo Link module.

On this tab, you can:

- view the type and description of the module being created
- view the vendor of the module being created
- enter the name of the module
- enter a description for the module
- set the Base Node for the module
- select the minor revision number of your module
- select Electronic Keying (Exact Match, Compatible Module, or Disable Keying)
- view the status the controller has about the module (you can only view the status while online)
Type
Displays the module type of the module being created: 1394C-SJT05-D 5 KW, 1394C-SJT10-D 10 KW, or 1394C-SJT22-D 22 KW digital servo drive module (read only).

Vendor
Displays the vendor of the module being created (read only).

Name
Enter the name of the module. The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

Description
Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

Base Node
Type or select the Base Node number of the drive module. This node number is determined by multiplying the node number from the module’s rotary switch (1 to 9) by a factor of ten. Thus, valid Base Node values are 10, 20, 30, 40, 50, 60, 70, 80 or 90.

Revision
Select the minor revision number of your module.

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

The major revision is used to indicate the revision of the interface to the module. The minor revision is used to indicate the firmware revision.
Electronic Keying

Select one of these keying options for your module during initial module configuration:

- **Exact Match** - all of the parameters described below must match or the inserted module will reject the connection.

- **Compatible Module**
  - the Module Types, Catalog Number, and Major Revision must match
  - the Minor Revision of the physical module must be equal to or greater than the one specified in the software
  or the inserted module will reject the connection

- **Disable Keying** - Logix5550 will not employ keying at all.

---

**WARNING**

Changing the RPI and Electronic Keying selections may cause the connection to the module to be broken and may result in a loss of data.

Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.

---

When you insert a module into a slot in a ControlLogix chassis, RSLogix 5000 compares the following information for the inserted module to that of the configured slot:

- **Vendor**
- **Product Type**
- **Catalog Number**
- **Major Revision**
- **Minor Revision**

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.
Connection Tab

Use this tab to define controller to drive module behavior.

On this tab, you can:

- choose to inhibit the module
- configure the controller so loss of the connection to this module causes a major fault
- view module faults

TIP

The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.

Requested packet Interval

The amount of time (in milliseconds) between updates of data from the remote controller. This value is scaled as a real value from the microseconds stored in the controller. The local controller will receive data at least this fast. (Not enabled for this release.)
Inhibit Module checkbox

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

**IMPORTANT** Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module.

**WARNING** Inhibiting the module causes the connection to the module to be broken and may result in loss of data.

When you check this box and go online, the icon representing this module in the controller organizer displays the Attention Icon.

**If you are:** **Check this checkbox to:**

- **offline** put a place holder for a module you are configuring
- **online** stop communication to a module. If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs will go to the last configured Program mode state. If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information will change to indicate that the module is 'Inhibited' and not 'Faulted'. If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module. If you are a listener (have chosen a "Listen Only" Communications Format), you can not re-configure the module. If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.
Major Fault on Controller if Connection Fails checkbox

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

Module Fault

Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the module and has received an error. The connection was not made.

- Service Request Error - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.

- Module Configuration Invalid - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).

- Electronic Keying Mismatch - Electronic Keying is enabled and some part of the keying information differs between the.
Associated Axes Tab

Use this tab to configure the selected 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module by associating up to four AXIS_SERVO_DRIVE axis tags with configured axis modules.

Node X0

Represents Axis 0 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (1). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 0. This field transitions to a read only state while online. Click on the Ellipses (...) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X1

Represents Axis 1 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (1). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 1. This field transitions to a read only state while online. Click on the Ellipses (...) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X2

Represents Axis 2 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (2). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 2. This field transitions
to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X3

Represents Axis 3 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (3). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 3. This field transitions to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

New Axis button

Click this button to navigate to the New Tag dialog to create an AXIS_SERVO_DRIVE tag to associate with one of the channels.

**Power Tab**

Use this tab to select a bus regulator for your 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module.

![Module Properties - m08se3 (1394C-SJT10 D 1.1)](image)

**Bus Regulator ID**

Select the catalog number that describes bus regulator device used by the 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module.
Depending upon the Drive Module you have selected, one or more of the following are available:

**Bus Regulator ID** | **Description**
---|---
1394-SR10A | 1400 Watt Resistor, for 5 and 10 kW modules
1394-SR9A | 300 Watt External Shunt, No Fan, for 22 kW modules
1394-SR9AF | 900 Watt External Shunt, No Fan, for 22 kW modules
1394-SR36A | 1800 Watt External Shunt, No Fan, for 22 kW modules
1394-SR36AF | 3600 Watt External Shunt, No Fan, for 22 kW modules
<none> | No bus regulator
Internal | The bus regulator is internal to the drive and need not be specified
Custom | A bus regulator not listed above

**Module Info tab**

Use this tab to display identifying and status information about the 1394C-SJT05/10/22-D drive module. It also allows you to refresh a module and reset a module to its power-up state.
The information on this tab is not displayed if you are:

- offline, or
- currently creating a module

**TIP**

The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.

**Identification**

Displays the module’s:

- Vendor
- Product Type
- Product Code
- Revision
- Serial Number

**Product Name**

The name displayed in the Product Name field is read from the module. This name displays the series of the module.

**Major/Minor Fault Status**

Statuses are: EEPROM fault, Backplane fault, None
Internal State Status

Displays the module’s current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault (please refer to “Major/Minor Fault Status” above)
- Run mode
- Program mode

(16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

Configured

Displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is reset or power is cycled, even if the owner drops connection to the module. This information does not apply to adapters.

Owned

Displays a yes or no value indicating whether an owner controller is currently connected to the module. This information does not apply to adapters.
Module Identity

Displays: If the module in the physical slot:

Match agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree: Vendor Module Type (the combination of Product Type and Product Code for a particular Vendor) Major Revision

Mismatch does not agree with what is specified on the General Tab

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

Refresh

Click on this button to refresh the tab with the new data from the module.

Reset Module

Click on this button to return a module to its power-up state by emulating the cycling of power.

WARNING

Resetting a module causes all connections to or through the module to be closed; this may result in loss of control.
Chapter 7

Motion Instructions

This chapter describes the 32 motion instructions for RSLogix 5000 programming software.

The motion instructions for the RSLogix 5000 programming software consist of five main categories:

- Motion state instructions – to control or change the operating state of an axis.
- Motion move instructions – to control all aspects of axis position.
- Motion group instructions – to control a group of axes.
- Motion event instructions – control the arming and disarming of special event checking functions.
- Motion configuration instructions – to tune an axis and to run diagnostic tests for the system.

For more information about Refer to

<table>
<thead>
<tr>
<th>For more information about</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion instructions</td>
<td>The Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM 007</td>
</tr>
<tr>
<td>Types of motion instruction timing</td>
<td>Appendix E - Instruction Timing</td>
</tr>
</tbody>
</table>

Motion State Instructions

Motion state instructions directly control or change the operating state of an axis.

The motion state instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Type of Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Servo On</td>
<td>MSO</td>
<td>Enables the servo drive and activates the axis servo loop</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Servo Off</td>
<td>MSF</td>
<td>Disables the servo drive and deactivates the axis servo loop</td>
<td>Message</td>
</tr>
</tbody>
</table>
Motion Instructions

For more information about motion state instructions, refer to the Motion State Instructions chapter of the Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - Instruction Timing.

Motion Move Instructions

Motion move instructions control all aspects of axis position.

The motion move instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Type of Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Axis Stop</td>
<td>MAS</td>
<td>Initiates a controlled stop of any motion process on an axis</td>
<td>Immediate Process</td>
</tr>
<tr>
<td>Motion Axis Home</td>
<td>MAH</td>
<td>Homes an axis</td>
<td>Message Process</td>
</tr>
<tr>
<td>Motion Axis Jog</td>
<td>MAJ</td>
<td>Initiates a jog motion profile for an axis</td>
<td>Immediate Process</td>
</tr>
<tr>
<td>Motion Axis Move</td>
<td>MAM</td>
<td>Initiates a move profile for an axis</td>
<td>Immediate Process</td>
</tr>
<tr>
<td>Motion Axis Gear</td>
<td>MAG</td>
<td>Enables electronic gearing between two axes</td>
<td>Immediate Process</td>
</tr>
<tr>
<td>Motion Change Dynamics</td>
<td>MCD</td>
<td>Changes the speed, acceleration rate, or deceleration rate of a move profile or jog profile in progress</td>
<td>Immediate</td>
</tr>
<tr>
<td>Motion Redefine Position</td>
<td>MRP</td>
<td>Changes the command or actual position of an axis</td>
<td>Message</td>
</tr>
</tbody>
</table>

Motion Axis Shutdown

Forces an axis into the shutdown operating state

Once the axis is in the shutdown state, the controller will block any instructions that initiate axis motion.

Motion Axis Shutdown Reset

Changes an axis from an existing shutdown operating state to an axis ready operating state.

If all of the axes of a servo module are removed from the shutdown state as a result of this instruction, the OK relay contacts for the module will close.

Motion Direct Drive On

Enables the servo drive and sets the servo output voltage of an axis.

Motion Direct Drive Off

Disables the servo drive and sets the servo output voltage to the output offset voltage.

Motion Axis Fault Reset

Clears all motion faults.
For more information about motion state instructions, refer to the Motion Move Instructions chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - Instruction Timing.

### Motion Group Instructions

Motion group instructions initiate action on all axes in a group.

The motion group instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Type of Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Group Stop</td>
<td>MGS</td>
<td>Initiates a stop of motion on a group of axes</td>
<td>Process</td>
</tr>
<tr>
<td>Motion Group Programmed Stop</td>
<td>MGPS</td>
<td>Initiates a stop of all motion on all the axes in a group using the method that you set for each axis.</td>
<td>Message Process</td>
</tr>
<tr>
<td>Motion Group Shutdown</td>
<td>MGSD</td>
<td>Forces all the axes in a group into the shutdown operating state</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Group Shutdown Reset</td>
<td>MGSR</td>
<td>Transitions a group of axes from the shutdown operating state to the axis ready operating state</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Group Strobe Position</td>
<td>MGSP</td>
<td>Latches the current command and actual positions of all the axes in a group</td>
<td>Immediate</td>
</tr>
</tbody>
</table>

For more information about motion state instructions, refer to the Motion Group Instructions chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - Instruction Timing.

### Motion Event Instructions

Motion event instructions control the arming and disarming of special event checking functions, such as registration and watch position.
The motion event instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Type of Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Arm Watch Position</td>
<td>MAW</td>
<td>Arms watch-position event checking for an axis</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Disarm Watch Position</td>
<td>MDW</td>
<td>Disarms watch-position event checking for an axis</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Arm Registration</td>
<td>MAR</td>
<td>Arms servo module registration event checking for an axis</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Disarm Registration</td>
<td>MDR</td>
<td>Disarms servo module registration event checking for an axis</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Arm Output Cam</td>
<td>MAOC</td>
<td>Arms an Output Cam for a particular Axis and Output as determined by the operands for the instruction.</td>
<td>Immediate Process</td>
</tr>
<tr>
<td>Motion Disarm Output Cam</td>
<td>MDOC</td>
<td>Disarms either one or all Output Cams connected to a specified axis depending on the selection in the Disarm Type operand.</td>
<td>Immediate Process</td>
</tr>
</tbody>
</table>

For more information about motion state instructions, refer to the Motion Event Instructions chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - Instruction Timing.

Understanding Motion Configuration Instructions

Motion configuration instructions allow you to tune an axis and to run diagnostic tests for your control system. These tests include:

- A motor/encoder hookup test
- An encoder hookup test
- A marker test

The motion configuration instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Type of Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Apply Axis Tuning</td>
<td>M AAT</td>
<td>Computes a complete set of servo gains and dynamic limits based on a previously executed M RAT instruction. The M AAT instruction also updates the servo module with new gain parameters.</td>
<td>Message</td>
</tr>
<tr>
<td>Motion Run Axis Tuning</td>
<td>M RAT</td>
<td>Commands the servo module to run a tuning motion profile for an axis</td>
<td>Message</td>
</tr>
</tbody>
</table>
For more information about motion state instructions, refer to the Motion Configuration Instructions chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - Instruction Timing.
Troubleshooting

This chapter describes how to troubleshoot your ControlLogix motion control system.

### 1756-M02AE Module Status Using the OK Indicator

<table>
<thead>
<tr>
<th>OK LED displays</th>
<th>Module status is</th>
<th>Take this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>The module is not operating.</td>
<td>• Apply chassis power.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Verify the module is completely inserted into the chassis and backplane.</td>
</tr>
<tr>
<td>Flashing green light</td>
<td>The module has passed internal diagnostics, but it is not communicating axis data over the backplane.</td>
<td>• None, if you have not configured the module.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If you have configured the module, check the slot number in the 1756-M02AE Properties dialog box.</td>
</tr>
<tr>
<td>Steady green light</td>
<td>• Axis data is being exchanged with the module.</td>
<td>None. The module is ready for action.</td>
</tr>
<tr>
<td></td>
<td>• The module is in the normal operating state.</td>
<td></td>
</tr>
<tr>
<td>Flashing red light</td>
<td>• A major recoverable failure has occurred.</td>
<td>• Check the servo fault word for the source of the error.</td>
</tr>
<tr>
<td></td>
<td>• A communication fault, timer fault, or NVS update is in progress.</td>
<td>• Clear the fault condition using the motion instructions.</td>
</tr>
<tr>
<td></td>
<td>• The OK contact has opened.</td>
<td>• Resume normal operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the flashing persists, reconfigure the module.</td>
</tr>
<tr>
<td>Solid red light</td>
<td>• A potential non-recoverable fault has occurred.</td>
<td>• Reboot the module.</td>
</tr>
<tr>
<td></td>
<td>• The OK contact has opened.</td>
<td>• If the solid red persists, replace the module.</td>
</tr>
</tbody>
</table>
### 1756-M 02AE Module Status Using the FDBK Indicator

<table>
<thead>
<tr>
<th>If the FDBK LED displays</th>
<th>Then the module status is</th>
<th>Take this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>The axis is not used.</td>
<td>• None, if you are not using this axis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If you are using this axis, make sure you configured the module and associated an axis tag with the module.</td>
</tr>
<tr>
<td>Flashing green light</td>
<td>The axis is in the normal servo loop inactive state.</td>
<td>None. You can change the servo axis state by executing motion instructions.</td>
</tr>
<tr>
<td>Steady green light</td>
<td>The axis is in the normal servo loop active state.</td>
<td>None. You can change the servo axis state by executing motion instructions.</td>
</tr>
<tr>
<td>Flashing red light</td>
<td>The axis servo loop error tolerance has been exceeded.</td>
<td>• Correct the source of the problem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Clear the servo fault using a fault reset instruction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Resume normal operation.</td>
</tr>
<tr>
<td>Solid red light</td>
<td>An axis encoder feedback fault has occurred.</td>
<td>• Correct the source of the problem by checking the encoder and power connections.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Clear the servo fault using the MAFR instruction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Resume normal operation.</td>
</tr>
</tbody>
</table>

### 1756-M 02AE Module Status Using the DRIVE Indicator

<table>
<thead>
<tr>
<th>If the DRIVE LED displays</th>
<th>Then the module status is</th>
<th>Take this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>The axis is not used.</td>
<td>• None, if you are not using the axis or have configured it as a position-only axis.</td>
</tr>
<tr>
<td></td>
<td>The axis is a position-only axis type.</td>
<td>• Otherwise, make sure you have configured the module, associated an axis tag with the module, and configured the axis as a servo axis.</td>
</tr>
</tbody>
</table>
## Troubleshooting

<table>
<thead>
<tr>
<th>If the DRIVE LED displays</th>
<th>Then the module status is</th>
<th>Take this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashing green light</td>
<td>The axis drive is in the normal disabled state.</td>
<td>None. You can change the servo axis state by executing a motion instruction.</td>
</tr>
<tr>
<td>Steady green light</td>
<td>The axis drive is in the normal enabled state.</td>
<td>None. You can change the servo axis state by executing a motion instruction.</td>
</tr>
</tbody>
</table>
| Flashing red light        | The axis drive output is in the Shutdown state. | • Check for faults that may have generated this state.  
• Execute the shutdown reset motion instruction.  
• Resume normal operation. |
| Solid red light           | The axis drive is faulted. | • Check the drive status.  
• Clear the drive fault condition at the drive.  
• Execute a fault reset motion instruction.  
• Resume normal operation.  
• Check the configuration for the Drive Fault.  
• If configured to be normally open and there is no voltage, this is the normal condition.  
• If configured to be normally closed and there is 24V applied, this is the normal condition. |

### 1756-M 08SE SERCOS Communication Phase Status Using the CP Indicator

<table>
<thead>
<tr>
<th>If the CP LED displays:</th>
<th>Then the module status is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>The module is not operating.</td>
</tr>
<tr>
<td>Solid red light</td>
<td>• In Phase 0: looking for a closed ring.</td>
</tr>
<tr>
<td>Flashing red light</td>
<td>• In Phase 1: looking for active nodes.</td>
</tr>
<tr>
<td>Alternating Red/Green light</td>
<td>• In Phase 2: configuring nodes for communication.</td>
</tr>
<tr>
<td>Flashing green light</td>
<td>• In Phase 3: configuring device specific parameters</td>
</tr>
<tr>
<td>Solid green light</td>
<td>• In Phase 4: configured and active.</td>
</tr>
</tbody>
</table>
### 1756-M08SE Module Status

#### Using the OK Indicator

<table>
<thead>
<tr>
<th>If the OK LED displays:</th>
<th>Then the module status is:</th>
<th>Take this action:</th>
</tr>
</thead>
</table>
| Off                    | The module is not operating. | • Apply chassis power.  
|                        |                            | • Verify the module is completely inserted into the chassis and backplane. |
| Flashing green light   | The module has passed internal diagnostics, but has not established active communications. | • None, if you have not configured the module. |
| Solid green light      | • Data is being exchanged.  
|                        | • The module is in the normal operating state. | None. The module is ready for action. |
| Flashing red light     | • A major recoverable failure has occurred.  
|                        | • An NVS update is in progress. | If an NVS update is in progress, complete the NVS update.  
|                        |                            | If an NVS update is not in progress:  
|                        |                            | • Reboot |
| Solid red light        | • A potential nonrecoverable fault has occurred. | • Reboot the module.  
|                        |                            | • If the solid red persists, replace the module. |

### 1756-M08SE SERCOS Ring Status

<table>
<thead>
<tr>
<th>If the SERCOS Ring LED displays:</th>
<th>Then the ring status is:</th>
<th>Take this action:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid green light</td>
<td>The ring, drive, and axes are configured and are actively communicating through to the nodes on the ring.</td>
<td>None.</td>
</tr>
</tbody>
</table>
| Flashing red light               | The module has detected a setup or configuration fault with the ring. | Check your system setup and configuration as follows:  
|                                  |                        | • Ensure drive and axes addresses are correct.  
|                                  |                        | • Eliminate duplicate drive or axes addresses on ring.  
|                                  |                        | • Remove excess axes from ring.  
<p>|                                  |                        | • Make sure application program has selected the proper Ring Cycle Period and Baud Rate. |</p>
<table>
<thead>
<tr>
<th>If the SERCOS Ring LED displays:</th>
<th>Then the ring status is:</th>
<th>Take this action:</th>
</tr>
</thead>
</table>
| **Solid red light**              | The module has detected a hardware or installation fault with the ring. | Check your system hardware and installation as follows:  
• Make sure all cables are properly installed.  
• Make sure cable is of the correct type and length.  
• Make sure application program has configured the module’s ring transmit level to High when using specified cables.  
• Make sure the drive’s transmit levels are set appropriately.  
• Inspect cables for degradation.  
• Inspect drives for any faults and correct them. |
| **Off**                          | The module has detected no ring data on its receiver. | Check your system and installation as follows:  
• Make sure all cables are properly installed  
• Inspect cable for degradation and breakage.  
• Inspect drives for faults. |
| **Flashing green light**         | The ring, drive, or axes are not configured. | Not a problem if the system has not been configured. If you are having trouble configuring the ring, drive, and axes:  
• Make sure that the application program is setup properly for the equipment in use. |
# Specifications and Performance

This appendix shows specifications and performance guidelines for the motion modules.

## 1756-M02AE Motion Module Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Specification Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of axes per chassis</td>
<td>Configurable</td>
</tr>
<tr>
<td>Motion commands</td>
<td>32</td>
</tr>
<tr>
<td>Number of axes per module</td>
<td>2 axes maximum</td>
</tr>
<tr>
<td>Servo loop</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Nested PI digital position and velocity servo</td>
</tr>
<tr>
<td>Gain resolution</td>
<td>32-bit floating point</td>
</tr>
<tr>
<td>Absolute position range</td>
<td>±1,000,000,000 encoder counts</td>
</tr>
<tr>
<td>Rate</td>
<td>5 kHz</td>
</tr>
<tr>
<td>Module location</td>
<td>1756 ControlLogix chassis</td>
</tr>
<tr>
<td>Module keying</td>
<td>Electronic</td>
</tr>
<tr>
<td>Power dissipation</td>
<td>5.5W maximum</td>
</tr>
<tr>
<td>Backplane current</td>
<td></td>
</tr>
<tr>
<td>Voltage range</td>
<td></td>
</tr>
<tr>
<td>On state</td>
<td>3.4V to 5.0V</td>
</tr>
<tr>
<td>Off state</td>
<td>0V to 1.8V</td>
</tr>
<tr>
<td>Input impedance</td>
<td>531 Ohms differential</td>
</tr>
<tr>
<td>Encoder input</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Incremental AB quadrature with marker</td>
</tr>
<tr>
<td>Mode</td>
<td>4X quadrature</td>
</tr>
<tr>
<td>Rate</td>
<td>4 MHz counts per second maximum</td>
</tr>
<tr>
<td>Electrical interface</td>
<td>Optically isolated 5V differential</td>
</tr>
<tr>
<td>Voltage range</td>
<td></td>
</tr>
<tr>
<td>On state</td>
<td></td>
</tr>
<tr>
<td>Off state</td>
<td></td>
</tr>
<tr>
<td>Input impedance</td>
<td></td>
</tr>
<tr>
<td>24V input voltage</td>
<td>+24V dc nominal</td>
</tr>
<tr>
<td>Maximum</td>
<td>26.4V</td>
</tr>
<tr>
<td>Minimum on</td>
<td>18.5V</td>
</tr>
<tr>
<td>Maximum off</td>
<td>3.5V</td>
</tr>
<tr>
<td>5V input voltage</td>
<td>+5V dc nominal</td>
</tr>
<tr>
<td>Maximum</td>
<td>5.5V</td>
</tr>
<tr>
<td>Minimum on</td>
<td>3.7V</td>
</tr>
<tr>
<td>Maximum off</td>
<td>1.5V</td>
</tr>
<tr>
<td>Input impedance</td>
<td></td>
</tr>
<tr>
<td>24V input</td>
<td>1.2 kOhms</td>
</tr>
<tr>
<td>5V input</td>
<td>9.5 kOhms</td>
</tr>
<tr>
<td>Response time</td>
<td></td>
</tr>
<tr>
<td>(position latched)</td>
<td>1µs</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>All other inputs</th>
<th>Optically isolated, current sinking input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>Input voltage</td>
<td>+24V dc nominal</td>
</tr>
<tr>
<td>Maximum</td>
<td>26.4V</td>
</tr>
<tr>
<td>Minimum on</td>
<td>17.0V</td>
</tr>
<tr>
<td>Maximum off</td>
<td>8.5V</td>
</tr>
<tr>
<td>Input impedance</td>
<td>7.5 kOhms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Servo output</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Analog voltage</td>
</tr>
<tr>
<td>Isolation</td>
<td>200 kOhms</td>
</tr>
<tr>
<td>Voltage range</td>
<td>±10V</td>
</tr>
<tr>
<td>Voltage resolution</td>
<td>16 bits</td>
</tr>
<tr>
<td>Load</td>
<td>5.6 kOhms resistive minimum</td>
</tr>
<tr>
<td>Maximum offset</td>
<td>25 mV</td>
</tr>
<tr>
<td>Gain error</td>
<td>±4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All other outputs</th>
<th>Solid-state isolated relay contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>Operating voltage</td>
<td>+24V dc nominal</td>
</tr>
<tr>
<td>Maximum</td>
<td>26.4V</td>
</tr>
<tr>
<td>Operating current</td>
<td>75 mA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RTB keying</th>
<th>User-defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field wiring arm</td>
<td>36-position RTB (1756-TBCH or -TBS6H)²</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RTB screw torque</th>
<th>5lb-in. (0.5 Nm) maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cage clamp)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conductors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire size</td>
<td>22 gauge (3.1 mm²) minimum to copper¹</td>
</tr>
<tr>
<td></td>
<td>3/64 inch (1.2 mm) insulation maximum</td>
</tr>
<tr>
<td>Category</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screwdriver blade width for RTB</th>
<th>1/8 inch (3.2 mm) maximum</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Environmental conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature</td>
<td>0 to 60°C (32 to 140°F)</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>-40 to 85°C (-40 to 185°F)</td>
</tr>
<tr>
<td>Relative humidity</td>
<td>5 to 95% noncondensing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Agency certification (when product or packaging is marked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UL Class 1, Division 2, hazardous location</td>
</tr>
<tr>
<td>CE marked for all applicable directives</td>
</tr>
</tbody>
</table>

¹ Maximum wire size will require the extended depth RTB housing (1756-TBE).
² Use this conductor category information for planning conductor routing as described in the system level installation manual.
³ Refer to Industrial Automation Wiring and Grounding Guidelines, publication number 1770-4.1.
Understanding Coarse Update Period Calculations

To calculate the coarse update period for the number axes in your application, you can use the following formula:

\[
2 \times \left( \frac{\text{Baseline task time}}{1000} \right) + \left( \frac{(\text{Actions for axis 1})}{1000} \right) + \left( \frac{(\text{Actions for axis 2})}{1000} \right) + \ldots + \left( \frac{(\text{Actions for axis } n)}{1000} \right) = \text{Coarse Update Period}
\]

The result of the above calculation must be divided by 1000 rounded up to the nearest milliseconds.

You can use the sample calculation worksheet in this section to determine your coarse update period. To determine the values for your equation, refer to the following table.

The coarse update period can have a significant bearing on the quality of motion control for a given application. If an application requires commanded acceleration or deceleration times that are comparable to the coarse update period, significant velocity and position overshoot can occur as the axis attempts to follow the command profile. The amount of velocity overshoot can be calculated as the product of the acceleration or deceleration rate and the coarse update period. As a general rule of thumb the acceleration and deceleration times for a motion application should be at least 10 times the coarse update period to avoid significant velocity or position overshoot.

For example, an application that requires an axis to accelerate and decelerate to full speed in 100 milliseconds is best handled by choosing a coarse update period of 10 milliseconds or shorter. Be sure to check what the minimum coarse update period is for the associated controller before selecting the coarse update period. The minimum coarse update period can be determined based on the number of axes and the worst case motion activity according to the execution time tables provided.

Understanding Action Timing

Every action performed by an axis requires an amount of motion task time. For example to perform a trapezoidal move requires 310 µs.
The following table shows execution times for common motion actions.

<table>
<thead>
<tr>
<th>State/Action (Δ = per axis)</th>
<th>Typical Execution Time (in µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Task Overhead</td>
<td>190</td>
</tr>
<tr>
<td>Servo Axis Δ</td>
<td>205</td>
</tr>
<tr>
<td>Virtual Axis Δ</td>
<td>175</td>
</tr>
<tr>
<td>Consumed Axis Δ*</td>
<td>900*</td>
</tr>
<tr>
<td>Servo On Δ</td>
<td>40</td>
</tr>
<tr>
<td>Trap Move Δ</td>
<td>310</td>
</tr>
<tr>
<td>S-Curve Move Δ</td>
<td>435</td>
</tr>
<tr>
<td>Trap Jog Δ</td>
<td>210</td>
</tr>
<tr>
<td>S-Curve Jog Δ</td>
<td>340</td>
</tr>
<tr>
<td>Gearing (Actual) Δ</td>
<td>290</td>
</tr>
<tr>
<td>Clutch Δ</td>
<td>105</td>
</tr>
<tr>
<td>Gearing (Command) Δ</td>
<td>125</td>
</tr>
<tr>
<td>Clutch Δ</td>
<td>110</td>
</tr>
<tr>
<td>Position Camming (Actual,linear) Δ</td>
<td>495</td>
</tr>
<tr>
<td>Position Camming (Actual,cubic) Δ</td>
<td>550</td>
</tr>
<tr>
<td>Position Camming (Command,linear) Δ</td>
<td>295</td>
</tr>
<tr>
<td>Position Camming (Command,cubic) Δ</td>
<td>380</td>
</tr>
<tr>
<td>Time Camming (linear) Δ</td>
<td>260</td>
</tr>
<tr>
<td>Time Camming (cubic) Δ</td>
<td>320</td>
</tr>
</tbody>
</table>

* The task time of a consumed axis can be lowered to approximately 650 µs by using a consumer coarse update period that is an integer multiple of the producer coarse update period. The value of 900µs reported above is the worst case (producer/consumer update ratio = 2/3). Using 2/4 would yield ~650µs.
Using the Sample Calculations Worksheet

You can use this sample calculation worksheet to determine the coarse update period for the group in your application.

1. Complete the following table.

<table>
<thead>
<tr>
<th>System</th>
<th>Describe the type of system you are using.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter baseline task time = Motion Task Overhead + (# of Servo Axes * Servo Axis ∆) + (# of Virtual Axes * Virtual Axis ∆) + (# of Consumed Axes * Consumed Axis ∆) μs</td>
</tr>
</tbody>
</table>

2. For each axis in your application, use the following table to determine the action value for each axis.

<table>
<thead>
<tr>
<th>Actions</th>
<th>If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Servo on</td>
<td>μs</td>
</tr>
<tr>
<td>3 Trapezoidal move</td>
<td>μs</td>
</tr>
<tr>
<td>4 S-curve move</td>
<td>μs</td>
</tr>
<tr>
<td>5 Trapezoidal jog</td>
<td>μs</td>
</tr>
<tr>
<td>6 S-curve jog</td>
<td>μs</td>
</tr>
<tr>
<td>7 Actual gear</td>
<td>μs</td>
</tr>
<tr>
<td>8 Command gear</td>
<td>μs</td>
</tr>
<tr>
<td>9 Actual PCAM</td>
<td>μs</td>
</tr>
<tr>
<td>10 Command PCAM</td>
<td>μs</td>
</tr>
<tr>
<td>11 TCAM</td>
<td>μs</td>
</tr>
<tr>
<td>12 Total Axis Action Value (Add lines 2 through 11). Place total here.</td>
<td>μs</td>
</tr>
</tbody>
</table>

3. Calculate the Total Axes Action Value by adding all of the Total Axis Action Values (line 12) for all axes in the group.

4. The Recommended Coarse Update Period is calculated by: (Baseline Task Time(line 1) + Total of Axes Action Values(line 12) ) * 2 / 1000 then round up to the milliseconds to get your Recommended Coarse Update Period.
Sample Calculation

You have the following situation:

- You have a system consisting of 2 modules, 3 Servo axes, and 1 virtual axis.
- You are turning the servo on and performing a trapezoidal move for axis1.
- You are turning the servo on and doing an S-Curve jog for axis2.
- Axis3 is a virtual axis and you are doing a command gear for on it.
- You are turning the servo on and doing a cubic command position cam for axis4.

1. Complete the following table.

<table>
<thead>
<tr>
<th>System</th>
<th>Describe the type of system you are using.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter baseline task time = Motion Task Overhead + (3 * Servo Axis) + (1 * Virtual Axis).</td>
</tr>
</tbody>
</table>

2. For each axis in your application, use the following tables to determine the action value for each axis.

**Axis 1:**

<table>
<thead>
<tr>
<th>Actions</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Servo on</td>
<td>40 µs</td>
</tr>
<tr>
<td>3 Trapezoidal move</td>
<td>310 µs</td>
</tr>
<tr>
<td>4 S-curve move</td>
<td>0 µs</td>
</tr>
<tr>
<td>5 Trapezoidal jog</td>
<td>0 µs</td>
</tr>
<tr>
<td>6 S-curve jog</td>
<td>0 µs</td>
</tr>
<tr>
<td>7 Actual gear</td>
<td>0 µs</td>
</tr>
<tr>
<td>8 Command gear</td>
<td>0 µs</td>
</tr>
<tr>
<td>9 Actual PCAM</td>
<td>0 µs</td>
</tr>
<tr>
<td>10 Command PCAM</td>
<td>0 µs</td>
</tr>
<tr>
<td>11 TCAM</td>
<td>0 µs</td>
</tr>
<tr>
<td>12 Total Axis Action Value (Add lines 2 through 11)</td>
<td>350 µs</td>
</tr>
</tbody>
</table>
### Axis 2:

<table>
<thead>
<tr>
<th>Actions</th>
<th>Execution Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Servo on</td>
<td>40</td>
</tr>
<tr>
<td>3 Trapezoidal move</td>
<td>0</td>
</tr>
<tr>
<td>4 S-curve move</td>
<td>0</td>
</tr>
<tr>
<td>5 Trapezoidal jog</td>
<td>0</td>
</tr>
<tr>
<td>6 S-curve jog</td>
<td>340</td>
</tr>
<tr>
<td>7 Actual gear</td>
<td>0</td>
</tr>
<tr>
<td>8 Command gear</td>
<td>0</td>
</tr>
<tr>
<td>9 Actual PCAM</td>
<td>0</td>
</tr>
<tr>
<td>10 Command PCAM</td>
<td>0</td>
</tr>
<tr>
<td>11 TCAM</td>
<td>0</td>
</tr>
<tr>
<td>12 Total Axis Action Value (Add lines 2 through 11)</td>
<td>380 µs</td>
</tr>
</tbody>
</table>

### Axis 3:

<table>
<thead>
<tr>
<th>Actions</th>
<th>Execution Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Servo on</td>
<td>0</td>
</tr>
<tr>
<td>3 Trapezoidal move</td>
<td>0</td>
</tr>
<tr>
<td>4 S-curve move</td>
<td>0</td>
</tr>
<tr>
<td>5 Trapezoidal jog</td>
<td>0</td>
</tr>
<tr>
<td>6 S-curve jog</td>
<td>0</td>
</tr>
<tr>
<td>7 Actual gear</td>
<td>0</td>
</tr>
<tr>
<td>8 Command gear</td>
<td>125</td>
</tr>
<tr>
<td>9 Actual PCAM</td>
<td>0</td>
</tr>
<tr>
<td>10 Command PCAM</td>
<td>0</td>
</tr>
<tr>
<td>11 TCAM</td>
<td>0</td>
</tr>
<tr>
<td>12 Total Axis Action Value (Add lines 2 through 11)</td>
<td>125 µs</td>
</tr>
</tbody>
</table>
A-8 Specifications and Performance

Axes 4:

<table>
<thead>
<tr>
<th>Actions</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Servo on</td>
<td>40</td>
</tr>
<tr>
<td>3 Trapezoidal move</td>
<td>0</td>
</tr>
<tr>
<td>4 S-curve move</td>
<td>0</td>
</tr>
<tr>
<td>5 Trapezoidal jog</td>
<td>0</td>
</tr>
<tr>
<td>6 S-curve jog</td>
<td>0</td>
</tr>
<tr>
<td>7 Actual gear</td>
<td>0</td>
</tr>
<tr>
<td>8 Command gear</td>
<td>0</td>
</tr>
<tr>
<td>9 Actual PCAM</td>
<td>0</td>
</tr>
<tr>
<td>10 Command PCAM</td>
<td>380</td>
</tr>
<tr>
<td>11 TCAM</td>
<td>0</td>
</tr>
<tr>
<td>12 Total Axis Action Value (Add lines 2 through 11)</td>
<td>420</td>
</tr>
</tbody>
</table>

3. The calculated coarse rate for this application is

<table>
<thead>
<tr>
<th>Description</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline task time (line 2)</td>
<td>980</td>
</tr>
<tr>
<td>Total Axis Action Value for axis 1 (line 12)</td>
<td>350</td>
</tr>
<tr>
<td>Total Axis Action Value for axis 2 (line 12)</td>
<td>380</td>
</tr>
<tr>
<td>Total Axis Action Value for axis 3 (line 12)</td>
<td>125</td>
</tr>
<tr>
<td>Total Axis Action Value for axis 4 (line 12)</td>
<td>420</td>
</tr>
<tr>
<td>Total Axes Action Value (add all of the above)</td>
<td>2255</td>
</tr>
<tr>
<td>TOTAL (Total Axes Action Value * 2)</td>
<td>4510</td>
</tr>
<tr>
<td>Recommended Coarse Update Period = (TOTAL / 1000) rounded up to nearest ms</td>
<td>5 ms</td>
</tr>
</tbody>
</table>

Output Cam Timing For 1756 Controller

The impact on the coarse update period is primarily dependent on three factors:

- total number of output cam array entries
- total number of output compensation array entries
- number of outputs compensation array entries with non-zero latch/unlatch time delay values
The following formulas ballpark additional coarse update time required for each unique Output Cam execution target being used.

In the following formulas:

- \( A \) = number of Output Cam array elements
- \( B \) = number of Output Compensation array elements
- \( C \) = number of Output Compensation array elements with Latch/Unlatch non-zero delay values

All values are expressed in micro-seconds.

**For the 1756-L50 Controller**

\[ 1200 + (A \times 130) + (B \times 140) + (C \times 90) \]

**For the 1756-L53 Controller**

\[ 1000 + (A \times 100) + (B \times 110) + (C \times 60) \]
Appendix B

Loop and Interconnect Diagrams

This appendix shows the loop interconnect diagrams for common motion configurations.

Understanding Block Diagrams

The control block diagrams in this section use the following terms for motion attributes.

<table>
<thead>
<tr>
<th>Diagram term</th>
<th>Motion attribute name (as used in the GSV and SSV instructions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc FF Gain</td>
<td>AccelerationFeedforwardGain</td>
</tr>
<tr>
<td>Vel FF Gain</td>
<td>VelocityFeedforwardGain</td>
</tr>
<tr>
<td>Pos P Gain</td>
<td>PositionProportionalGain</td>
</tr>
<tr>
<td>Pos I Gain</td>
<td>PositionIntegralGain</td>
</tr>
<tr>
<td>Vel P Gain</td>
<td>VelocityProportionalGain</td>
</tr>
<tr>
<td>Vel I Gain</td>
<td>VelocityIntegralGain</td>
</tr>
<tr>
<td>Output Filter BW</td>
<td>OutputFilterBandwidth</td>
</tr>
<tr>
<td>Output Scaling</td>
<td>OutputScaling</td>
</tr>
<tr>
<td>Friction Comp</td>
<td>FrictionCompensation</td>
</tr>
<tr>
<td>Output Limit</td>
<td>OutputLimit</td>
</tr>
<tr>
<td>Output Offset</td>
<td>OutputOffset</td>
</tr>
<tr>
<td>Position Error</td>
<td>PositionError</td>
</tr>
<tr>
<td>Position Integrator Error</td>
<td>PositionIntegratorError</td>
</tr>
<tr>
<td>Velocity Error</td>
<td>VelocityError</td>
</tr>
<tr>
<td>Velocity Integrator Error</td>
<td>VelocityIntegratorError</td>
</tr>
<tr>
<td>Velocity Feedback</td>
<td>VelocityFeedback</td>
</tr>
<tr>
<td>Velocity Command</td>
<td>VelocityCommand</td>
</tr>
<tr>
<td>Servo Output Level</td>
<td>ServoOutputLevel</td>
</tr>
<tr>
<td>Registration Position</td>
<td>RegistrationPosition</td>
</tr>
<tr>
<td>Watch Position</td>
<td>WatchPosition</td>
</tr>
</tbody>
</table>
Using a 1756-M02AE Module With a Torque Servo Drive

- Command Acceleration
- Acc FF Gain
- d/dt
- Command Velocity
- Vel FF Gain
- d/dt
- Velocity Command
- Vel P Gain
- Velocity Error
- Vel I Gain
- Error Accumulator
- Pos P Gain
- Error Integrator Error
- Pos I Gain
- Position Command
- Position Error
- Position Accumulator
- Velocity Feedback
- Velocity Error Accumulator
- Low Pass Filter
- Velocity Integrator Error
- d/dt
- Output Filter Bit
- Friction Comp.
- Output Offset & Servo Polarity
- Output Limit
- 16 Bit DAC
- Servo Output Level
- Torque Servo Drive
- Optical Encoder
- Servo Motor
- Marker Input
- Marker Event Handler
- Marker Latch
- Registration Event and Position
- Regist. Event Handler
- Regist. Latch
- Home Input
- Home Event
- Watch Event
- Watch Position
- Coarse Actual Position (Relative)
- Coarse Command Position (Relative)
Using a 1756-M02AE Module With a Velocity Servo Drive
Understanding Wiring Diagrams

Wiring to a Servo Module RTB

This is a general wiring example illustrating Axis 1 wiring only. Other configurations are possible with Axis 0 wiring identical to Axis 1.
Wiring to an Ultra 100 Series Drive

This is a general wiring example only. Other configurations are possible. For more information, refer to the Ultra 100 Series Drive Installation Manual, publication number 1398-5.2.
Wiring to an Ultra 200 Series Drive

This is a general wiring example only. Other configurations are possible. For more information, refer to the Ultra 200 Series Drive Installation Manual, publication number 1398-5.0.
1398-CFLAE Exx Cable Diagram

The 1398-CFLAE Cable is available in 10, 25, and 50 foot lengths.

Pinouts for 1398-CFLAE Exx Cable

<table>
<thead>
<tr>
<th>Wires Stripped Back .25 in.</th>
<th>Wires Terminated with Ferrules</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHT/ORG 22GA</td>
<td>GREEN 28GA</td>
</tr>
<tr>
<td>WHT/YEL 22GA</td>
<td>BLUE 28GA</td>
</tr>
<tr>
<td>DRAIN</td>
<td>VIOLET 28GA</td>
</tr>
<tr>
<td>TAN 28GA</td>
<td>GRAY 28GA</td>
</tr>
<tr>
<td>DRAIN</td>
<td>WHITE 28GA</td>
</tr>
<tr>
<td>WHT/RED 22GA</td>
<td>BLACK 28G</td>
</tr>
<tr>
<td>DRAIN</td>
<td></td>
</tr>
<tr>
<td>WHT/BLK 22GA</td>
<td></td>
</tr>
<tr>
<td>DRAIN</td>
<td></td>
</tr>
<tr>
<td>WHT/GRN 22GA</td>
<td></td>
</tr>
<tr>
<td>WHT/BLU 22GA</td>
<td></td>
</tr>
<tr>
<td>DRAIN</td>
<td></td>
</tr>
<tr>
<td>BROWN 28GA</td>
<td></td>
</tr>
<tr>
<td>RED 28GA</td>
<td></td>
</tr>
<tr>
<td>ORANGE 28GA</td>
<td></td>
</tr>
<tr>
<td>YELLOW 28GA</td>
<td></td>
</tr>
<tr>
<td>DRAIN</td>
<td></td>
</tr>
<tr>
<td>24VDC</td>
<td>Enable</td>
</tr>
<tr>
<td>24VCOM</td>
<td>Ready +</td>
</tr>
<tr>
<td>COMMAND +</td>
<td>Ready -</td>
</tr>
<tr>
<td>COMMAND -</td>
<td>24VCOM</td>
</tr>
</tbody>
</table>
Wiring to a 1394 Servo Drive (in Torque Mode only)

The wiring diagram illustrates Axis 1 wiring only. Other configurations are possible.

The 1394CCAExx cable is wired to connect to torque command reference input pins.

An external +5V power supply is required to power the encoder driver circuit of the 1394 servo drive. Because this connection is shared by all four axis encoder driver circuits, only one connection is needed to the +5V field supply.

The xx in the cable number is the length of the cable. Options are 5, 10, 25, and 50 feet.
The 1394-CFLAExx Cable Wiring Diagram

The 1394-CFLAE cable is available in 1, 3, 8, and 15 meter lengths.

Pinouts for the 1394-CFLAE

- +5V
- +5VCOM
- CHANNEL A HIGH
- CHANNEL A LOW
- CHANNEL B HIGH
- CHANNEL B LOW
- CHANNEL Z HIGH
- CHANNEL Z LOW
- VREF+
- TREF+
- VREF-
- TREF-
- (DROK-0)
- (24V EN COM)
- (24V)
- (AX-ENABLE)
- TO SYSTEM
- FAULT STRING
- RED 22GA
- BLACK 22GA
- DRAIN
- ORANGE 22GA
- WHT/ORG 22GA
- YELLOW 22GA
- WHT/YEL 22GA
- GREEN 22GA
- WHT/GRN 22GA
- BLUE 22GA
- WHT/BLU 22GA
- DRAIN
- VIOLET 22GA
- WHT/VIO 22GA
- GRAY 22GA
- WHT/GRY 22GA
- DRAIN
- RED 22GA
- BLACK 22GA
- DRAIN
Wiring Registration Sensors

The registration inputs to the servo module can support 24V or 5V registration sensors. These inputs should be wired to receive source current from the sensor. Current sinking sensor configurations are not allowed because the registration input common (IN_COM) is shared with the other 24V servo module inputs.

24V Registration Sensor

5V Registration Sensor
Wiring the Home Limit Switch Input

The home limit switch inputs to the servo module are designed for 24V nominal operation. These inputs should be wired for current sourcing operation.

Wiring the OK Contacts

A set of isolated solid-state OK relay contacts is provided for optional interface to an E-stop string, which controls power to the associated drives. The OK contacts are rated to drive an external 24V pilot relay (for example, Allen-Bradley 700-HA32Z24) whose contacts can be incorporated into the E-Stop string as shown below.
Appendix C

The Motion Control Structures

This appendix shows the structures for the AXIS, MOTION_GROUP, MOTION_INSTRUCTION, CAM, and CAM_PROFILE data tags.

The AXIS Structure

The AXIS structure contains status and configuration information for your motion axis. You can directly access this information in your application program. For example, if you want to use the AccelStatus attribute for Axis_X, you would use Axis_X.AccelStatus to gain access to the attribute.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.AccelStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if the axis has been commanded to accelerate. If neither this bit nor the .DecelStatus bit is set, the axis is running at the steady-state velocity or is at rest.</td>
</tr>
<tr>
<td>.ACAsyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of asynchronous communication. When the controller detects that one of the servo module parameters failed to update because of an asynchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
<tr>
<td>.ACSyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of synchronous communication. When the controller detects that the servo module has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
<tr>
<td>.AxisHomedStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of a homing sequence. During power-up or reconnection, the controller clears this bit. The Motion Axis Home (MAH) instruction sets this bit when a homing sequence completes successfully. After this bit sets, if the axis enters the shutdown state, the controller clears this bit.</td>
</tr>
<tr>
<td>.DecelStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if the axis has been commanded to decelerate. If neither this bit nor the .AccelStatus bit is set, the axis is running at the steady-state velocity or is at rest.</td>
</tr>
<tr>
<td>.DriveEnableStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the drive enable output. If this bit is set, you have activated the drive enable output for your axis. This bit is clear if you have deactivated the drive enable output for your axis.</td>
</tr>
<tr>
<td>.DriveFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the external drive. If this bit is set, the external drive detected a fault. This bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHALossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel A. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
</tbody>
</table>
### The Motion Control Structures

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.EncCHBLossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel B. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHZLossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel Z. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncNsFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of encoder channels A and B. If the servo module detects simultaneous transitions of channels A and B, this bit sets. This bit is clear after the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EventStatus</td>
<td>DINT</td>
<td>The servo event bits for your servo loop.</td>
</tr>
<tr>
<td>Bit</td>
<td>Number</td>
<td>Data type</td>
</tr>
<tr>
<td>.WatchEvArmStatus</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>.WatchEvStatus</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>.RegEvArmStatus</td>
<td>02</td>
<td>BOOL</td>
</tr>
<tr>
<td>.RegEvStatus</td>
<td>03</td>
<td>BOOL</td>
</tr>
<tr>
<td>.HomeEvArmStatus</td>
<td>04</td>
<td>BOOL</td>
</tr>
<tr>
<td>.HomeEvStatus</td>
<td>05</td>
<td>BOOL</td>
</tr>
<tr>
<td>Bits 06 through 31 are reserved.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.GearingStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if electronic gearing is enabled. If this bit is set, the axis is currently gearing to another axis. This bit is clear when the gearing operation stops or when another motion operation supersedes the gearing operation.</td>
</tr>
<tr>
<td>.GearingLockStatus</td>
<td>BOOL</td>
<td>You can use this bit to clutch to a new gear speed. This bit is set when the axis is clutching to a new gear speed. When the clutch ramp completes, the slave is locked to the master according to the gear ratio.</td>
</tr>
<tr>
<td>.Hardfault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the servo module. If this bit is set, the servo module detected a hardware problem that typically requires the replacement of the servo module.</td>
</tr>
<tr>
<td>.HomeEvArmStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of a homing event. If this bit is set, a Motion Axis Home (MAH) instruction has armed a home event. This bit clears when a home event occurs.</td>
</tr>
<tr>
<td>.HomeEvStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of a homing event. If this bit is set, a home event has occurred. This bit clears when another Motion Axis Home (MAH) instruction executes.</td>
</tr>
<tr>
<td>.HomingStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if a homing profile is in progress. If this bit is set, a homing profile is currently in progress. This bit is clear when the homing operation completes or when another motion operation supersedes the homing operation.</td>
</tr>
<tr>
<td>.JogStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if a jog profile is in progress. If this bit is set, a jog profile is currently in progress. This bit is clear when the jog completes or when another motion operation supersedes the jog operation.</td>
</tr>
<tr>
<td>.MotionFault</td>
<td>DINT</td>
<td>The motion fault bits for your axis.</td>
</tr>
<tr>
<td>Bit</td>
<td>Number</td>
<td>Data type</td>
</tr>
<tr>
<td>.ACAsyncConnFault</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>.ACSyncConnFault</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>Bits 02 through 31 are reserved.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MotionStatus

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.MotionStatus</td>
<td>DINT</td>
<td>The motion status bits for your axis.</td>
</tr>
<tr>
<td><strong>Bit</strong></td>
<td><strong>Number</strong></td>
<td><strong>Data type</strong></td>
</tr>
<tr>
<td>AccelStatus</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>DecelStatus</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>MoveStatus</td>
<td>02</td>
<td>BOOL</td>
</tr>
<tr>
<td>JogStatus</td>
<td>03</td>
<td>BOOL</td>
</tr>
<tr>
<td>GearingStatus</td>
<td>04</td>
<td>BOOL</td>
</tr>
<tr>
<td>HomingStatus</td>
<td>05</td>
<td>BOOL</td>
</tr>
<tr>
<td>StoppingStatus</td>
<td>06</td>
<td>BOOL</td>
</tr>
<tr>
<td>AxisHomedStatus</td>
<td>07</td>
<td>BOOL</td>
</tr>
<tr>
<td>PositionCamStatus</td>
<td>08</td>
<td>BOOL</td>
</tr>
<tr>
<td>TimeCamStatus</td>
<td>09</td>
<td>BOOL</td>
</tr>
<tr>
<td>PositionCamPendingStatus</td>
<td>10</td>
<td>BOOL</td>
</tr>
<tr>
<td>TimeCamPendingStatus</td>
<td>11</td>
<td>BOOL</td>
</tr>
<tr>
<td>GearingLockStatus</td>
<td>12</td>
<td>BOOL</td>
</tr>
<tr>
<td>PositionCamLockStatus</td>
<td>13</td>
<td>BOOL</td>
</tr>
<tr>
<td>MasterOffsetMoveStatus</td>
<td>14</td>
<td>BOOL</td>
</tr>
<tr>
<td>OutputCamStatus</td>
<td>set of bits*</td>
<td>DINT</td>
</tr>
<tr>
<td>OutputCamPendingStatus</td>
<td>set of bits*</td>
<td>DINT</td>
</tr>
<tr>
<td>OutputCamLockStatus</td>
<td>set of bits*</td>
<td>DINT</td>
</tr>
<tr>
<td>OutputCamTransitionStatus</td>
<td>set of bits*</td>
<td>DINT</td>
</tr>
</tbody>
</table>

* The bit number corresponds with the execution target number.

### MoveStatus

- **BOOL**: You can use this bit to determine if a move profile is in progress. If this bit is set, a move profile is currently in progress. This bit is clear when the move completes or when another motion operation supersedes the move operation.

### NOtrvlFault

- **BOOL**: You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumNegativeOvertravel value. This bit is clear when the axis moves within the MaximumNegativeOvertravel values.

### OutLmtStatus

- **BOOL**: You can use this bit to determine the status of servo loop output. If the magnitude of the servo loop output reaches or exceeds the OutputLimit value, this bit sets. This bit is clear when the magnitude of the servo loop output is within the OutputLimit value.

### PositionCamLockStatus

- **BOOL**: You can use this bit to determine the start status of the current Position Cam Profile. This bit is set when the master axis satisfies the starting condition of the currently active Position Cam Profile. The starting condition is established by the Start Control and the Start Position parameters of the MAPC instruction. This bit is cleared when the current position cam profile completes or is superseded by another motion operation.

### PositionCamPendingStatus

- **BOOL**: You can use this bit to see if a position cam motion profile is currently waiting for the completion of an executing cam profile. This bit is set when a Position Cam motion profile is waiting the completion of an executing cam profile. It is initiated by executing an MAPC instruction with Pending execution selected. As soon as the current position cam profile completes, the pending cam profile is started, and the bit is cleared. This bit is also cleared when the position cam profile completes, or is superseded by some other motion operation.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PositionCamStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine whether or not a Position Cam profile is currently in progress. This bit is set when a Position Cam motion profile is currently in progress. The Position Cam Status bit is cleared as soon as the position cam completes or is superseded by some other motion operation.</td>
</tr>
<tr>
<td>PosErrorFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis position error. If this bit is set, the servo module has detected that axis position error exceeds the PositionErrorTolerance value. This bit is clear when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>PosLockStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the axis position error. If this bit is set, the magnitude of the axis position error is less than or equal to the PositionLockTolerance value. This bit is clear when the magnitude of the axis position error is greater than the PositionLockTolerance value.</td>
</tr>
<tr>
<td>POtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumPositiveOvertravel value. This bit is clear when the axis moves within the MaximumPositiveOvertravel values</td>
</tr>
<tr>
<td>RegEvArmStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of a registration event. If this bit is set, the execution of a Motion Arm Registration (MAR) instruction has armed a registration event. This bit clears when a registration event occurs or the controller executes a Motion Disarm Registration (MDR) instruction.</td>
</tr>
<tr>
<td>RegEvStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of a registration event. If this bit is set, a registration event has occurred. This bit clears when the controller executes another Motion Arm Registration (MAR) instruction or a Motion Disarm Registration (MDR) instruction.</td>
</tr>
<tr>
<td>ServoActStatus</td>
<td>BOOL</td>
<td>You can use this bit to determine if servo action is enabled for your axis. If this bit is set, servo action is currently enabled. This bit is clear when servo action is disabled.</td>
</tr>
<tr>
<td>ServoFault</td>
<td>DINT</td>
<td>The servo fault bits for your servo loop.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Number</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>POtrvlFault</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>NOtrvlFault</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>PosErrorFault</td>
<td>02</td>
<td>BOOL</td>
</tr>
<tr>
<td>EncCHALossFault</td>
<td>03</td>
<td>BOOL</td>
</tr>
<tr>
<td>EncCHBLossFault</td>
<td>04</td>
<td>BOOL</td>
</tr>
<tr>
<td>EncCHZLossFault</td>
<td>05</td>
<td>BOOL</td>
</tr>
<tr>
<td>EncNsFault</td>
<td>06</td>
<td>BOOL</td>
</tr>
<tr>
<td>DriveFault</td>
<td>07</td>
<td>BOOL</td>
</tr>
<tr>
<td>SyncConnFault</td>
<td>08</td>
<td>BOOL</td>
</tr>
<tr>
<td>Hardfault</td>
<td>09</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

Bits 10 through 31 are reserved.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ServoStatus</td>
<td>DINT</td>
<td>The status bits for your servo loop.</td>
</tr>
<tr>
<td>Bit</td>
<td></td>
<td>Number</td>
</tr>
<tr>
<td>.ServoActStatus</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>.DriveEnableStatus</td>
<td></td>
<td>01</td>
</tr>
<tr>
<td>.OutLmtStatus</td>
<td></td>
<td>02</td>
</tr>
<tr>
<td>.PosLockStatus</td>
<td></td>
<td>03</td>
</tr>
<tr>
<td>.TuneStatus</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>.TestStatus</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>.ShutdownStatus</td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

Bits 04 through 12 and bits 16 through 31 are reserved.

| .ShutdownStatus  | BOOL      | You can use this bit to determine if your axis is in the shutdown state. If this bit is set, the axis is in the shutdown state. This bit is clear when the axis transitions from the shutdown state to another state. |
| .StoppingStatus  | BOOL      | You can use the Stopping Status bit attribute to see if there is a stopping process currently in progress. This bit is set when a stopping process is in progress. The bit is cleared as soon as the stopping process is complete. The stopping process is used to stop an axis (initiated by an M AS, M GS, M GPS, Stop M otion fault action, or mode change). |
| .SyncConnFault    | BOOL      | You can use this bit to determine the status of synchronous communication. When the servo module detects that it has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears. |
| .TestStatus      | BOOL      | You can use this bit to determine the status of diagnostic tests. If this bit is set, a diagnostic test operation is in progress for the servo module. |
| .TimeCamPendingStatus | BOOL | You can use the Time Cam Pending Status bit to see if a Time Cam motion profile is waiting for an executing cam profile to finish. This bit is set when a Time Cam motion profile is currently waiting for the currently executing cam profile to complete. This is initiated by executing an MATC instruction with Pending execution selected. The Time Cam Pending bit is cleared as soon as the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation. |
| .TimeCamStatus   | BOOL      | You can use the Time Cam Status bit to see if a Time Cam motion profile is currently being executed. This bit is set when a Time Cam motion profile is currently in progress. The Time Cam Status bit is cleared as soon as the Time Cam is complete or superseded by another motion operation. |
| .TuneStatus      | BOOL      | You can use this bit to determine the status of axis tuning. If this bit is set, an auto tuning operation is in progress for the servo module. |
You can use the servo configuration update status bits attributes to monitor the progress of servo configuration attribute updates, which are initiated by an SSV instruction in your application program.

When the SSV instruction initiates an update, the controller sets the update status bit associated with the attribute. The update status bit remains set until the servo module indicates that the data update was successful.

For example, if you use an SSV instruction to change the PositionProportionalGain attribute of an axis and follow it with logic based on the completion of the SSV instruction, you can check for the resetting of the .PosPGainStatus bit to ensure that the servo module attribute is updated.

The following is a list of the servo configuration update status bits attributes.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.AccFfGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the AccelerationFeedforwardGain attribute.</td>
</tr>
<tr>
<td>.AxisTypeStatus</td>
<td>BOOL</td>
<td>The status of an update to the AxisType attribute.</td>
</tr>
<tr>
<td>.DriveFaultActStatus</td>
<td>BOOL</td>
<td>The status of an update to the DriveFaultAction attribute.</td>
</tr>
</tbody>
</table>
The MOTION_GROUP Structure

The MOTION_GROUP structure contains status and configuration information for your motion group. You can directly access this information in your motion control program. For example, if you want to use the DriveFault attribute for Motion_Group, you would use Motion_Group.DriveFault to gain access to the attribute.

The bits in the MOTION_GROUP structure are set when any axis in the group experiences the conditions required to set the bit. For example, if one axis in a group of ten axes developed the conditions to set the .POtrvlFault bit, the controller would set the .POtrvlFault bit in the MOTION_GROUP structure.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.FricCompStatus</td>
<td>BOOL</td>
<td>The status of an update to the FrictionCompensation attribute.</td>
</tr>
<tr>
<td>.MaxNTrvlStatus</td>
<td>BOOL</td>
<td>The status of an update to the MaximumNegativeTravel attribute.</td>
</tr>
<tr>
<td>.MaxPTrvlStatus</td>
<td>BOOL</td>
<td>The status of an update to the MaximumPositiveTravel attribute.</td>
</tr>
<tr>
<td>.OutFiltBWStatus</td>
<td>BOOL</td>
<td>The status of an update to the OutputFilterBandwidth attribute.</td>
</tr>
<tr>
<td>.OutLimitStatus</td>
<td>BOOL</td>
<td>The status of an update to the OutputLimit attribute.</td>
</tr>
<tr>
<td>.OutOffsetStatus</td>
<td>BOOL</td>
<td>The status of an update to the OutputOffset attribute.</td>
</tr>
<tr>
<td>.OutScaleStatus</td>
<td>BOOL</td>
<td>The status of an update to the OutputScaling attribute.</td>
</tr>
<tr>
<td>.PosErrorTolStatus</td>
<td>BOOL</td>
<td>The status of an update to the PositionErrorTolerance attribute.</td>
</tr>
<tr>
<td>.PosGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the PositionIntegralGain attribute.</td>
</tr>
<tr>
<td>.PosLockTolStatus</td>
<td>BOOL</td>
<td>The status of an update to the PositionLockTolerance attribute.</td>
</tr>
<tr>
<td>.PosPGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the PositionProportionalGain attribute.</td>
</tr>
<tr>
<td>.PosUnwindStatus</td>
<td>BOOL</td>
<td>The status of an update to the PositionUnwind attribute.</td>
</tr>
<tr>
<td>.POtrvlFactActStatus</td>
<td>BOOL</td>
<td>The status of an update to the SoftOvertravelFaultAction attribute.</td>
</tr>
<tr>
<td>.VelFFGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the VelocityFeedforwardGain attribute.</td>
</tr>
<tr>
<td>.VelIGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the VelocityIntegralGain attribute.</td>
</tr>
<tr>
<td>.VelPGainStatus</td>
<td>BOOL</td>
<td>The status of an update to the VelocityProportionalGain attribute.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ACAsyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of asynchronous communication. When the controller detects that one of the servo module parameters failed to update because of an asynchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
<tr>
<td>.ACSyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of synchronous communication. When the controller detects that the servo module has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>.DriveFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the external drive. If this bit is set, the external drive detected a fault. This bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHA LossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel A. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHB LossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel B. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHZ LossFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the encoder channel Z. This bit sets if both of the differential signals are at the same level or if the servo module or encoder loses encoder power or common. The bit clears when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncN s Fault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of encoder channels A and B. If the servo module detects simultaneous transitions of channels A and B, this bit sets. This bit is clear after the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.GroupFault</td>
<td>DINT</td>
<td>The fault bits for your motion group.</td>
</tr>
<tr>
<td>Bit Number</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>GroupOverlapFault</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>GroupOverlapFault</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>GroupStatus</td>
<td>DINT</td>
<td>The status bits for your motion group.</td>
</tr>
<tr>
<td>Bit Number</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>.InhibitStatus</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>.GroupSynced</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td>GroupSynced</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the group connection to the controller. This bit is set the first time all the axes in a group are connected and synchronized to the controller. This bit remains set until you download a new program, clear the controller memory, or powercycle the controller.</td>
</tr>
<tr>
<td>Hardfault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the servo module. If this bit is set, the servo module detected a hardware problem that typically requires the replacement of the servo module.</td>
</tr>
<tr>
<td>InhibitStatus</td>
<td>BOOL</td>
<td>Not used by the controller.</td>
</tr>
<tr>
<td>MotionFault</td>
<td>DINT</td>
<td>The motion fault bits for your axis.</td>
</tr>
<tr>
<td>Bit Number</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>.ACAsyncConnFault</td>
<td>00</td>
<td>BOOL</td>
</tr>
<tr>
<td>.ACSyncConnFault</td>
<td>01</td>
<td>BOOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bits 02 through 31 are reserved.</td>
</tr>
</tbody>
</table>
The MOTION_INSTRUCTION Structure

The controller uses the MOTION_INSTRUCTION tag (structure) to store status information during the execution of motion instructions.

### The Motion Control Structures

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.NOtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumNegativeOvertravel value. This bit is clear when the axis moves within the MaximumNegativeOvertravel values</td>
</tr>
<tr>
<td>.PosErrorFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis position error. If this bit is set, the servo module has detected that axis position error exceeds the PositionErrorTolerance value. This bit is clear when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.POtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumPositiveOvertravel value. This bit is clear when the axis moves within the MaximumPositiveOvertravel values</td>
</tr>
<tr>
<td>.ServoFault</td>
<td>DINT</td>
<td>The servo fault bits for your servo loop.</td>
</tr>
<tr>
<td>.SyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of synchronous communication. When the servo module detects that it has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.NOtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumNegativeOvertravel value. This bit is clear when the axis moves within the MaximumNegativeOvertravel values</td>
</tr>
<tr>
<td>.POtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumPositiveOvertravel value. This bit is clear when the axis moves within the MaximumPositiveOvertravel values</td>
</tr>
<tr>
<td>.ServoFault</td>
<td>DINT</td>
<td>The servo fault bits for your servo loop.</td>
</tr>
<tr>
<td>.SyncConnFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of synchronous communication. When the servo module detects that it has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
</tbody>
</table>

### Variable Table

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.NOtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumNegativeOvertravel value. This bit is clear when the axis moves within the MaximumNegativeOvertravel values</td>
</tr>
<tr>
<td>.POtrvlFault</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumPositiveOvertravel value. This bit is clear when the axis moves within the MaximumPositiveOvertravel values</td>
</tr>
<tr>
<td>.ServoFault</td>
<td>DINT</td>
<td>The servo fault bits for your servo loop.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Number</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.POtrvlFault</td>
<td>00</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumPositiveOvertravel value. This bit is clear when the axis moves within the MaximumPositiveOvertravel values</td>
</tr>
<tr>
<td>.NOtrvlFault</td>
<td>01</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis travel. If this bit is set, the axis has moved or has attempted to move beyond the MaximumNegativeOvertravel value. This bit is clear when the axis moves within the MaximumNegativeOvertravel values</td>
</tr>
<tr>
<td>.PosErrorFault</td>
<td>02</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of axis position error. If this bit is set, the servo module has detected that axis position error exceeds the PositionErrorTolerance value. This bit is clear when the controller executes a Motion Axis Fault Reset (MAFR) instruction.</td>
</tr>
<tr>
<td>.EncCHALossFault</td>
<td>03</td>
<td>BOOL</td>
<td>The encoder channel A has lost its synchronization. This bit is set when the encoder channel A loses its synchronization with the servo loop.</td>
</tr>
<tr>
<td>.EncCHBLossFault</td>
<td>04</td>
<td>BOOL</td>
<td>The encoder channel B has lost its synchronization. This bit is set when the encoder channel B loses its synchronization with the servo loop.</td>
</tr>
<tr>
<td>.EncCHZLossFault</td>
<td>05</td>
<td>BOOL</td>
<td>The encoder channel Z has lost its synchronization. This bit is set when the encoder channel Z loses its synchronization with the servo loop.</td>
</tr>
<tr>
<td>.EncNsFault</td>
<td>06</td>
<td>BOOL</td>
<td>The encoder channel NS has lost its synchronization. This bit is set when the encoder channel NS loses its synchronization with the servo loop.</td>
</tr>
<tr>
<td>.DriveFault</td>
<td>07</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the drive. If this bit is set, the drive has failed or has attempted to move beyond the MaximumOvertravel value. This bit is clear when the drive moves within the MaximumOvertravel values</td>
</tr>
<tr>
<td>.SyncConnFault</td>
<td>08</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of synchronous communication. When the servo module detects that it has missed several position updates in a row because of a synchronous communication failure, this bit sets. When you reestablish the connection, the bit clears.</td>
</tr>
<tr>
<td>.HardFault</td>
<td>09</td>
<td>BOOL</td>
<td>You can use this bit to determine the status of the hard fault. If this bit is set, the hard fault has occurred. This bit is clear when the hard fault is cleared.</td>
</tr>
</tbody>
</table>

Bits 10 through 31 are reserved.
Every motion instruction has a motion control parameter that requires a MOTION_INSTRUCTION tag for this purpose.

The motion control parameter

**WARNING** Tags used for the motion control attribute of instructions should only be used once. Re-use of the motion control attribute in other instructions can cause unintended operation of the control variables.

The structure of the motion instruction structure is shown below:

<table>
<thead>
<tr>
<th>bit number</th>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>16</th>
<th>15</th>
<th>.DECEL</th>
<th>ACCEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>DN</td>
<td>ER</td>
<td>IP</td>
<td>PC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>error code (.ERR) (16 bits)</td>
<td>message</td>
<td>execution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mnemonic</td>
<td>Data Type</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.EN</td>
<td>BOOL</td>
<td>The enable bit indicates that the instruction is enabled (the rung-in and rung-out condition is true).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.DN</td>
<td>BOOL</td>
<td>The done bit indicates that all calculations and messaging (if any) are complete.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.ER</td>
<td>BOOL</td>
<td>The error bit indicates when the instruction is used illegally.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IP</td>
<td>BOOL</td>
<td>The in process bit indicates that a process is being executed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.PC</td>
<td>BOOL</td>
<td>The process complete bit indicates that the operation is complete. The .DN bit sets when an instruction has completed execution. The .PC bit sets when the initiated process has completed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.ACCEL</td>
<td>BOOL</td>
<td>The .ACCEL bit indicates that the velocity has increased for the individual instruction that it is tied to i.e jog, move, gearing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.DECEL</td>
<td>BOOL</td>
<td>The .DECEL bit indicates that the velocity has decreased for the individual instruction that it is tied to i.e jog, move, gearing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.ERR</td>
<td>INT</td>
<td>The error value contains the error code associated with a motion function. See page 1-8.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.STATUS</td>
<td>SINT</td>
<td>The message status value indicates the status condition of any message associated with the motion function. See page 1-10.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.STATE</td>
<td>SINT</td>
<td>The execution status value keeps track of the execution state of a function. Many motion functions have several steps and this value tracks these steps. See page 1-10.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.SEGMENT</td>
<td>DINT</td>
<td>A segment is the distance from one point up to but, not including the next point. A .SEGMENT gives the relative position by segment number as the Cam is executing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Error codes (.ERR)

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Execution Collision</td>
<td>The instruction tried to execute while another instance of this instruction was executing. This can occur when the controller executes a messaging instruction without checking the .DN bit of the preceding instruction.</td>
</tr>
<tr>
<td>4</td>
<td>Servo On State Error</td>
<td>The instruction tried to execute on an axis with a closed servo loop.</td>
</tr>
<tr>
<td>5</td>
<td>Servo Off State Error</td>
<td>The instruction tried to execute on an axis with a servo loop that is not closed.</td>
</tr>
<tr>
<td>6</td>
<td>Drive On State Error</td>
<td>The axis drive is enabled.</td>
</tr>
<tr>
<td>7</td>
<td>Shutdown State Error</td>
<td>The axis is in the shutdown state.</td>
</tr>
<tr>
<td>8</td>
<td>Illegal Axis Type</td>
<td>The configured axis type is not correct.</td>
</tr>
<tr>
<td>9</td>
<td>Overtravel Condition</td>
<td>The instruction tried to execute in a direction that aggravates the current overtravel condition.</td>
</tr>
<tr>
<td>10</td>
<td>Master Axis Conflict</td>
<td>The master axis reference is the same as the slave axis reference.</td>
</tr>
<tr>
<td>11</td>
<td>Axis Not Configured</td>
<td>The axis is not configured.</td>
</tr>
<tr>
<td>12</td>
<td>Servo Message Failure</td>
<td>Messaging to the servo module failed.</td>
</tr>
<tr>
<td>13</td>
<td>Parameter Out Of Range</td>
<td>The instruction tried to use a parameter that is outside the range limit.</td>
</tr>
<tr>
<td>14</td>
<td>Tune Process Error</td>
<td>The instruction cannot apply the tuning parameters because of an error in the run tuning instruction.</td>
</tr>
<tr>
<td>15</td>
<td>Test Process Error</td>
<td>The instruction cannot apply the diagnostic parameters because of an error in the run diagnostic test instruction.</td>
</tr>
<tr>
<td>16</td>
<td>Home In Process Error</td>
<td>The instruction tried to execute with homing in progress.</td>
</tr>
<tr>
<td>17</td>
<td>Axis Mode Not Rotary</td>
<td>The instruction tried to execute a rotary move on an axis that is not configured for rotary operation.</td>
</tr>
<tr>
<td>18</td>
<td>Axis Type Unused</td>
<td>The axis type is configured as unused.</td>
</tr>
<tr>
<td>19</td>
<td>Group Not Synchronized</td>
<td>The motion group is not in the synchronized state. This could be caused by a missing servo module or a misconfiguration.</td>
</tr>
<tr>
<td>20</td>
<td>Axis In Faulted State</td>
<td>The axis is in the faulted state.</td>
</tr>
<tr>
<td>21</td>
<td>Group In Faulted State</td>
<td>The group is in the faulted state.</td>
</tr>
<tr>
<td>22</td>
<td>Axis In Motion</td>
<td>An MSO (Motion Servo On) or MAH (Motion Axis Home) instruction was attempted while the axis was in motion.</td>
</tr>
<tr>
<td>23</td>
<td>Illegal Dynamic Change</td>
<td>An instruction attempted an illegal change of dynamics.</td>
</tr>
<tr>
<td>24</td>
<td>Illegal AC Mode Op</td>
<td>The controller attempted to execute an M DO, M SO, M AH, M AJ, M AM, M CD, M AP, M ATC, M AG, M RAT, or M RHD instruction when the controller was in the test mode.</td>
</tr>
<tr>
<td>25</td>
<td>Illegal Instruction</td>
<td>You attempted to execute an instruction that is not correct.</td>
</tr>
<tr>
<td>26</td>
<td>Illegal Cam Length</td>
<td>The cam array is of an illegal length.</td>
</tr>
<tr>
<td>27</td>
<td>Illegal Cam Profile Length</td>
<td>The cam profile array is of an illegal length.</td>
</tr>
<tr>
<td>28</td>
<td>Illegal Cam Type</td>
<td>You have an illegal segment type in the cam element.</td>
</tr>
<tr>
<td>29</td>
<td>Illegal Cam Order</td>
<td>You have an illegal order of cam elements.</td>
</tr>
<tr>
<td>30</td>
<td>Cam Profile Being Calculated</td>
<td>You tried to execute a cam profile while it is being calculated.</td>
</tr>
<tr>
<td>31</td>
<td>Cam Profile Being Used</td>
<td>The cam profile array you tried to execute is in use.</td>
</tr>
<tr>
<td>32</td>
<td>Cam Profile Not Calculated</td>
<td>The cam profile array you tried to execute has not been calculated.</td>
</tr>
<tr>
<td>33</td>
<td>Position Cam Not Enabled</td>
<td>It attempted to execute an MAH instruction without a position cam in process.</td>
</tr>
</tbody>
</table>
The Motion Control Structures

Message status (.STATUS)

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Registration in Progress</td>
<td>A MAH instruction is trying to start while a registration is already running.</td>
</tr>
<tr>
<td>35</td>
<td>Illegal Execution Target</td>
<td>Either the Logix controller or the Output Cam module does not support the specified Output Cam, axis, input or output.</td>
</tr>
<tr>
<td>36</td>
<td>Illegal Output Cam</td>
<td>Either the size of the Output Cam array is not supported or the value of one of its members is out of range.</td>
</tr>
<tr>
<td>37</td>
<td>Illegal Output Compensation</td>
<td>Either the size of the Output Compensation array is not supported or the value of one of its members is out of range.</td>
</tr>
</tbody>
</table>

Execution status (.STATE)

The execution status is always set to 0 when the controller sets the .EN bit for a motion instruction. Other execution states depend on the motion instruction.

Profile Segment (.SEGMENT)

A segment is the distance from one point up to but, not including the next point. A .SEGMENT instruction gives the relative position by segment number as the Cam is executing.

CAM Structure

The Cam data type consists of slave and master point pairs as well as an interpolation type. Since there is no association with a specific axis position or time, the point values are unit-less. The interpolation type
can be specified for each segment as either linear or cubic. The format of the cam array element is shown in the following table.

<table>
<thead>
<tr>
<th>Mnemonic:</th>
<th>Data Type:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER</td>
<td>REAL</td>
<td>The x value of the point.</td>
</tr>
<tr>
<td>SLAVE</td>
<td>REAL</td>
<td>The y value of the point.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Segment Type</th>
<th>DINT</th>
<th>The type of interpolation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>linear.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>cubic.</td>
<td></td>
</tr>
</tbody>
</table>

**CAM_PROFILE Structure**

The CAM_PROFILE data type is an array of coefficients representing a calculated cam profile that can be used as input to a time cam or position cam instruction. The only element available to the user is Status which is defined in the following table.

<table>
<thead>
<tr>
<th>Mnemonic:</th>
<th>Data Type:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>DINT</td>
<td>The status parameter is used to indicate that the Cam Profile array element has been calculated. If execution of a camming instruction is attempted using an uncalculated element in a Cam Profile, the instruction produces an error.</td>
</tr>
<tr>
<td>Value:</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Cam profile element has not been calculated.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Cam profile element is being calculated.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Cam profile element has been calculated.</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>Cam profile element has been calculated and is currently being used by (n-2) MAPC and MATC instructions.</td>
<td></td>
</tr>
</tbody>
</table>
The Motion Attributes

This appendix describes the motion attributes, their data types, and their access rules.

The Logix5550 controller stores motion status and configuration information in the AXIS and MOTION_GROUP objects. To directly access this information, you can select the object (AXIS or MOTION_GROUP) and select the attribute. You can also use the GSV and SSV instructions to access these objects. See Input/Output Instructions in the Logix550 Controller Instruction Set Reference Manual, publication 1756-6.4.1 for more information about the GSV and SSV instructions.

Motion Instance Variables

To use the motion instance variables, choose AXIS from the object list of the GSV and SSV instructions.

When an attribute is marked with an asterisk (*), it means that the attribute is located in both the ControlLogix controller and in the motion module. When you use an SSV instruction to write one of these values, the controller will automatically update the copy in the module. However, this process is not immediate. To be sure that the new value has been updated in the module, use an interlock mechanism using the boolean bits in the Servo Configuration Update Status Bits of the AXIS structure.

For example, if you perform an SSV instruction on the PositionLockTolerance, the PositionLockTolStatus of the Axis tag will be set until an update to the module is successful. Therefore, the logic following the SSV could wait on this bit resetting before continuing in the program.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* AccelerationFeedforwardGain</td>
<td>REAL</td>
<td>GSV, SSV</td>
<td>The value used to provide the torque command output to generate the command acceleration.</td>
</tr>
<tr>
<td>ActualPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The actual position of your axis.</td>
</tr>
<tr>
<td>ActualVelocity</td>
<td>REAL</td>
<td>GSV</td>
<td>The actual velocity of your axis. The internal resolution limit of the actual velocity is 1 encoder count per coarse update.</td>
</tr>
<tr>
<td>AverageVelocity</td>
<td>REAL</td>
<td>GSV</td>
<td>The average velocity of your axis.</td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AverageVelocityTimebase</td>
<td>REAL</td>
<td>GSV</td>
<td>The timebase of the average velocity of your axis.</td>
</tr>
<tr>
<td>AxisConfigurationState</td>
<td>SINT</td>
<td>GSV</td>
<td>The state of the axis configuration.</td>
</tr>
<tr>
<td>* AxisType</td>
<td>INT</td>
<td>GSV</td>
<td>The type of axis that you are using.</td>
</tr>
<tr>
<td>CommandPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The command position of your axis.</td>
</tr>
<tr>
<td>CommandVelocity</td>
<td>REAL</td>
<td>GSV</td>
<td>The command velocity of your axis.</td>
</tr>
<tr>
<td>ConversionConstant</td>
<td>REAL</td>
<td>GSV</td>
<td>The conversion factor used to convert from your units to feedback counts.</td>
</tr>
<tr>
<td>DampingFactor</td>
<td>REAL</td>
<td>GSV</td>
<td>The value used in calculating the maximum position servo bandwidth.</td>
</tr>
<tr>
<td>* DriveFaultAction</td>
<td>SINT</td>
<td>GSV</td>
<td>The operation performed when a drive fault occurs.</td>
</tr>
<tr>
<td>EffectiveInertia</td>
<td>REAL</td>
<td>GSV</td>
<td>The inertia value for the axis as calculated from the measurements the controller made during the last Motion Run Axis Tuning (M RAT) instruction.</td>
</tr>
<tr>
<td>* EncoderLossFaultAction</td>
<td>SINT</td>
<td>GSV</td>
<td>The operation performed when an encoder loss fault occurs.</td>
</tr>
<tr>
<td>* EncoderNoiseFaultAction</td>
<td>SINT</td>
<td>GSV</td>
<td>The operation performed when an encoder noise fault occurs.</td>
</tr>
<tr>
<td>* FrictionCompensation</td>
<td>REAL</td>
<td>GSV</td>
<td>The fixed output level used to compensate for static friction.</td>
</tr>
<tr>
<td>GroupInstance</td>
<td>DINT</td>
<td>GSV</td>
<td>The instance number of the motion group that contains your axis.</td>
</tr>
<tr>
<td>HomeMode</td>
<td>SINT</td>
<td>GSV</td>
<td>The homing mode for your axis.</td>
</tr>
<tr>
<td>HomePosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The homing position of your axis.</td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HomeReturnSpeed</td>
<td>REAL</td>
<td>GSV</td>
<td>The homing return speed of your axis.</td>
</tr>
<tr>
<td>HomeSequenceType</td>
<td>SINT</td>
<td>GSV</td>
<td>The homing sequence type for your axis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSV</td>
<td>Value: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0  immediate homing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  switch homing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2  marker homing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3  switch-marker homing (default)</td>
</tr>
<tr>
<td>HomeSpeed</td>
<td>REAL</td>
<td>GSV</td>
<td>The homing speed of your axis.</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>DINT</td>
<td>GSV</td>
<td>The instance number of the axis.</td>
</tr>
<tr>
<td>MapTableInstance</td>
<td>DINT</td>
<td>GSV</td>
<td>The I/O map instance of the servo module.</td>
</tr>
<tr>
<td>MapTableInstance</td>
<td></td>
<td>SSV</td>
<td>This attribute can only be set if you did not assign the axis to a group or</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>if you assigned it to a group in the group inhibit mode.</td>
</tr>
<tr>
<td>MaximumAcceleration</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum acceleration of your axis.</td>
</tr>
<tr>
<td>MaximumDeceleration</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum deceleration of your axis.</td>
</tr>
<tr>
<td>MaximumNegativeTravel</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum negative travel limit.</td>
</tr>
<tr>
<td>MaximumPositiveTravel</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum positive travel limit.</td>
</tr>
<tr>
<td>MaximumSpeed</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum speed of your axis.</td>
</tr>
<tr>
<td>ModuleChannel</td>
<td>SINT</td>
<td>GSV</td>
<td>The module channel of your servo module.</td>
</tr>
<tr>
<td>MotionConfigurationBits</td>
<td>DINT</td>
<td>GSV</td>
<td>The motion configuration bits for your axis.</td>
</tr>
<tr>
<td>MotionFaultBits</td>
<td>DINT</td>
<td>AXIS</td>
<td>The motion fault bits for your axis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
<td>Bit: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0  home direction reverse</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  home switch normally closed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2  home marker edge negative</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BitName: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0  ACAsyncConnFault asynchronous connection fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  ACSyncConnFault synchronous connection fault</td>
</tr>
</tbody>
</table>
### Variable | Data Type | Access | Description
---|---|---|---
MotionStatusBits | DINT | AXIS structure | The motion status bits for your axis.  

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AccelStatus</td>
<td>velocity increase</td>
</tr>
<tr>
<td>1</td>
<td>DecelStatus</td>
<td>velocity decrease</td>
</tr>
<tr>
<td>2</td>
<td>MoveStatus</td>
<td>move motion profile in progress</td>
</tr>
<tr>
<td>3</td>
<td>JogStatus</td>
<td>jog motion profile in progress</td>
</tr>
<tr>
<td>4</td>
<td>GearingStatus</td>
<td>axis is gearing to another axis</td>
</tr>
<tr>
<td>5</td>
<td>HomingStatus</td>
<td>home motion profile in progress</td>
</tr>
<tr>
<td>6</td>
<td>StoppingStatus</td>
<td>stopping process in progress</td>
</tr>
<tr>
<td>7</td>
<td>AxisHomedStatus</td>
<td>absolute position reference established</td>
</tr>
<tr>
<td>8</td>
<td>PositionCamStatus</td>
<td>Pcam in progress</td>
</tr>
<tr>
<td>9</td>
<td>TimeCamStatus</td>
<td>Tcam in progress</td>
</tr>
<tr>
<td>10</td>
<td>PositionCamPendingStatus</td>
<td>Pcam profile waiting for another to end</td>
</tr>
<tr>
<td>11</td>
<td>TimeCamPendingStatus</td>
<td>Tcam profile waiting for another to end</td>
</tr>
<tr>
<td>12</td>
<td>GearingLockedStatus</td>
<td>clutching to a new gear speed</td>
</tr>
<tr>
<td>13</td>
<td>PositionCamLockStatus</td>
<td>master axis meets Pcam condition</td>
</tr>
</tbody>
</table>

MotorEncoderTestIncrement | REAL | GSV SSV | The amount of motion that is necessary to initiate the Motion Run Hookup Diagnostic (MRHD) test.  

* OutputFilterBandwidth | REAL | GSV SSV | The bandwidth of the servo low-pass digital output filter.  

* OutputLimit | REAL | GSV SSV | The value of the maximum servo output voltage of your axis.  

* OutputOffset | REAL | GSV SSV | The value used to offset the effects of the cumulative offsets of the servo module DAC output and the servo drive input.  

* OutputScaling | REAL | GSV SSV | The value used to convert the output of the servo loop into the equivalent voltage to the drive.  

<table>
<thead>
<tr>
<th>For a velocity servo drive, the output scaling is:</th>
</tr>
</thead>
</table>
| \[
\text{Output Scaling} = \frac{10 \text{Volts}}{\text{Speed at 10 Volts} \times \text{Conversion Constant}}
\]|

<table>
<thead>
<tr>
<th>For a torque servo drive, the output scaling is:</th>
</tr>
</thead>
</table>
| \[
\text{Output Scaling} = \frac{10 \text{Volts}}{\text{Acceleration at 10 Volts} \times \text{Conversion Constant}}
\]|

PositionError | REAL | GSV | The difference between the actual and command position of an axis.  

You can use this value to drive the motor to where the actual position equals the command position.  

* PositionErrorFaultAction | SINT | GSV SSV | The operation performed when a position error fault occurs.  

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>shutdown the axis</td>
</tr>
<tr>
<td>1</td>
<td>disable the drive</td>
</tr>
<tr>
<td>2</td>
<td>stop the commanded motion</td>
</tr>
<tr>
<td>3</td>
<td>change the status bit only</td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>* PositionErrorTolerance</td>
<td>REAL</td>
</tr>
<tr>
<td>* PositionIntegralGain</td>
<td>REAL</td>
</tr>
<tr>
<td>PositionIntegratorError</td>
<td>REAL</td>
</tr>
<tr>
<td>PositionLockTolerance</td>
<td>REAL</td>
</tr>
<tr>
<td>* PositionProportionalGain</td>
<td>REAL</td>
</tr>
<tr>
<td>PositionServoBandwidth</td>
<td>REAL</td>
</tr>
<tr>
<td>* PositionUnwind</td>
<td>DINT</td>
</tr>
<tr>
<td>ProgrammedStopMode</td>
<td>SINT</td>
</tr>
</tbody>
</table>
| RegistrationPosition           | REAL      | GSV    | The registration position for your axis. You can use the following equation to determine the maximum registration position error based on your axis speed:  
\[
\text{MaximumSpeed} \left( \frac{\text{PositionUnits}}{\text{Seconds}} \right) = \frac{\text{Accuracy}(\text{PositionUnits})}{0.000001\text{Seconds}}
\]  |
<p>| * ServoConfigurationBits       | DINT      | GSV    | The servo configuration bits for your servo loop. Bit: Meaning: 0 rotary axis 1 external velocity servo drive 2 encoder polarity negative 3 servo polarity negative 4 soft overtravel checking 5 position error checking 6 encoder loss fault checking 7 encoder noise fault checking 8 drive fault checking 9 drive fault normally closed |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Access</th>
<th>Description</th>
<th>Bit: Bit Name: Meaning:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServoConfigurationUpdateBits</td>
<td>DINT</td>
<td>AXIS</td>
<td>The servo configuration status bits for your servo loop.</td>
<td>Bit: Bit Name: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
<td></td>
<td>0 AxisTypeStatus: axis type</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 PosUnwindStatus: position unwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 MaxPTvlStatus: maximum positive travel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 MaxNTvlStatus: maximum negative travel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 PosErrorTolStatus: position error tolerance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 PosLockTolStatus: position lock tolerance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 PosPGainStatus: position proportional gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7 PosiGainStatus: position integral gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 VelFFGainStatus: velocity feedforward gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9 AccFFGainStatus: acceleration feedforward gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10 VelPGainStatus: velocity proportional gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11 VellGainStatus: velocity integral gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12 OutFiltBwStatus: output filter bandwidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13 OutScaleStatus: output scaling</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14 OutLimitStatus: output limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 OutOffsetStatus: output offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16 FricCompStatus: friction compensation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17 P0trvlFaultActStatus: soft overtravel fault action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18 PosErrorFaultActStatus: position error fault action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19 EncLossFaultActStatus: encoder loss fault action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20 EncNsFaultActStatus: encoder noise fault action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21 DriveFaultActStatus: drive fault action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22 ServoConfigBitsStatus: update to Servo config bits</td>
</tr>
<tr>
<td>ServoEventBits</td>
<td>DINT</td>
<td>AXIS</td>
<td>The servo event bits for your servo loop.</td>
<td>Bit: Bit Name: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
<td></td>
<td>0 WatchEvArmStatus: watch event armed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 WatchEvStatus: watch event</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 RegEvArmStatus: registration event armed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 RegEvStatus: registration event</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 HomeEvArmStatus: home event armed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 HomeEvStatus: home event</td>
</tr>
<tr>
<td>ServoFaultBits</td>
<td>DINT</td>
<td>AXIS</td>
<td>The servo fault bits for your servo loop.</td>
<td>Bit: Bit Name: Meaning:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
<td></td>
<td>0 P0trvlFault: positive overtravel fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 N0trvlFault: negative overtravel fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 PosErrorFault: position error fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 EncCHAlossFault: encoder channel A loss fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 EncCHBlossFault: encoder channel B loss fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 EncCHZlossFault: encoder channel Z loss fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 EncNsFault: encoder noise fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7 DriveFault: drive fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 SyncConnFault: synchronous connection fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9 HardFault: servo hardware fault</td>
</tr>
<tr>
<td>ServoOutputLevel</td>
<td>REAL</td>
<td>GSV</td>
<td>The output voltage level for your axis servo loop.</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
<td>Access</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ServoStatusBits</td>
<td>DINT</td>
<td>AXIS</td>
<td>The status bits for your servo loop.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
<td><strong>Bit</strong>: <strong>BitName</strong>: <strong>Meaning:</strong></td>
<td></td>
</tr>
<tr>
<td>0 ServoActStatus</td>
<td></td>
<td></td>
<td>0 ServoActStatus</td>
<td>servo action</td>
</tr>
<tr>
<td>1 DriveEnableStatus</td>
<td></td>
<td></td>
<td>1 DriveEnableStatus</td>
<td>drive enable</td>
</tr>
<tr>
<td>2 OutLmtStatus</td>
<td></td>
<td></td>
<td>2 OutLmtStatus</td>
<td>output limit</td>
</tr>
<tr>
<td>3 PosLockStatus</td>
<td></td>
<td></td>
<td>3 PosLockStatus</td>
<td>position lock</td>
</tr>
<tr>
<td>5 HomeSwitchStatus</td>
<td></td>
<td></td>
<td>5 HomeSwitchStatus</td>
<td>state of home input switch</td>
</tr>
<tr>
<td>13 TuneStatus</td>
<td></td>
<td></td>
<td>13 TuneStatus</td>
<td>tuning process</td>
</tr>
<tr>
<td>14 TestStatus</td>
<td></td>
<td></td>
<td>14 TestStatus</td>
<td>test diagnostic</td>
</tr>
<tr>
<td>15 ShutdownStatus</td>
<td></td>
<td></td>
<td>15 ShutdownStatus</td>
<td>axis shutdown</td>
</tr>
<tr>
<td>ServoStatusUpdateBits</td>
<td>DINT</td>
<td>GSV</td>
<td>The servo status update bits for your axis.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSV</td>
<td><strong>Bit</strong>: <strong>Meaning:</strong></td>
<td></td>
</tr>
<tr>
<td>0 position error update</td>
<td></td>
<td></td>
<td>0 position error update</td>
<td>position error update</td>
</tr>
<tr>
<td>1 position integrator error update</td>
<td></td>
<td></td>
<td>1 position integrator error update</td>
<td>velocity error update</td>
</tr>
<tr>
<td>2 velocity error update</td>
<td></td>
<td></td>
<td>2 velocity error update</td>
<td>velocity integrator error update</td>
</tr>
<tr>
<td>3 velocity integrator error update</td>
<td></td>
<td></td>
<td>3 velocity integrator error update</td>
<td>velocity command update</td>
</tr>
<tr>
<td>4 velocity command update</td>
<td></td>
<td></td>
<td>4 velocity command update</td>
<td>velocity feedback update</td>
</tr>
<tr>
<td>5 velocity feedback update</td>
<td></td>
<td></td>
<td>5 velocity feedback update</td>
<td>servo output level update</td>
</tr>
<tr>
<td>6 servo output level update</td>
<td></td>
<td></td>
<td>6 servo output level update</td>
<td></td>
</tr>
<tr>
<td>* SoftOvertravelFaultAction</td>
<td>SINT</td>
<td>GSV</td>
<td>The operation performed when a soft overtravel fault occurs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSV</td>
<td><strong>Value</strong>: <strong>Meaning:</strong></td>
<td></td>
</tr>
<tr>
<td>0 shutdown the axis</td>
<td></td>
<td></td>
<td>0 shutdown the axis</td>
<td>shutdown the axis</td>
</tr>
<tr>
<td>1 disable the drive</td>
<td></td>
<td></td>
<td>1 disable the drive</td>
<td>disable the drive</td>
</tr>
<tr>
<td>2 stop the commanded motion</td>
<td></td>
<td></td>
<td>2 stop the commanded motion</td>
<td>stop the commanded motion</td>
</tr>
<tr>
<td>3 change the status bit only</td>
<td></td>
<td></td>
<td>3 change the status bit only</td>
<td>change the status bit only</td>
</tr>
<tr>
<td>StartActualPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The actual position of your axis when new commanded motion starts for the axis.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>You can use this value to correct for any motion occurring between the detection of an event and the action initiated by the event.</td>
<td></td>
</tr>
<tr>
<td>StartCommandPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The command position of your axis when new commanded motion starts for the axis.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>You can use this value to correct for any motion occurring between the detection of an event and the action initiated by the event.</td>
<td></td>
</tr>
<tr>
<td>StrobeActualPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The actual position of an axis when the Motion Group Strobe Position (M GSP) instruction executes.</td>
<td></td>
</tr>
<tr>
<td>StrobeCommandPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The command position of an axis when the Motion Group Strobe Position (M GSP) instruction executes.</td>
<td></td>
</tr>
<tr>
<td>TestDirectionForward</td>
<td>BOOL</td>
<td>GSV</td>
<td>The direction of axis travel during the Motion Run Hookup Diagnostic (M RHD) instruction as seen by the servo module.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Value</strong>: <strong>Meaning:</strong></td>
<td></td>
</tr>
<tr>
<td>0 negative (reverse) direction</td>
<td></td>
<td></td>
<td>0 negative (reverse) direction</td>
<td>negative (reverse) direction</td>
</tr>
<tr>
<td>1 positive (forward) direction</td>
<td></td>
<td></td>
<td>1 positive (forward) direction</td>
<td>positive (forward) direction</td>
</tr>
<tr>
<td>TestStatus</td>
<td>UINT16</td>
<td>GSV</td>
<td>The status of the last Motion Run Hookup Diagnostic (M RHD) instruction.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Value</strong>: <strong>Meaning:</strong></td>
<td></td>
</tr>
<tr>
<td>0 test process successful</td>
<td></td>
<td></td>
<td>0 test process successful</td>
<td>test process successful</td>
</tr>
<tr>
<td>1 test in progress</td>
<td></td>
<td></td>
<td>1 test in progress</td>
<td>test in progress</td>
</tr>
<tr>
<td>2 test process aborted by the user</td>
<td></td>
<td></td>
<td>2 test process aborted by the user</td>
<td>test process aborted by the user</td>
</tr>
<tr>
<td>3 test exceeded 2-second time-out</td>
<td></td>
<td></td>
<td>3 test exceeded 2-second time-out</td>
<td>test exceeded 2-second time-out</td>
</tr>
<tr>
<td>4 test process failed due to servo fault</td>
<td></td>
<td></td>
<td>4 test process failed due to servo fault</td>
<td>test process failed due to servo fault</td>
</tr>
<tr>
<td>5 insufficient test increment</td>
<td></td>
<td></td>
<td>5 insufficient test increment</td>
<td>insufficient test increment</td>
</tr>
</tbody>
</table>

Publication 1756-UM 006A-EN-P - April 2001
<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TuneAcceleration</td>
<td>REAL</td>
<td>GSV</td>
<td>The acceleration value measured during the last Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuneAccelerationTime</td>
<td>REAL</td>
<td>GSV</td>
<td>The acceleration time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuneDeceleration</td>
<td>REAL</td>
<td>GSV</td>
<td>The deceleration value measured during the last Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuneDecelerationTime</td>
<td>REAL</td>
<td>GSV</td>
<td>The deceleration time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuneRiseTime</td>
<td>REAL</td>
<td>GSV</td>
<td>The axis rise time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction. This value only applies to axes that you configure to work with an external velocity servo drive.</td>
</tr>
<tr>
<td>TuneSpeedScaling</td>
<td>REAL</td>
<td>GSV</td>
<td>The axis drive scaling factor measured during the last Motion Run Axis Tuning (MRAT) instruction. This value only applies to axes that you configure to work with an external velocity servo drive.</td>
</tr>
<tr>
<td>TuneStatus</td>
<td>UINT16</td>
<td>GSV</td>
<td>The status of the last Motion Run Axis Tuning (MRAT) instruction. <strong>Value:</strong> <strong>Meaning:</strong> 0: tune process successful 1: tuning in progress 2: tune process aborted by user 3: tune exceeded 2-second time-out 4: tune process failed due to servo fault 5: axis reached tuning travel limit 6: axis polarity set incorrectly 7: tune speed is too small to make measurements</td>
</tr>
<tr>
<td>TuneVelocityBandwidth</td>
<td>REAL</td>
<td>GSV</td>
<td>The bandwidth of the drive as calculated from the measurements made during the last Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuningConfigurationBits</td>
<td>DINT</td>
<td>GSV</td>
<td>The tuning configuration bits for your axis. <strong>Bit:</strong> <strong>Meaning:</strong> 0: tuning direction (0=forward, 1=reverse) 1: tune position error integrator 2: tune velocity error integrator 3: tune velocity feedforward 4: acceleration feedforward 5: tune velocity low-pass filter</td>
</tr>
<tr>
<td>TuningSpeed</td>
<td>REAL</td>
<td>GSV</td>
<td>The maximum speed reached by the Motion Run Axis Tuning (MRAT) instruction.</td>
</tr>
<tr>
<td>TuningTravelLimit</td>
<td>REAL</td>
<td>GSV</td>
<td>The travel limit used by the Motion Run Axis Tuning (MRAT) instruction to limit the action of the axis during tuning.</td>
</tr>
<tr>
<td>VelocityCommand</td>
<td>REAL</td>
<td>GSV</td>
<td>The current velocity reference to the velocity servo loop for an axis.</td>
</tr>
<tr>
<td>VelocityError</td>
<td>REAL</td>
<td>GSV</td>
<td>The difference between the commanded and actual velocity of a servo axis. You can use this value to drive the motor to where the velocity feedback equals the velocity command.</td>
</tr>
<tr>
<td>VelocityFeedback</td>
<td>REAL</td>
<td>GSV</td>
<td>The actual velocity of your axis as estimated by the servo module. To estimate the velocity, the servo module applies a 1 kHz low-pass filter to the change in actual position in one update interval.</td>
</tr>
<tr>
<td>Variable</td>
<td>Data Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>* VelocityFeedforwardGain</td>
<td>REAL</td>
<td>GSV</td>
<td>The value used to provide the velocity command output to generate the command velocity.</td>
</tr>
<tr>
<td>* VelocityIntegralGain</td>
<td>REAL</td>
<td>GSV</td>
<td>The value that the controller multiplies with the VelocityIntegralError value to correct the velocity error.</td>
</tr>
<tr>
<td>VelocityIntegralError</td>
<td>REAL</td>
<td>GSV</td>
<td>The sum of the velocity error for a specified axis. You can use this value to drive the motor to where the velocity feedback equals the velocity command.</td>
</tr>
<tr>
<td>* VelocityProportionalGain</td>
<td>REAL</td>
<td>GSV</td>
<td>The value that the controller multiplies with the VelocityError to correct the velocity error.</td>
</tr>
<tr>
<td>WatchPosition</td>
<td>REAL</td>
<td>GSV</td>
<td>The watch position of your axis.</td>
</tr>
</tbody>
</table>
Instruction Timing

This appendix describes motion instruction timing types.

Motion instructions use three types of timing sequences:

<table>
<thead>
<tr>
<th>Timing type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td>The instruction completes in one scan.</td>
</tr>
<tr>
<td>Message</td>
<td>The instruction completes over several scans because the instruction sends messages to the servo module.</td>
</tr>
<tr>
<td>Process</td>
<td>The instruction could take an indefinite amount of time to complete.</td>
</tr>
</tbody>
</table>

Understanding Immediate Type Instructions

Immediate type motion instructions execute to completion in one scan. If the controller detects an error during the execution of these instructions, the error status bit sets and the operation ends.

Examples of immediate type instructions include the:

- Motion Change Dynamics (MCD) instruction
- Motion Group Strobe Position (MGSP) instruction

Immediate instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (EN) bit.
   - Clears the done (DN) bit.
   - Clears the error (ER) bit.

2. The controller executes the instruction completely.

3. If the controller
   - Does not detect an error when the instruction executes, then the controller sets the DN bit.
   - Detects an error when the instruction executes, then the controller sets the ER bit and stores an error code in the control structure.
4. The next time the rung becomes false after either the .DN or .ER bit sets, the controller clears the .EN bit.

5. The controller can execute the instruction again when the rung becomes true.

---

Understanding Message Type Instructions

Message type motion instructions send one or more messages to the servo module.

Examples of message type instructions include the:

- Motion Direct Drive On (MDO) instruction
- Motion Redefine Position (MRP) instruction

Message type instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (EN) bit.
   - Clears the done (DN) bit.
   - Clears the error (ER) bit.

2. The controller begins to execute the instruction by setting up a message request to the servo module.

The remainder of the instruction executes in parallel to the program scan.

3. The controller checks if the servo module is ready to receive a new message.

4. The controller places the results of the check in the message status word of the control structure.
5. When the module is ready, the controller constructs and transmits the message to the module. This process may repeat several times if the instruction requires multiple messages.

6. | If the controller                                | Then                                                                 |
    |-----------------------------------------------|----------------------------------------------------------------------|
    | Does not detect an error when the instruction executes | The controller sets the .DN bit.                                    |
    | Detects an error when the instruction executes | The controller sets the .ER bit and stores an error code in the control structure. |

7. The next time the rung becomes false after either the .DN or .ER bit sets, the controller clears the .EN bit.

8. When the rung becomes true, the controller can execute the instruction again.

---

**Understanding Process Type Instructions**

Process type motion instructions initiate motion processes that can take an indefinite amount of time to complete.

Examples of process type instructions include:

- Motion Arm Watch Position (MAW) instruction
- Motion Axis Move (MAM) instruction
Process type instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (.EN) bit.
   - Clears the done (.DN) bit.
   - Clears the error (.ER) bit.
   - Clears the process complete (.PC) bit.

2. The controller initiates the motion process.

3. The controller does not detect an error when the instruction executes
   - Sets the .DN bit.
   - Sets the in process (.IP) bit.

The controller detects an error when the instruction executes
   - Sets the .ER bit.
   - Stores an error code in the control structure.
   - Does not change the .IP and .PC bits.

The controller detects another instance of the motion instruction
   - Clears the .IP bit for that instance.

The motion process reaches the point where the instruction can be executed again
   - Sets the .DN bit.
   - For some process type instructions, like MAM, this will occur on the first scan. For others, like MAH, the .DN bit will not be set until the entire homing process is complete.

One of the following occurs during the motion process:
   - The motion process completes
   - Another instance of the instruction executes
   - Another instruction stops the motion process
   - A motion fault stops the motion process
   - Sets the .DN bit.
   - Sets the .PC bit.
   - Clears the .IP bit.

4. Once the initiation of the motion process completes, the program scan can continue.

The remainder of the instruction and the control process continue in parallel with the program scan.
5. The next time the rung becomes false after either the .DN bit or the .ER bit sets, the controller clears the .EN bit.

6. When the rung becomes true, the instruction can execute again.
Fault Handling

This appendix describes motion errors and faults.

Handling Motion Faults

Two types of motion faults exist.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors</td>
<td>• Do not impact controller operation • Should be corrected to optimize execution time and ensure program accuracy</td>
<td>A Motion Axis Move (MAM) instruction with a parameter out of range</td>
</tr>
<tr>
<td>Minor/Major</td>
<td>• Caused by a problem with the servo loop • Can shutdown the controller if you do not correct the fault condition</td>
<td>The application exceeded the PositionErrorTolerance value</td>
</tr>
</tbody>
</table>

Understanding Errors

Executing a motion instruction within an application program can generate errors. The MOTION_INSTRUCTION tag has a field that contains the error code. For more information on error codes for individual instructions, refer to the motion instruction chapters in the Logix5550 Controller Instruction Set Reference Manual.

Understanding Minor/Major Faults

Several faults can occur that are not caused by motion instructions. For example, a loss of encoder feedback or actual position exceeding an overtravel limit will cause faults. The motion faults are considered Type 11 faults with error codes from 1 to 32. For more information about motion error codes, refer to Handling Controller Faults in the Logix5550 Controller User Manual.

TIP

You can configure a fault as either minor (non major) or major by using the Axis Wizard-Group window.
For more information about handling faults, see Handling Controller Faults in the Logix5550 Controller User Manual, publication 1756-6.5.12.
Symbols
.C-12
.C-13
.C-13
.C-13

Numerics
1394C Drive module
Associated Axes Tab 6-10
New Axis button 6-11
Node X0 6-10
Node X1 6-10
Node X2 6-10
Node X3 6-11
Connection Tab 6-7
Inhibit Module checkbox 6-8
Major Fault on Controller if Connection Fails checkbox 6-9
Module Fault 6-9
Connection Request Error 6-9
Electronic Keying Mismatch 6-9
Module Configuration Invalid 6-9
Service Request Error 6-9
Requested packet interval 6-7
General Tab 6-4
Base Node 6-5
Description 6-5
Electronic Keying 6-6
Compatible Module 6-6
Disable Keying 6-6
Exact Match 6-6
Name 6-5
Revision 6-5
Type 6-5
Vendor 6-5
Module Info tab 6-12
(16#xxxx) unknown 6-14
Configured 6-14
Internal State Status 6-14
Major/Minor Fault Status 6-13
Module Identity 6-15
Owned 6-14
Product Name 6-13
Refresh 6-15
Reset Module 6-15
Power Tab 6-11
Bus Regulator ID 6-11
1394-CFLAExx Cable

Pinouts B-9
Wiring Diagram B-9
1394C-SJ T05/10/22-D Digital Servo Drive Overview 6-3
1398-CFLAExx
Cable Diagram B-7
Pinouts B-7
1756-M02AE servo module 1-1
Adding to a program 2-1, 5-1
Additional modules and axes 2-18
Auto tuning 2-18
Block diagrams
Torque servo drive B-2
Velocity servo drive B-3
Coarse update rate calculations A-3
Configuring a motion axis 2-1
Features 1-2
Hookup diagnostics 2-18
Loop and interconnect diagrams B-1
Specifications A-1
Troubleshooting 8-1
Wiring diagrams
1394 drive B-8
24V registration sensor B-10
5V registration sensor B-10
Home limit switch B-11
OK contacts B-11
Servo module RTB B-4
Ultra 100 drive B-5
Ultra 200 drive B-6
1756-M08SE 3-1
Adding the module 3-1
configuring module 3-1
General Tab
Description 3-5
Electronic Keying 3-6
Compatible Module 3-6
Disable Keying 3-6
Exact Match 3-6
Name 3-5
Revision 3-6
Slot 3-5
Type 3-5
Vendor 3-5
Motion Module Overview 3-4
Properties
SERCOS Interface Info Tab 3-8
SERCOS Interface Info Tab
Fault Type 3-9
Refresh 3-9
M

Major faults F-1
Marker test
    Starting 2-23
Message instruction timing E-2
Message status C-13
Minor faults F-1
Module Properties
    Backplane Tab 2-16
    Connection Tab 2-8
    General Tab 2-6
    Module Info Tab 2-12
Module Properties window 2-20
Motion attributes D-1
    Changing configuration parameters 1-5
    Motion instance variables D-1
    Understanding status and configuration parameters 1-5
Motion configuration instructions 7-4
    Motion Apply Axis Tuning (MAAT) 7-4
    Motion Apply Hookup Diagnostic (MAHD) 7-5
    Motion Run Axis Tuning (MRT) 7-4
    Motion Run Hookup Diagnostic (MRHD) 7-5
Motion event instructions 7-3
    Motion Arm Output Cam (MAOC) 7-4
    Motion Arm Registration (MAR) 7-4
    Motion Arm Watch Position (MAW) 7-4
    Motion Disarm Output Cam (MDOC) 7-4
    Motion Disarm Registration (MDR) 7-4
    Motion Disarm Watch Position (MDW) 7-4
Motion group instructions 7-3
    Motion Group Programmed Stop (MGPS) 7-3
    Motion Group Shutdown (MGS) 7-3
    Motion Group Shutdown Reset (MGSR) 7-3
    Motion Group Stop (MGS) 7-3
    Motion Group Strobe Position (MGSP) 7-3
    Motion instance variables D-1
Motion instructions 7-1
    Motion Apply Axis Tuning (MAAT) 7-4
    Motion Apply Hookup Diagnostic (MAHD) 7-5
    Motion Arm Output Cam (MAOC) 7-4
    Motion Arm Registration (MAR) 7-4
    Motion Arm Watch Position (MAW) 7-4
    Motion Axis Fault Reset (MAFR) 7-2
    Motion Axis Gear (MAG) 7-2
    Motion Axis Home (MAH) 7-2
    Motion Axis Jog (MAJ) 7-2
    Motion Axis Move (MAM) 7-2
    Motion Axis Position Cam (MAPC) 7-3
    Motion Axis Shutdown (MASD) 7-2
    Motion Axis Shutdown Reset (MASR) 7-2
    Motion Axis Stop (MAS) 7-2
    Motion Axis Time Cam (MATC) 7-3
    Motion Calculate Cam Profile (MATC) 7-3
    Motion Change Dynamics (MCD) 7-2
    Motion configuration instructions 7-4
    Motion Direct Drive Off (MDF) 7-2
    Motion Direct Drive On (MDO) 7-2
    Motion Disarm Output Cam (MDOC) 7-4
    Motion Disarm Registration (MDR) 7-4
    Motion Disarm Watch Position (MDW) 7-4
    Motion event instructions 7-3
    Motion group instructions 7-3
    Motion Group Programmed Stop (MGPS) 7-3
    Motion Group Shutdown (MGS) 7-3
    Motion Group Shutdown Reset (MGSR) 7-3
    Motion Group Strobe Position (MGSP) 7-3
    Motion move instructions 7-2
    Motion Redefine Position (MRP) 7-2
    Motion Run Axis Tuning (MRT) 7-4
    Motion Run Hookup Diagnostic (MRHD) 7-5
    Motion Servo Off (MSF) 7-1
    Motion Servo On (MSO) 7-1
    Motion state instructions 7-1
    Timing E-1
    Immediate E-1
    Message E-2
    Process E-3
Motion move instructions 7-2
    Motion Axis Gear (MAG) 7-2
    Motion Axis Home (MAH) 7-2
    Motion Axis Jog (MAJ) 7-2
    Motion Axis Move (MAM) 7-2
    Motion Axis Stop (MAS) 7-2
    Motion Axis Time Cam (MATC) 7-3
    Motion Calculate Cam Profile (MATC) 7-3
    Motion Change Dynamics (MCD) 7-2
    Motion Redefine Position (MAPC) 7-3
    Motion Redefine Position (MRP) 7-2
Motion state instructions 7-1
    Motion Axis Fault Reset (MAFR) 7-2
    Motion Axis Shutdown (MASD) 7-2
    Motion Axis Shutdown Reset (MASR) 7-2
    Motion Direct Drive Off (MDF) 7-2
    Motion Direct Drive On (MDO) 7-2
    Motion Servo Off (MSF) 7-1
    Motion Servo On (MSO) 7-1
MOTION_GROUP control structure C-7
MOTION_INSTRUCTION control structure C-9
Understanding 1-4
Motor/encoder test
Starting 2-21

N
Naming & Configuring Motion Axes 5-1
Naming an Axis 5-1
New Module window 2-4
New Tag window
MOTION_GROUP 5-8

O
OK LED indicator 8-1
Output Cam Timing A-8
1756-L50 Controller A-9

P
Performance guidelines A-1
Process instruction timing E-3
product support, local
telephone number P-3
Profile Segment C-13
Proportional (Velocity) Gain 5-38

R
Rockwell Automation support
Technical product assistance P-3
RSLogix 5000 programming software 1-1
Application program
Developing 5-54
Example 5-55
Control structures C-1
Downloading an application program 2-18, 2-19
Fault handling F-1
Features 1-3
GSV/SSV instructions D-1
Instruction timing E-1
Motion attributes D-1
Motion instructions 7-1
Running Hookup Diagnostics and Auto Tuning 5-54

S
Select Module Type window 2-2
Servo Configuration Update Status Bits attributes C-6
Specifications A-1
SSV instruction D-1
Changing configuration parameters 1-5
structures
CAM C-13
CAM_PROFILE C-14
support
On the Web P-3

T
Tags
New Tag window
MOTION_GROUP 5-8
Troubleshooting 8-1
DRIVE LED indicator 8-2
FDBK LED indicator 8-2
OK LED indicator 8-1

W
Windows
Axis properties 2-20
Module properties 2-20
New module 2-4
New tag 5-8
Select module type 2-2
Tune bandwidth 2-25
Wiring diagrams B-4
1394 drive B-8
24V registration sensor B-10
5V registration sensor B-10
Home limit switch B-11
OK contacts B-11
Servo module RTB B-4
Ultra 100 drive B-5
Ultra 200 drive B-6
World Wide Web site P-3