



EtherNet/IP Add-On Instructions for Allen Bradley® RSLogix5000® Software

Table of Contents

1	INTRODUCTION.....	5
1.1	REFERENCE DOCUMENTATION	5
1.2	EXAMPLE HARDWARE.....	6
2	SPECIFICATION OF AOIS.....	7
2.1	GENERAL AOI DESIGN AND USE	7
2.1.1	<i>PLC code examples</i>	<i>7</i>
2.1.2	<i>Typical Command Sequence</i>	<i>8</i>
2.1.3	<i>Exceptions/Notes</i>	<i>9</i>
2.1.3.1	<i>Error output</i>	<i>9</i>
2.1.3.2	<i>Buffering and Sent output.....</i>	<i>9</i>
2.1.3.3	<i>Using Done output with buffering</i>	<i>9</i>
2.1.3.4	<i>Trigger signal is edge triggered.....</i>	<i>9</i>
2.1.3.5	<i>Abortion of command</i>	<i>9</i>
2.1.3.6	<i>Using Done output with Interruption/Stopping</i>	<i>10</i>
2.1.4	<i>Message Type 1 or Type 2 Commands.....</i>	<i>10</i>
3	EXAMPLE PROJECT	11
3.1	NEW PROJECT SETUP IN RSLOGIX5000®	11
3.2	ADD AOIS THAT REQUIRE MESSAGING.....	25
4	APPENDIX: ADD-ON INSTRUCTIONS.....	37
4.1	AMP_INPUT_ASSEMBLY AOI	39
4.1.1	<i>Overview</i>	<i>39</i>
4.1.2	<i>Functionality</i>	<i>40</i>
4.1.3	<i>Parameters.....</i>	<i>40</i>
4.2	AMP_STATUS_CODE AOI	41
4.2.1	<i>Overview</i>	<i>41</i>
4.2.2	<i>Functionality</i>	<i>41</i>
4.2.3	<i>Parameters.....</i>	<i>42</i>
4.3	AMP_ALARM_CODE AOI.....	43
4.3.1	<i>Overview</i>	<i>43</i>
4.3.2	<i>Functionality</i>	<i>43</i>
4.3.3	<i>Parameters.....</i>	<i>44</i>
4.4	AMP_MOTOR_ENABLE AOI.....	45
4.4.1	<i>Overview</i>	<i>45</i>
4.4.2	<i>Functionality</i>	<i>45</i>
4.4.3	<i>Parameters.....</i>	<i>45</i>
4.5	AMP_MOTOR_DISABLE AOI.....	47
4.5.1	<i>Overview</i>	<i>47</i>
4.5.2	<i>Functionality</i>	<i>47</i>
4.5.3	<i>Parameters.....</i>	<i>47</i>

4.6	AMP_ABSOLUTE_MOVE AOI	49
4.6.1	Overview	49
4.6.2	Functionality	49
4.6.3	Parameters.....	50
4.7	AMP_RELATIVE_MOVE AOI	51
4.7.1	Overview	51
4.7.2	Functionality	51
4.7.3	Parameters:.....	52
4.8	AMP_NORMAL_STOP AOI.....	53
4.8.1	Overview	53
4.8.2	Functionality	53
4.8.3	Parameters.....	53
4.9	AMP_CRASH_STOP AOI	55
4.9.1	Overview	55
4.9.2	Functionality	55
4.9.3	Parameters:.....	55
4.10	AMP_ALARM_RESET AOI	57
4.10.1	Overview	57
4.10.2	Functionality	57
4.10.3	Parameters:.....	57
4.11	AMP_SCL_COMMAND_EXECUTE AOI	59
4.11.1	Overview	59
4.11.2	Functionality	59
4.11.3	Exceptions/Notes	60
4.11.4	Parameters:.....	60
4.12	AMP_POINT_TO_POINT_IO AOI.....	61
4.12.1	Overview	61
4.12.2	Functionality	62
4.12.3	Parameters.....	63
4.13	AMP_POINT_TO_POINT_FC AOI.....	65
4.13.1	Overview	65
4.13.2	Functionality	65
4.13.3	Parameters.....	66
4.14	AMP_JOG_MOVE AOI	67
4.14.1	Overview	67
4.14.2	Functionality	67
4.14.3	Exceptions/Notes	68
4.14.4	Parameters.....	68
4.15	AMP_UPDATE_JOG_SPEED AOI	69
4.15.1	Overview	69
4.15.2	Functionality	69
4.15.3	Parameters.....	69
4.16	AMP_Q_SEGMENT_EXECUTE AOI	71
4.16.1	Overview	71
4.16.2	Functionality	71
4.16.3	Parameters:.....	71

4.17	AMP_HARD_STOP_HOMING AOI.....	73
4.17.1	Overview	73
4.17.2	Functionality	74
4.17.3	Parameters.....	74
4.18	AMP_TUNING_PARAMS_SERVO AOI	76
4.18.1	Overview	76
4.18.2	Functionality	76
4.18.3	Parameters:.....	77
4.19	AMP_TUNING_PARAMS_STEPSERVO AOI	78
4.19.1	Overview	78
4.19.2	Functionality	78
4.19.3	Parameters:.....	79
4.20	AMP_CONFIGURATION_SETTINGS AOI.....	80
4.20.1	Overview	80
4.20.2	Functionality	80
4.20.3	Parameters.....	81

1 INTRODUCTION

This application note will show how to setup an RSLogix5000® project that uses Add-On Instructions (AOI) to control an EtherNet/IP-enabled drive from Applied Motion Products.

Please note that the majority of AOIs in this document use Implicit Messaging, which was released in AMP EtherNet/IP firmware versions 251.xx and higher. For older firmware versions, or for compatibility with PLCs which do not support Implicit Messaging, please see Application Note 24, which performs many of the same tasks using Explicit Messaging.

Starting in firmware version 251.xx, AMP's EtherNet/IP interface was redesigned to make use of Implicit Messaging for sending commands to a drive. In addition to adding Implicit Messaging support, drive data (in terms of byte ordering) is was updated to be more PLC-friendly with the intention to lighten the work required to integrate AMP products into EtherNet/IP PLC programs.

The output assembly being implicitly sent to the drive allows access to the majority of drive features with minimal overhead, since there is no longer a need to include explicit messaging and therefore passing message data structures to the AOI.

Besides simplifying usage, implicit messaging also introduces new challenges. This is because Ethernet/IP Class 1 lacks a built-in acknowledgment mechanism. Unlike the ACK/NAK mechanism of the 1st generation interface, the 2nd generation interface requires the control logic to include time delays to pace the transmission of data, to ensure the drive sees each command before sending the next one.

These AOIs are intended to demonstrate the features of the Applied Motion Products EtherNet/IP implementation. They are being distributed unlocked and unprotected, and may be used as-is in your PLC program or modified as needed. This document outlines the supplied AOIs, and provides the steps needed to get the motor spinning. The functional descriptions found in the appendix provide details about the input and output variables of each AOI. In addition, because the AOIs are provided unprotected, we encourage you to view the source ladder logic inside RSLogix5000® once you import the AOIs into your program.

1.1 Additional Content

Within the zip file that contained this PDF, you can find the following other contents:

- A folder named *AOI*, which contains the AOI files to be imported into your program

- The file *AMP_Demo_APPN0046_RevA.ACD*, which is an example program with all the AOIs imported, and two pre-configured drives
 - This demo file assumes a PLC IP address of 10.10.10.150, and drive IP addresses of 10.10.10.10, and 10.10.10.11

1.2 Reference Documentation

This document only explains each AOI's purpose and functionality. It is not intended to be fully inclusive of all the drive's features and functionality. For more information regarding drive commands, input/output data structures, acronyms, operands, etc. please refer to:

- Host Command Reference, 920-0002 Rev. P, 1/17/2018 (https://www.applied-motion.com/sites/default/files/hardware-manuals/Host-Command-Reference_920-0002P.PDF)
- Application Note APPN0040: "Hard-Stop Homing Commands for StepSERVO and SV200 Series" (https://www.applied-motion.com/sites/default/files/APPN0040A_Hard-Stop_Homing_StepSERVO-SV200.pdf)
- Application Note APPN0024: "Add-On Instructions for Allen Bradley® RSLogix5000® Software" (https://www.applied-motion.com/sites/default/files/APPN0024_AOIs-for-RSLogix5000.zip)
- Specific drive hardware manuals. These can be found in the *Downloads* section of the drive Product Page on the Applied Motion Products website (<http://www.applied-motion.com>)

1.3 Example Hardware

This application note was developed and primarily tested on the following hardware:

- Allen Bradley® CompactLogix® L16ER – Firmware v.20 – Configured to IP address 10.10.10.150
- Applied Motion Products SSM23IP-3EG – IP address 10.10.10.10
- Applied Motion products STM23IP-2EE – IP Address 10.10.10.11

For determining compatibility of individual commands and features used by the AOIs with other hardware from Applied Motion Products, always refer to the Host Command Reference manual.

Allen Bradley®, RSLogix5000®, and CompactLogix® are registered trademarks of Rockwell Automation, Inc., <http://ab.rockwellautomation.com/>

2 SPECIFICATION OF AOIS

The majority of AOIs provided are similar with each other in terms of parameters and common functionality. In this section we therefore document the basic common features of the AOIs

- Preview of AOI on a rung
- Common sequence of operations
- Exceptions and notes

The remainder of the section will describe AOIs in specific terms: of their general description and parameters. In the [appendix](#), you can find a specification of each of the 20 AOIs provided. For each AOI we provide

- A short description
- The list of parameters
- A normal sequence of operations
- Some exceptions and notes, if applicable

2.1 General AOI Design and Use

2.1.1 PLC code examples

In Figure 1 and Figure 2 are previews of how the AOI will appear to the PLC user.

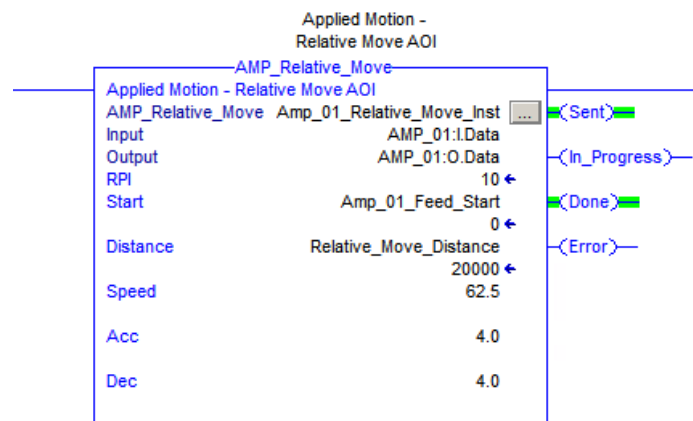


Figure 1 - AMP_Relative_Move AOI instance example

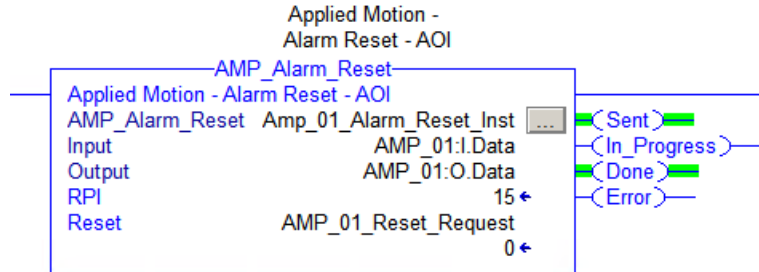


Figure 2 - AMP_Alarm_Reset AOI instance example

2.1.2 Typical Command Sequence

Many AOIs are simple wrappers for “native” drive commands (e.g. FL=Feed to Length, full details of which can be found in the Host Command Reference). Such AOIs have an appropriately named parameter used for triggering the start of the command sequence.

The built-in EnableIn parameter is only used to enable the execution of the AOI, not to trigger the drive to perform an action.

When the “trigger” input parameter transitions from false to true, the following sequence of steps occur:

- If the drive is not in the correct state (not enabled, faulted, etc), the AOI sets its Error output and takes no further action.
- Otherwise, the AOI outputs a command value of 0 (Idle), in case it was not already 0.
- The AOI ensures that the drive command word has been idle for a set time (equal to the RPI parameter times 1.3), or waits for that length of time before proceeding.
- The AOI copies or transforms AOI inputs and outputs these to the drive (e.g. for a Relative Move it outputs the setpoint distance and the correctly scaled values of speed, acceleration and deceleration parameters).
- The AOI outputs the command value corresponding to the AOI basic function (e.g. a Relative Move corresponds to a "Feed to Length" (FL) command, i.e. setting the command word to a value of 0x8).
- The AOI waits at least a full I/O cycle (RPI parameter times 1.3) to ensure the drive saw the command, then resets the command word value to 0 (Idle) to prepare for the next command.
- The AOI sets its Sent output.
- The drive continually updates the status as the command progresses to completion (the completion condition varies per AOI and is detailed within each individual specification further below).

- The AOI waits for the drive status to indicate completion of the command (e.g. for a Relative Move this would be "In Position").
- The AOI sets its Done output.

2.1.3 Exceptions/Notes

2.1.3.1 Error output

If an error occurs during execution, the AOI sets its Error output. The Error output is reset when the “trigger” parameter transitions from false to true.

2.1.3.2 Buffering and Sent output

Depending on the type of commands, the drive may sometimes buffer the command if it is not ready to execute it immediately. When using buffering, the AOI’s Sent output can be integrated into the flow control logic to trigger several commands in rapid succession. Full details of Buffered vs Immediate type commands can be found in the Host Command Reference, and the AOI descriptions in the appendix indicate whether the AOIs use Buffered or Immediate commands.

2.1.3.3 Using Done output with buffering

When using buffering, the Done output may not reflect intermediate completion states between successive commands. If intermediate completion checkpoints are desired, the user will need to develop custom completion logic.

2.1.3.4 Trigger signal is edge triggered

When the “trigger” signal parameter transitions from true to false, the command is not stopped, but rather proceeds normally. **For users of AMP’s original Explicit Messaging AOIs, this is a key difference in operation.**

2.1.3.5 Abortion of command

Since the “trigger” signal parameter is rising edge triggered and cannot be used to abort a command (some 1st generation AOIs used the false state of the “Execute” signal to stop/abort a move command), the user will need to use other means to

stop/abort. For simple commands that involve moves, this can be done by executing the AMP_Normal_Stop or the AMP_Crash_Stop AOI. For AOI's handling more complex sequence of commands, a dedicated "stop" signal parameters is used. **For users of AMP's original Explicit Messaging AOIs, this is a key difference in operation.**

2.1.3.6 Using Done output with Interruption/Stopping

The interruption of a command using "Stop" AOIs will render invalid the Done output of the original AOI command. To prevent confusion by the user, AOI documentation in the appendix is clear about the exact meaning of the Done output for each given AOI, and whether a command is subject to buffering or not.

2.1.4 Message Type 1 or Type 2 Commands

As a matter of design choice, when Message Type 1 or 2 Commands are required for the internal functionality in any AOI, then that AOI will exclusively use explicit messaging internally and not attempt to leverage the simplicity of the new output assembly. The rationale is that by doing so, the AOI no longer needs the Input, Output and RPI parameters, and solely requires the MSG_Write, SendMsg and MSG_Response parameters. These controller tags are required whenever an AOI use a MSG (Message) instruction.

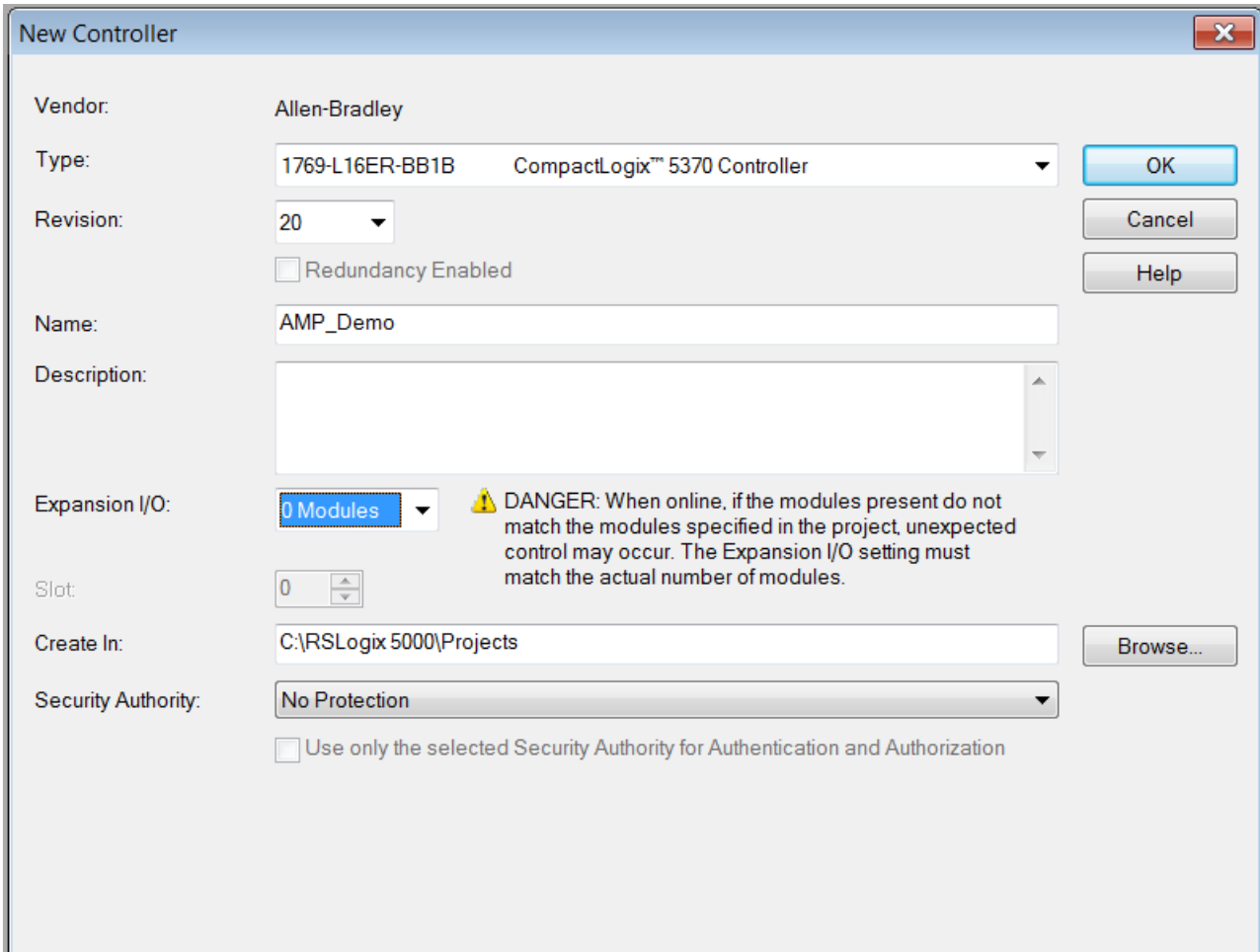
3 EXAMPLE PROJECT

3.1 New Project Setup in RSLogix5000®

This section features a step-by-step procedure for setting up a new project that can communicate to an Applied Motion EtherNet/IP drive and use a selection of Add-On Instructions (AOIs) to handle common tasks. The procedure can also be used to add a new drive and AOIs to an existing project. Note that a demo project pre-loaded with all the AOIs is also included in the ZIP folder with this application note (*AMP_Demo_APPN0046_RevA.ACD*).

STEP 1 **Create a project**

Create a new RSLogix project (v20 or later)



The "New Controller" dialog box is used to configure a new RSLogix controller. It includes fields for Vendor, Type, Revision, Name, Description, Expansion I/O, Slot, Create In, and Security Authority. A warning message is displayed next to the Expansion I/O field.

Vendor: Allen-Bradley

Type: 1769-L16ER-BB1B CompactLogix™ 5370 Controller

Revision: 20

☐ Redundancy Enabled

Name: AMP_Demo

Description:

Expansion I/O: 0 Modules

Slot: 0

Create In: C:\RSLogix 5000\Projects

Security Authority: No Protection

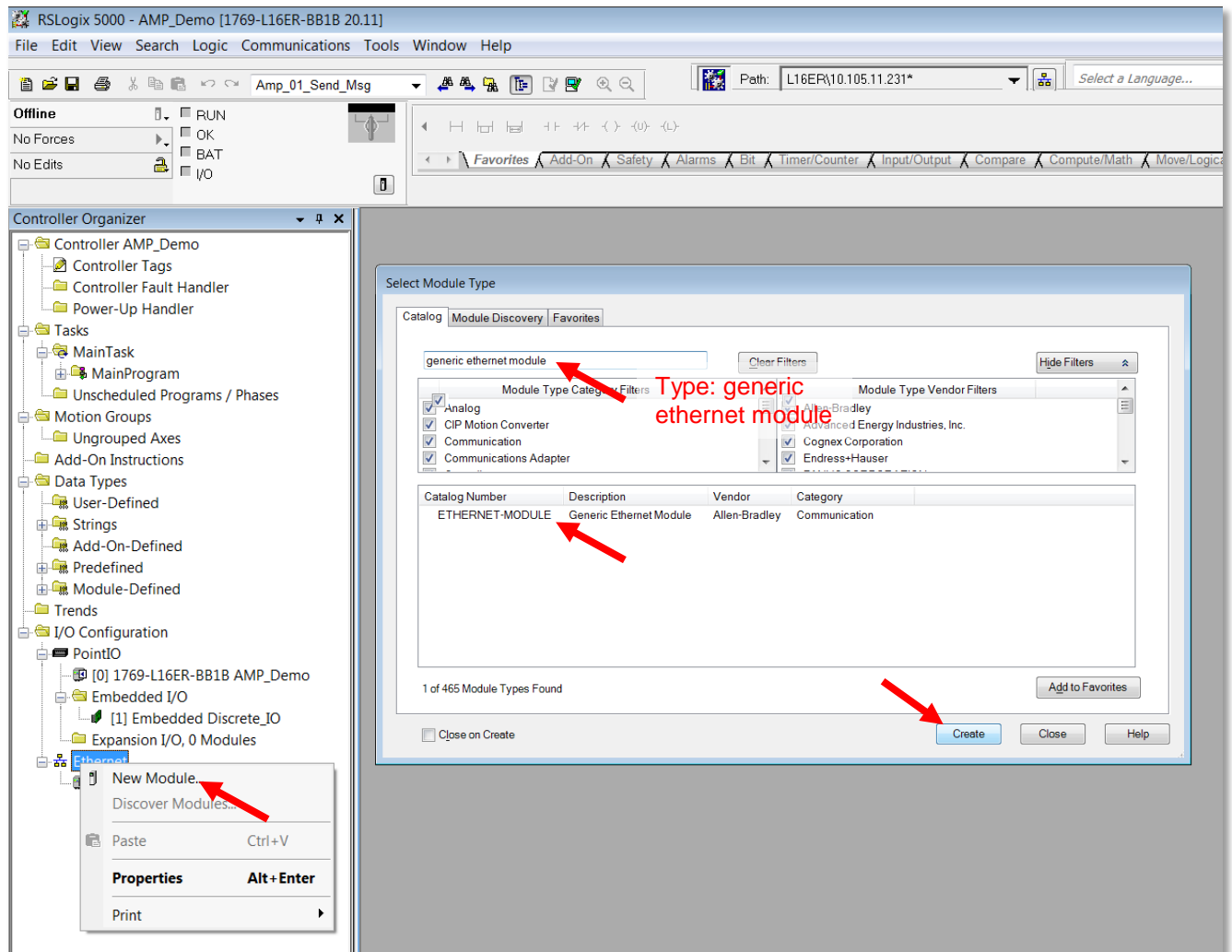
☐ Use only the selected Security Authority for Authentication and Authorization

Warning: DANGER: When online, if the modules present do not match the modules specified in the project, unexpected control may occur. The Expansion I/O setting must match the actual number of modules.

Buttons: OK, Cancel, Help, Browse...

STEP 2 I/O Configuration

In the I/O Configuration under Ethernet, add a new module: ETHERNET-MODULE – Generic Ethernet Module

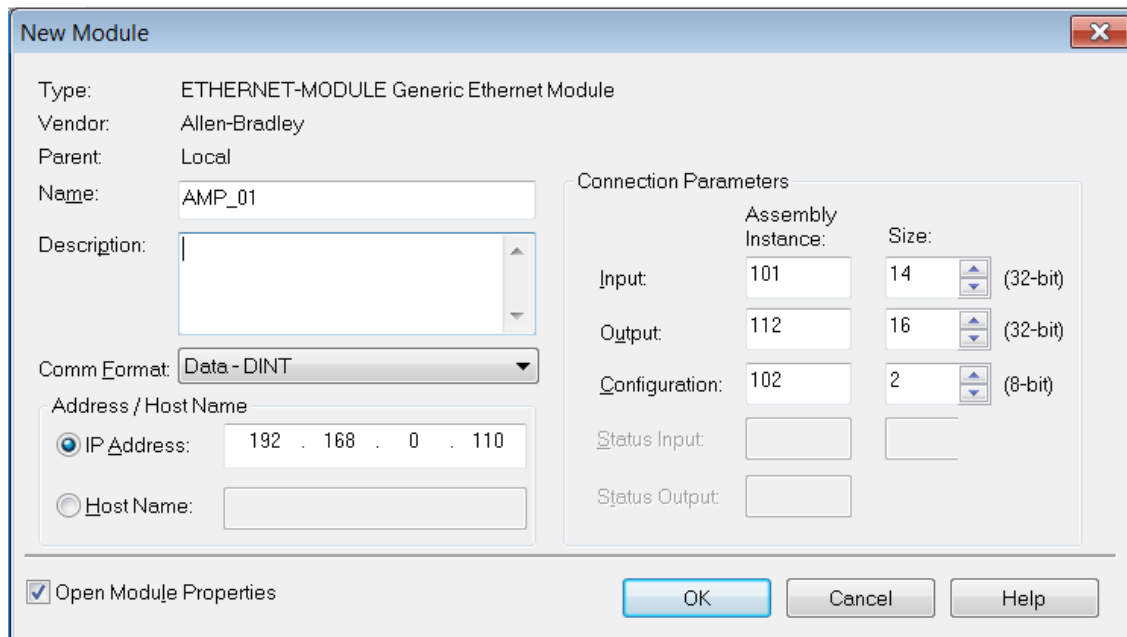


STEP 3 Configure the Ethernet Hardware Module

Fill in the address of the AMP drive and configure the Assembly Instance.

Note: The Comm Format entry of Data – DINT is critical in this step.

Name:	AMP_01 (arbitrary)
Comm Format:	Data – DINT
IP Address:	192.168.0.110 (target drive IP address, refer to your drive's hardware manual for details on setting this)
Input Assembly Instance:	101
Input Size:	14
Output Assembly Instance:	112
Output Size:	16
Configuration Assembly Instance:	102
Configuration Size:	2

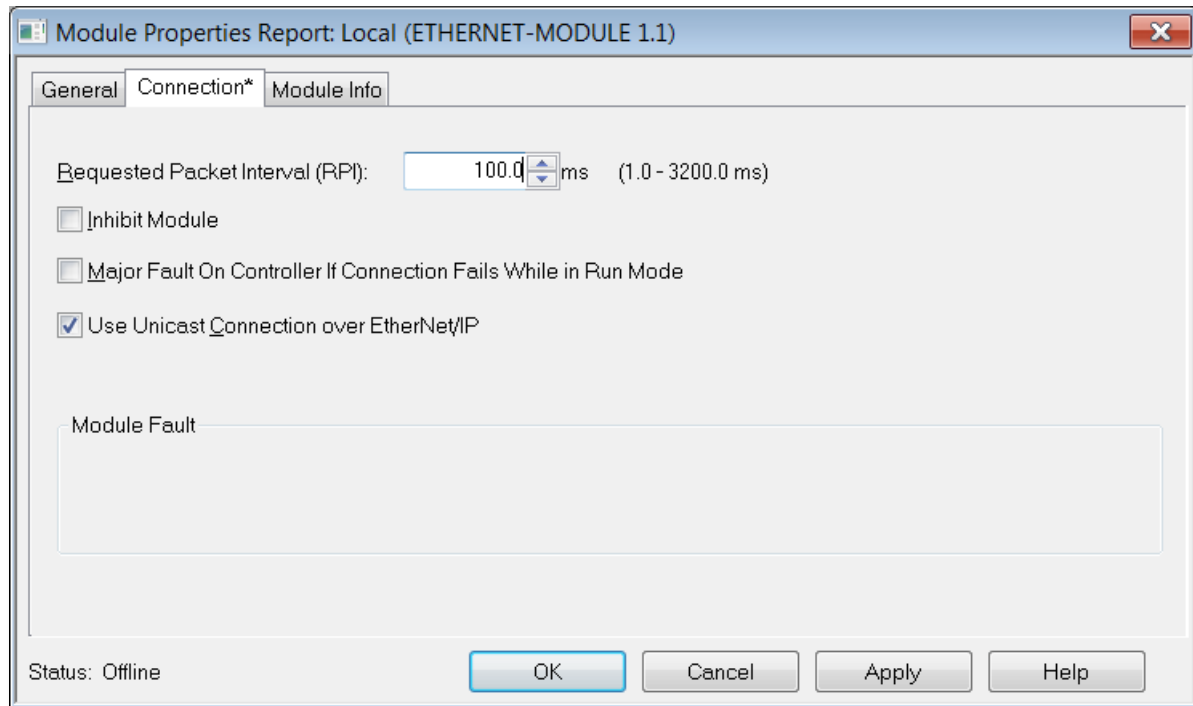


New Module

Type: ETHERNET-MODULE Generic Ethernet Module
 Vendor: Allen-Bradley
 Parent: Local
 Name: AMP_01
 Description:
 Comm Format: Data - DINT
 Address / Host Name
☒ IP Address: 192 . 168 . 0 . 110
☐ Host Name:
 Connection Parameters
 Input Assembly Instance: 101 Size: 14 (32-bit)
 Output Assembly Instance: 112 Size: 16 (32-bit)
 Configuration Assembly Instance: 102 Size: 2 (8-bit)
 Status Input:
 Status Output:
☒ Open Module Properties
 OK Cancel Help

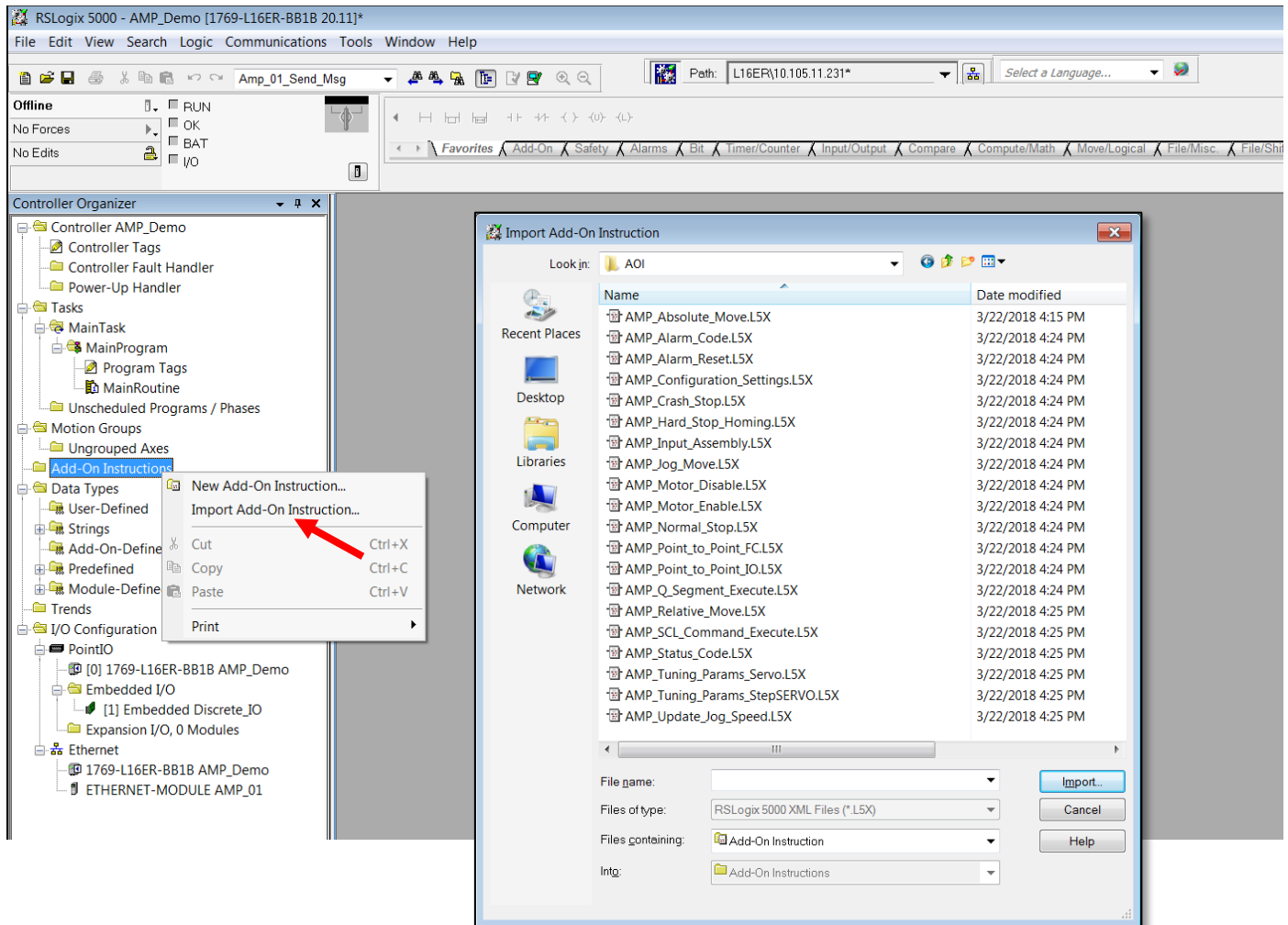
STEP 4 **Setting the RPI**

After creating the module, navigate to the Connection tab of the module properties and set the Requested Packet Interval (RPI) to the desired value. Larger values such as 250ms will minimize the load on the network, while smaller values such as 25ms can provide quicker response times between PLC and drive.



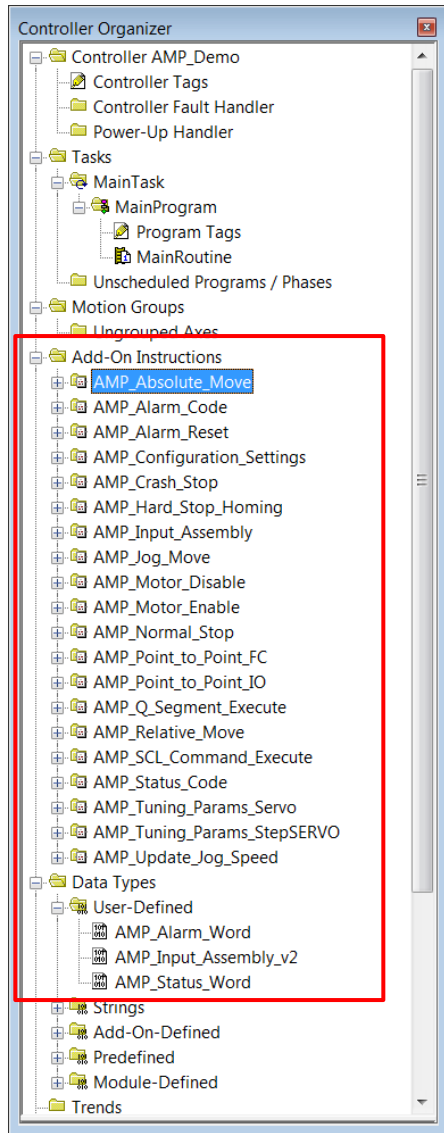
STEP 5 Add-On Instruction L5X Files

Import the Add-On Instructions into your new program following the screenshot below as an example. Locate the AOI folder containing .L5X files that was packaged together with this application note inside the ZIP file.



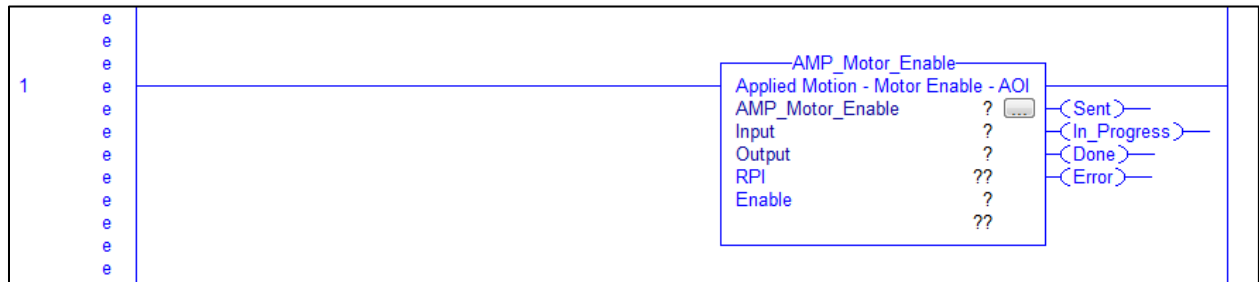
STEP 6 **Confirm all the Imports**

Import all AOIs that you plan to use. At a minimum to complete this introduction: AMP_Motor_Enable, AMP_Motor_Disable and AMP_Status_Code. In the appendix of this application note, you will find detailed information about all included AOIs.



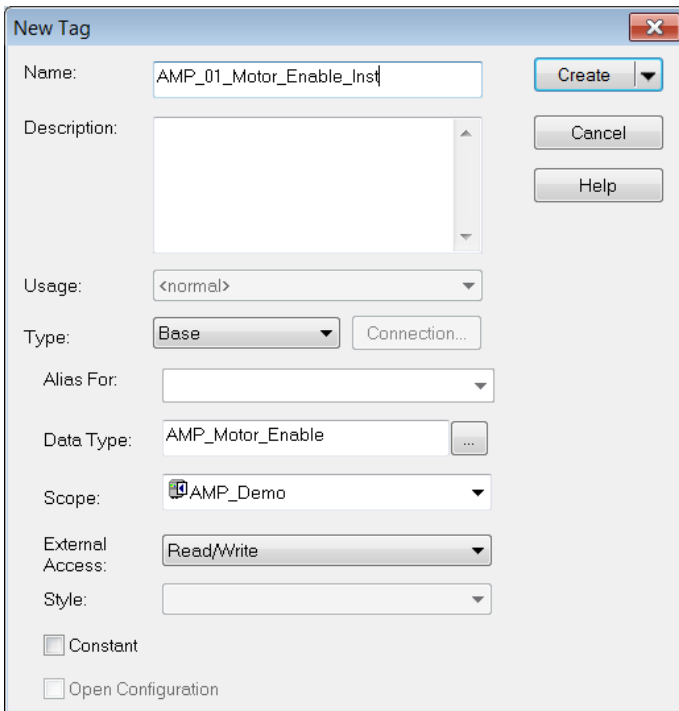
STEP 7 Place an AOI on a Rung

Place an AMP_Motor_Enable AOI on a rung



STEP 8 **Declare the AOI Instance Tag**

Declare the AOI instance tag: AMP_Motor_Enable_Inst



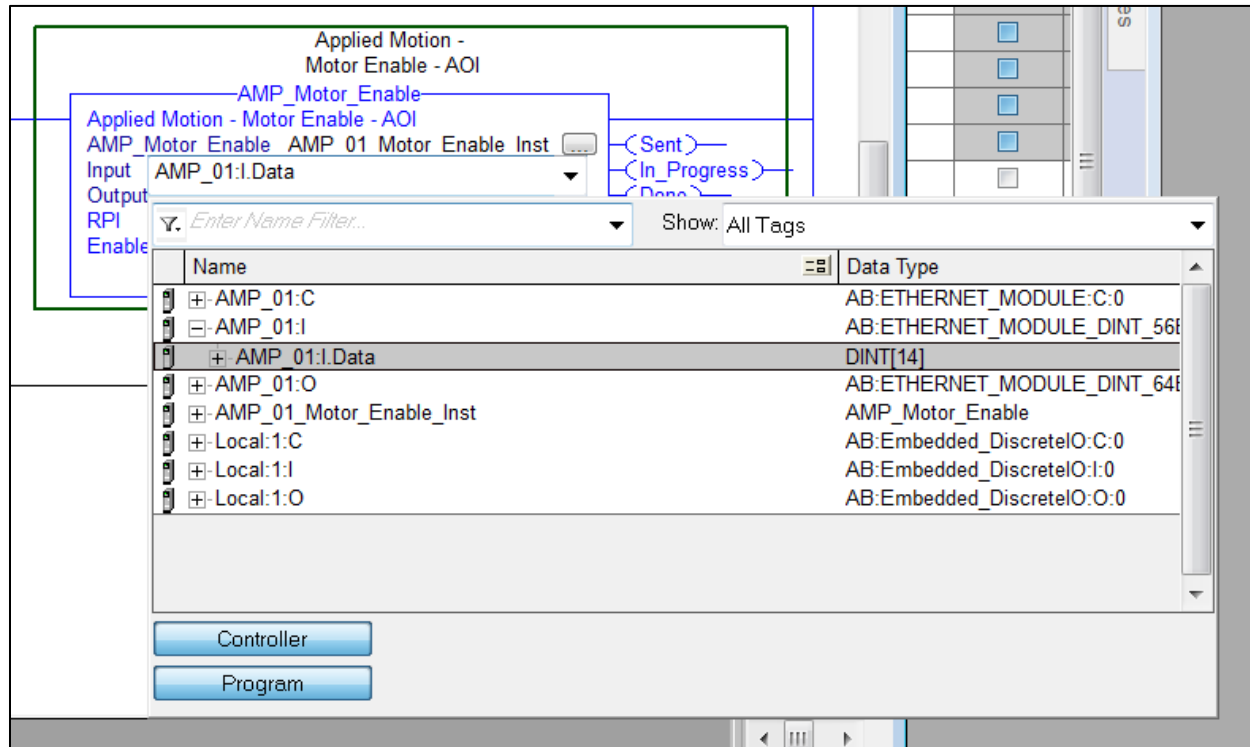
The "New Tag" dialog box is shown with the following fields and options:

- Name:** AMP_01_Motor_Enable_Inst
- Description:** (Empty text area)
- Usage:** <normal>
- Type:** Base (dropdown menu) and Connection... (button)
- Alias For:** (Empty dropdown menu)
- Data Type:** AMP_Motor_Enable (dropdown menu) and ... (button)
- Scope:** AMP_Demo (dropdown menu)
- External Access:** Read/Write (dropdown menu)
- Style:** (Empty dropdown menu)
- ☐ Constant
- ☐ Open Configuration

Buttons: Create (dropdown menu), Cancel, Help.

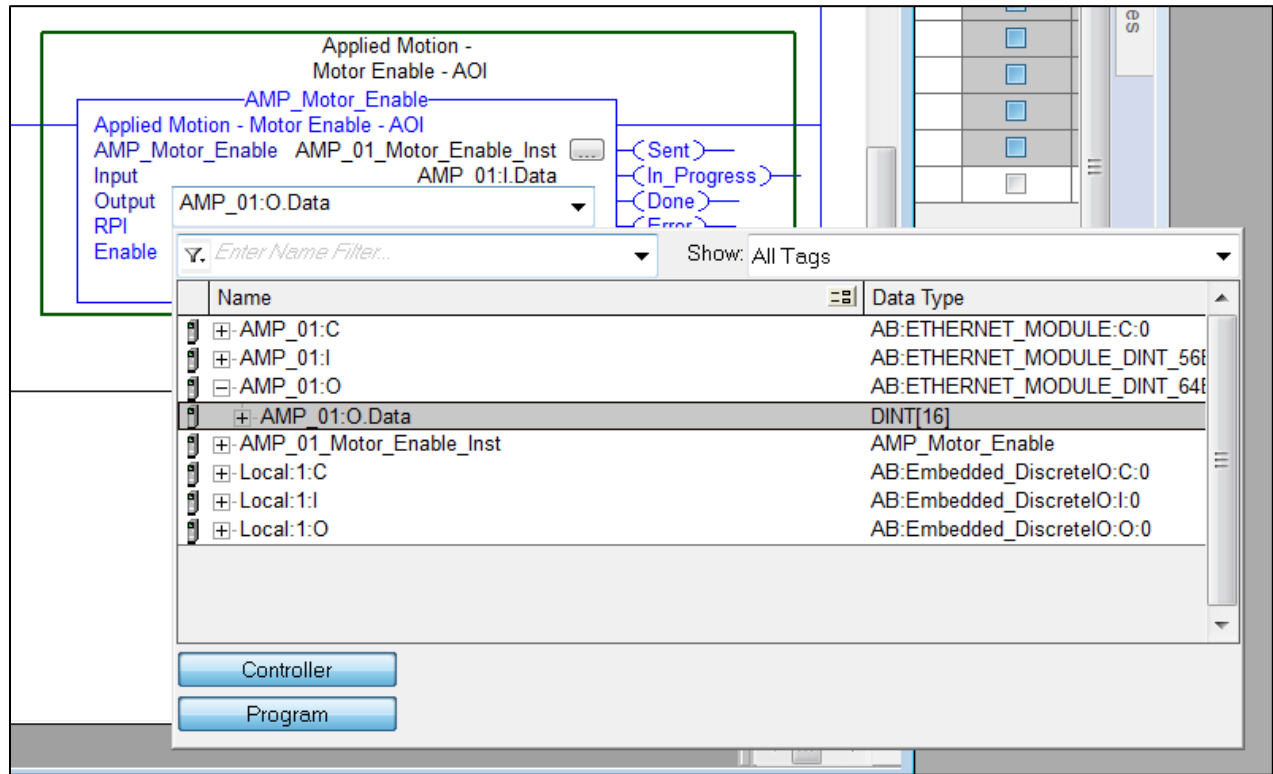
STEP 9 Specifying Parameters – Input

In the Input parameter field, specify the Data member of the servo/stepper drive's input module that was defined in STEP 3, i.e. AMP_01:I.Data, which should be of type DINT[14].



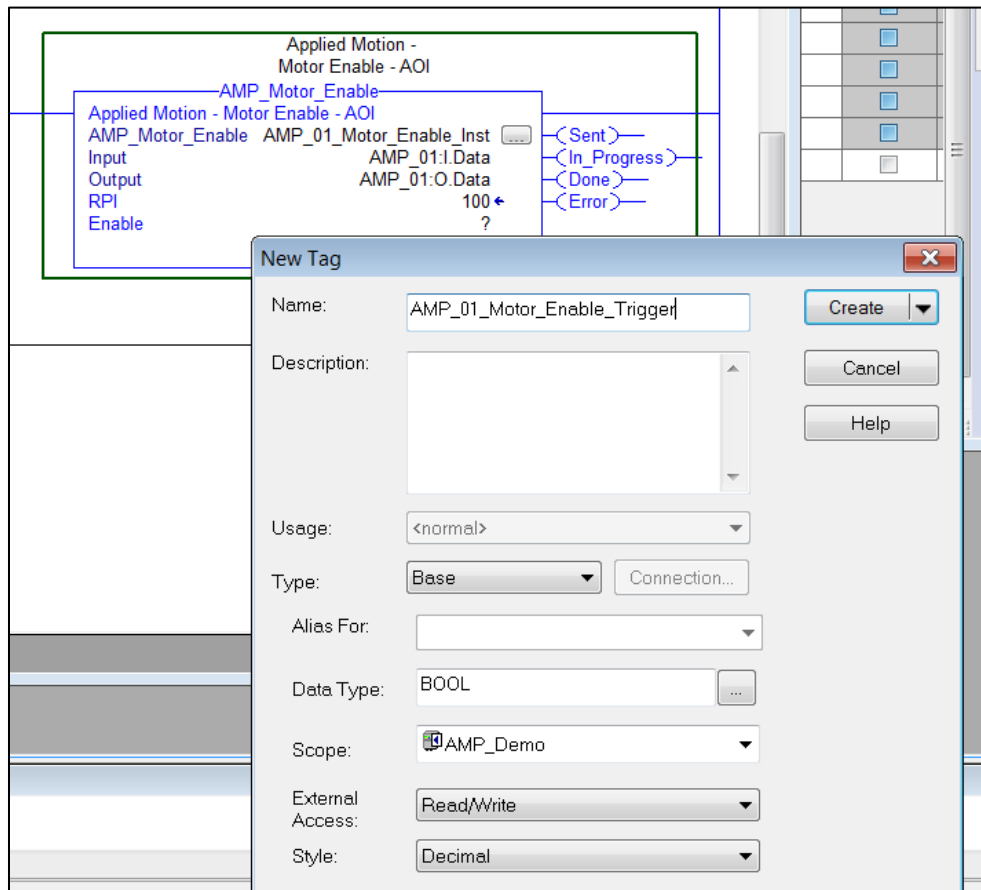
STEP 10 Specifying Parameters – Output

In the Output parameter field, specify the Data member of the servo/stepper drive's output module that was defined in STEP 3, i.e. AMP_01:O.Data, which should be of type DINT[16].



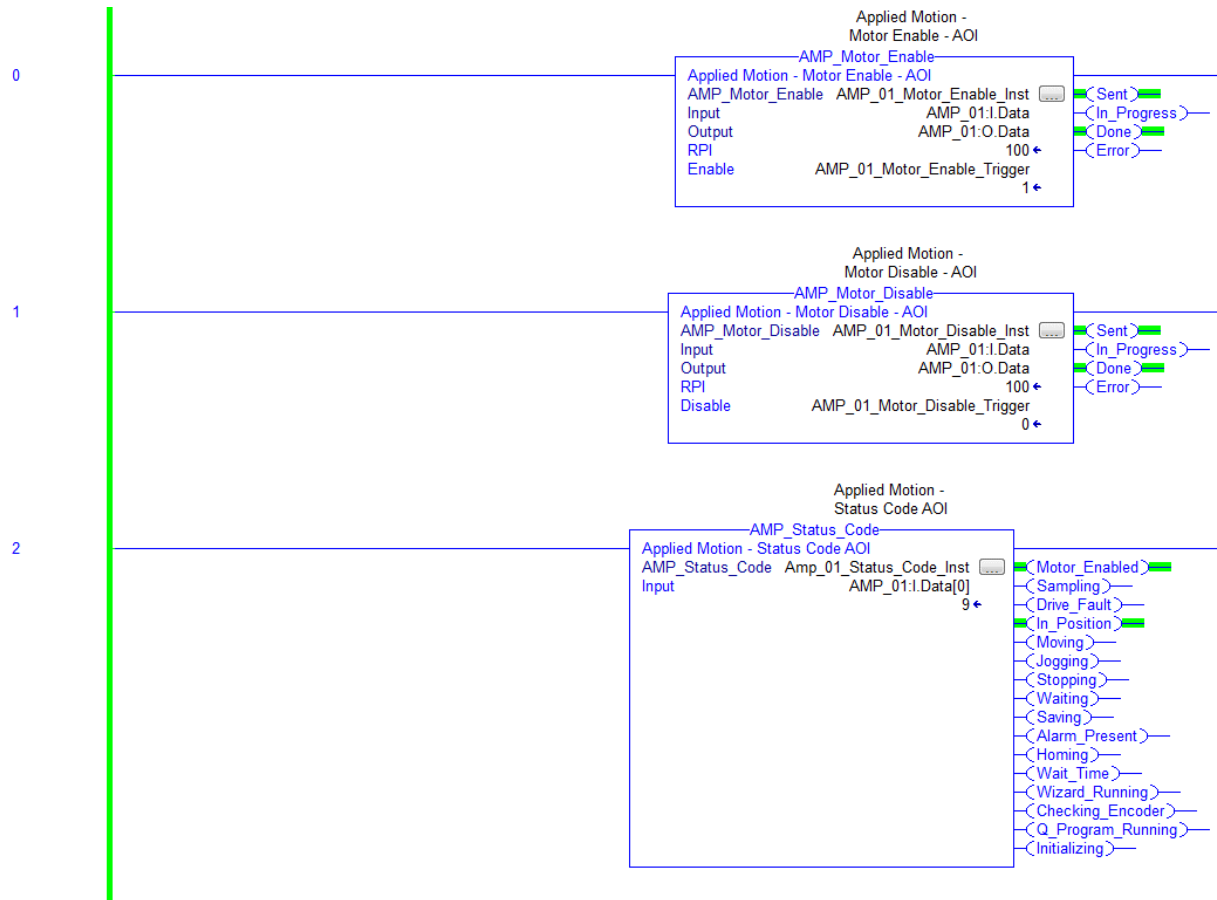
STEP 12 Specifying Parameters – Enable

Create a new tag (e.g. AMP_01_Motor_Enable_Trigger) or specify an existing tag of BOOL type, which will be used to trigger the execution of the AOI's intended functionality.



STEP 13 Add More AOIs

Repeat STEP 7 through STEP 12 twice for adding AMP_Motor_Disable and the AMP_Status_Code AOIs to your routine. The AMP_Status_Code requires only one parameter: word 0 of the input assembly, e.g. AMP_01:I.Data[0]. The code can now be tested by manually changing the values of the Enable and Disable trigger tags from false to true. Observe the effect of these commands by monitoring the Motor_Enabled output value of the AMP_Status_Code AOI.



3.2 Add AOIs that require Messaging

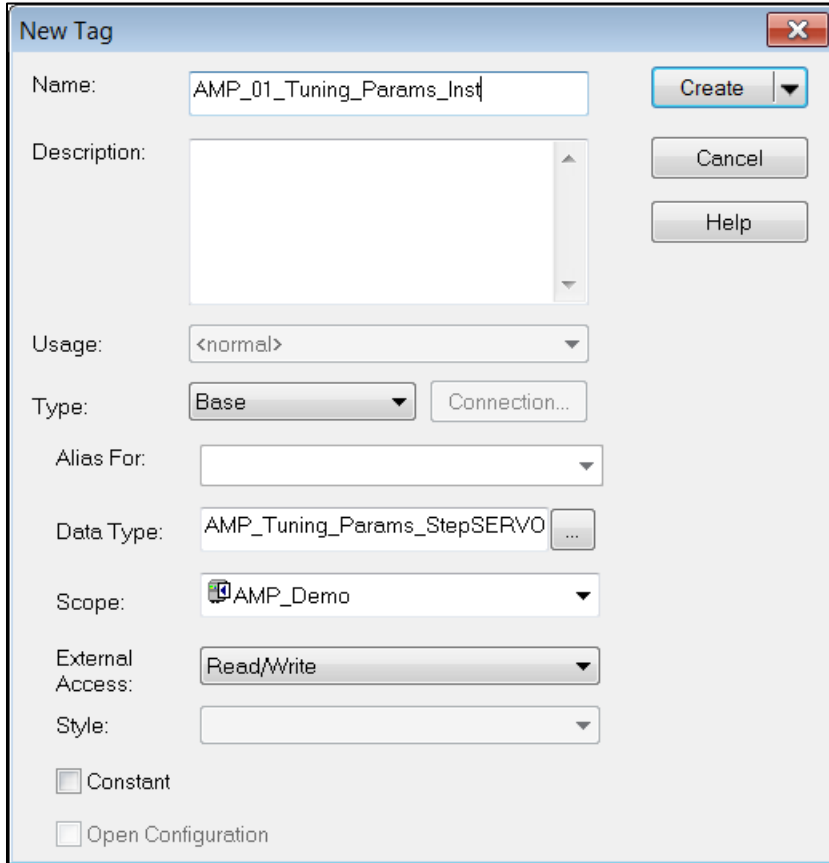
The following AOIs need special parameters because they incorporate messaging:

- AMP_Configuration_Settings
- AMP_Tuning_Params_Servo
- AMP_Tuning_Params_StepSERVO

These AOIs require that the user set up the MESSAGE tag as well as a write buffer tag and a response buffer tag. This section will show how to set up one of these AOIs.

STEP 2 **Declare Variables – AOI Instance Tag**

Declare the AOI instance tag: AMP_Tuning_Params_Inst



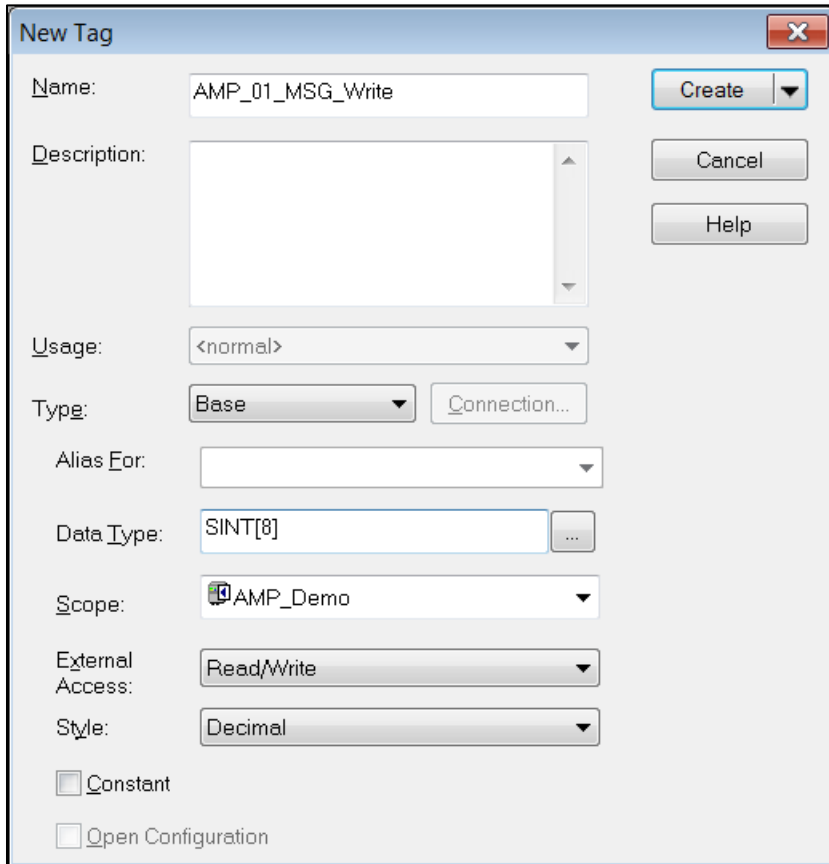
The "New Tag" dialog box is shown with the following fields and values:

- Name:** AMP_01_Tuning_Params_Inst
- Description:** (Empty text area)
- Usage:** <normal>
- Type:** Base (selected in dropdown), Connection... (button)
- Alias For:** (Empty dropdown)
- Data Type:** AMP_Tuning_Params_StepSERVO (selected in dropdown)
- Scope:** AMP_Demo (selected in dropdown)
- External Access:** Read/Write (selected in dropdown)
- Style:** (Empty dropdown)
- ☐ Constant
- ☐ Open Configuration

Buttons: Create, Cancel, Help

STEP 3 **Declare Variables – MSG_Write**

Declare a controller scoped array of 8 SINT which will be used as the Message's Source Element, e.g. AMP_01_MSG_Write.



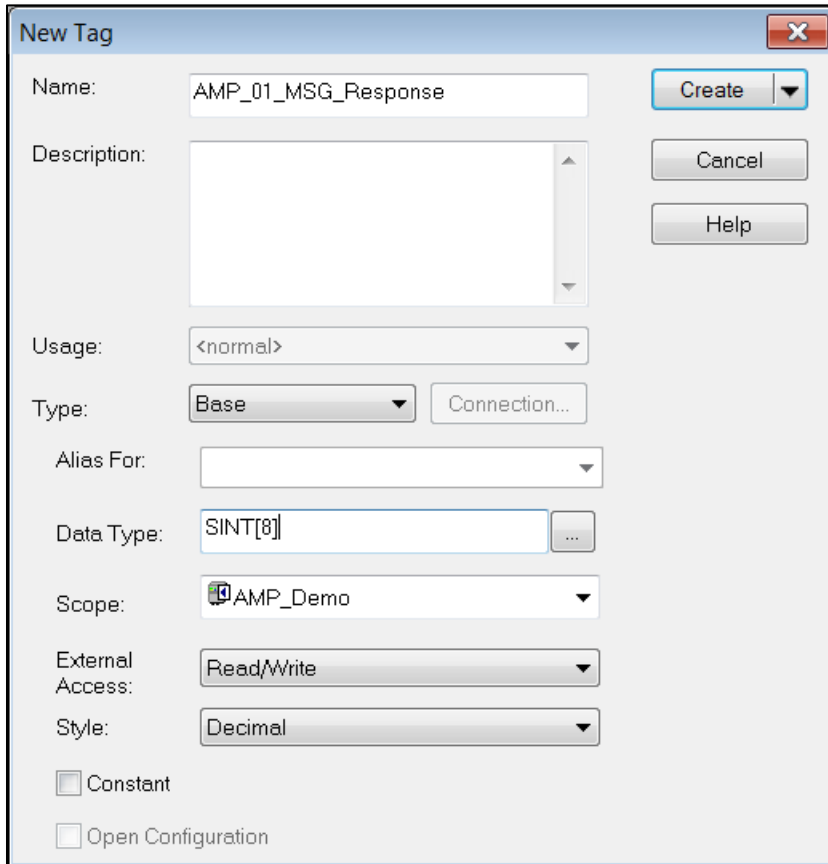
The "New Tag" dialog box is shown with the following configuration:

- Name:** AMP_01_MSG_Write
- Description:** (Empty text area)
- Usage:** <normal>
- Type:** Base
- Alias For:** (Empty dropdown)
- Data Type:** SINT[8]
- Scope:** AMP_Demo
- External Access:** Read/Write
- Style:** Decimal
- ☐ Constant
- ☐ Open Configuration

Buttons: Create, Cancel, Help

STEP 4 **Declare Variables – MSG_Response**

Declare a controller scoped array of 8 SINT, e.g. AMP_01_MSG_Response, which will be used as the Message's Destination Element.



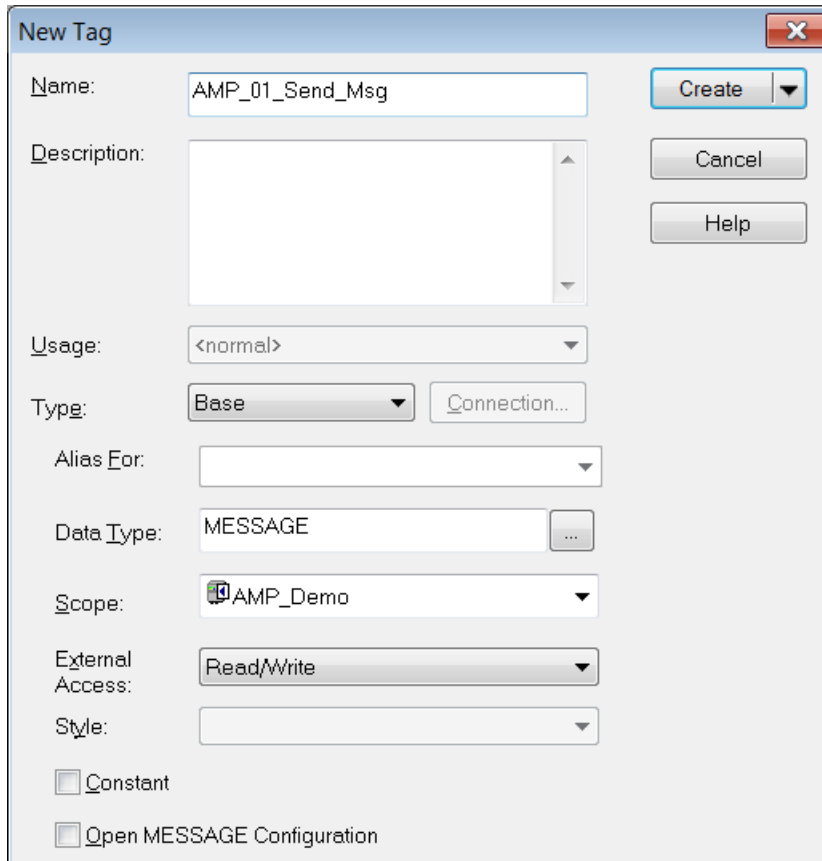
The "New Tag" dialog box is shown with the following configuration:

- Name:** AMP_01_MSG_Response
- Description:** (Empty text area)
- Usage:** <normal>
- Type:** Base (with a "Connection..." button)
- Alias For:** (Empty dropdown)
- Data Type:** SINT[8] (with a selection button "...")
- Scope:** AMP_Demo (with a selection button "...")
- External Access:** Read/Write
- Style:** Decimal
- ☐ Constant
- ☐ Open Configuration

Buttons on the right: Create (with a dropdown arrow), Cancel, and Help.

STEP 5 **Declare Variables – MSG_Tag**

Declare a controller scoped MESSAGE typed tag, e.g. AMP_01_Send_Msg



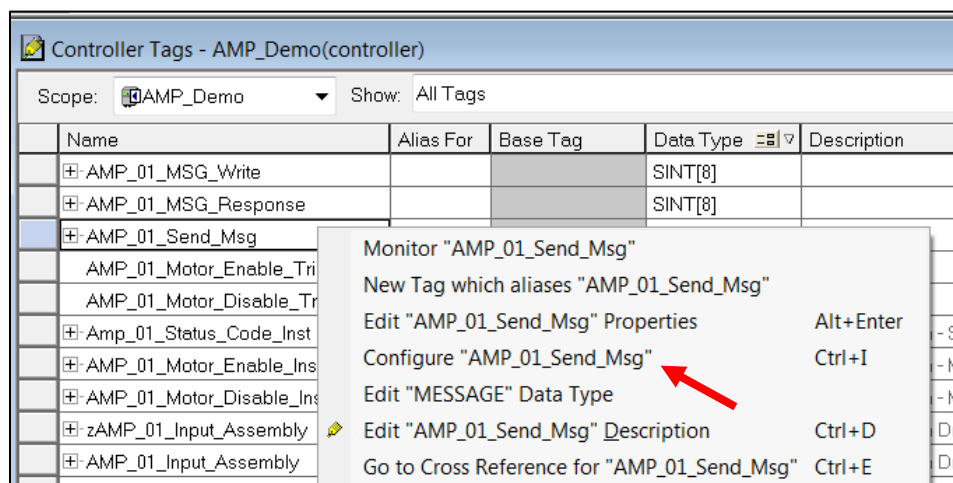
The "New Tag" dialog box is shown with the following configuration:

- Name:** AMP_01_Send_Msg
- Description:** (Empty text area)
- Usage:** <normal>
- Type:** Base (dropdown menu)
- Alias For:** (Empty dropdown menu)
- Data Type:** MESSAGE (dropdown menu)
- Scope:** AMP_Demo (dropdown menu)
- External Access:** Read/Write (dropdown menu)
- Style:** (Empty dropdown menu)
- ☐ Constant
- ☐ Open MESSAGE Configuration

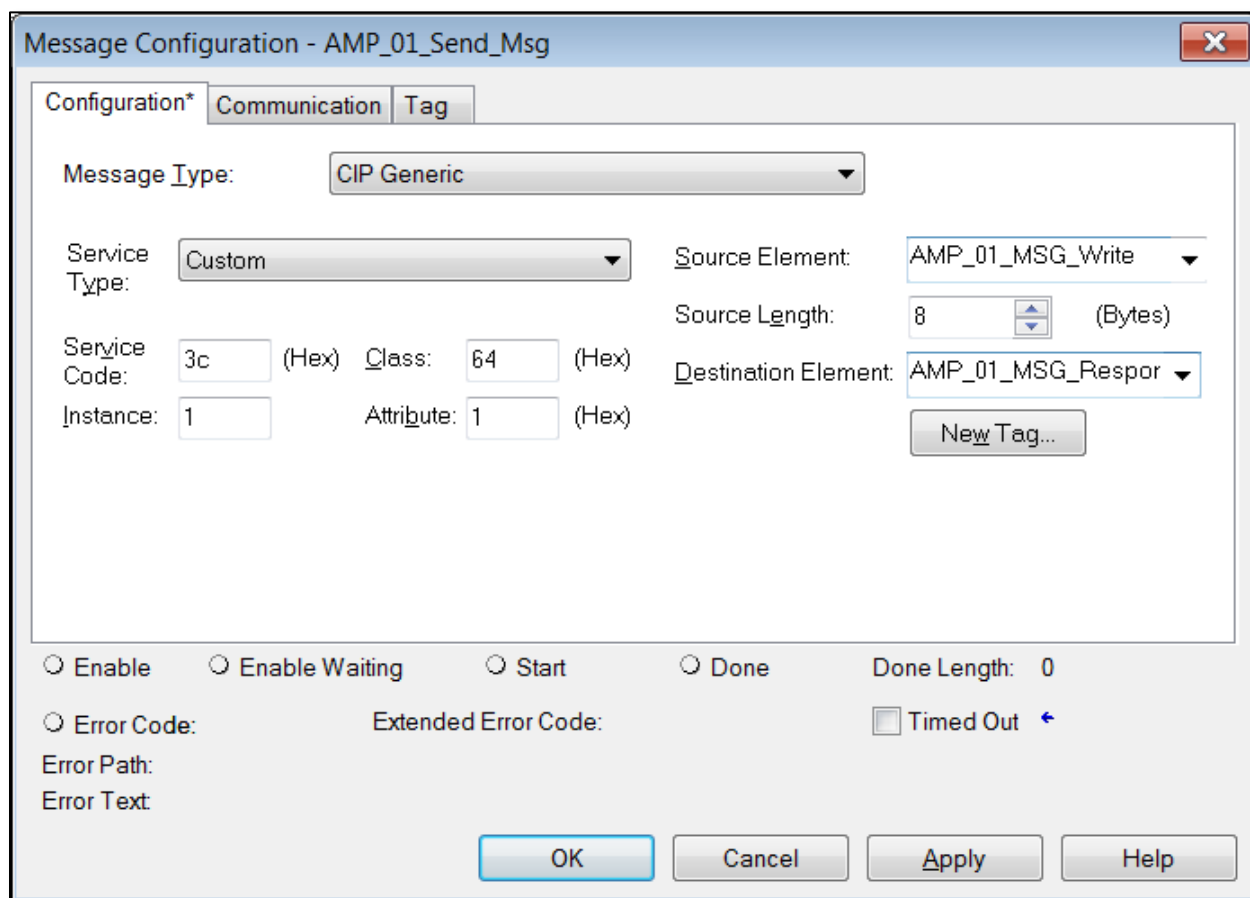
Buttons: Create, Cancel, Help

STEP 6 Configure the Message Tag

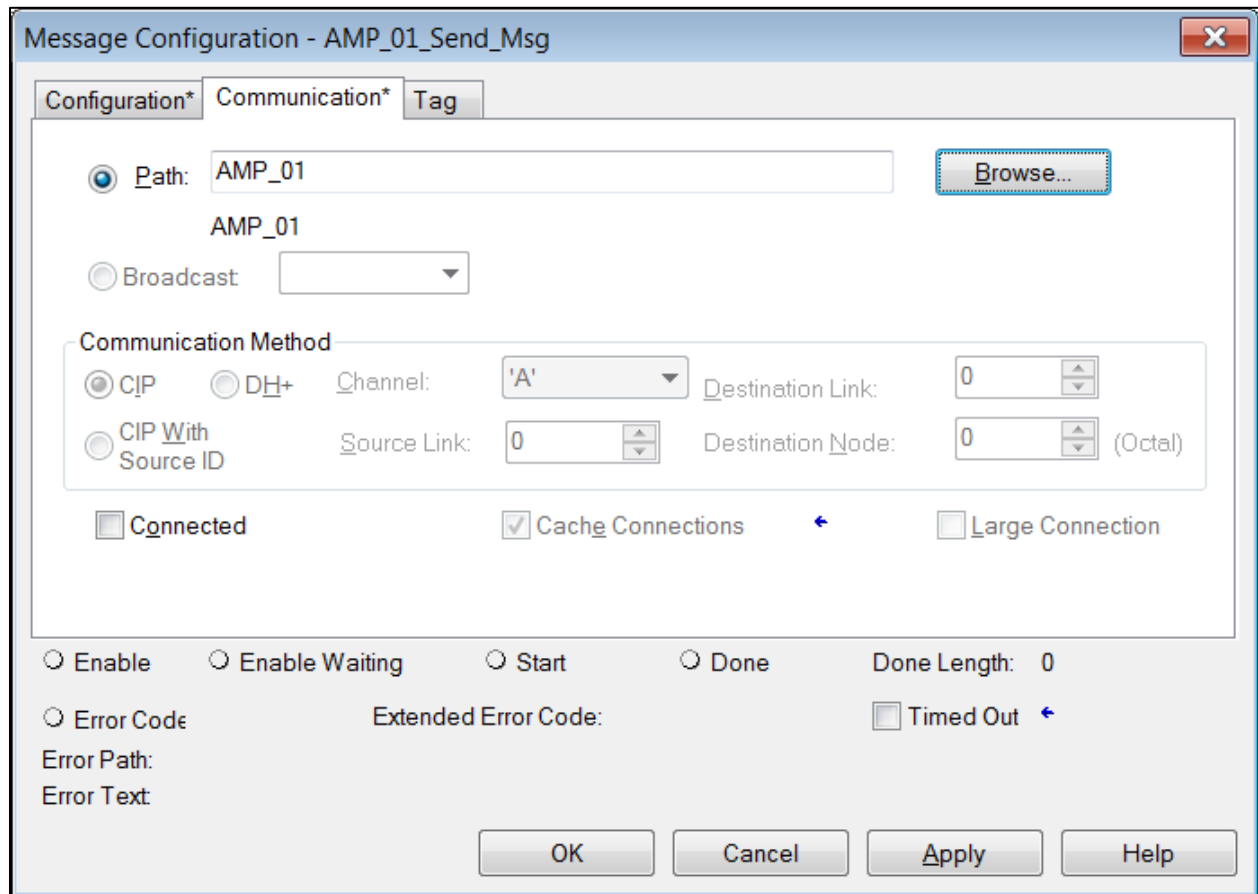
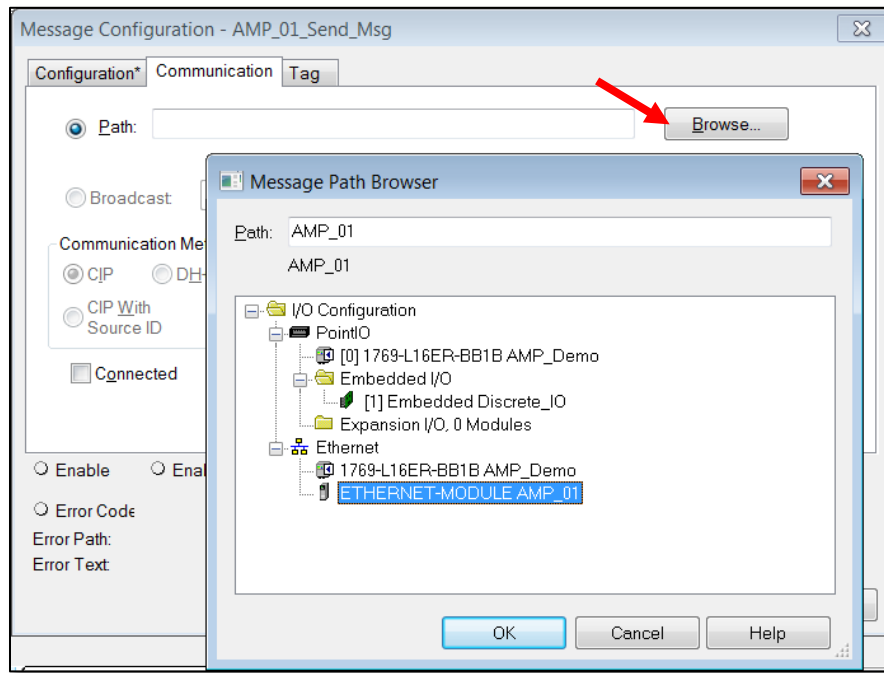
Open the tag monitoring, right-click the message tag and select Configure



Set the Configuration tab elements as follows:

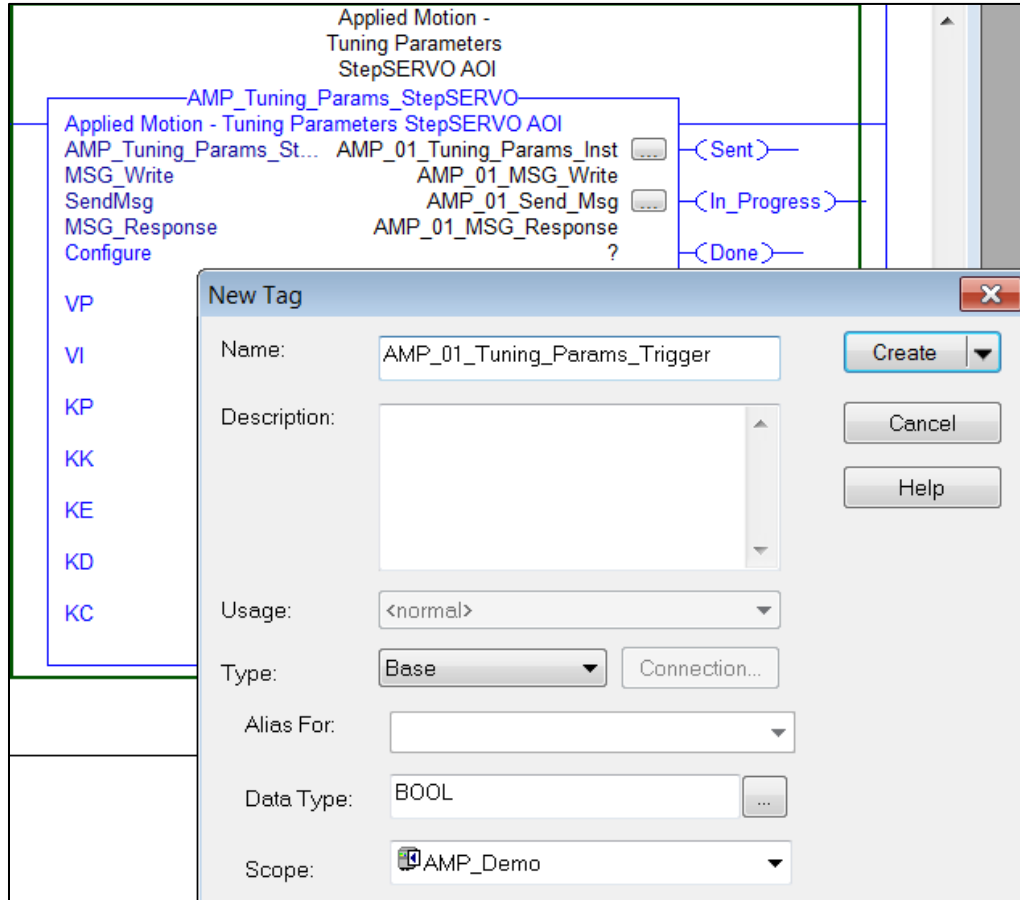


Set the Communication tab elements as follows:



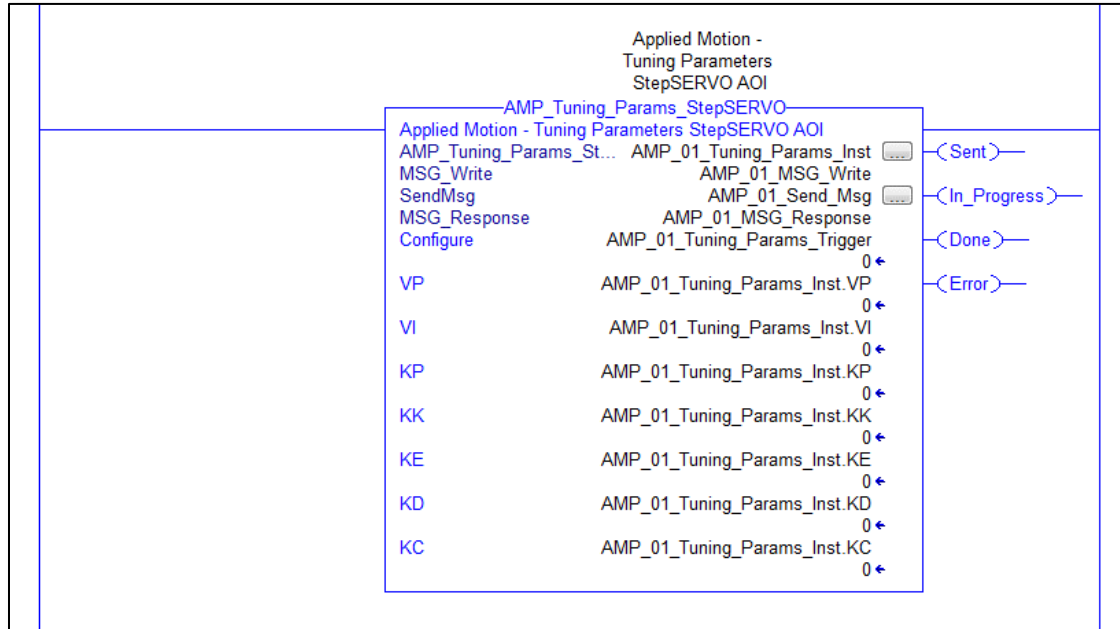
STEP 8 Specifying Parameters – Configure

Create a new tag (e.g. AMP_01_Tuning_Params_StepSERVO_Trigger) or specify an existing tag of BOOL type, which will be used to trigger the execution of the AOI's intended functionality.



STEP 9 Specifying Other Parameters

For the other parameters, either right-click to create a “new tag” for each of the remaining fields, or alternatively, enter AMP_01_Tuning_Params_Inst.<Parameter> as demonstrated in the following figure:



STEP 10 Specify Parameter Values

Specify appropriate values for the drive configuration parameters either via programming or by hand directly into the variables listed within the AOI.

Applied Motion - Tuning Parameters StepSERVO AOI		
AMP_Tuning_Params_StepSERVO		
Applied Motion - Tuning Parameters StepSERVO AOI	AMP_01_Tuning_Params_Inst	(Sent)
AMP_Tuning_Params_St...	AMP_01_MSG_Write	
MSG_Write	AMP_01_Send_Msg	(In_Progress)
SendMsg	AMP_01_MSG_Response	
MSG_Response	AMP_01_Tuning_Params_Trigger	(Done)
Configure	0	
VP	AMP_01_Tuning_Params_Inst.VP	(Error)
	15000	
VI	AMP_01_Tuning_Params_Inst.VI	
	1000	
KP	AMP_01_Tuning_Params_Inst.KP	
	5000	
KK	AMP_01_Tuning_Params_Inst.KK	
	0	
KE	AMP_01_Tuning_Params_Inst.KE	
	15000	
KD	AMP_01_Tuning_Params_Inst.KD	
	2000	
KC	AMP_01_Tuning_Params_Inst.KC	
	15000	

4 APPENDIX: ADD-ON INSTRUCTIONS

Following is a comprehensive list of all AOIs included in the ZIP folder accompanying this Application Note:


4.1 AMP_Input_Assembly AOI	39
4.1.1 Overview	39
4.1.2 Functionality	40
4.1.3 Parameters	40
4.2 AMP_STATUS_CODE AOI	41
4.2.1 Overview	41
4.2.2 Functionality	41
4.2.3 Parameters	42
4.3 AMP_ALARM_CODE AOI	43
4.3.1 Overview	43
4.3.2 Functionality	43
4.3.3 Parameters	44
4.4 AMP_MOTOR_ENABLE AOI	45
4.4.1 Overview	45
4.4.2 Functionality	45
4.4.3 Parameters	45
4.5 AMP_MOTOR_DISABLE AOI	47
4.5.1 Overview	47
4.5.2 Functionality	47
Parameters	47
4.5.3	47
4.6 AMP_ABSOLUTE_MOVE AOI	49
4.6.1 Overview	49
4.6.2 Functionality	49
4.6.3 Parameters	50
4.7 AMP_RELATIVE_MOVE AOI	51
4.7.1 Overview	51
4.7.2 Functionality	51
4.7.3 Parameters:	52
4.8 AMP_NORMAL_STOP AOI	53
4.8.1 Overview	53
4.8.2 Functionality	53
4.8.3 Parameters	53
4.9 AMP_CRASH_STOP AOI	55
4.9.1 Overview	55
4.9.2 Functionality	55
4.9.3 Parameters:	55
4.10 AMP_ALARM_RESET AOI	57
4.10.1 Overview	57
4.10.2 Functionality	57
4.10.3 Parameters:	57
4.11 AMP_SCL_COMMAND_EXECUTE AOI	59

4.11.1	Overview	59
4.11.2	Functionality	59
4.11.3	Exceptions/Notes	60
4.11.4	Parameters:	60
4.12	AMP_POINT_TO_POINT_IO AOI	61
4.12.1	Overview	61
4.12.2	Functionality	62
4.12.3	Parameters	63
4.13	AMP_POINT_TO_POINT_FC AOI	65
4.13.1	Overview	65
4.13.2	Functionality	65
	Parameters	66
4.13.3	66
4.14	AMP_JOG_MOVE AOI	67
4.14.1	Overview	67
4.14.2	Functionality	67
4.14.3	Exceptions/Notes	68
	Parameters	68
4.14.4	68
4.15	AMP_UPDATE_JOG_SPEED AOI	69
4.15.1	Overview	69
4.15.2	Functionality	69
	Parameters	69
4.15.3	69
4.16	AMP_Q_SEGMENT_EXECUTE AOI	71
4.16.1	Overview	71
4.16.2	Functionality	71
4.16.3	Parameters:	71
4.17	AMP_HARD_STOP_HOMING AOI	73
4.17.1	Overview	73
4.17.2	Functionality	74
4.17.3	Parameters	74
4.18	AMP_TUNING_PARAMS_SERVO AOI	76
4.18.1	Overview	76
4.18.2	Functionality	76
4.18.3	Parameters:	77
4.19	AMP_TUNING_PARAMS_STEPSERVO AOI	78
4.19.1	Overview	78
4.19.2	Functionality	78
4.19.3	Parameters:	79
4.20	AMP_CONFIGURATION_SETTINGS AOI	80
4.20.1	Overview	80
4.20.2	Functionality	80
4.20.3	Parameters	81

4.1 AMP_Input_Assembly AOI

4.1.1 Overview

The Input Assembly AOI takes raw input data from the drive (Input Assembly 0x65) and re-interprets the data, using either scaling or copying, to appropriately named outputs.

Applied Motion - Input Assembly AOI	
AMP_Input_Assembly	
Applied Motion - Input Assembly AOI	
AMP_Input_Assembly	AMP_01_Input_Assembly_Inst 
Input	AMP_01:I.Data
Output	AMP_01_Input_Assembly
Status_Code	16#0000_0009+
Alarm_Code	16#0000_0000+
Voltage	48.3+
Current	-59.0+
Temp	53.0+
Encoder_Position	0+
Absolute_Position	0+
Actual_Velocity	0.0+
Position_Error	0+
Extended_Inputs	16#8000+
Main_Board_Inputs	16#0007+
Analog_Input_1	156+
Analog_Input_2	0+

4.1.2 Functionality

The Input Assembly AOI does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The AOI copies the raw data from the input assembly (Input parameter) into the provided user defined data type tag (Input_Assembly_Tag InOut parameter)
- The AOI copies and/or scales data from the Input and outputs them to named fields.

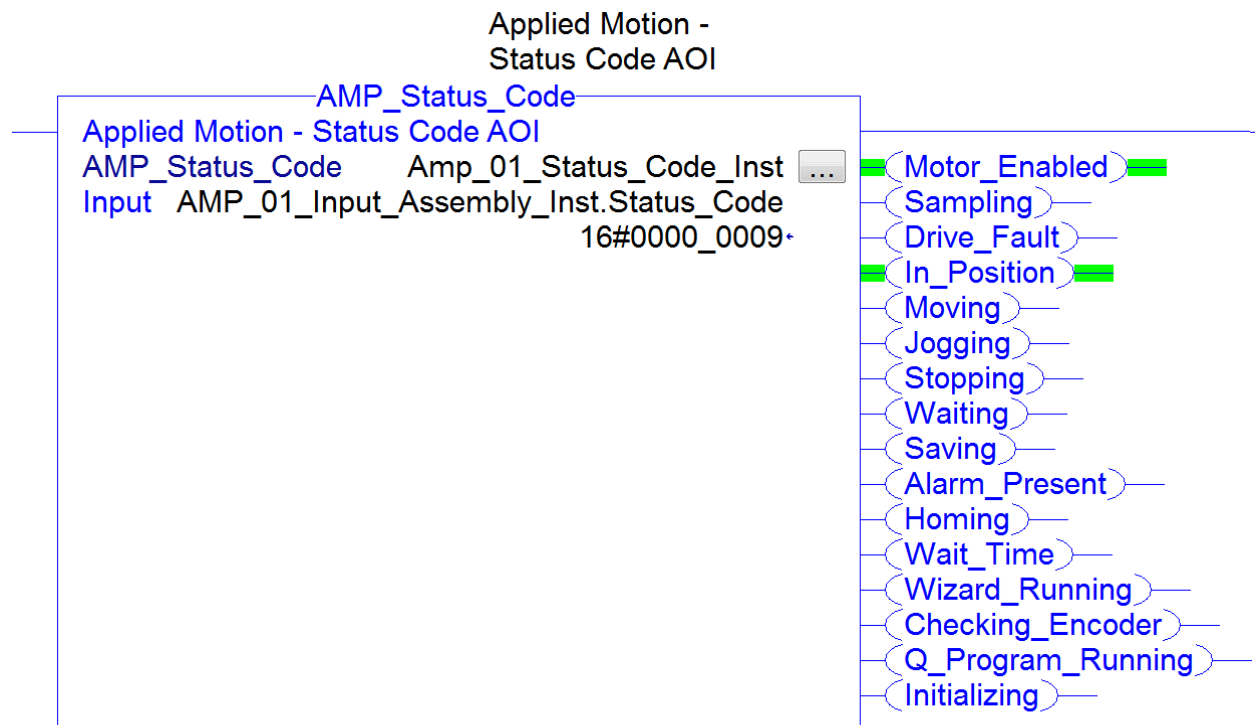
4.1.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Input_Assembly_Tag	InOut	AMP_Input_Assembly_v2	Copy of Drive Input Assembly as UDDT
Status_Code	Output	DINT	Status Code
Alarm_Code	Output	DINT	Alarm Code
Voltage	Output	REAL	Supply Voltage (V)
Current	Output	REAL	Actual Current (mA)
Temp	Output	REAL	Drive Temperature (deg C)
Encoder_Position	Output	DINT	Encoder Position (20,000 counts = 1 shaft turn)
Absolute_Position	Output	DINT	Absolute Position (20,000 counts = 1 shaft turn)
Velocity	Output	REAL	Actual Velocity (rev/sec, 0 when no encoder)
Entended_Inputs	Output	DINT	Input Status (Extended)
Main_Board_Inputs	Output	DINT	Input Status (Main board)
Analog_Input_1	Output	DINT	Analog Input 1 (ADC counts, 0=min V, 16383 = max V)
Analog_Input_2	Output	DINT	Analog Input 2 (ADC counts, 0=min V, 16383 = max V)

4.2 AMP_Status_Code AOI

4.2.1 Overview

The Status Code AOI uses the raw data stored in the Status Code word (Element 0 from Input Assembly 0x65) to set individual appropriately named boolean outputs. The equivalent drive command is SC.



4.2.2 Functionality

The Status Code AOI does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The AOI copies individual booleans from the Status Code word (Element 0 from Input Assembly 0x65) and sets appropriately named outputs.

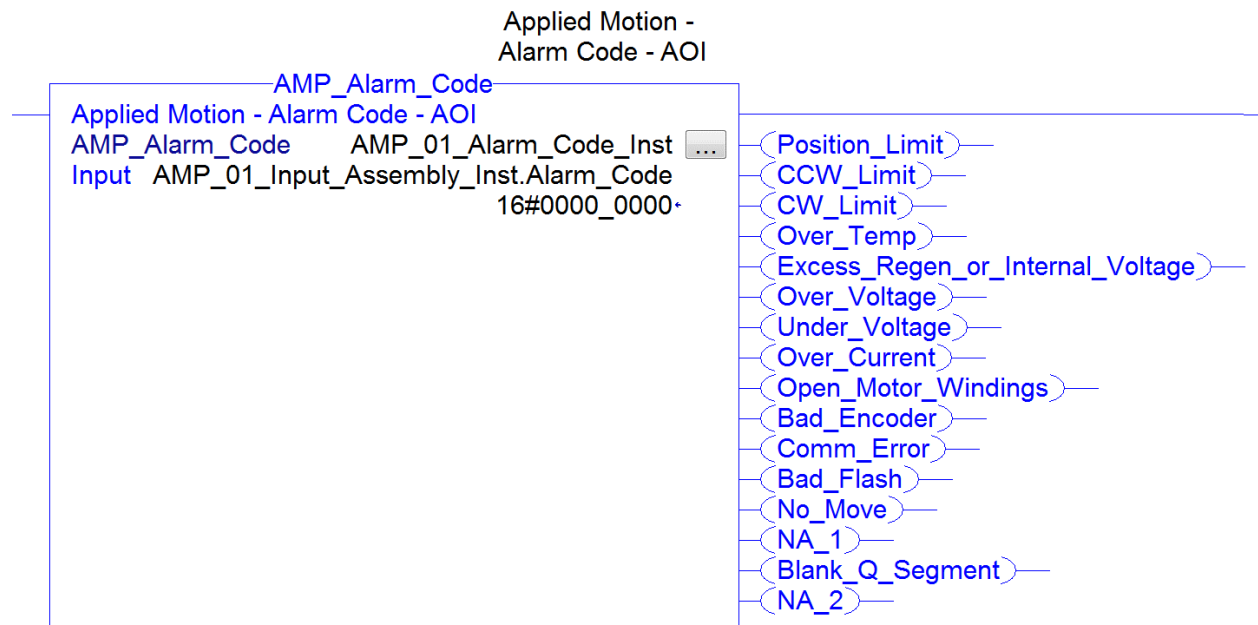
4.2.3 Parameters

Name	Usage	Data Type	Description
Input	Input	DINT	Drive Status Code word (Element 0 from Input Assembly 0x65)
Enabled	Output	BOOL	Motor Enabled (Motor Disabled if this bit = 0)
Sampling	Output	BOOL	Sampling (for Quick Tuner)
Drive_Fault	Output	BOOL	Drive Fault (check Alarm Code)
In_Position	Output	BOOL	In Position (motor is in position)
Moving	Output	BOOL	Moving (motor is moving)
Jogging	Output	BOOL	Jogging (currently in jog mode)
Stopping	Output	BOOL	Stopping (in the process of stopping from a stop command)
Waiting	Output	BOOL	Waiting (for an input; executing a WI command)
Saving	Output	BOOL	Saving (parameter data is being saved)
Alarm_Present	Output	BOOL	Alarm present (check Alarm Code)
Homing	Output	BOOL	Homing (executing an SH command)
Wait_Time	Output	BOOL	Waiting (for time; executing a WD or WT command)
Wizard_Running	Output	BOOL	Wizard running (Timing Wizard is running)
Checking_Encoder	Output	BOOL	Checking encoder (Timing Wizard is running)
Q_Program_Running	Output	BOOL	Q Program is running
Initializing	Output	BOOL	Initializing (happens at power up) ; Servo Ready (for SV200 drives only)

4.3 AMP_Alarm_Code AOI

4.3.1 Overview

The Alarm Code AOI uses the raw data stored in the Alarm Code word (Element 1 from Input Assembly 0x65) to set individual appropriately named boolean outputs. The equivalent drive command is AC.



4.3.2 Functionality

The Alarm Code AOI does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The AOI copies individual booleans from the Alarm Code word (Element 1 from Input Assembly 0x65) and sets appropriately named outputs.

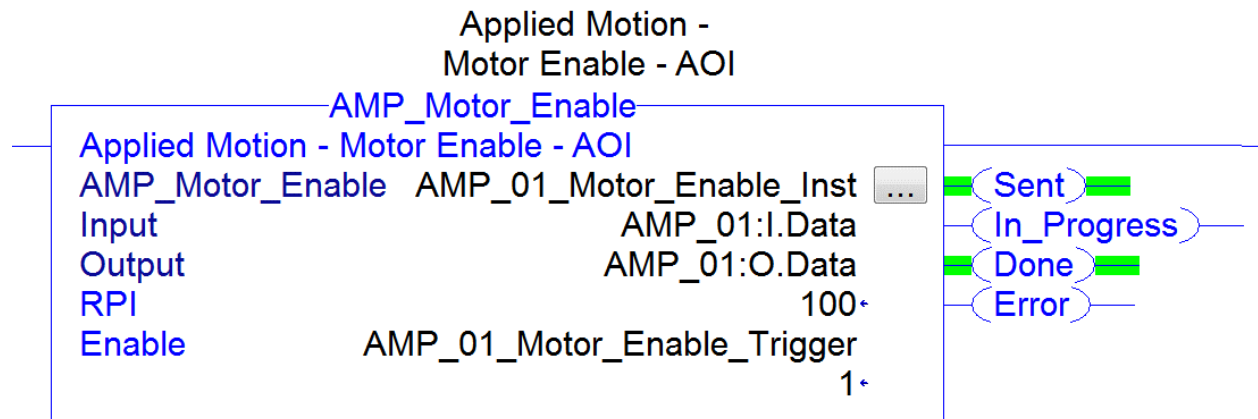
4.3.3 Parameters

Name	Usage	Data Type	Description
Input	Input	DINT	Drive Alarm Code word (Element 1 from Input Assembly 0x65)
Position_Limit	Output	BOOL	Position Limit
CCW_Limit	Output	BOOL	CCW Limit
CW_Limit	Output	BOOL	CW Limit
Over_Temp	Output	BOOL	Over Temperature
Excess_Regen_or_Internal_Voltage	Output	BOOL	Excess Regen or Internal Voltage
Over_Voltage	Output	BOOL	Over Voltage
Under_Voltage	Output	BOOL	Under Voltage
Over_Current	Output	BOOL	Over Current
Open_Motor_Windings	Output	BOOL	Bad Hall Sensor / Open Motor Windings
Bad_Encoder	Output	BOOL	Bad Encoder
Comm_Error	Output	BOOL	Comm Error
Bad_Flash	Output	BOOL	Bad Flash
Wizard_Failed	Output	BOOL	Wizard Failed / No Move
Current_Foldback	Output	BOOL	Current Foldback / Motor Resistance
Blank_Q_Segment	Output	BOOL	Blank Q Segment
No_Move	Output	BOOL	No Move

4.4 AMP_Motor_Enable AOI

4.4.1 Overview

The Motor Enable AOI restores drive current to motor. The equivalent drive command is ME.



4.4.2 Functionality

The Motor Enable AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: none
- Drive Command: "Motor Enable" (command word value of 0x2).
- Done output: when drive status indicates it is "Enabled".

Type: Buffered

4.4.3 Parameters

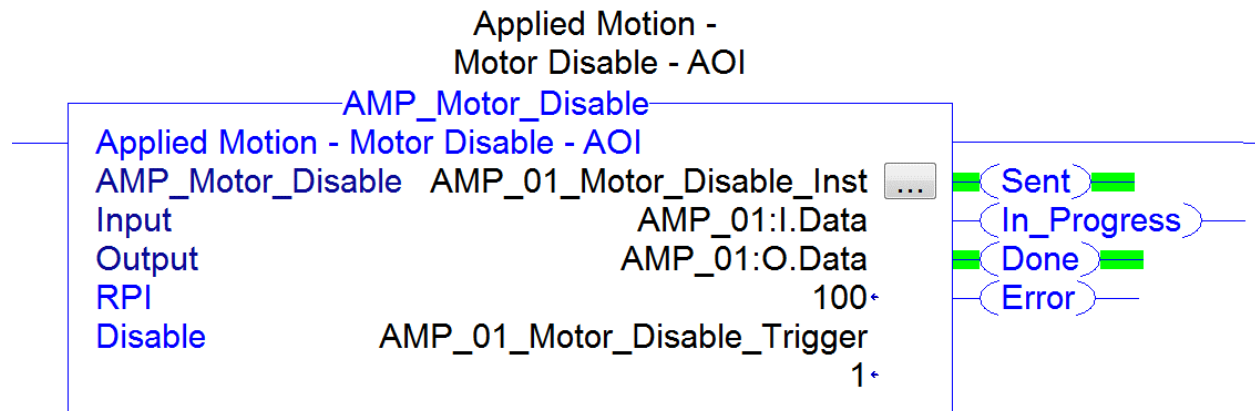
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)

Name	Usage	Data Type	Description
Enable	Input	BOOL	Enables the drive
Sent	Output	BOOL	Enable has been sent to drive
In_Progress	Output	BOOL	Enable in progress
Done	Output	BOOL	Enable completed without errors – Motor is Enabled
Error	Output	BOOL	Enable failed to complete (Always false – Future use only)
Output	DINT	Internal state of progress	

4.5 AMP_Motor_Disable AOI

4.5.1 Overview

The Motor Disable AOI disables motor outputs (reduces motor current to zero). The equivalent drive command is MD.



4.5.2 Functionality

The Motor Disable AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: none
- Drive Command: "Motor Disable" (command word value of 0x4).
- Done output: when drive status indicates it is not "Enabled".
- Type: Buffered

4.5.3 Parameters

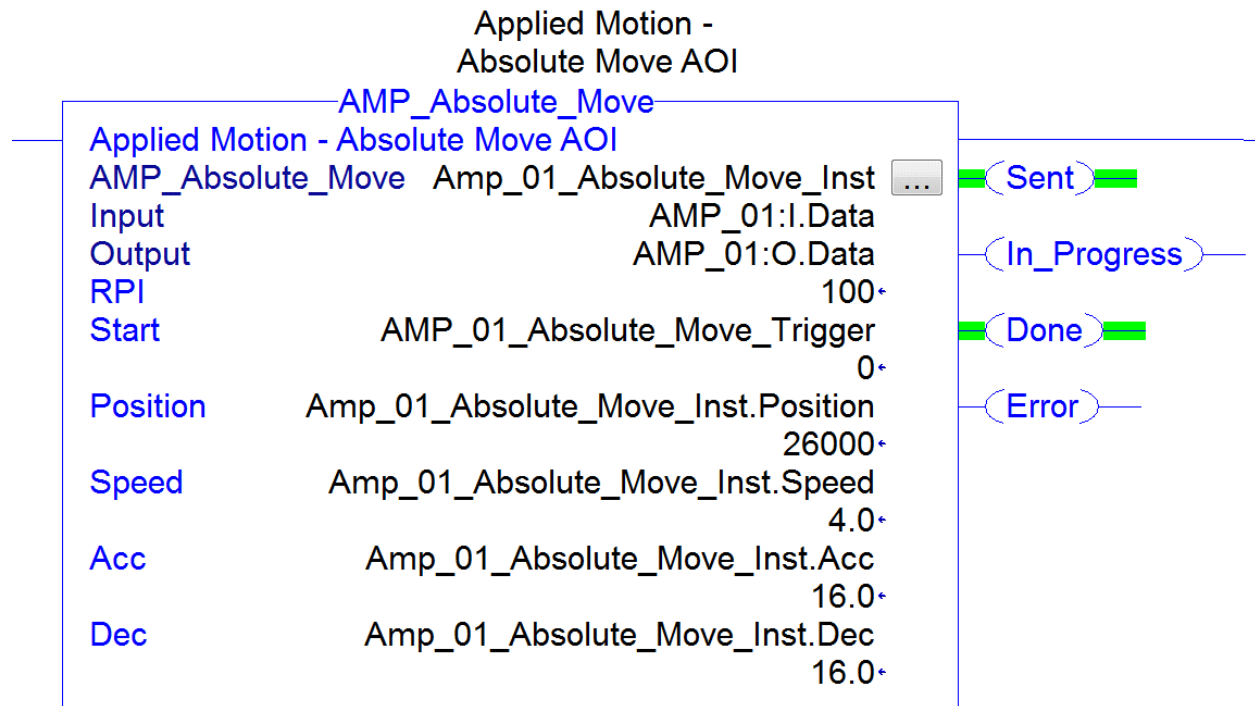
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Disable	Input	BOOL	Disables the drive

Name	Usage	Data Type	Description
Sent	Output	BOOL	Disable has been sent to drive
In_Progress	Output	BOOL	Disable in progress
Done	Output	BOOL	Disable completed without errors – Motor is not Enabled
Error	Output	BOOL	Disable failed to complete (Always false – Future use only)
State	Output	DINT	Internal state of progress

4.6 AMP_Absolute_Move AOI

4.6.1 Overview

The Absolute Move AOI is used to move an axis to a specified absolute position. The equivalent drive command is FP.



4.6.2 Functionality

The Absolute Move uses a typical command sequence with the following specific differences:

- Parameters output to drive: Distance (=Target absolute position), as well as the correctly scaled values of parameters: Speed, Acc and Dec are sent via the output assembly.
- Drive Command: "Feed to Position" (FP) (command word value of 0x10).
- Done output: when drive reaches "In Position"
- Type: Buffered

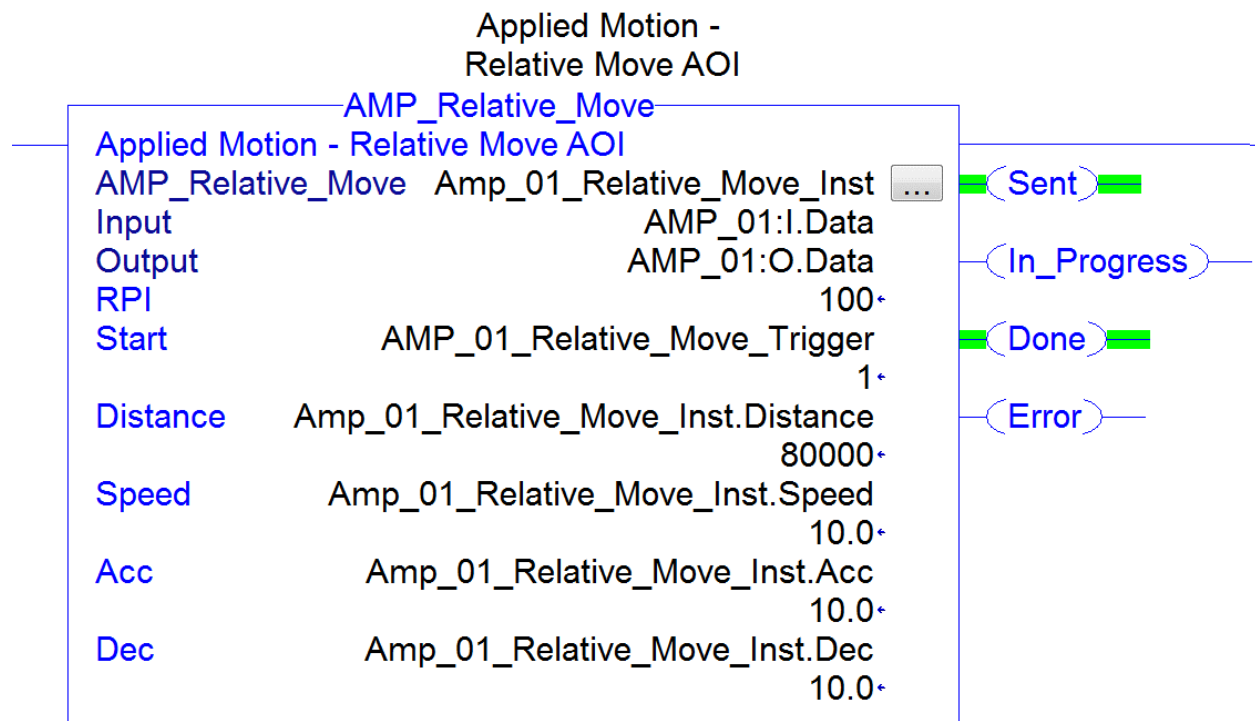
4.6.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Start	Input	BOOL	Start the Absolute Move
Position	Input	DINT	Position (motor/encoder steps, negative = reverse)
Speed	Input	REAL	Velocity of Move (rev/s)
Acc	Input	REAL	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Deceleration (rev/s/s)
Sent	Output	BOOL	Move has been sent to drive
In_Progress	Output	BOOL	Move in progress
Done	Output	BOOL	Move completed without errors - desired position reached
Error	Output	BOOL	Move failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.7 AMP_Relative_Move AOI

4.7.1 Overview

The Relative Move AOI is used to move an axis by a specified distance (positive or negative) relative to its currently commanded position. The equivalent drive command is FL.



4.7.2 Functionality

The Relative Move uses a typical command sequence with the following specific differences:

- Parameters output to drive: relative Distance, as well as the correctly scaled values of parameters: Speed, Acc and Dec are sent via the output assembly.
- Drive Command: "Feed to Length" (FL) (command word value of 0x8).
- Done output: when drive reaches "In Position"
- Type: Buffered

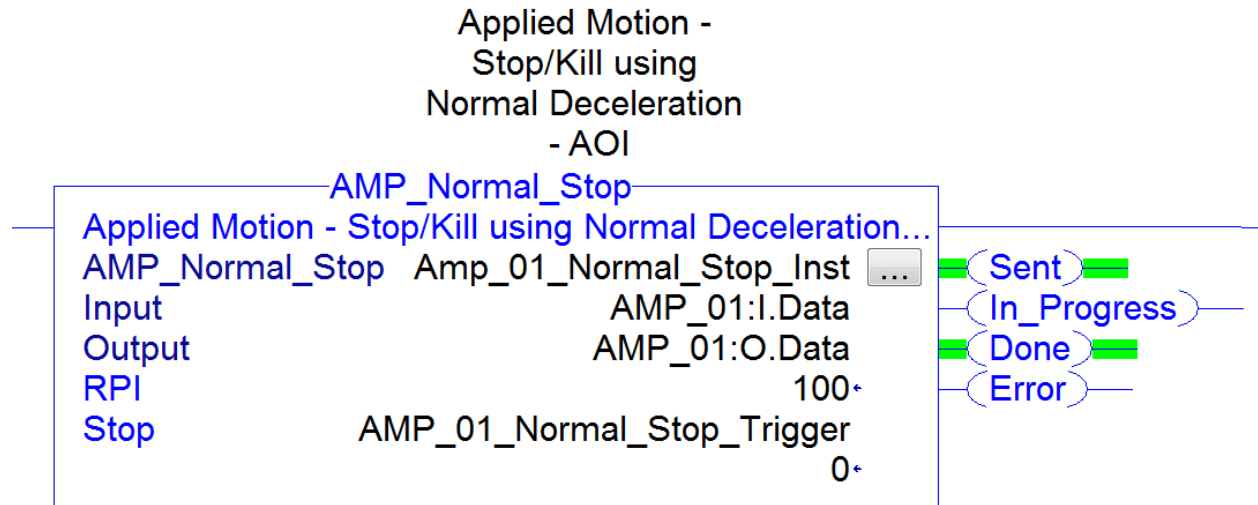
4.7.3 Parameters:

Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Start	Input	BOOL	Start the Relative Move
Distance	Input	DINT	Distance (motor/encoder steps, negative = reverse)
Speed	Input	REAL	Velocity of Move (rev/s)
Acc	Input	REAL	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Deceleration (rev/s/s)
Sent	Output	BOOL	Move has been sent to drive
In_Progress	Output	BOOL	Move in progress
Done	Output	BOOL	Move completed without errors - desired position reached
Error	Output	BOOL	Move failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.8 AMP_Normal_Stop AOI

4.8.1 Overview

The Normal Stop AOI is used to stop and kill the drive using the normal deceleration set by the DE command. The equivalent drive command is SKD.



4.8.2 Functionality

The Normal Stop AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: none.
- Drive Command: "Stop/Kill - DE " (command word value of 0x8000)
- Done output: when drive status indicates it is no longer "Moving"
- Type: Immediate

4.8.3 Parameters

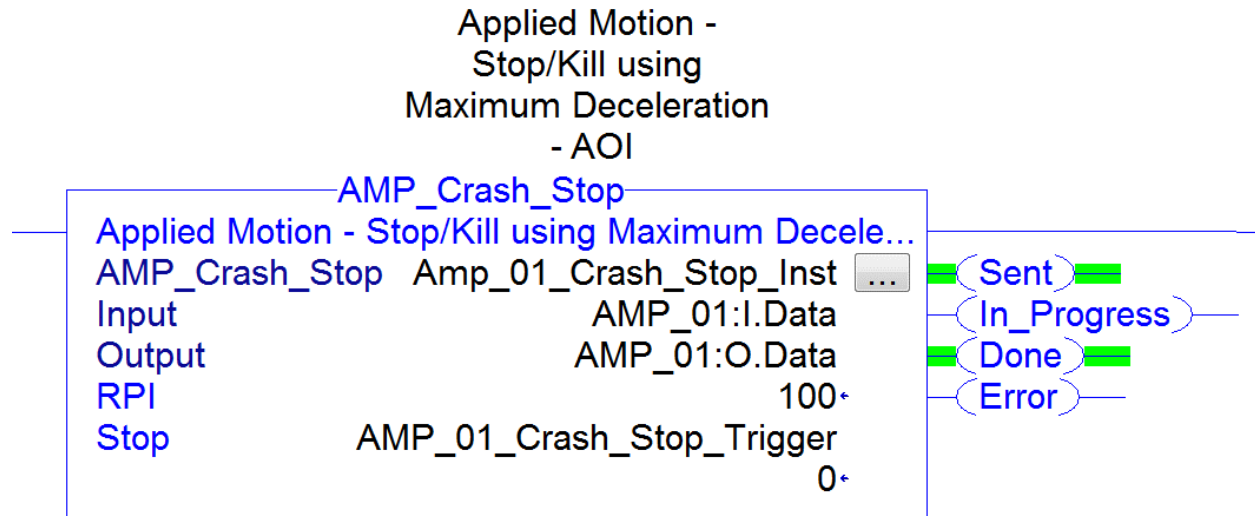
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly

Name	Usage	Data Type	Description
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Stop	Input	BOOL	Stop the Drive (on rising edge)
Sent	Output	BOOL	Stop has been sent to drive
In_Progress	Output	BOOL	Stop in progress
Done	Output	BOOL	Stop completed without errors – Not moving
Error	Output	BOOL	Stop failed to complete (Always false – Future use only)
State	Output	DINT	Internal state of progress

4.9 AMP_Crash_Stop AOI

4.9.1 Overview

The Crash Stop AOI is used to stop and kill the drive using the maximum deceleration as set by the AM command.



4.9.2 Functionality

The Crash Stop AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: none.
- Drive Command: "Stop/Kill - AM" (command word value of 0x4000).
- Done output: when drive status indicates it is no longer "Moving".
- Type: Immediate.

4.9.3 Parameters:

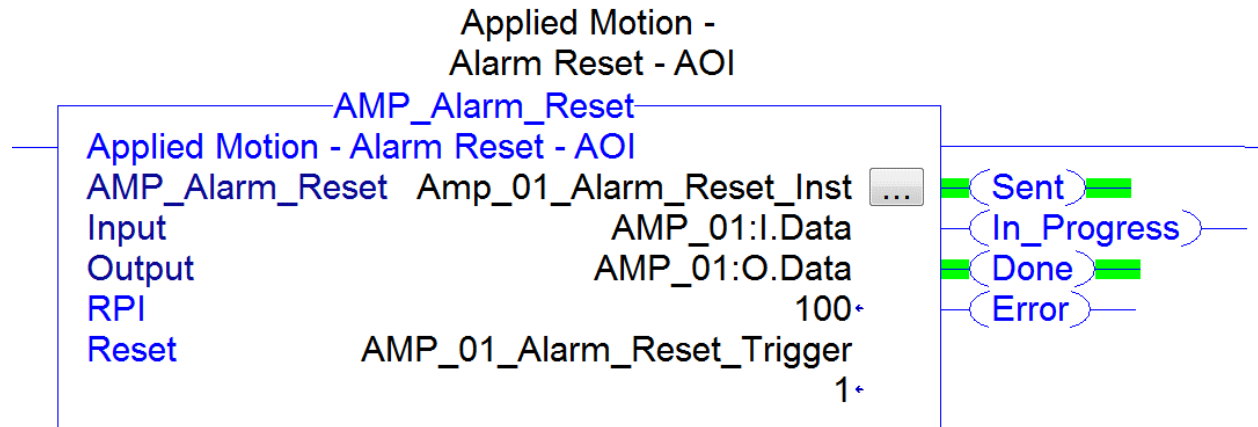
Name	Usage	Data Type	Description
------	-------	-----------	-------------

Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Stop	Input	BOOL	Stop the Drive (on rising edge)
Sent	Output	BOOL	Stop has been sent to drive
In_Progress	Output	BOOL	Stop in progress
Done	Output	BOOL	Stop completed without errors – Not moving
Error	Output	BOOL	Stop failed to complete (Always false – Future use only)
State	Output	DINT	Internal state of progress

4.10 AMP_Alarm_Reset AOI

4.10.1 Overview

The Alarm Reset AOI is used to reset Drive Fault and clear Alarm Code (if possible). The equivalent drive command is AR.



4.10.2 Functionality

The Alarm Reset AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: not changed.
- Drive Command: "Alarm Reset" (AR) (command word value of 0x100000).
- Done output: when drive no longer indicates "Drive Faulted" or "Alarm Present".
- Type: Immediate.

4.10.3 Parameters:

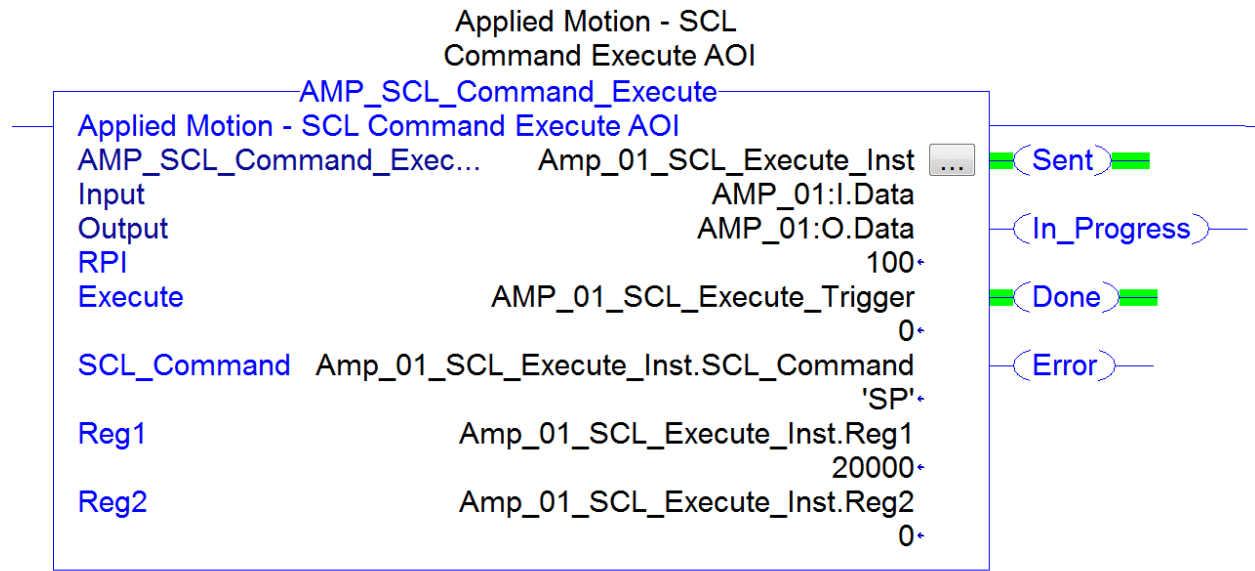
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly

Name	Usage	Data Type	Description
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Reset	Input	BOOL	Reset the Drive (on rising edge)
Sent	Output	BOOL	Reset has been sent to drive
In_Progress	Output	BOOL	Reset in progress
Done	Output	BOOL	Reset completed without errors – no alarm present
Error	Output	BOOL	Reset failed to complete (Always false – Future use only)
State	Output	DINT	Internal state of progress

4.11 AMP_SCL_Command_Execute AOI

4.11.1 Overview

The SCL Command Execute AOI executes selected SCL commands (as listed under “Available SCL Commands” in the Host Command Reference).



4.11.2 Functionality

The SCL Command Execute AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: SCL Command sent as two ASCII characters into lower 16 bits of output assembly element 9. First character is copied into bits 8-15, second character is copied into bits 0-7. SCL Data/Registers Reg1 and Reg2 parameters are copied unchanged to output assembly elements 10 and 11.
- Drive Command: the command word value is set to 0x40000.
- Done output: when command is sent.
- Type: Immediate

4.11.3 Exceptions/Notes

The user can set Reg1 and/or Reg2 parameters to ASCII values when required by enclosing the character value in single quotes.

4.11.4 Parameters:

Name	Usage	Data Type	Style	Description
Input	InOut	DINT[14]	Decimal	Drive Input Assembly
Output	InOut	DINT[16]	Decimal	Drive Output Assembly
RPI	Input	DINT	Decimal	Drive Connection Request Packet Interval (milliseconds)
Execute	Input	BOOL	Decimal	Execute the SCL command
SCL_Command	Input	INT	ASCII	SCL Command Characters (always 2)
Reg1	Input	DINT	Decimal	Data / SCL Register 1*
Reg2	Input	DINT	Decimal	Data / SCL Register 2*
Sent	Output	BOOL	Decimal	Command has been sent to drive
In_Progress	Output	BOOL	Decimal	In progress = sending command
Done	Output	BOOL	Decimal	Command sent
Error	Output	BOOL	Decimal	Command failed to be sent
State	Output	DINT	Decimal	Internal state of progress

* Reg1 and/or Reg2 parameters can be set to ASCII values when required by enclosing the character value in single quotes.

4.12 AMP_Point_to_Point_IO AOI

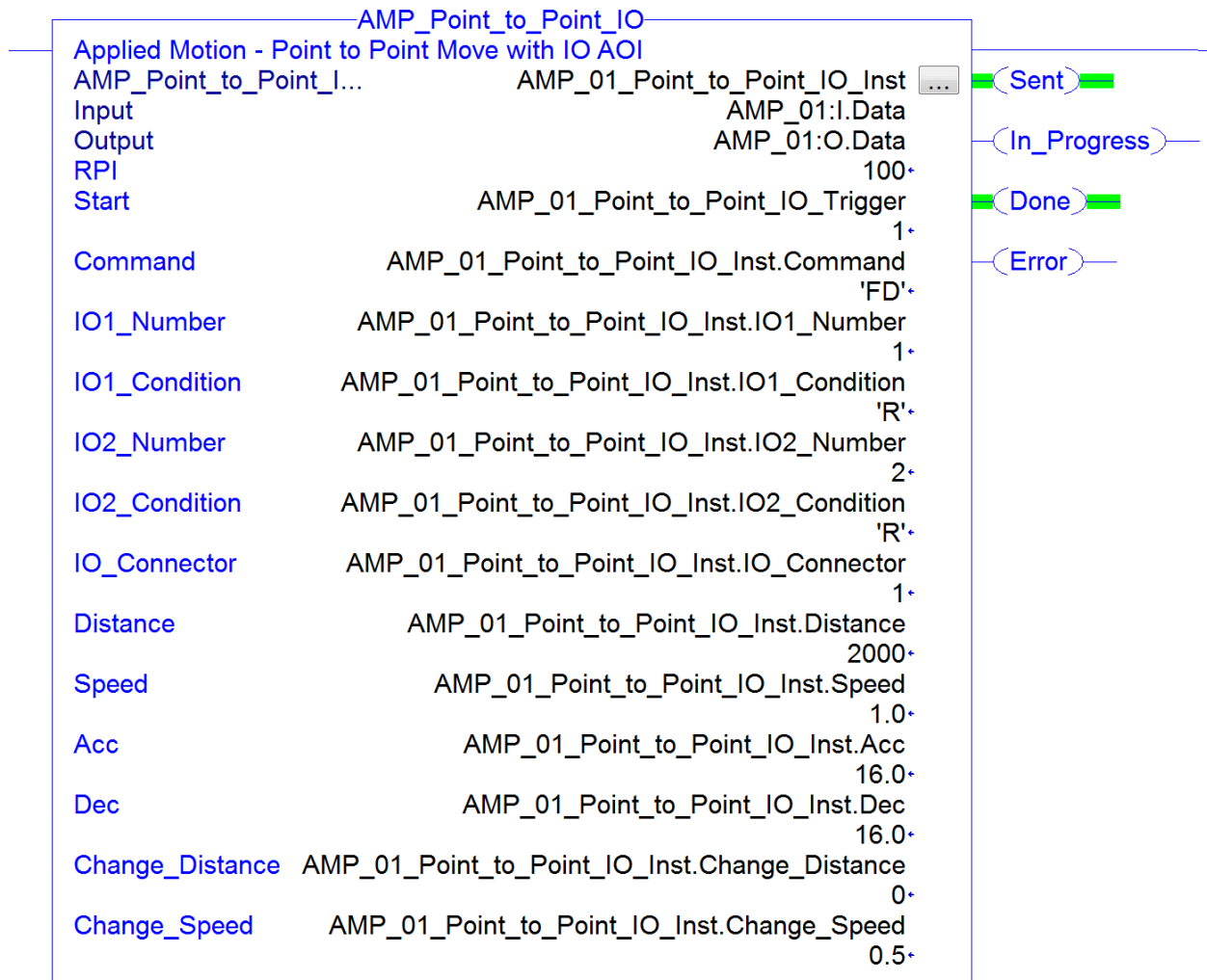
4.12.1 Overview

The Point-to-Point Move with I/O AOI executes moves involving up to 2 Input conditions and a possible output state change. The move variant can be any of the following:

- FS - Feed to Sensor
- FD - Feed to Double Sensor
- FM - Feed to Sensor with Mask Distance
- FY - Feed to Sensor with Safety Distance
- FO - Feed to Length and Set Output
- FE - Follow Encoder
- SH - Seek Home

All seven (7) of the above variants need exactly one I/O point except for FD which requires a second I/O point to be specified in the parameters. The equivalent drive command is a combination of the chosen mnemonic among the 7 variants, along with either DC or VC.

Applied Motion -
Point to Point Move
with IO AOI



4.12.2 Functionality

The Point-to-Point Move with I/O uses a typical command sequence with the following specific differences:

- For the FD (Double Sensor) variation, the Change_Speed parameter is converted as required and sent using the VC SCL command (without requiring the user to perform this externally). This parameter is optional: it can be set to 0 to save execution time. In that case, the preset/previously set value will be used by the drive.
- For the FY, FM and FO variations, the Change_Distance parameter is converted as required and sent using the DC SCL command (without requiring the user to

perform this externally). This parameter is optional: it can be set to 0 to save execution time. In that case, the preset/previously set value will be used by the drive.

- Parameters output to drive: Distance to move past the I/O detection, as well as the correctly scaled values of parameters: Speed, Acc and Dec are sent via the output assembly, Input/Output numbers, connector and conditions (FD requires both I/O points defined).
- Drive Command: depending on value of Command parameter (FS/FD/FY/FM/FO/SH/FE), the command word value is correspondingly set to one of 0x20, 0x40, 0x80, 0x100, 0x200, 0x800 or 0x2000.
- Done output: when drive reaches "In Position"
- Type: Buffered

4.12.3 Parameters

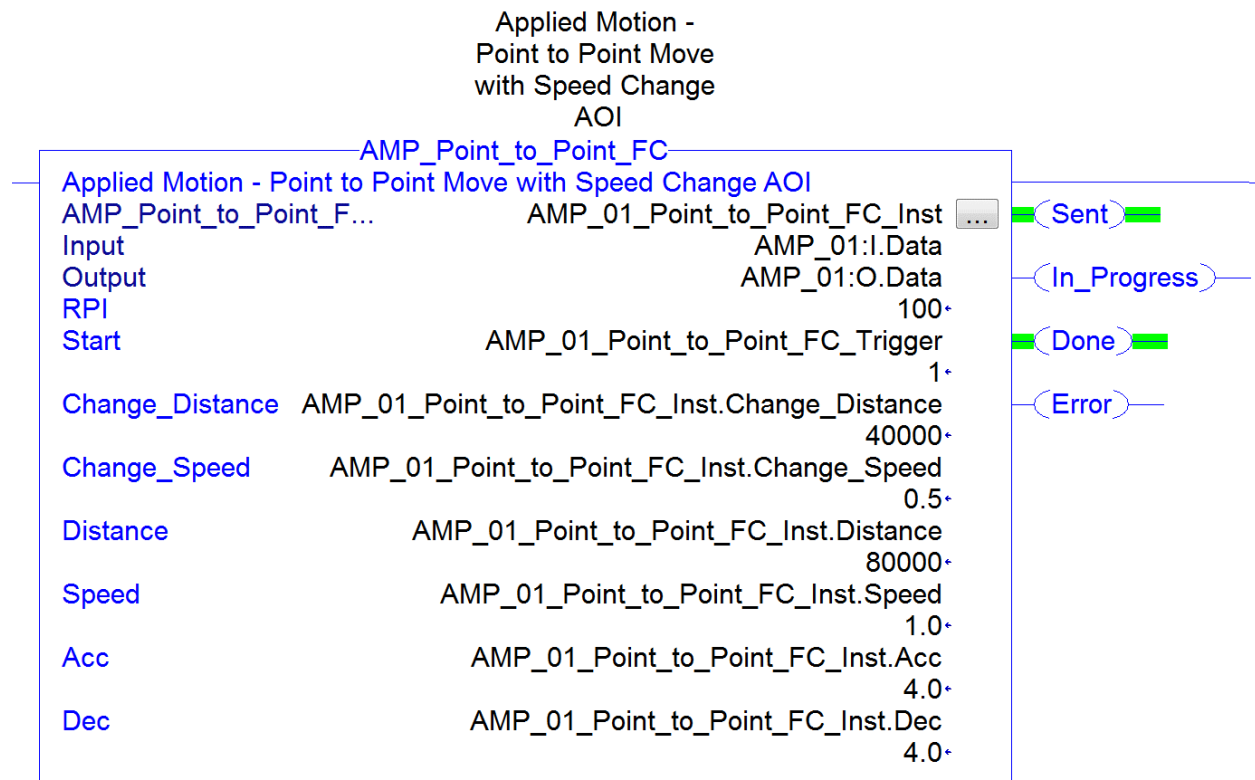
Name	Usage	Data Type	Style	Description
Input	InOut	DINT[14]	Decimal	Drive Input Assembly
Output	InOut	DINT[16]	Decimal	Drive Output Assembly
RPI	Input	DINT	Decimal	Drive Connection Request Packet Interval (milliseconds)
Start	Input	BOOL	Decimal	Start the Move
Command	Input	INT	ASCII	Command variant: FS/FD/FY/FM/FO/SH/FE
IO1_Number	Input	DINT	Decimal	Main Input/Output Number (0 = Encoder index, 1-8 = In/Out 1-8)
IO1_Condition	Input	SINT	ASCII	Main Input/Output Condition/State H/L/R/F (High/Low/Rising Edge/Falling Edge)
IO2_Number	Input	DINT	Decimal	Second Input/Output Number (0 = Encoder index, 1-8 = In/Out 1-8)
IO2_Condition	Input	SINT	ASCII	Second Input/Output Condition/State H/L/R/F (High/Low/Rising Edge/Falling Edge)
IO_Connector	Input	DINT	Decimal	I/O Connector (1=IN/OUT1, 2=IN/OUT2)
Distance	Input	DINT	Decimal	Distance to Move (negative = reverse)
Speed	Input	REAL	Float	Velocity of Move (rev/s)
Acc	Input	REAL	Float	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Float	Deceleration (rev/s/s)

Name	Usage	Data Type	Style	Description
Change_Distance	Input	DINT	Decimal	Change Distance (FY/FM/FO only - 0=Use existing DC set value)
Change_Speed	Input	REAL	Float	Velocity of Move after Change (rev/s – FD only -). 0=Use existing VC set value)
Sent	Output	BOOL	Decimal	Move has been sent to drive
In_Progress	Output	BOOL	Decimal	Move in progress
Done	Output	BOOL	Decimal	Move completed without errors - desired position reached
Error	Output	BOOL	Decimal	Move failed to complete - drive not ready or faulted
State	Output	DINT	Decimal	Internal state of progress

4.13 AMP_Point_to_Point_FC AOI

4.13.1 Overview

The Point-to-Point Move with Speed Change AOI executes a feed to length (relative move) with a speed change after an initial travel distance (FC command). The equivalent drive commands are DC, VC and FC.



4.13.2 Functionality

The Point-to-Point Move with Speed Change uses a typical command sequence with the following specific differences:

- Change_Distance and Change_Speed parameters are converted as required and sent using SCL (without requiring the user to perform this externally). These parameters are optional: they can be set to 0 to save execution time. In that case, the preset/previously set values will be used by the drive.

- Parameters output to drive: Total distance to move, as well as the correctly scaled values of parameters: Speed, Acc and Dec are sent via the output assembly.
- Drive Command: FC, the command word value is set to 0x400.
- Done output: when drive reaches "In Position".
- Type: Buffered

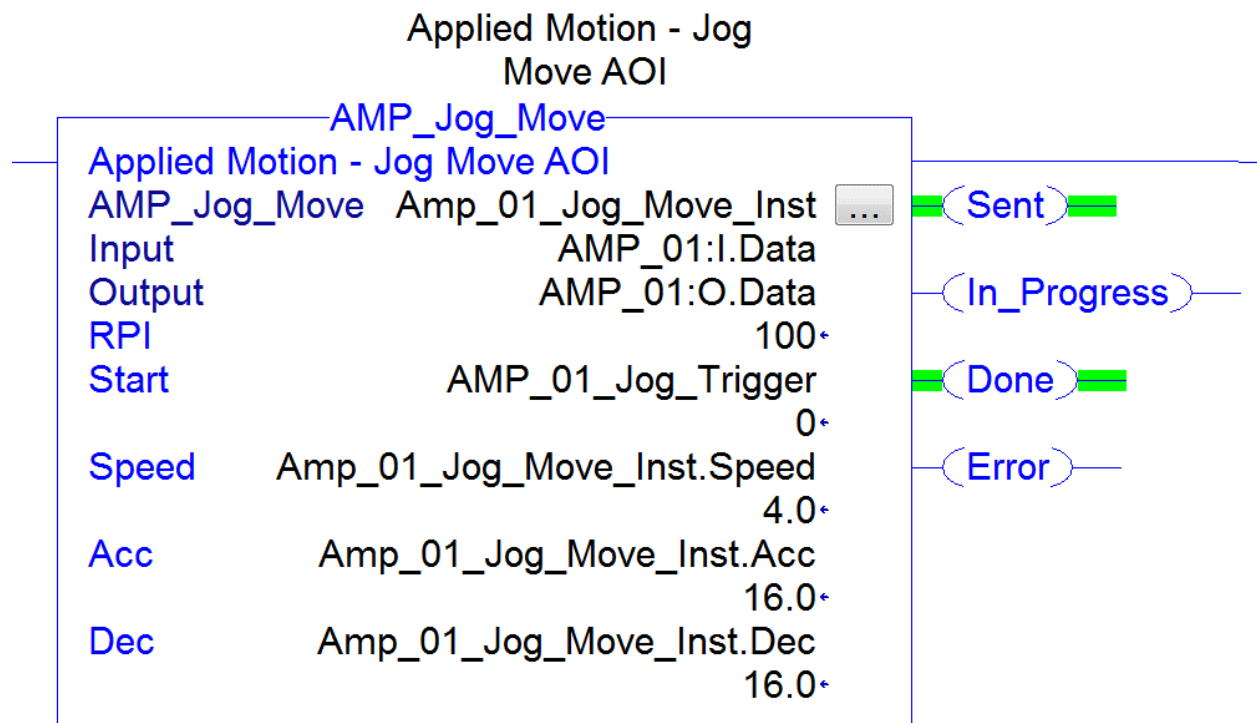
4.13.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Start	Input	BOOL	Start the Move
Change_Distance	Input	DINT	Distance to Move before Speed Change (negative = reverse). (0=Use existing DC set value)
Change_Speed	Input	REAL	Velocity of Move after Change (rev/s). (0=Use existing VC set value)
Distance	Input	DINT	Overall Distance to Move (negative = reverse)
Speed	Input	REAL	Velocity of Move before Change (rev/s)
Acc	Input	REAL	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Deceleration (rev/s/s)
Sent	Output	BOOL	Move has been sent to drive
In_Progress	Output	BOOL	Move in progress
Done	Output	BOOL	Move completed without errors - desired position reached
Error	Output	BOOL	Move failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.14 AMP_Jog_Move AOI

4.14.1 Overview

The Jog Move AOI is used to start jogging an axis at a specified speed and direction (a negative speed set point implies a CCW direction). The equivalent drive commands are JA, JL, JS and CJ.



4.14.2 Functionality

The Jog Move AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: The correctly scaled values of parameters: Speed, Acc and Dec are written to the output assembly elements 1, 2 and 3, respectively.
- Drive Command: "Start Jogging" (command word value of 0x10000).
- Done output: when drive status indicates it is "Moving".

- Type: Buffered

4.14.3 Exceptions/Notes

The Jog Move only uses the rising edge of the Start parameter to start the jogging. To stop the motion, the user will need to use either the AMP_Normal_Stop or the AMP_Crash_Stop AOI. Alternately, you may use AMP_Update_Jog_Speed to set the speed to 0, this will indeed cause the motor to stop, but the drive will still report that it is jogging – this may cause confusion in some cases.

4.14.4 Parameters

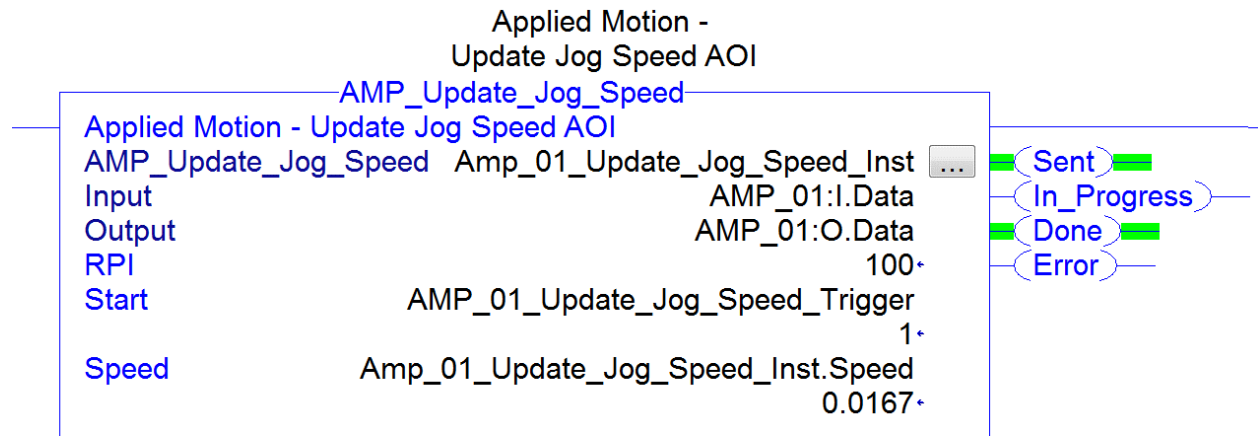
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
Start	Input	BOOL	Start Jogging*
Speed	Input	REAL	Velocity of Move (rev/s) (negative = CCW)
Acc	Input	REAL	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Deceleration (rev/s/s)
Sent	Output	BOOL	Move has been sent to drive
In_Progress	Output	BOOL	Move in progress
Done	Output	BOOL	Move completed without errors - desired position reached
Error	Output	BOOL	Move failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

*The Jog Move AOI uses the rising edge of the Start parameter to start the jogging. To stop the motion, use either the AMP_Normal_Stop AOI or the AMP_Crash_Stop AOI.

4.15 AMP_Update_Jog_Speed AOI

4.15.1 Overview

The Update Jog Speed AOI is used to modify the speed of an ongoing jog motion. The equivalent drive command is CS.



4.15.2 Functionality

The Update Jog Speed AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: The correctly scaled values of parameter “Speed” is written to the output assembly element 1.
- Drive Command: "Update Jog Speed" (command word value of 0x20000).
- Done output: immediately when command is sent.
- Type: Immediate

4.15.3 Parameters

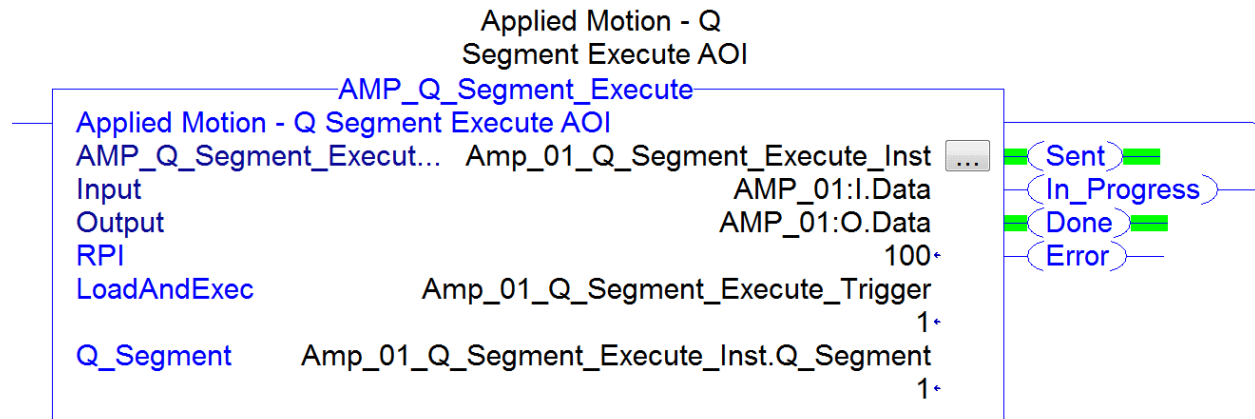
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)

Name	Usage	Data Type	Description
Update	Input	BOOL	Update the Jogging Speed
Speed	Input	REAL	Updated Jog Speed (rev/s), (negative = CCW)
Sent	Output	BOOL	Update has been sent to drive
In_Progress	Output	BOOL	Update in progress
Done	Output	BOOL	Update completed without errors
Error	Output	BOOL	Update failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.16 AMP_Q_Segment_Execute AOI

4.16.1 Overview

The Queue Segment Execute AOI is used to load a specified segment into the queue and execute it. The equivalent drive command is QX.



4.16.2 Functionality

The Queue Segment Execute AOI uses a typical command sequence with the following specific differences:

- Parameters output to drive: The Q_Segment parameters is copied to output assembly element 10.
- Drive Command: "Q Load and Execute (QX)", command word value of 0x80000.
- Done output: immediately when command is sent.
- Type: Buffered

4.16.3 Parameters:

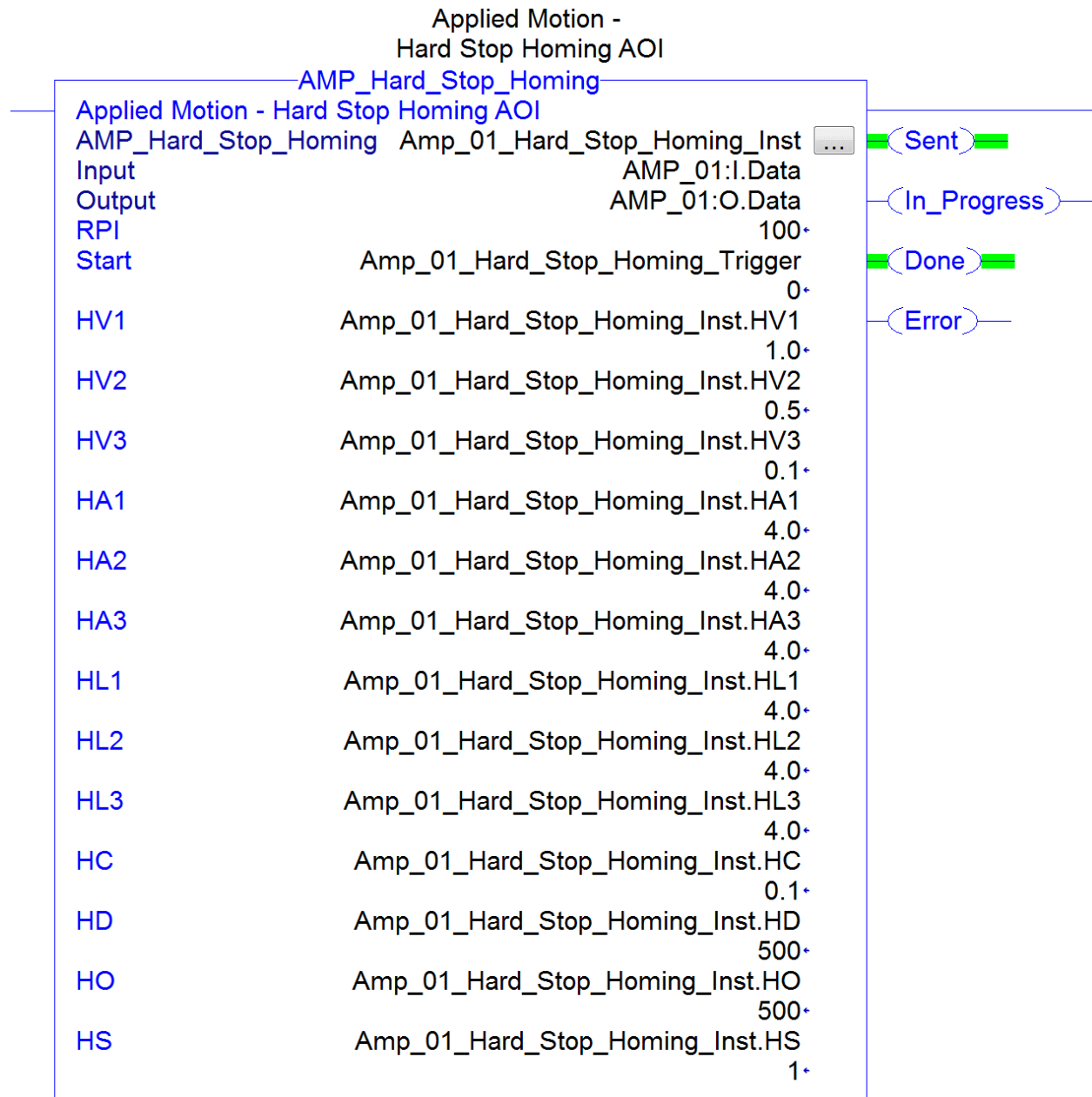
Name	Usage	Data Type	Description
Input	InOut	DINT[14]	Drive Input Assembly
Output	InOut	DINT[16]	Drive Output Assembly

Name	Usage	Data Type	Description
RPI	Input	DINT	Drive Connection Request Packet Interval (milliseconds)
LoadAndExec	Input	BOOL	Load and Execute the provided Q segment
Q_segment	Input	DINT	Q segment number to load and execute
Sent	Output	BOOL	Q Load and Execute has been sent to drive
In_Progress	Output	BOOL	Q Load and Execute in progress
Done	Output	BOOL	Q Load and Execute completed without errors
Error	Output	BOOL	Q Load and Execute failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.17 AMP_Hard_Stop_Homing AOI

4.17.1 Overview

The Hard Stop Homing AOI accepts all the required parameters for SCL commands HV1, HV2, HV3, HA1, HA2, HA3, HL1, HL2, HL3, HC, HD, HO and then sends the HS command to initiate the Hard Stop Homing move.



4.17.2 Functionality

The Hard Stop Homing AOI uses a typical command sequence with the following specific differences:

- The command sequence is repeated for each of HV1, HV2, HV3, HA1, HA2, HA3, HL1, HL2, HL3, HC, HD, HO, HS in order to set parameters and finally start the homing.
- The SCL Command letters (output element 9) are set for the corresponding parameter
- The Data/SCL Register 1 (output element 10) is set to the ASCII values of '1', '2' or '3' for commands that require this, otherwise it is set to the correctly scaled value supplied as the corresponding parameter.
- The Data/SCL Register 2 (output element 11) is set as required to the correctly scaled value of the corresponding parameter.
- Drive Command: "Send Host Command", i.e. the command word value is set to 0x40000. (The value will return to 0=idle between successive SCL command execution cycles).
- Done output: when drive reaches "In Position".
- Sent output: when the HS command has been sent
- Type: Buffered

4.17.3 Parameters

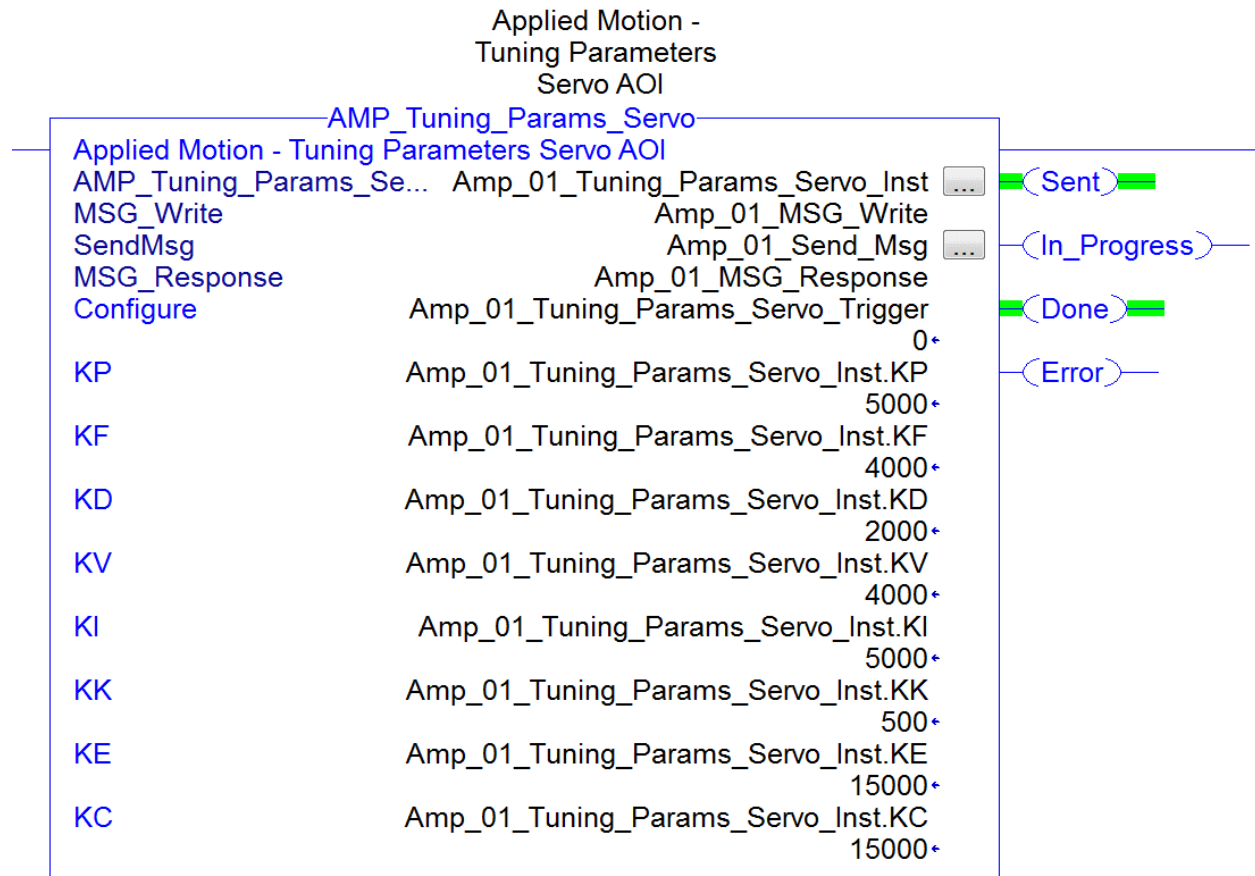
Name	Usage	Data Type	Description
MSG_Write	InOut	SINT[8]	Message Write data tag
SendMsg	InOut	MESSAGE	Message tag
MSG_Response	InOut	SINT[8]	Message Response data tag
Configure	Input	BOOL	Start the Configuration
VP	Input	INT	Velocity Mode Proportional Constant
VI	Input	INT	Velocity Integrator Constant
KK	Input	INT	Inertia Feedforward Constant
KC	Input	INT	Overall Servo Filter
KP	Input	INT	Proportional Constant
KD	Input	INT	Differential Constant
KE	Input	INT	Differential Filter

Name	Usage	Data Type	Description
Sent	Output	BOOL	All parameter settings have been sent to drive
In_Progress	Output	BOOL	The setting of parameters is in progress
Done	Output	BOOL	Parameters set without errors
Error	Output	BOOL	Configuration failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.18 AMP_Tuning_Params_Servo AOI

4.18.1 Overview

The Tuning Parameters for Servo AOI accepts all the required parameters for SCL commands KP, KF, KD, KV, KI, KK, KE and KC, and then scales and issues these to the drive, one at a time.



4.18.2 Functionality

The Tuning Parameters for Servo AOI uses a typical command sequence with the following specific differences:

- For each of these settings: KP, KF, KD, KV, KI, KK, KE and KC, the following is repeated:

- An explicit Message Type 2 Command is executed with Opcode 83 (Parameter Write) with the correctly chosen parameter index as the operand and the correctly scaled setting value supplied as the data.
- The AOI waits for a valid response before continuing.
- A communication error (“MSG.ER”) results in the Error output being set and this command sequence being aborted
- Done output: when all parameters have been sent.
- Sent output: when all parameters have been sent.
- Type: Buffered

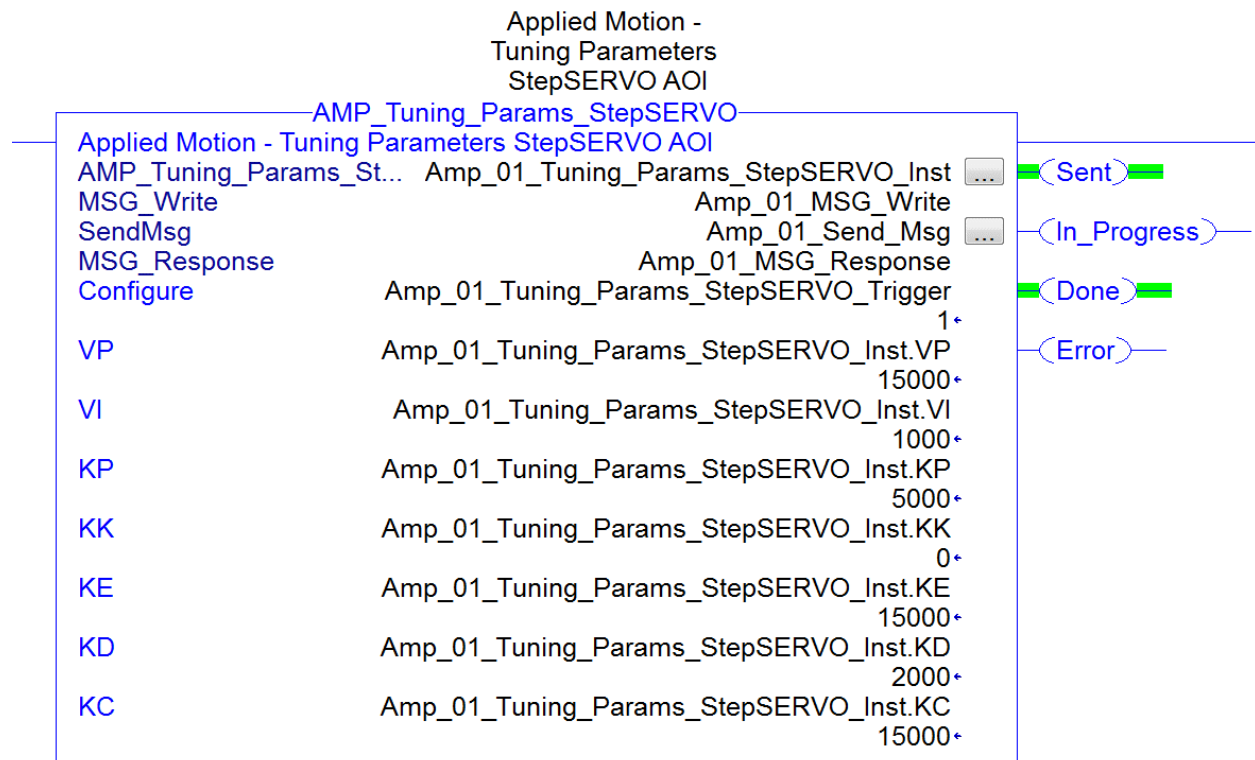
4.18.3 Parameters:

Name	Usage	Data Type	Description
MSG_Write	InOut	SINT[8]	MSG_Write
SendMsg	InOut	MESSAGE	SendMsg
MSG_Response	InOut	SINT[8]	MSG_Response
Configure	Input	BOOL	Start the Configuration
KP	Input	INT	Proportional Constant
KF	Input	INT	Velocity Feedforward Constant
KD	Input	INT	Differential Constant
KV	Input	INT	Velocity Feedback Constant
KI	Input	INT	Integrator Constant
KK	Input	INT	Inertia Feedforward Constant
KE	Input	INT	Differential Filter
KC	Input	INT	Overall Servo Filter
Sent	Output	BOOL	All parameter settings have been sent to drive
In_Progress	Output	BOOL	The setting of parameters is in progress
Done	Output	BOOL	Parameters set without errors
Error	Output	BOOL	Configuration failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.19 AMP_Tuning_Params_StepSERVO AOI

4.19.1 Overview

The Tuning Parameters for StepSERVO AOI accepts all the required parameters for SCL commands VP, VI, KK, KC, KP, KD and KE, and then scales and issues these to the drive, one at a time.



4.19.2 Functionality

The Tuning Parameters for StepSERVO AOI uses a typical command sequence with the following specific differences:

- For each of these settings: VP, VI, KK, KC, KP, KD and KE, the following is repeated:
- An explicit Message Type 2 Command is executed with Opcode 83 (Parameter Write) with the correctly chosen parameter index as the operand and the correctly scaled setting value supplied as the data.
- The AOI waits for a valid response before continuing.

- A communication error (“MSG.ER”) results in the Error output being set and this command sequence being aborted
- Done output: when all parameters have been sent.
- Sent output: when all parameters have been sent.
- Type: Buffered

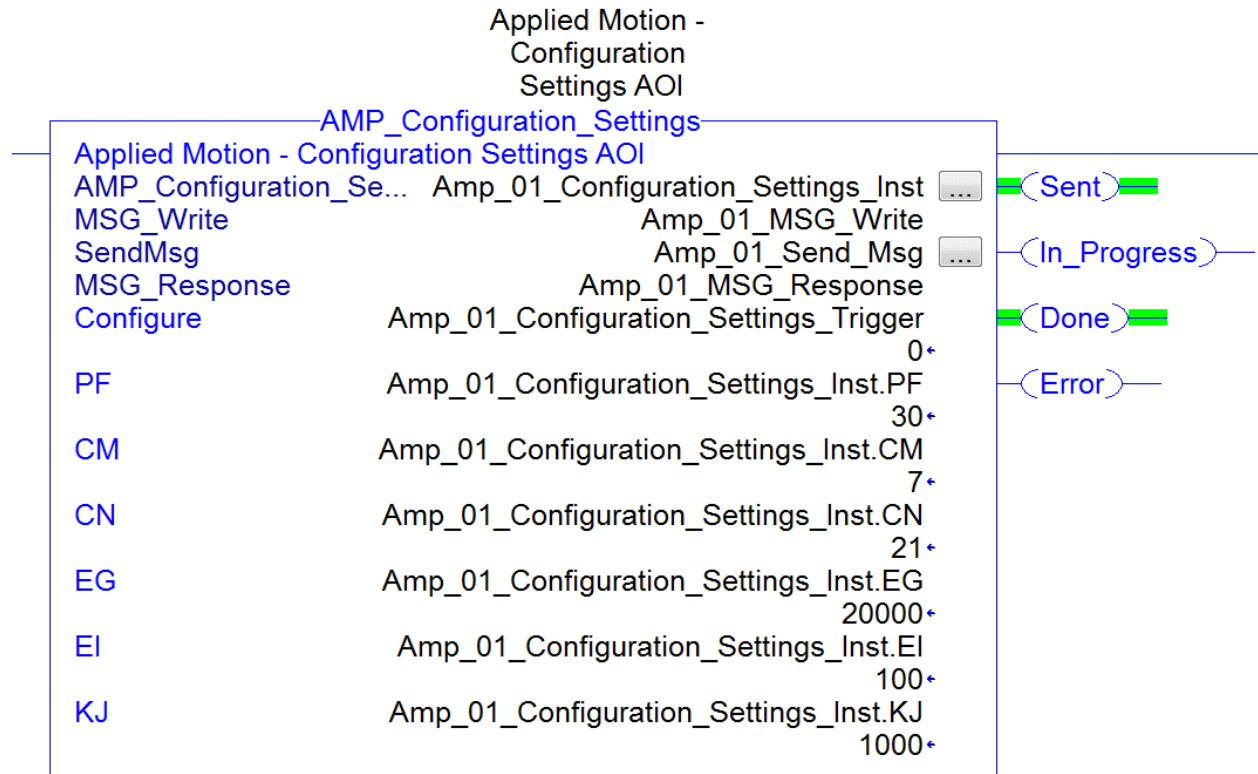
4.19.3 Parameters:

Name	Usage	Data Type	Description
MSG_Write	InOut	SINT[8]	Message Write data tag
SendMsg	InOut	MESSAGE	Message tag
MSG_Response	InOut	SINT[8]	Message Response data tag
Configure	Input	BOOL	Start the Configuration
VP	Input	INT	Velocity Mode Proportional Constant
VI	Input	INT	Velocity Integrator Constant
KK	Input	INT	Inertia Feedforward Constant
KC	Input	INT	Overall Servo Filter
KP	Input	INT	Proportional Constant
KD	Input	INT	Differential Constant
KE	Input	INT	Differential Filter
Sent	Output	BOOL	All parameter settings have been sent to drive
In_Progress	Output	BOOL	The setting of parameters is in progress
Done	Output	BOOL	Parameters set without errors
Error	Output	BOOL	Configuration failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

4.20 AMP_Configuration_Settings AOI

4.20.1 Overview

The Configuration Settings AOI accepts all the required parameters for: Position Fault (PF), Control Mode (CM), Secondary Control Mode (CN), Electronic Gearing (EG), Input Noise Filter (EI), Jerk Filter (KJ), and then scales and issues these to the drive, one at a time.



4.20.2 Functionality

The Configuration Settings AOI uses the following command sequence:

- For each of these settings: Position Fault (PF), Control Mode (CM), Secondary Control Mode (CN), Electronic Gearing (EG), Input Noise Filter (EI), Jerk Filter (KJ), the following is repeated:
- An explicit Message Type 2 Command is executed with Opcode 83 (Parameter Write) with the correctly chosen parameter index as the operand and the correctly scaled setting value supplied as the data.
- The AOI waits for a valid response before continuing.

- A communication error (“MSG.ER”) results in the Error output being set and this command sequence being aborted
- Done output: when all settings have been sent.
- Sent output: when all settings configuration commands have been sent.
- Type: Immediate

4.20.3 Parameters

Name	Usage	Data Type	Description
MSG_Write	InOut	SINT[8]	Message Write data tag
SendMsg	InOut	MESSAGE	Message tag
MSG_Response	InOut	SINT[8]	Message Response data tag
Configure	Input	BOOL	Configures the Drive (on rising edge)
PF	Input	INT	Position Fault (steps or counts)
CM	Input	INT	Control Mode (integer code)
CN	Input	INT	Secondary Control Mode (integer code)
EG	Input	INT	Electronic Gearing (steps/rev or counts/rev)
EI	Input	INT	Input Noise Filter Constant (0-255)
KJ	Input	INT	Jerk Filter (Hz)
Sent	Output	BOOL	Configuration has been sent to drive
In_Progress	Output	BOOL	Configuration in progress
Done	Output	BOOL	Configuration completed without errors – no alarm present
Error	Output	BOOL	Configuration failed to complete
State	Output	DINT	Internal state of progress