

AMCI
Integrated
Motion
Devices with
CIP Sync



AMCI Integrated Motion Devices with CIP Sync Owner's Manual

[Home](#) » [AMCI](#) » AMCI Integrated Motion Devices with CIP Sync Owner's Manual 

Contents

- [1 AMCI Integrated Motion Devices with CIP Sync](#)
- [2 Specifications](#)
- [3 Step 1: Enabling Time Synchronization](#)
- [4 Step 2: Scanner Module Configuration](#)
- [5 Step 3: Create an Event Driven Task](#)
- [6 Step 4: Add GSV Instruction](#)
- [7 FAQs](#)
- [8 Class Name: TimeSynchronize](#)
- [9 Documents / Resources](#)
 - [9.1 References](#)



AMCI Integrated Motion Devices with CIP Sync



Specifications

- Model Numbers: SD4840E2, SD17060E2, SD31045E2, SMD17E2, SMD23E2, SMD24E2, SMD34E2, SV160E2, SV400E2
- Network Firmware Version: 1.37 and later
- Feature: CIP Sync for synchronized motion

Step 1: Enabling Time Synchronization

To use CIP Sync, enable Time Synchronization in the host controller:

1. Select the Date/Time tab.
2. Check the Enable Time Synchronization field.

Step 2: Scanner Module Configuration

If using a separate Ethernet scanner module:

1. On the General Tab, click Change.
2. Select Time Sync and Motion from the Time Sync Connection dropdown.
3. Click OK and Apply.

Step 3: Create an Event Driven Task

To closely follow the master axis:

1. Right-click on the desired task and select Properties.
2. Set Type to Event and Trigger to Motion Group Execution.
3. Specify the motion axis in the Tag field.

Step 4: Add GSV Instruction

Add a GSV instruction to your logic:

- Class Name: TimeSynchronize
- Attribute Name: CurrentTimeNanoseconds
- Destination: Array of two DINT registers

FAQs

Using AMCI Motion Add-On Instructions with CIP Sync.

Devices

Beginning in September 2024, with network firmware version 1.37, AMC I's networked integrated motor drivers now include the ability to use CIP Sync to more closely and quietly follow a master axis. This document describes how to use the AMCI Motion Axis Add-On Instructions with CIP Sync to follow a master axis. These Add Instructions will work with the following AMCI-integrated motion devices, including,

- SD4840E2
- SD17060E2
- SD31045E2
- SMD17E2
- SMD23E2
- SMD24E2
- SMD34E2
- SV160E2
- SV400E2

step 1: Enabling the host controller for Time Synchronization

The first step in using CIP Sync is to Enable Time Synchronization in the host controller. Depending on the system being used, this may be located in the controller properties, the network properties, or the scanner module's properties. As shown in the following image, select the Date/Time tab and then place a checkmark in the Enable Time Synchronization field.

Controller Properties -

Nonvolatile Memory Capacity Internet Protocol Port Configuration Network Security Alarm Log
General Major Faults Minor Faults **Date/Time** Advanced SFC Execution Project

i The Date and Time displayed here is Controller local time, not workstation local time.
Use these fields to configure Time attributes of the Controller.

Set Date, Time and Zone from Workstation

Date and Time: 6/10/1998 7:22:36 PM Change Date and Time...

Time Zone: (UTC+00:00)

☐ Adjust for Daylight Saving (+01:00)

Time Synchronize

☒ Enable Time Synchronization

☒ Is the system time master
☐ Is a synchronized time slave
☐ Duplicate CST master detected
☐ CST Mastership disabled
☐ No CST master

! DANGER. If time synchronization is disabled online, active axes in any controller in this chassis, or any other synchronized device, may experience unexpected motion.

Advanced...

Step 2: Scanner Module Configuration

Perform this step only if a separate Ethernet scanner module, as opposed to a built-in Ethernet Port, is being used. On the General Tab of the Ethernet scanner module, click on the Change button.

Module Properties: Local:2 (1756-EN2TR 10.001)

General Connection RSNetWorx Module Info Internet Protocol Port Configuration Network Time Sync

Type: 1756-EN2TR 1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair Media Change Type...

Vendor: Rockwell Automation/Allen-Bradley

Parent: Local

Name: EN2TR

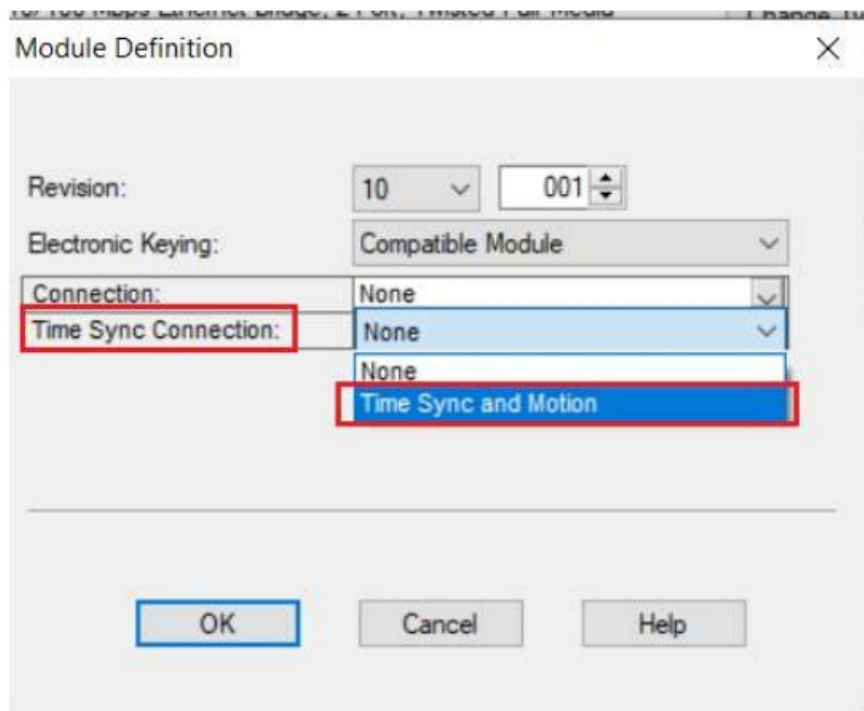
Description:

Ethernet Address
☐ Private Network: 192.168.1.
☒ IP Address: 192 . 168 . 0 . 5
☐ Host Name:

Slot: 2

Module Definition
 Revision: 10.001 Change ...
 Electronic Keying: Compatible Module
 Connection: None
 Time Sync Connection: None

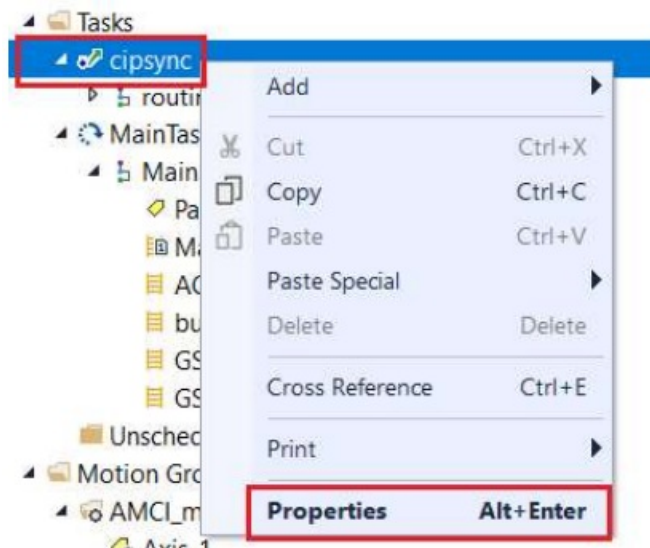
Click on the down arrow next to Time Sync Connection and select Time Sync and Motion. Click on the OK button to accept this change.



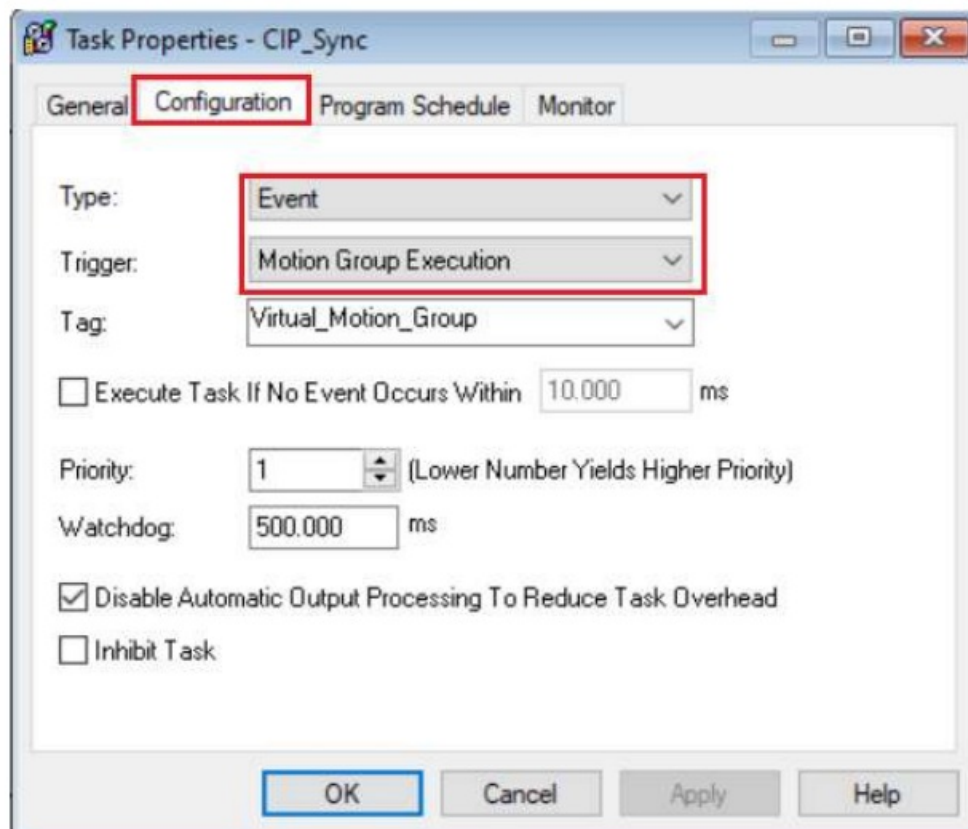
Back on the General tab, click on Apply.

Step 3: Create a Task

While not required, the AMCI motion device will more closely follow the master axis if the supporting logic, including the AMCI AOIs, is in an Event Driven Task, where the trigger for the task is Motion Group Execution. To create an Event Driven Task, right-click on the desired task in the project tree and select Properties.



Click on the Configuration tab and select the Type to be "Event" and the Trigger to be "Motion Group Execution". The Tag field will be the motion axis that is being followed.



Step 4: Add the GSV (Get System Value) instruction The final step in using the AMCI motion devices with CIP Sync is to add a GSV instruction to your logic. As shown in the following image.

Class Name: TimeSynchronize

Attribute Name: CurrentTimeNanoseconds

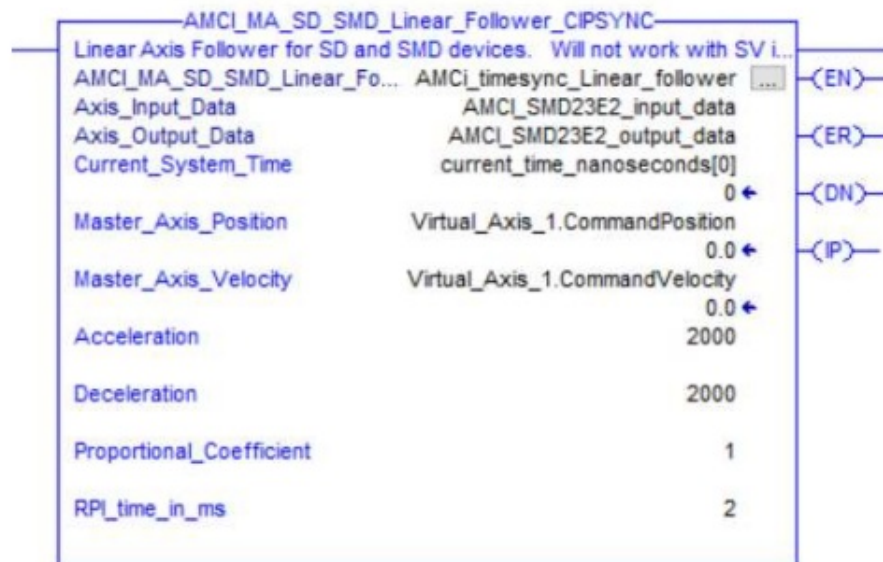
Destination: An array consisting of two DINT registers The GSV instruction should be located in an unconditional rung in the Event Driven Task created in Step 2 above.

GSV	
Class Name	TimeSynchronize
Instance Name	
Attribute Name	CurrentTimeNanoseconds
Dest	current_time_nanoseconds[0]
	0

AMCI_MA_SD_SMD_Linear Follower with CIP SYNC

AMCI_MA_SD_SMD_Circular_Follower with CIP SYNC

These AOIs will only work with the SD and SMD motion devices. The SV motion devices have their follower AOIs.



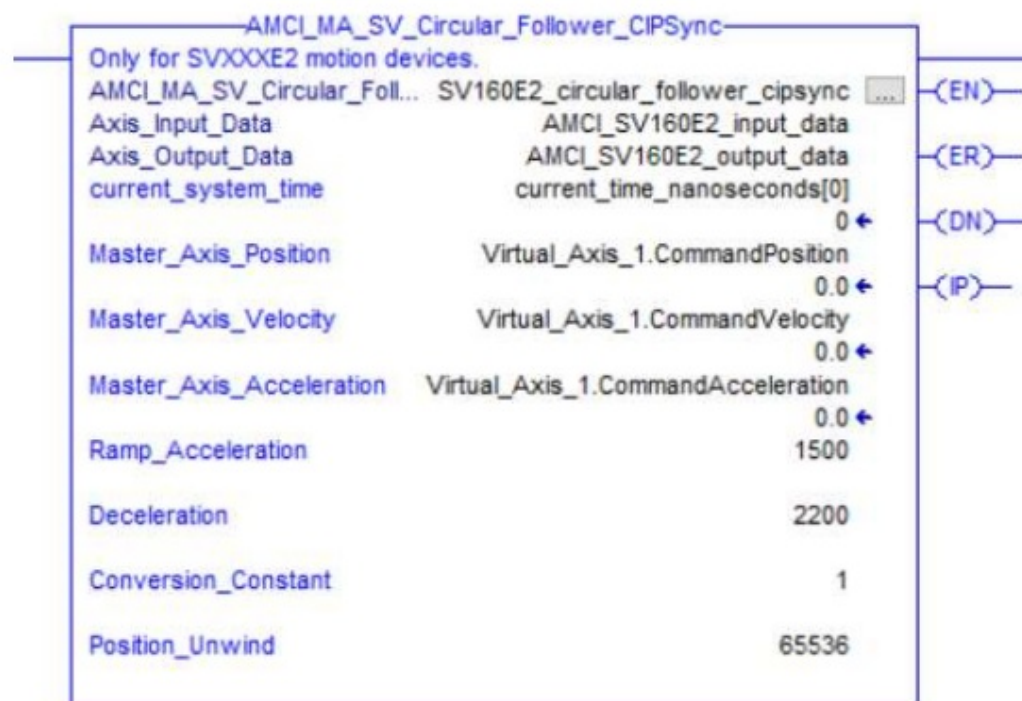
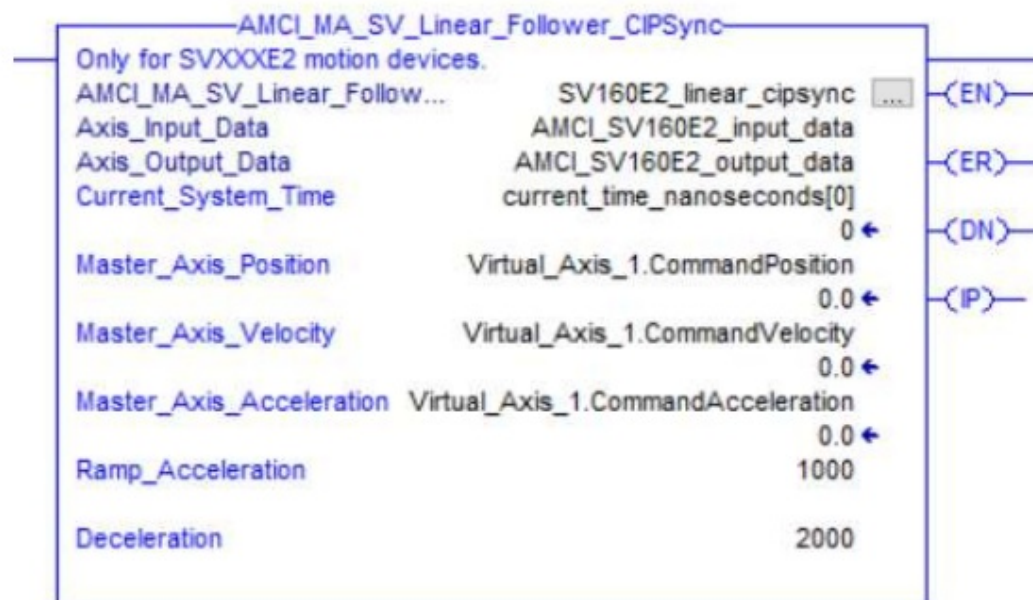
Parameter	
Axis_Input_Data	Input data from AMCI motion device. Uses the AMCI_Motion_Axis_Input_Data User Defined Data Type.
Axis_Output_Data	Output data from the AOI to the AMCI motion device. Uses the AMCI_Motion_Axis_Output_Data User Defined Data Type.
Current_System_Time	The first word of the two DINT word arrays of the current time in nanoseconds is read by a GSV instruction.
Master_Axis_Position	REAL DATA TYPE position directly from the motion axis.
Master_Axis_Velocity	REAL DATA TYPE velocity directly from the motion axis.
Acceleration & Deceleration	An actual value or an INT DATA TYPE. Larger acceleration and deceleration values will cause the motion device to more quickly react changes in the source position and velocity values.
Proportional_Coefficient	A value of 1 or 2 is recommended.
Conversion_Constant (Circular Follower Only)	<p>The data from a circular motion axis has units of revolutions and revolutions / second. However, the AMCI Motion Device requires that the position and velocity have units of counts and counts/sec. The AOI performs this conversion by multiplying both the position and velocity from the motion axis by the Conversion Constant parameter before sending them to the AMCI Motion Controller.</p> <p>This field is typically but does not have to be, set to the master axis' Conversion Constant. The Conversion Constant can be a fractional number. A negative Conversion Constant will cause the motor to turn in the opposite direction from the master axis.</p>
Position_Unwind (Circular Follower Only)	<p>Must be set to the Unwind Value of the motion axis and defines the point at which the position data will transition from its maximum to its minimum value. The Position Unwind Value MUST BE IN THE RANGE OF 21 TO 65535.</p>
RPI_Time_in_ms	<p>The RPI time used when the AMCI Motion Device was added to the network. Used by the AOI to control how long the Preset Command is sent to the Motion Device before motion begins. A value of zero will cause the AOI to assume that the default RPI of 8ms is being used.</p>

Enumerations	Set When.....	Reset When
EN (Enable)	Rung is true	Rung is false
DN (Done)	The command is sent to the motion device	Rung is false
ER (Error)	There is an Input, Command, or Configuration Error	Rung is false
IP (In Process)	The follower command is active, even if the master motion axis position and velocity are not changing.	Rung goes false

AMCI_MA_SV_Linear Follower with CIP SYNC

AMCI_MA_SV_Circular_Follower with CIP SYNC

These AOIs will only work with the SVXXE2 motion devices. The SD and SMD motion devices have their follower AOIs.



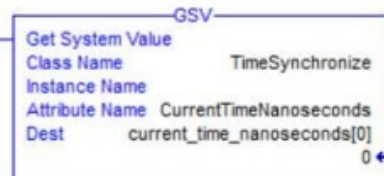
Parameter	
Axis_Input_Data	Input data from AMCI motion device. Uses the AMCI_Motion_Axis_Input_Data User Defined Data Type.
Axis_Output_Data	Output data from the AOI to the AMCI motion device. Uses the AMCI_Motion_Axis_Output_Data User Defined Data Type.
Current_System_Time	The first word of the two DINT word arrays of the current time in nanoseconds is read by a GSV instruction.
Master_Axis_Position	REAL DATA TYPE position directly from the motion axis.
Master_Axis_Velocity	REAL DATA TYPE velocity directly from the motion axis.
Master_Axis_Acceleration	REAL DATA TYPE acceleration directly from the motion axis.
Ramp Acceleration	An actual value or INT value is used to transition from no motion to motion. Once motion is occurring, the follower acceleration will be used. Range of 0 to 15,999.
Deceleration	An actual value or INT value is used to transition from motion to no motion. Range of 0 to 15,999.
Conversion_Constant (Circular Follower Only)	<p>An actual value or a REAL data type register. The value in this field is multiplied by the Position, Velocity, and Acceleration from the master axis before being sent to the servo and scales the supplied data to the servo motor counts per turn. This field is typically, but does not have to be, set to the master axis'</p> <p>Conversion Constant.</p>
Position_Unwind (Circular Follower Only)	An actual value or a DINT TYPE register. This parameter defines the point at which the position data will transition from its maximum to its minimum value.

Parameter	
Axis_Input_Data	Input data from AMCI motion device. Uses the AMCI_Motion_Axis_Input_Data User Defined Data Type.
Axis_Output_Data	Output data from the AOI to the AMCI motion device. Uses the AMCI_Motion_Axis_Output_Data User Defined Data Type.
Current_System_Time	The first word of the two DINT word arrays of the current time in nanoseconds is read by a GSV instruction.
Master_Axis_Position	REAL DATA TYPE position directly from the motion axis.
Master_Axis_Velocity	REAL DATA TYPE velocity directly from the motion axis.
Master_Axis_Acceleration	REAL DATA TYPE acceleration directly from the motion axis.
Ramp Acceleration	An actual value or INT value is used to transition from no motion to motion. Once motion is occurring, the follower acceleration will be used. Range of 0 to 15,999.
Deceleration	An actual value or INT value is used to transition from motion to no motion. Range of 0 to 15,999.
Conversion_Constant (Circular Follower Only)	An actual value or a REAL data type register. The value in this field is multiplied by the Position, Velocity, and Acceleration from the master axis before being sent to the servo and scales the supplied data to the servo motor counts per turn. This field is typically, but does not have to be, set to the master axis' Conversion Constant.
Position_Unwind (Circular Follower Only)	An actual value or a DINT TYPE register. This parameter defines the point at which the position data will transition from its maximum to its minimum value.

The following logic shows all of the elements required to use an AMCI motion device in a follower system with CIP Sync.

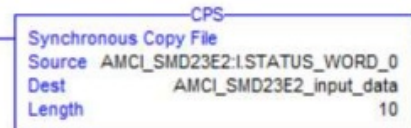
The following rung uses a GSV (Get System Value) instruction to read the system time, in nanoseconds, from the controller. The destination address must be two DINT registers.

This rung MUST NOT have any input conditions.



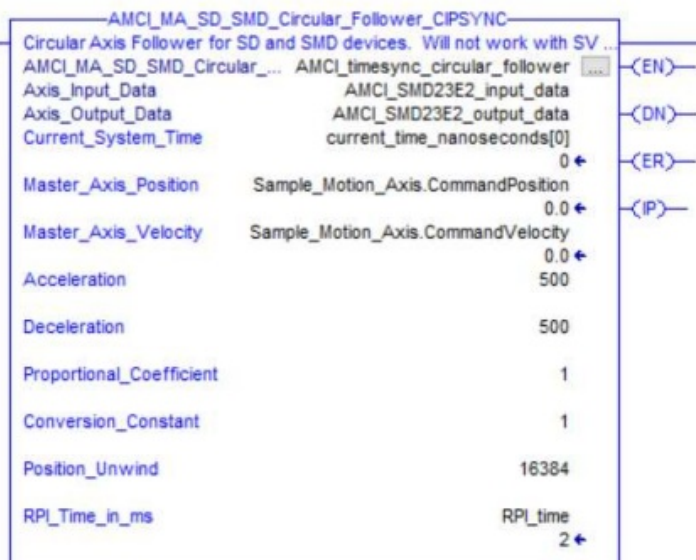
At the top of your program, BEFORE ALL OF THE ADD ON INSTRUCTIONS, use a CPS instruction to copy the input data from the AMCI motion device to a tag array that was created using the AMCI_Motion_Axis_Input_Data User Defined Data Type.

The input data in this tag array is made up of named tags and will also be used as the buffered data in your program. It is this buffered data that must be used in place of the input data directly from the AMCI motion device.



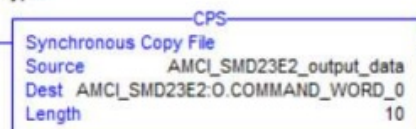
Circular Axis
Follower

Run_AMCI_SMD23_24E2.7



At the bottom of your program, after all of the Add On Instructions, use a CPS instruction to copy the data from the AOs to the output registers of the AMCI motion device.

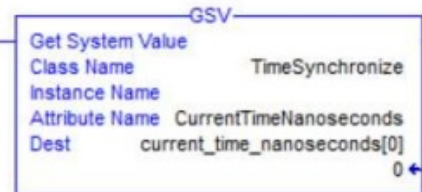
The source tag array that was created using the AMCI_Motion_Axis_Output_Data User Defined Data Type.



The logic on this and the following page shows how a single master axis can be used to control multiple AMCI motion devices.

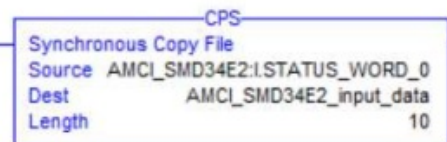
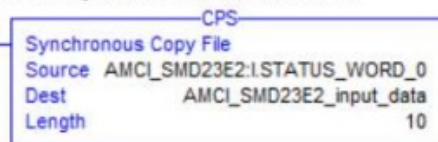
The following rung uses a GSV (Get System Value) instruction to read the system time, in nanoseconds, from the controller. The destination address must be two DINT registers.

This rung MUST NOT have any input conditions.



At the top of your program, BEFORE ALL OF THE ADD ON INSTRUCTIONS, use a CPS instruction to copy the input data from the AMCI motion device to a tag array that was created using the AMCI_Motion_Axis_Input_Data User Defined Data Type.

The input data in this tag array is made up of named tags and will also be used as the buffered data in your program. It is this buffered data that must be used in place of the input data directly from the AMCI motion device.



Set to have AMCI
SMD23E2 follow a
master axis

Run_AMCI_SMD23_24E2.7



File: FAQ_Using_AMCI_Motion_AOIs_with_cipsync.docx Date: 8/15/2024

20 Gear Drive, Plymouth Industrial Park, Terryville, CT 06786

Tel: [860-585-1254](tel:860-585-1254) Fax: [860-584-1973](tel:860-584-1973) Web: www.amci.com

Documents / Resources

	<p>AMCI Integrated Motion Devices with CIP Sync [pdf] Owner's Manual SD4840E2, SD17060E2, SD31045E2, SMD17E2, SMD23E2, SMD24E2, SMD34E2, SV160E2, SV400E2, Integrated Motion Devices with CIP Sync, Motion Devices with CIP Sync, Devices with CIP Sync, CIP Sync, Sync</p>
--	---

References

- [User Manual](#)

Manuals+. Privacy Policy

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.