

FAQ: Using AMCI Motion Add On Instructions with CIP Sync Devices

Beginning in September 2024, with network firmware version 1.37, AMCI’s networked integrated motor drivers now include the ability to use CIP Sync to more closely and quietly follow a master axis.

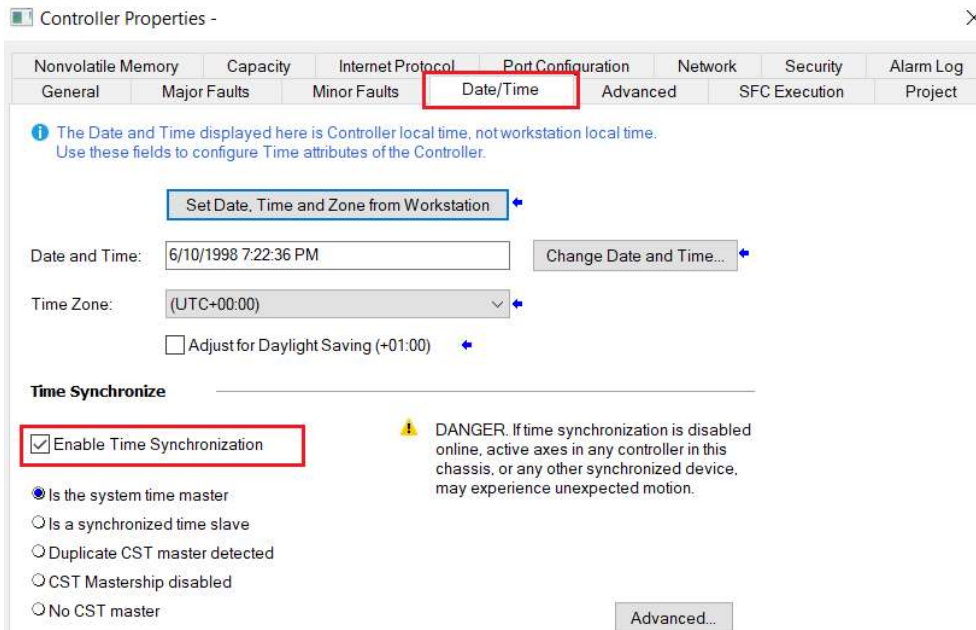
This document describes how to use the AMCI Motion Axis Add On Instructions with CIP Sync to follow a master axis. These Add On Instructions will work with the following AMCI integrated motion devices, including,

- 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, in the network properties, or in the scanner module’s properties.

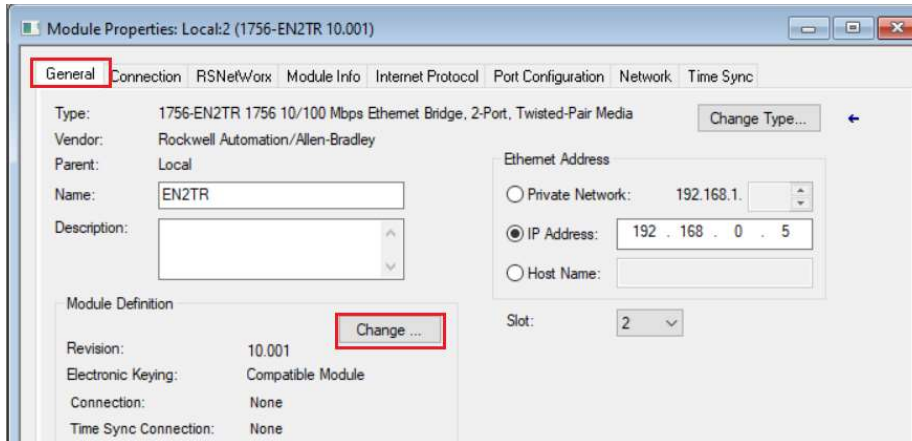
As shown in the following image, select the Date/Time tab and then place a check mark in the Enable Time Synchronization field.



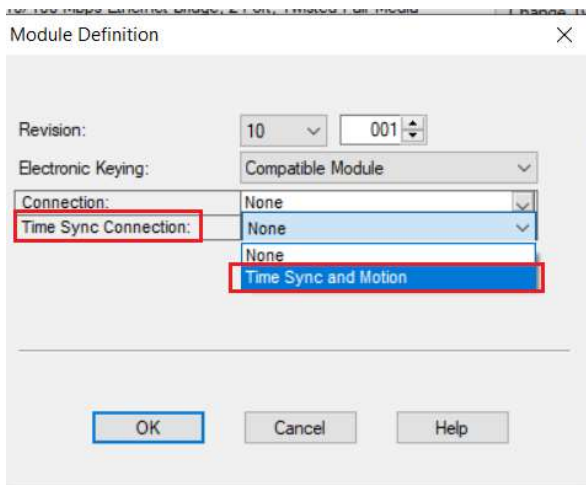
Step 2: Scanner Module Configuration

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.



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.

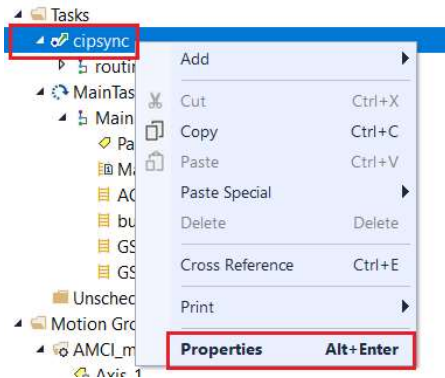


Back on the General tab, click on Apply.

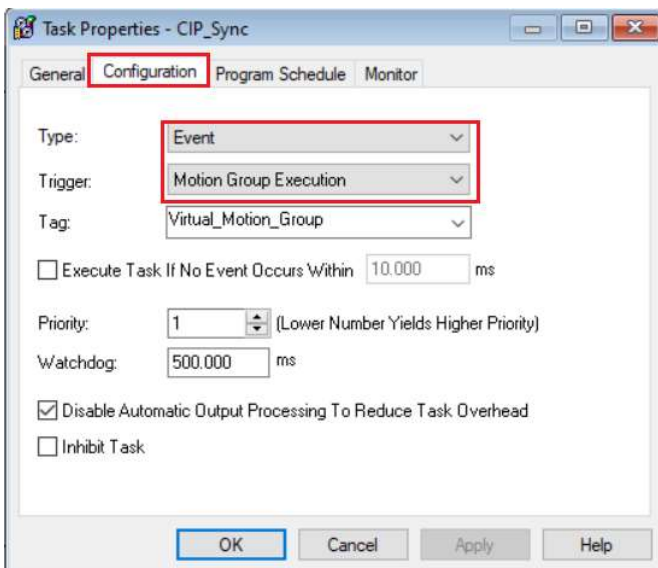
Step 3: Create a Task

While not absolutely 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,

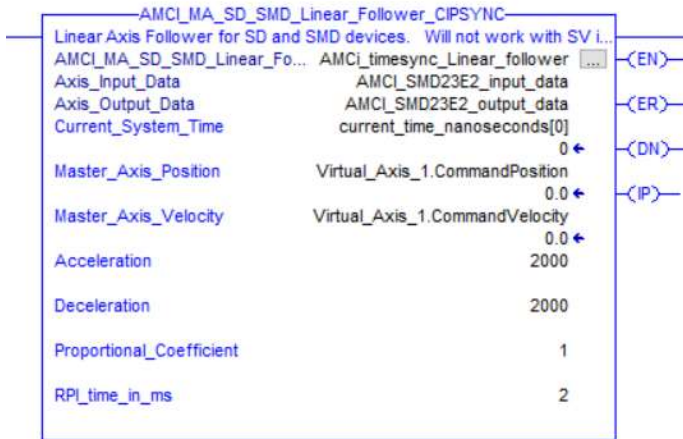
Class Name: TimeSynchronize
Attribute Name: CurrentTimeNanoseconds
Destination: An array consisting of two DINT registers

The GSV instruction should be located in an unconditional rung should be in the Event Driven Task created in Step 2 above.



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 own follower AOIs.

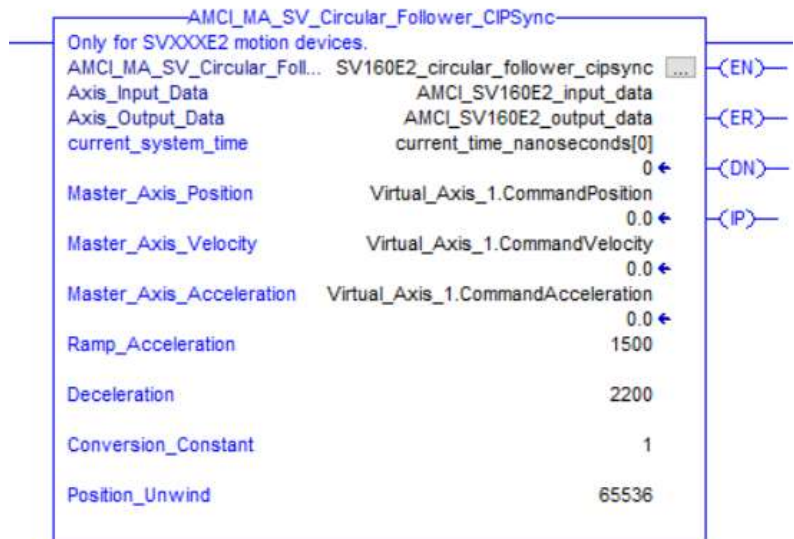
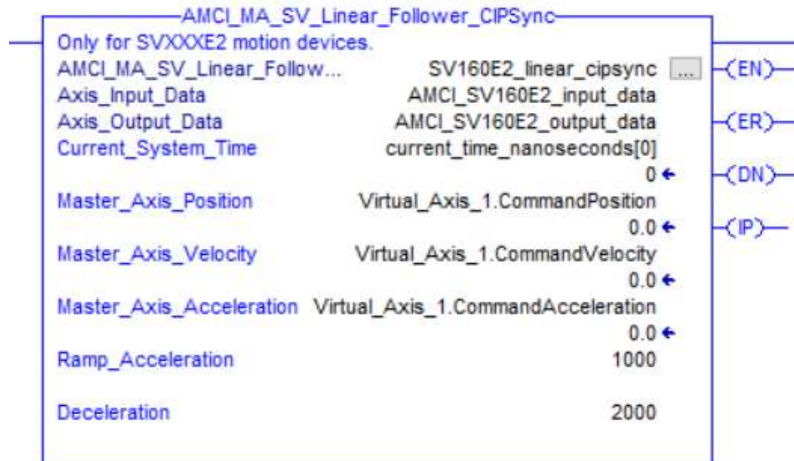


Parameter	
Axis_Input_Data	Input data from AMCI motion device. Uses the AMCI Motion Axis Input Data User Defined Data Type.
Axis_Output_Data	Output data from the AOI to the AMCI motion device. Uses the AMCI Motion Axis Output Data User Defined Data Type.
Current_System_Time	The first word of the two DINT word array of the current time in nanoseconds 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)	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 Conversion Constant parameter before sending them to the AMCI Motion Controller. 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
Position_Unwind (Circular Follower Only)	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.
RPI_Time_in_ms	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.

Enumerations	Set When.....	Reset When
EN (Enable)	Rung is true	Rung is false
DN (Done)	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 own follower AOIs.





Frequently Asked Questions

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 array of the current time in nanoseconds 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 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 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.

Enumerations	Set When.....	Reset When
EN (Enable)	Rung is true	Rung is false
DN (Done)	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)	Motion is occurring	Motion stops or the rung goes false

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.

GSV	
Get System Value	
Class Name	TimeSynchronize
Instance Name	
Attribute Name	CurrentTimeNanoseconds
Dest	current_time_nanoseconds[0]
	0 ←

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.

CPS	
Synchronous Copy File	
Source	AMCI_SMD23E2:I.STATUS_WORD_0
Dest	AMCI_SMD23E2_input_data
Length	10

Circular Axis
Follower
Run_AMCI_SMD23_24E2.7

AMCI_MA_SD_SMD_Circular_Follower_CIPSYNC	
Circular Axis Follower for SD and SMD devices. Will not work with SV ...	
AMCI_MA_SD_SMD_Circular_...	AMCI_timesync_circular_follower [EN] ←
Axis_Input_Data	AMCI_SMD23E2_input_data (DN) ←
Axis_Output_Data	AMCI_SMD23E2_output_data (ER) ←
Current_System_Time	current_time_nanoseconds[0]
	0 ← (IP) ←
Master_Axis_Position	Sample_Motion_Axis.CommandPosition
	0.0 ←
Master_Axis_Velocity	Sample_Motion_Axis.CommandVelocity
	0.0 ←
Acceleration	500
Deceleration	500
Proportional_Coefficient	1
Conversion_Constant	1
Position_Unwind	16384
RPI_Time_in_ms	RPI_time
	2 ←

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

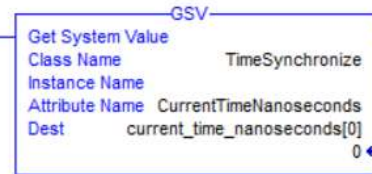
The source tag array that was created using the AMCI_Motion_Axis_Output_Data User Defined Data Type.

CPS	
Synchronous Copy File	
Source	AMCI_SMD23E2_output_data
Dest	AMCI_SMD23E2:O.COMMAND_WORD_0
Length	10

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

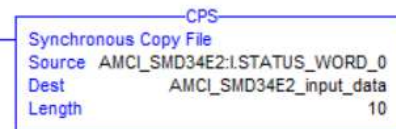
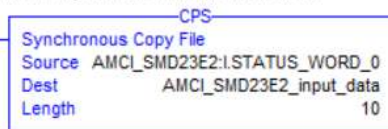
The following rung uses a GSV (Get System Value) instruction to read the system time, in nanoseconds, from the controller. The destination address must be two DINT registers.

This rung MUST NOT have any input conditions.



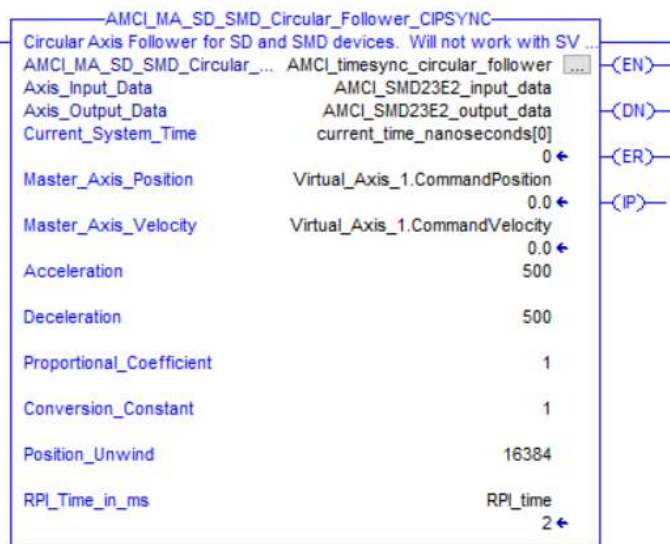
At the top of your program, BEFORE ALL OF THE ADD ON INSTRUCTIONS, use a CPS instruction to copy the input data from the AMCI motion device to a tag array that was created using the AMCI_Motion_Axis_Input_Data User Defined Data Type.

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



Set to have AMCI SMD34E2 follow a master axis

```
run_smd34e2
```

AMCI_MA_SD_SMD_Circular_Follower_CIPSYNC	
Circular Axis Follower for SD and SMD devices. Will not work with SV i...	
AMCI_MA_SD_SMD_Circular_...	AMCI_SMD34E2_circular_follower
Axis_Input_Data	AMCI_SMD34E2_input_data
Axis_Output_Data	AMCI_SMD34E2_output_data
Current_System_Time	current_time_nanoseconds[0]
Master_Axis_Position	Virtual_Axis_1.CommandPosition
Master_Axis_Velocity	Virtual_Axis_1.CommandVelocity
Acceleration	500
Deceleration	500
Proportional_Coefficient	1
Conversion_Constant	1
Position_Unwind	16384
RPL_Time_in_ms	2

At the bottom of your program, after all of the Add On Instructions, use a CPS instruction to copy the data from the AOIs 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.

CPS		CPS	
Synchronous Copy File		Synchronous Copy File	
Source	AMCI_SMD23E2_output_data	Source	AMCI_SMD34E2_output_data
Dest	AMCI_SMD23E2:O.COMMAND_WORD_0	Dest	AMCI_SMD34E2:O.COMMAND_WORD_0
Length	10	Length	10

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