

AKD Ethernet IP: Move Complete status when starting a chain of preset motion tasks

This application note is specific to running a stored motion task sequence per the AKD Ethernet IP Communications manual:

6.4.4 Running a Stored Motion Task Sequence

As an alternative to issuing a single point-to-point position commands, EtherNet/IP can be used to start a pre-defined motion task or sequence of motion tasks.

A motion tasking sequence may be setup in Workbench and then executed later through EtherNet/IP. Motion tasks may also be setup directly through EtherNet/IP as demonstrated in the sample programs.

To execute a motion task sequence, set Block Number equal to the index of the motion task to begin executing and transition the Start Block bit high. The drive must be enabled and the stop and Load/Start bits must be low.

When a stored motion task is running, the response assembly will report this with the Block in Execution status bit, and the executing task will be given in the Block # response byte.

To stop an executing sequence, set the Smooth Stop or Hard Stop bit.

There is no specific status keyword in the AKD that will tell you that a chain of motion tasks or a motion task is “complete”. The AKD also doesn’t have a keyword or status that indicates “currently executing motion task number”.

However when using AKD Ethernet IP communications you will note the following:

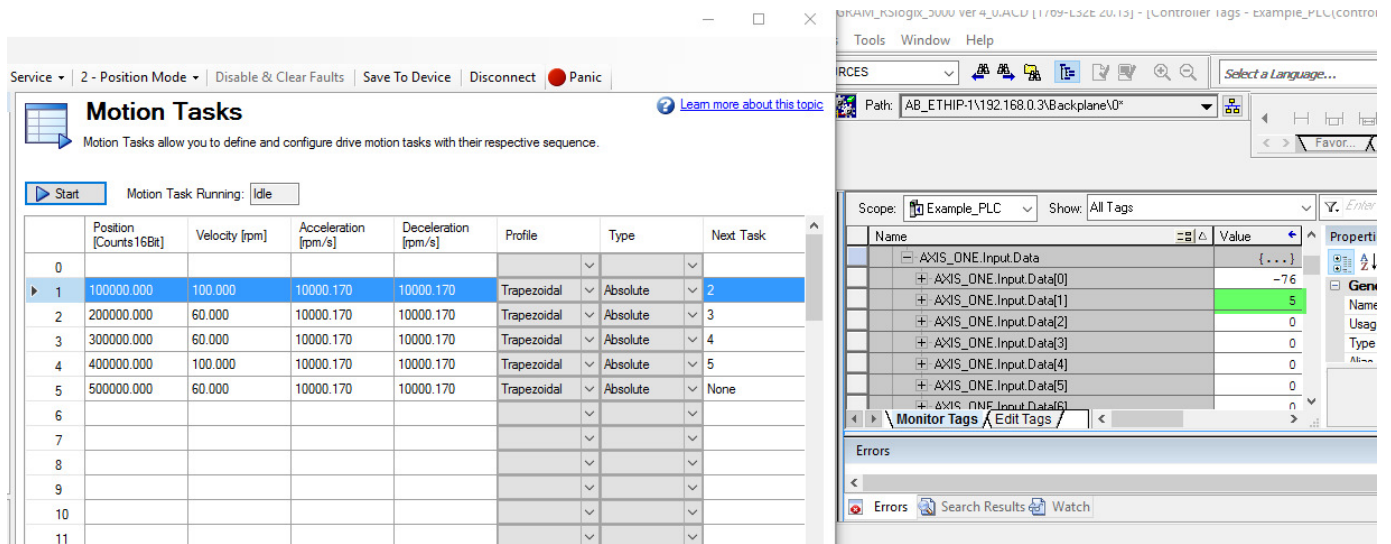
6.2.3 Response Assemblies

In I/O Assembly Messaging, the amplifier transmits a response assembly back to the controller. The response assembly has a number of pre-defined status words and data values. In addition, it can contain one data value which is selected by the Response Type field of the command assembly.

6.2.3.1 Response Assembly Data Structure

Byte	Data	Comment
0	Status Word 1	Various status bits
1	Executing Block #	The index of the Motion Task which is currently being executed
2	Status Word 2	Various status bits
3	Response Type	Specifies the response type of this assembly, echoing the Response Type set in the command assembly.

To test this I setup the following chain of motion tasks and monitored Byte 1 of the response assembly for AXIS_ONE in the sample project. From the screenshots below (Workbench on the left and RSLogix5000 on the right), when motion task 1 was started, byte 1 did step through 1, 2, 3, 4, and 5 as each motion task was executing but while executing the last motion task 5 in this example (the terminating task number) and also after it completed the value in byte 1 stayed 5. I rehommed and then re-executed motion task 1 and it stepped through the numbers again. Based on this you cannot confirm the chain is finished by looking at Byte 1 only since you can't tell the difference between the states of the final task in process executing or that it is complete.



However, the EIP manual shows the following status word in Byte 0 of the Response Assembly.

6.2.3.2 Status Word 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reserved	Homed	Current Direction	General Fault	In Position	Block in Execution	In Motion

Enable State: This bit reflects the enable state of the amplifier.

Homed: This bit is set when the drive has been successfully homed.

Current Direction: This bit reflects the actual direction of motion.

General Fault: This bit indicates whether or not a fault has occurred.

In Position: This bit indicates whether or not the motor is on the last targeted position (1=On Target).

Block in Execution: When set, indicates the amplifier is running a motion task.

Executing Block # (Byte 1 in Response Assembly): Indicates the index of the currently executing Motion Task when the Block in Execution bit is set.

In Motion: This bit indicates whether a trajectory is in progress (1) or has completed (0).

This bit is set immediately when motion begins and remains set for the entire motion.

The same test was run again and both bit 0 (In Motion) and Bit 1 (Block In Execution) turned on (1) as long as a motion task was running (executing) and then turned off (back to zero) when the Motion Task Running was Idle at the end of the chain of moves.

The description above is Block In Execution: When set indicates the amplifier is running a motion task.

Motion Tasks allow you to define and configure drive motion tasks with their respective sequence.

Stop Motion Task Running: 2

	Position [Counts/16Bit]	Velocity [rpm]	Acceleration [rpm/s]	Deceleration [rpm/s]	Profile	Type	Next Task
0							
1	100000.000	100.000	10000.170	10000.170	Trapezoidal	Absolute	2
2	200000.000	60.000	10000.170	10000.170	Trapezoidal	Absolute	3
3	300000.000	60.000	10000.170	10000.170	Trapezoidal	Absolute	4
4	400000.000	100.000	10000.170	10000.170	Trapezoidal	Absolute	5
5	500000.000	60.000	10000.170	10000.170	Trapezoidal	Absolute	None
6							
7							
8							
9							

In some cases the programmer may want to add code to validate that the position is at the final target position based on feedback position and the in position window status before declaring the chain of moves complete but this is up to the programmer.

Both the in position window and the current position feedback is built in to the static section of the response assembly.

<https://www.kollmorgen.com/en-us/developer-network/akd-ethernet-ip-position-window-status/>

6.2.3.2 Status Word 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reserved	Homed	Current Direction	General Fault	In Position	Block in Execution	In Motion

Enable State: This bit reflects the enable state of the amplifier.

Homed: This bit is set when the drive has been successfully homed.

Current Direction: This bit reflects the actual direction of motion.

General Fault: This bit indicates whether or not a fault has occurred.

In Position: This bit indicates whether or not the motor is on the last targeted position (1=On Target).

Block in Execution: When set, indicates the amplifier is running a motion task.

Executing Block # (Byte 1 in Response Assembly): Indicates the index of the currently executing Motion Task when the Block in Execution bit is set.

In Motion: This bit indicates whether a trajectory is in progress (1) or has completed (0).

This bit is set immediately when motion begins and remains set for the entire motion.