AKD Ethernet IP "In Position Status".

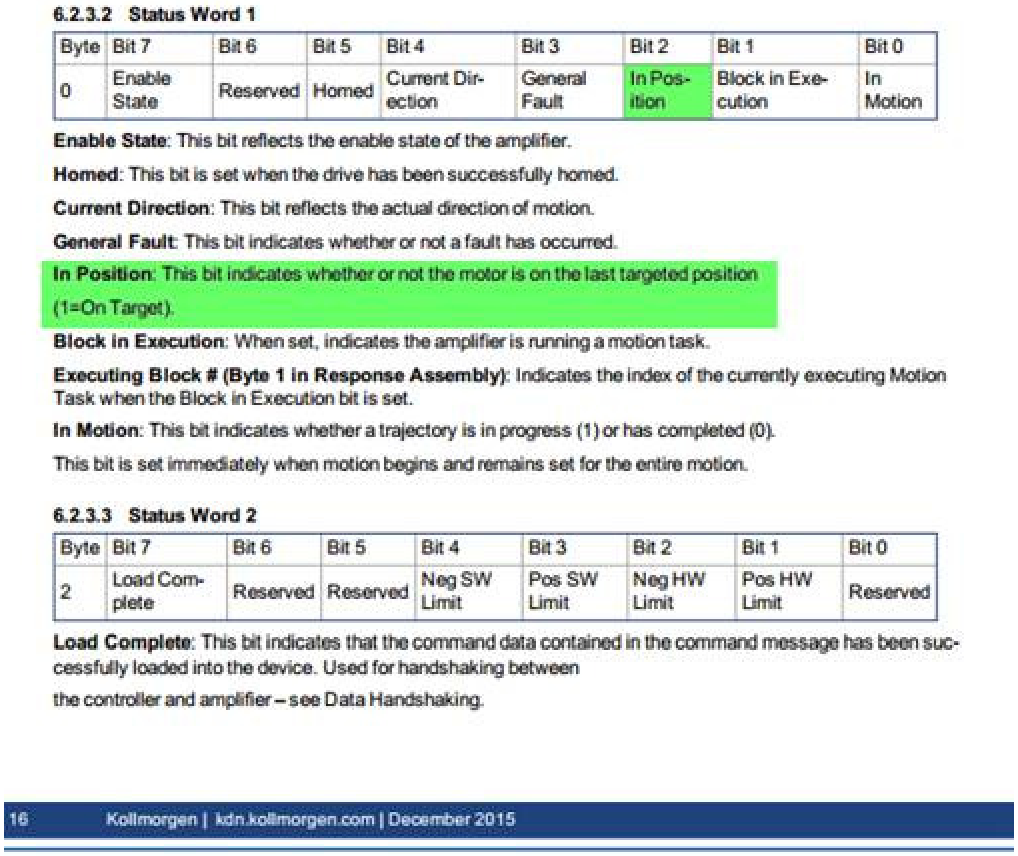
I've seen a few instances where the PLC programmer used the (axis_name).Status.On_Target_Position bit for logic to trigger other events in their ladder logic and in some cases they complained they observed the status bit flicker on and off or stay off, etc. Keep in mind the AKD_MOVE AOI already uses this status bit internally in tandem with the profile in progress status bit. When the AKD_Move block is enabled, the PC on the output of the given AKD_MOVE AOI is unlatched. On profile in progress (not in progress) and On target position, the PC (Process Complete) bit on he output of the given AKD_MOVE AOI is latched. In this way, the PC status can be used for "move complete".

A general rule of thumb for positioning is that your feedback resolution is 10 times the resolution you are trying to settle to.

That said, with an AKD Ethernet IP drive and using the On Target Position status or to compare target position with actual read over Ethernet IP it is important to understand how the feedback position is displayed both in Workbench and in the PLC over Ethernet IP. Ethernet IP position scaling has nothing to do with Workbench Units unless you set them up the same (recommended). The default scaling for Ethernet IP is 65536. Often this is adequate. In very high precision applications sometimes using 65536 (16 bit) resolution may be too low. See the AKD Getting Started quick start for more details on scaling (it shows an example of increasing the Ethernet IP scaling to 20 bit (1048576 counts per rev if required).

In Position Status (axis_name.Status.On_Target_Position)

The AKD Ethernet IP drive has status words in the response assembly where one of the bits is the "in position" status. This is detailed in the AKD Ethernet IP Communications manual as shown below.



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.

6.2.3.3 Status Word 2

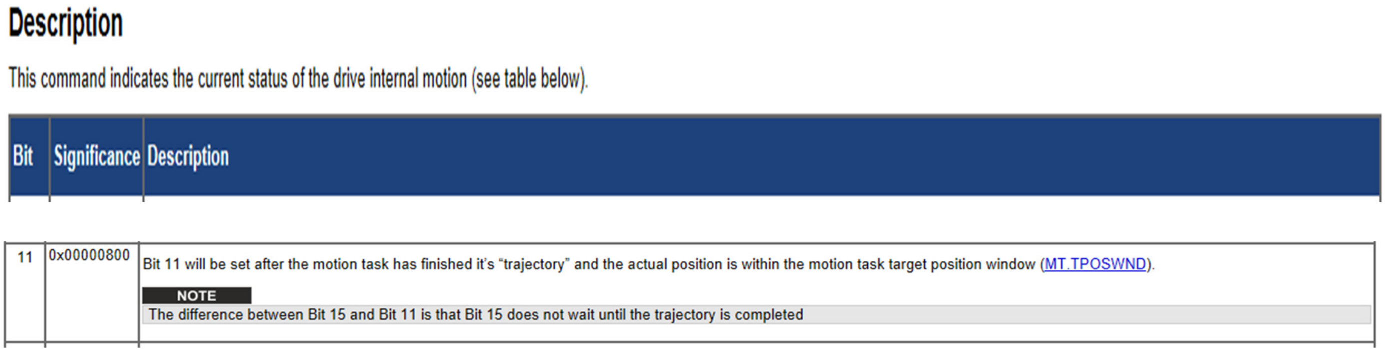
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2	Load Complete	Reserved	Reserved	Neg SW Limit	Pos SW Limit	Neg HW Limit	Pos HW Limit	Reserved

Load Complete: This bit indicates that the command data contained in the command message has been successfully loaded into the device. Used for handshaking between

the controller and amplifier – see Data Handshaking.

16 Kollmorgen | kdn.kollmorgen.com | December 2015

The In Position bit is based on bit 11 of the DRV.MOTIONSTAT in the AKD.

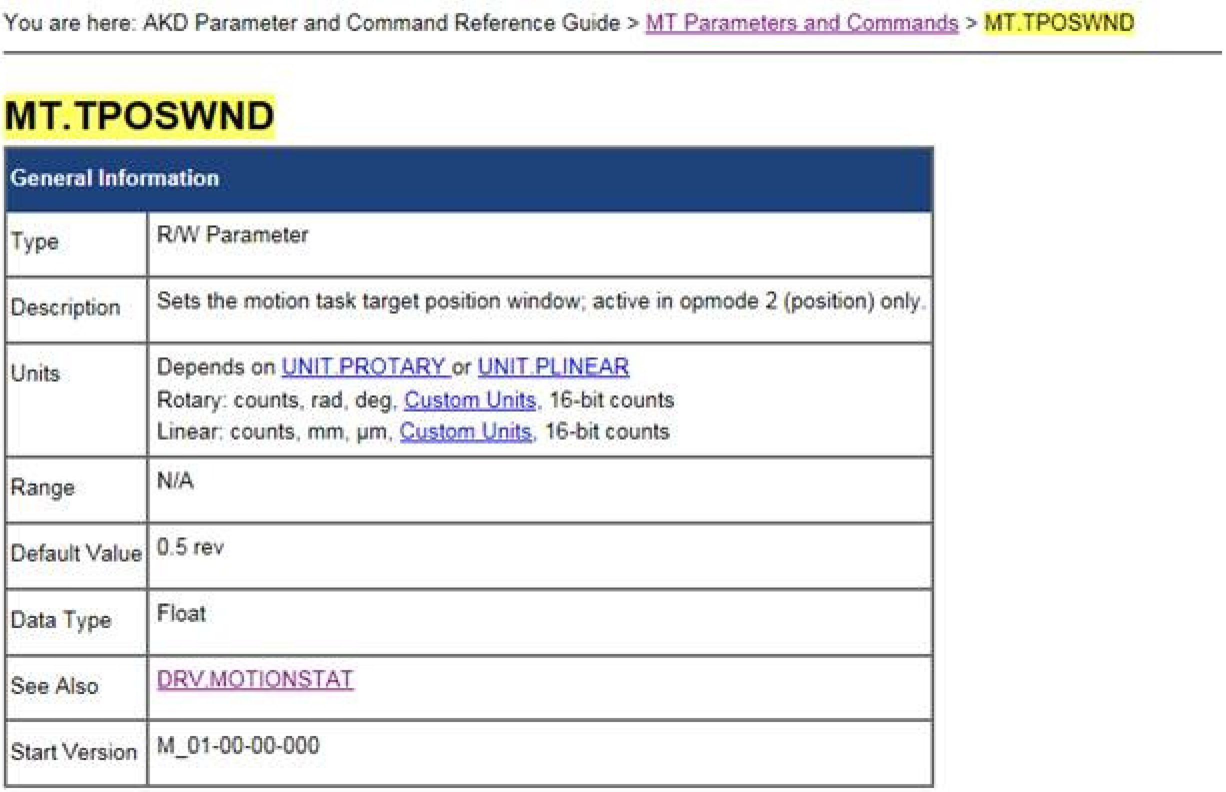


Description

This command indicates the current status of the drive internal motion (see table below).

Bit	Significance	Description
11	0x00000800	Bit 11 will be set after the motion task has finished it's "trajectory" and the actual position is within the motion task target position window (MT.TPOSWND). NOTE The difference between Bit 15 and Bit 11 is that Bit 15 does not wait until the trajectory is completed

The AKD parameter/keyword MT.TPOSWND uses the position units of the drive (Workbench units) to set the in position window.



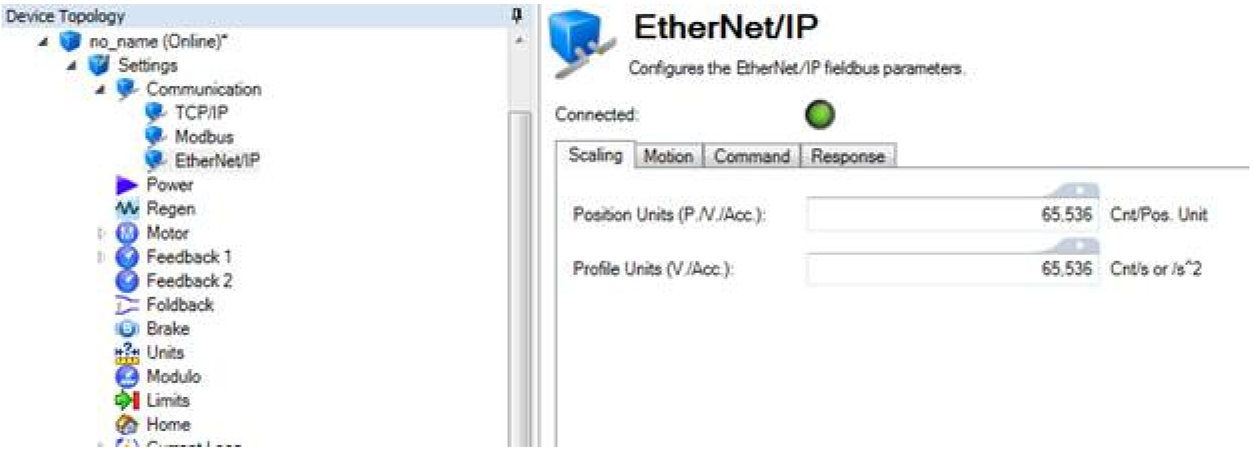
You are here: AKD Parameter and Command Reference Guide > MT Parameters and Commands > MT.TPOSWND

MT.TPOSWND

General Information	
Type	R/W Parameter
Description	Sets the motion task target position window; active in opmode 2 (position) only.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, Custom Units, 16-bit counts Linear: counts, mm, µm, Custom Units, 16-bit counts
Range	N/A
Default Value	0.5 rev
Data Type	Float
See Also	DRV.MOTIONSTAT
Start Version	M_01-00-00-000

If the EIP units and Workbench units are the same (recommended) then the window would be the same units as the PLC.

Assuming 16 bit resolution or 65536 counts per rev, the MT.POSWND default value is 32768.000 or 0.5 revolutions. If you are using some other units (i.e. cm, inches, etc.) then the value will be scaled accordingly.



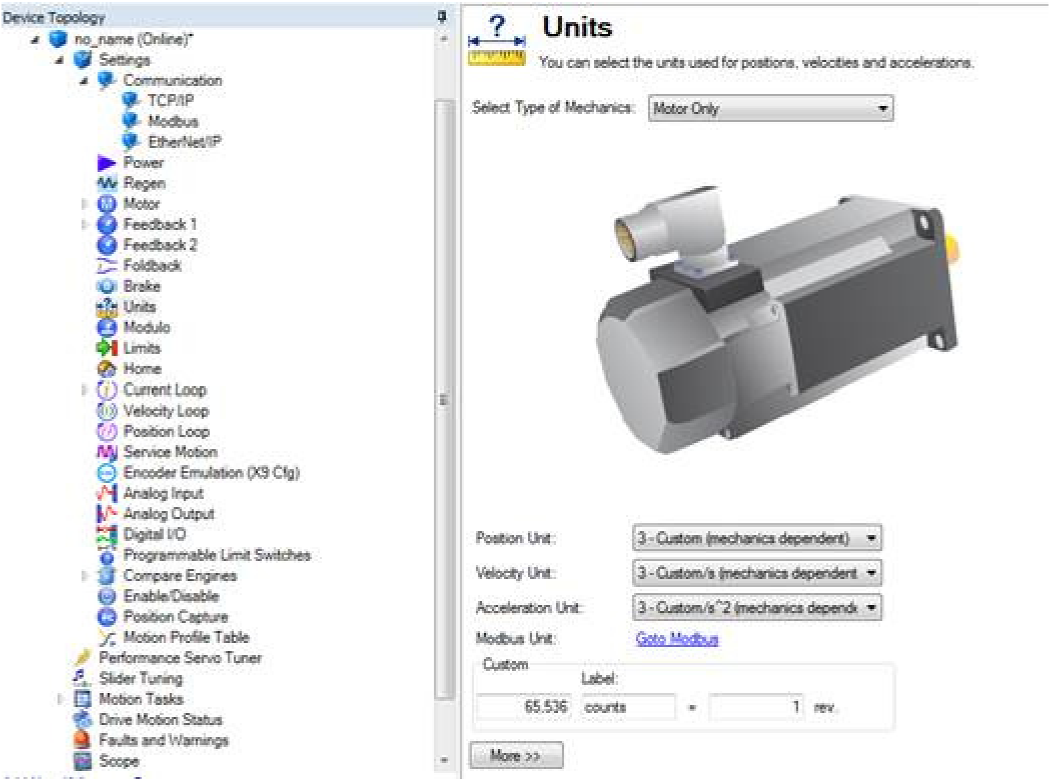
EtherNet/IP

Configures the EtherNet/IP fieldbus parameters.

Connected: ●

Position Units (P./V./Acc.): 65,536 Cnt/Pos. Unit

Profile Units (V./Acc.): 65,536 Cnt/s or /s^2



Units

You can select the units used for positions, velocities and accelerations.

Select Type of Mechanics: Motor Only

Position Unit: 3 - Custom (mechanics dependent)

Velocity Unit: 3 - Custom/s (mechanics dependent

Acceleration Unit: 3 - Custom/s^2 (mechanics depend

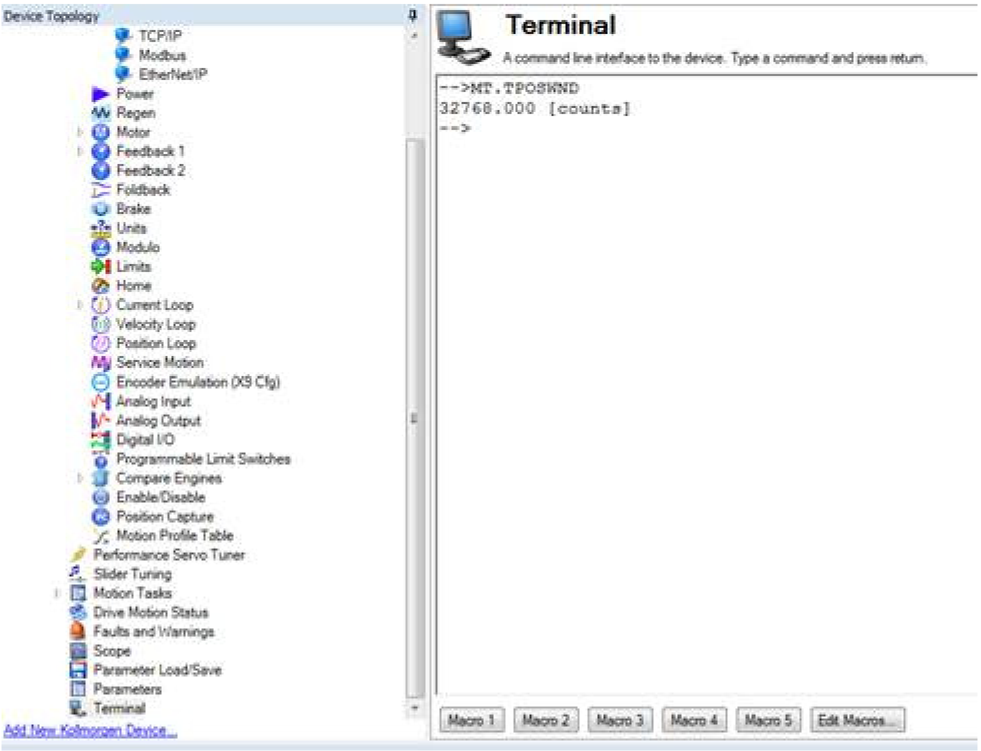
Modbus Unit: Goto Modbus

Custom

Label:

65,536 counts = 1 rev.

Querying or setting MT.TPOSWND:



Terminal

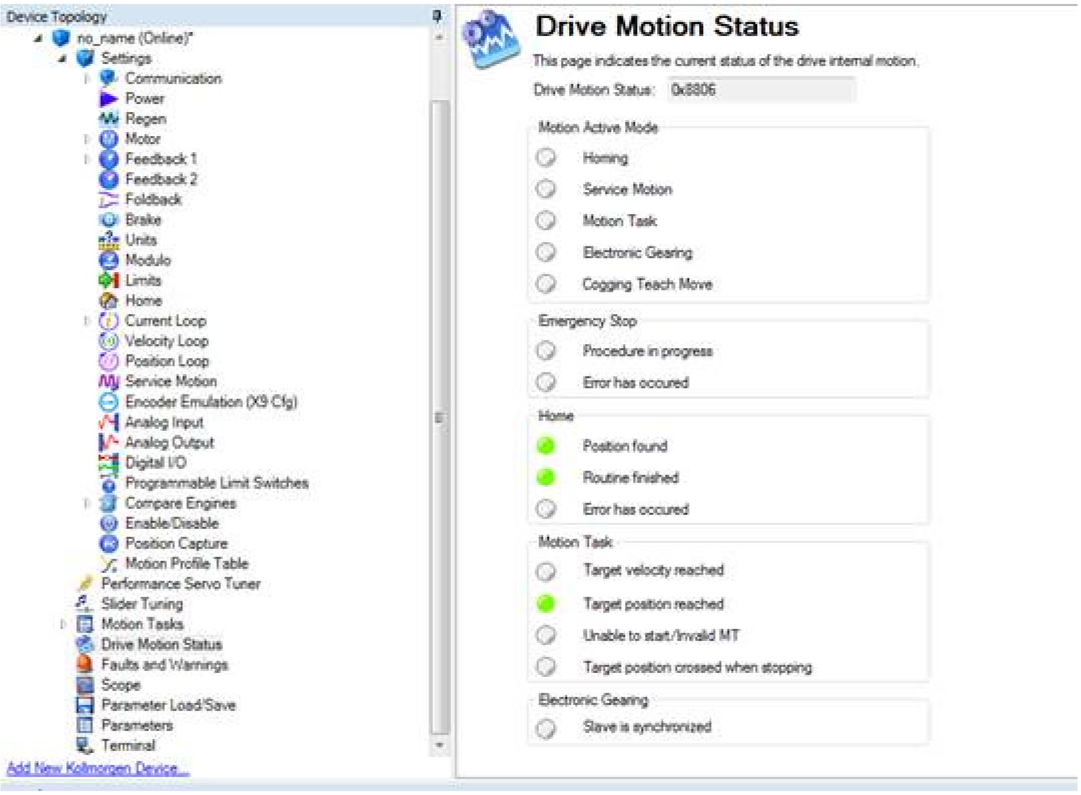
A command line interface to the device. Type a command and press return.

```
-->MT.TPOSWND
32768.000 [counts]
-->
```

<u>Bench Test</u>

I was able on my bench test to detune the drive and set the in position window (MT.TPOSWND=1 or 10 for example) sufficiently low using Workbench Terminal and then monitoring the In Position status where I could see the In Position bit toggle on and off as the position feedback was going in and out of the window even sitting still.

The In Position status in the drive can be viewed on the “Drive Motion Status” screen. Again I could increase or decrease the MT.TPOSWND in Workbench Terminal to adjust the in position window sensitivity and moved the motor shaft slightly and could see the status either stay on, blink, or go off.



Drive Motion Status

This page indicates the current status of the drive internal motion.

Drive Motion Status: 0x8806

Motion Active Mode
- ○ Homing
- ○ Service Motion
- ○ Motion Task
- ○ Electronic Gearing
- ○ Cogging Teach Move

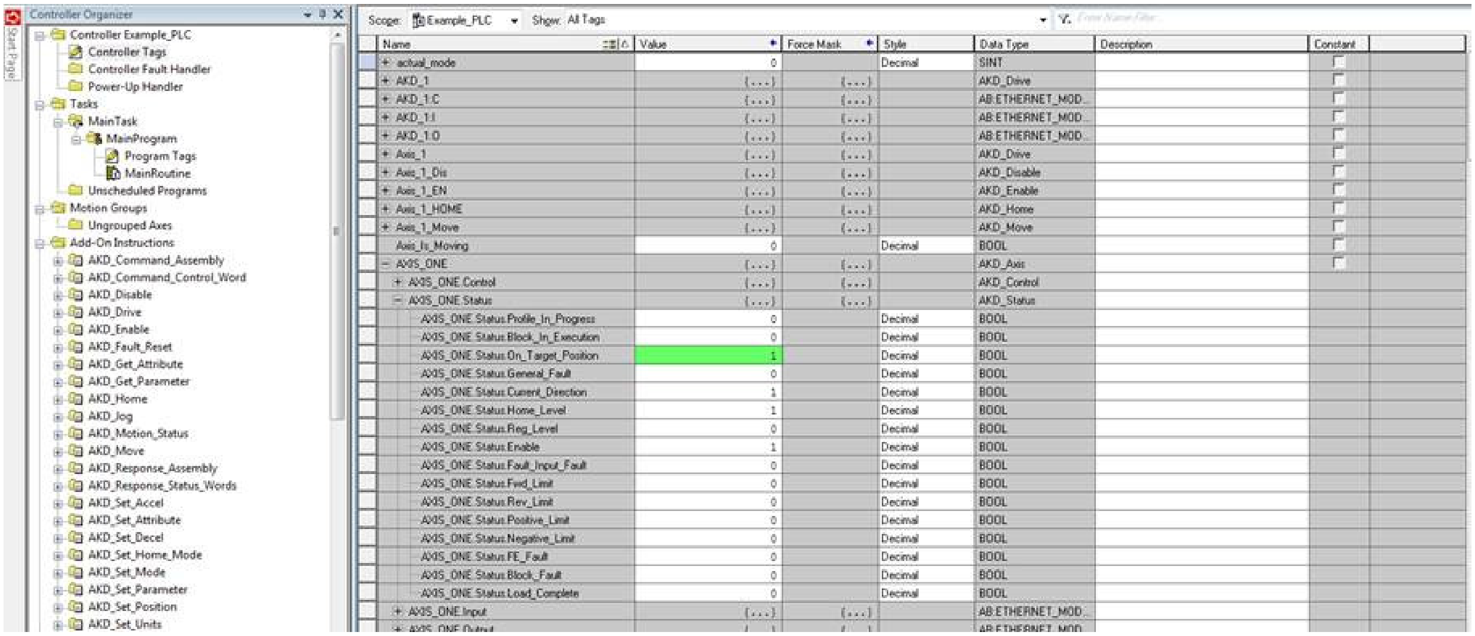
Emergency Stop
- ○ Procedure in progress
- ○ Error has occured

Home
- ● Position found
- ● Routine finished
- ○ Error has occured

Motion Task
- ○ Target velocity reached
- ● Target position reached
- ○ Unable to start/Invalid MT
- ○ Target position crossed when stopping

Electronic Gearing
- ○ Slave is synchronized

You can look at the same status in your ladder or alternatively in the Controller Tags. In my case as in the sample project my axis was called AXIS_ONE.

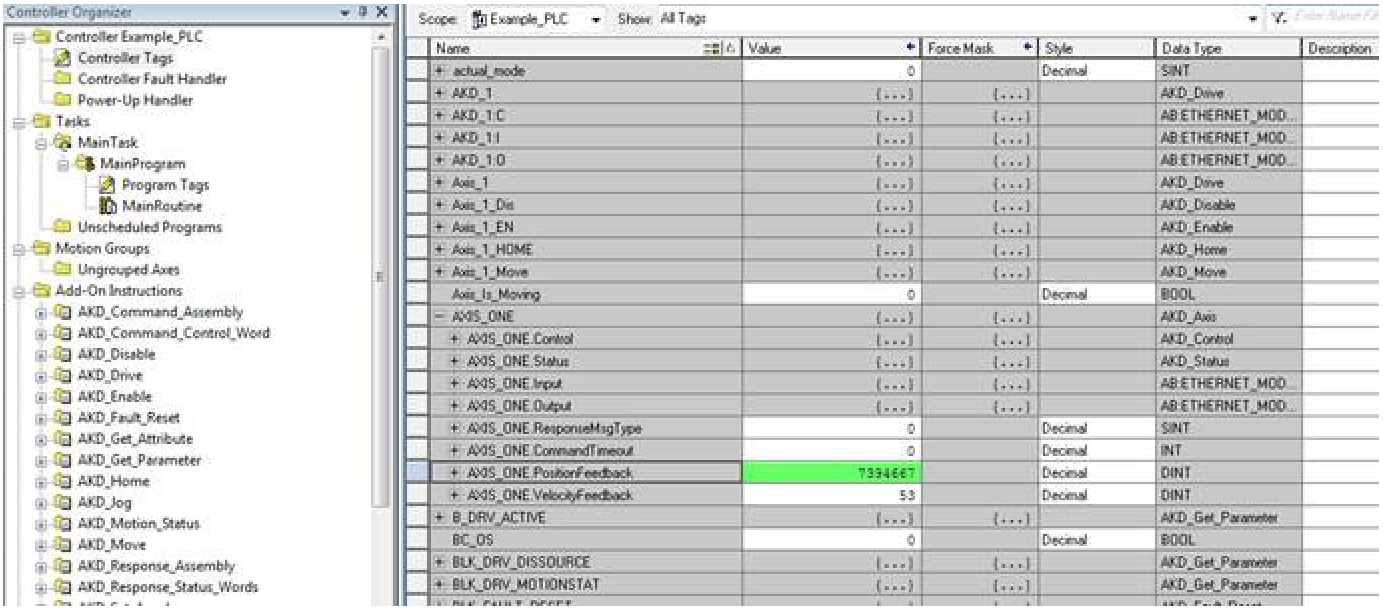


Scope: Example_PLC Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description	Constant
+ actual_mode	0		Decimal	SINT		□
+ AKD_1	{...}	{...}		AKD_Drive		□
+ AKD_1:C	{...}	{...}		AB:ETHERNET_MOD...		□
+ AKD_1:I	{...}	{...}		AB:ETHERNET_MOD...		□
+ AKD_1:O	{...}	{...}		AB:ETHERNET_MOD...		□
+ Axis_1	{...}	{...}		AKD_Drive		□
+ Axis_1_Dis	{...}	{...}		AKD_Disable		□
+ Axis_1_EN	{...}	{...}		AKD_Enable		□
+ Axis_1_HOME	{...}	{...}		AKD_Home		□
+ Axis_1_Move	{...}	{...}		AKD_Move		□
Axis_Is_Moving	0		Decimal	BOOL		□
− AXIS_ONE	{...}	{...}		AKD_Axis		□
+ AXIS_ONE.Control	{...}	{...}		AKD_Control		
− AXIS_ONE.Status	{...}	{...}		AKD_Status		
AXIS_ONE.Status.Profile_In_Progress	0		Decimal	BOOL		
AXIS_ONE.Status.Block_In_Execution	0		Decimal	BOOL		
AXIS_ONE.Status.On_Target_Position	1		Decimal	BOOL		
AXIS_ONE.Status.General_Fault	0		Decimal	BOOL		
AXIS_ONE.Status.Current_Direction	1		Decimal	BOOL		
AXIS_ONE.Status.Home_Level	1		Decimal	BOOL		
AXIS_ONE.Status.Reg_Level	0		Decimal	BOOL		
AXIS_ONE.Status.Enable	1		Decimal	BOOL		
AXIS_ONE.Status.Fault_Input_Fault	0		Decimal	BOOL		
AXIS_ONE.Status.Fwd_Limit	0		Decimal	BOOL		
AXIS_ONE.Status.Rev_Limit	0		Decimal	BOOL		
AXIS_ONE.Status.Positive_Limit	0		Decimal	BOOL		
AXIS_ONE.Status.Negative_Limit	0		Decimal	BOOL		
AXIS_ONE.Status.FE_Fault	0		Decimal	BOOL		
AXIS_ONE.Status.Block_Fault	0		Decimal	BOOL		
AXIS_ONE.Status.Load_Complete	0		Decimal	BOOL		
+ AXIS_ONE.Input	{...}	{...}		AB:ETHERNET_MOD...		
+ AXIS_ONE.Output	{...}	{...}		AB:ETHERNET_MOD...		

I have seen some implementations where the programmer also monitored the PL.FB (position feedback) in the PLC (this is already available in the response assembly on the RPI poll) to check the current position vs. the target position and also that the position is within a specified window. They essentially created their own "In Position Window" in the PLC.

From my sample project you can see the data under Controller Tags. We also bring this out to the ladder in our sample project.



Controller Organizer

- Controller Example_PLC
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - MainProgram
      - Program Tags
      - MainRoutine
  - Unscheduled Programs
- Motion Groups
  - Ungrouped Axes
- Add-On Instructions
  - AKD_Command_Assembly
  - AKD_Command_Control_Word
  - AKD_Disable
  - AKD_Drive
  - AKD_Enable
  - AKD_Fault_Reset
  - AKD_Get_Attribute
  - AKD_Get_Parameter
  - AKD_Home
  - AKD_Jog
  - AKD_Motion_Status
  - AKD_Move
  - AKD_Response_Assembly
  - AKD_Response_Status_Words

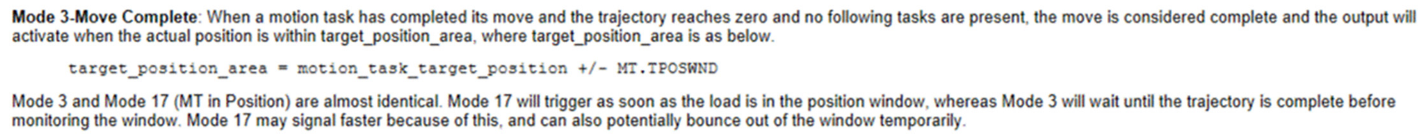
Scope: Example_PLC Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
+ actual_mode	0		Decimal	SINT	
+ AKD_1	{...}	{...}		AKD_Drive	
+ AKD_1:C	{...}	{...}		AB:ETHERNET_MOD...	
+ AKD_1:I	{...}	{...}		AB:ETHERNET_MOD...	
+ AKD_1:O	{...}	{...}		AB:ETHERNET_MOD...	
+ Axis_1	{...}	{...}		AKD_Drive	
+ Axis_1_Dis	{...}	{...}		AKD_Disable	
+ Axis_1_EN	{...}	{...}		AKD_Enable	
+ Axis_1_HOME	{...}	{...}		AKD_Home	
+ Axis_1_Move	{...}	{...}		AKD_Move	
Axis_Is_Moving	0		Decimal	BOOL	
− AXIS_ONE	{...}	{...}		AKD_Axis	
+ AXIS_ONE.Control	{...}	{...}		AKD_Control	
+ AXIS_ONE.Status	{...}	{...}		AKD_Status	
+ AXIS_ONE.Input	{...}	{...}		AB:ETHERNET_MOD...	
+ AXIS_ONE.Output	{...}	{...}		AB:ETHERNET_MOD...	
+ AXIS_ONE.ResponseMsgType	0		Decimal	SINT	
+ AXIS_ONE.CommandTimeout	0		Decimal	INT	
+ AXIS_ONE.PositionFeedback	7394667		Decimal	DINT	
+ AXIS_ONE.VelocityFeedback	53		Decimal	DINT	
+ B_DRV_ACTIVE	{...}	{...}		AKD_Get_Parameter	
BC_OS	0		Decimal	BOOL	
+ BLK_DRV_DISSOURCE	{...}	{...}		AKD_Get_Parameter	
+ BLK_DRV_MOTIONSTAT	{...}	{...}		AKD_Get_Parameter	

A final note is to keep in mind you can use the Workbench Scope to trace your moves and check

your settling into the target position and window.

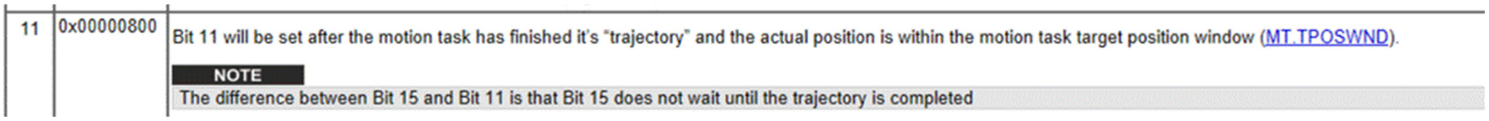
It is possible to Scope the in position status bit in Workbench. The requirement would be to set a spare output (if you have one) to Mode 3 Move Complete I believe (based on a bench test it seems the trajectory has to be completed AND in the target window).
Mode 3 is based on bit 11 in the DRV.MOTIONSTAT. There is a mode 17 based on bit 15 of the DRV.MOTIONSTAT as well.



Mode 3-Move Complete: When a motion task has completed its move and the trajectory reaches zero and no following tasks are present, the move is considered complete and the output will activate when the actual position is within target_position_area, where target_position_area is as below.

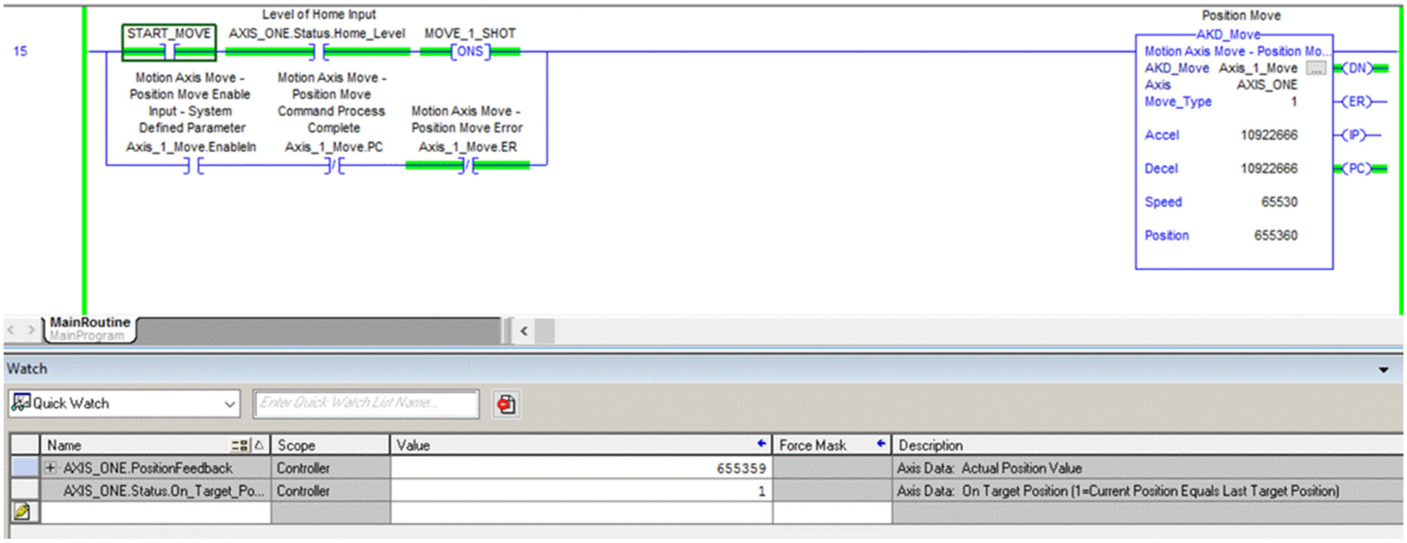
```
target_position_area = motion_task_target_position +/- MT.TPOSWND
```

Mode 3 and Mode 17 (MT in Position) are almost identical. Mode 17 will trigger as soon as the load is in the position window, whereas Mode 3 will wait until the trajectory is complete before monitoring the window. Mode 17 may signal faster because of this, and can also potentially bounce out of the window temporarily.



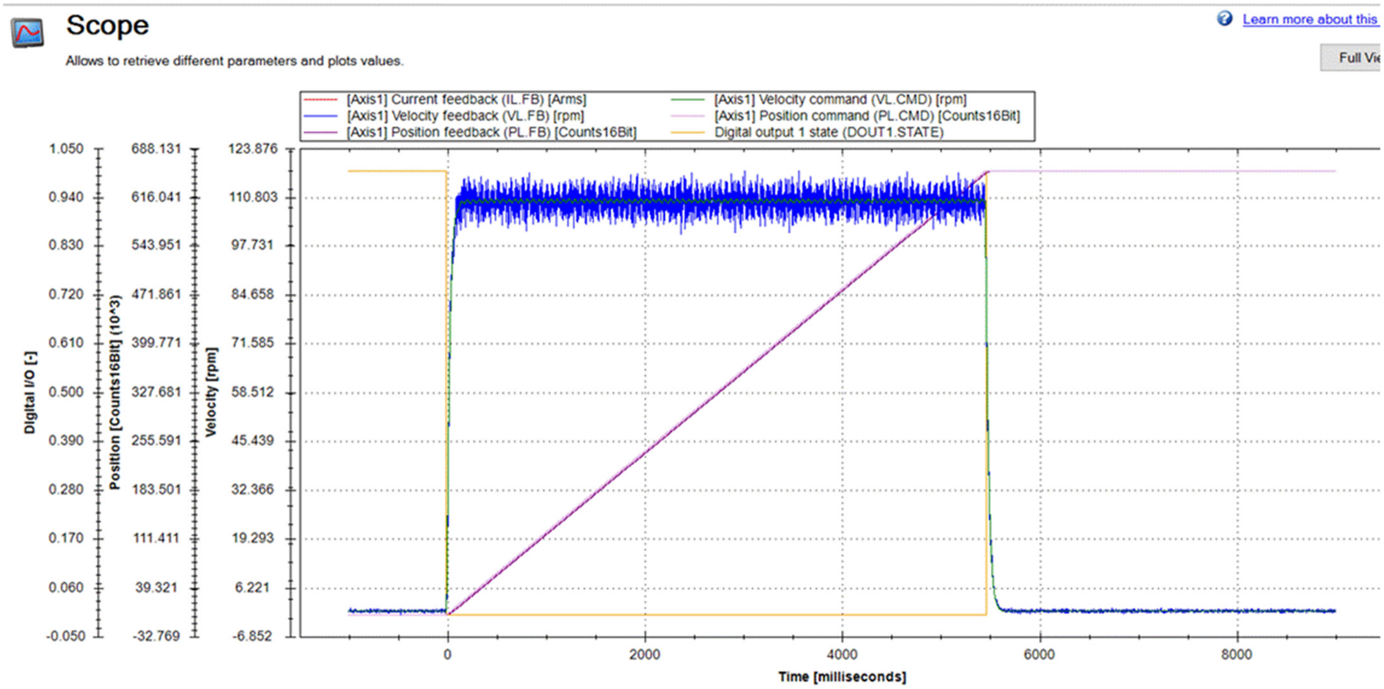
11	0x00000800	Bit 11 will be set after the motion task has finished it's "trajectory" and the actual position is within the motion task target position window (MT.TPOSWND). NOTE The difference between Bit 15 and Bit 11 is that Bit 15 does not wait until the trajectory is completed

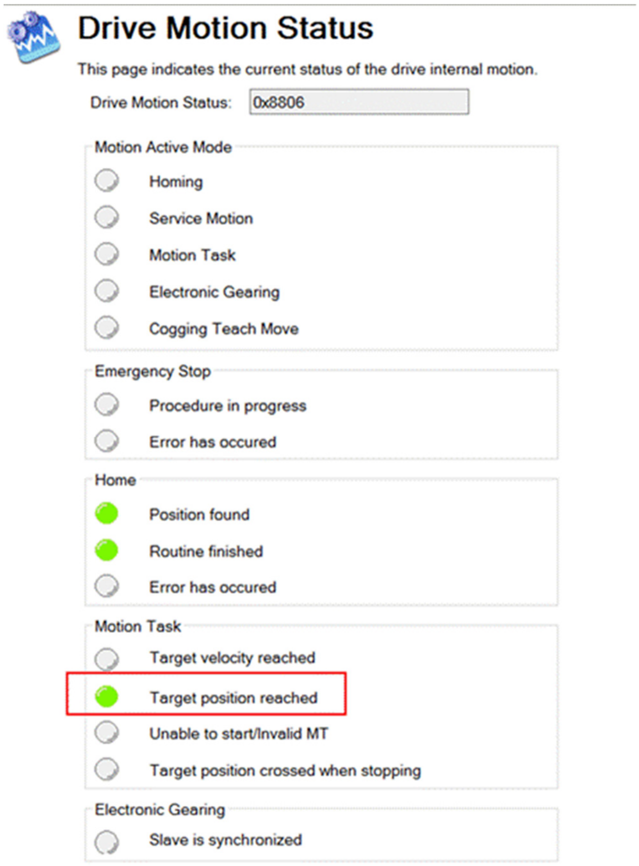
You can monitor the on target position bit and the feedback position using a Quick Watch in RSLogix5000.



Name	Scope	Value	Force Mask	Description
+ AXIS_ONE.PositionFeedback	Controller	655359		Axis Data: Actual Position Value
AXIS_ONE.Status.On_Target_Po...	Controller	1		Axis Data: On Target Position (1=Current Position Equals Last Target Position)

I set my Scope to record the DOUT1.STATE set for Move Complete and recorded velocity and position commands and feedbacks. You can see the output turn on at the end of the move.





Drive Motion Status

This page indicates the current status of the drive internal motion.

Drive Motion Status: 0x8806

Motion Active Mode

- ○ Homing
- ○ Service Motion
- ○ Motion Task
- ○ Electronic Gearing
- ○ Cogging Teach Move

Emergency Stop

- ○ Procedure in progress
- ○ Error has occured

Home

- ● Position found
- ● Routine finished
- ○ Error has occured

Motion Task

- ○ Target velocity reached
- ● Target position reached
- ○ Unable to start/Invalid MT
- ○ Target position crossed when stopping

Electronic Gearing

- ○ Slave is synchronized