APPLICATION NOTE YASKAWA

Title: SigmaLogic Example Code Example Manual

Product(s): SigmaLogic, LogicWorks | **Doc. No.** AN.SigmaLogic.01

1. Application Overview

SigmaLogic is an EtherNet/IP Indexer that was designed to work seamlessly with the CompactLogix and ControlLogix PLCs from Allen Bradley. Yaskawa has created Add-On Instructions (AOI) for easy import into the RSLogix programming environment. The AOIs as well as other documentation and videos can be found at www.yaskawa.com/sigmalogic.This application note highlights some key elements of the "EC.SigmaLogic.01_RSLogix_Example.ACD" code that is included in the SigmaLogic AOI zip file. This example code was created in RSLogix 5000 v20 to provide useful examples for RSLogix programmers. The code contains interlocking logic to help prevent alarms and faults from occurring when executing the AOIs.

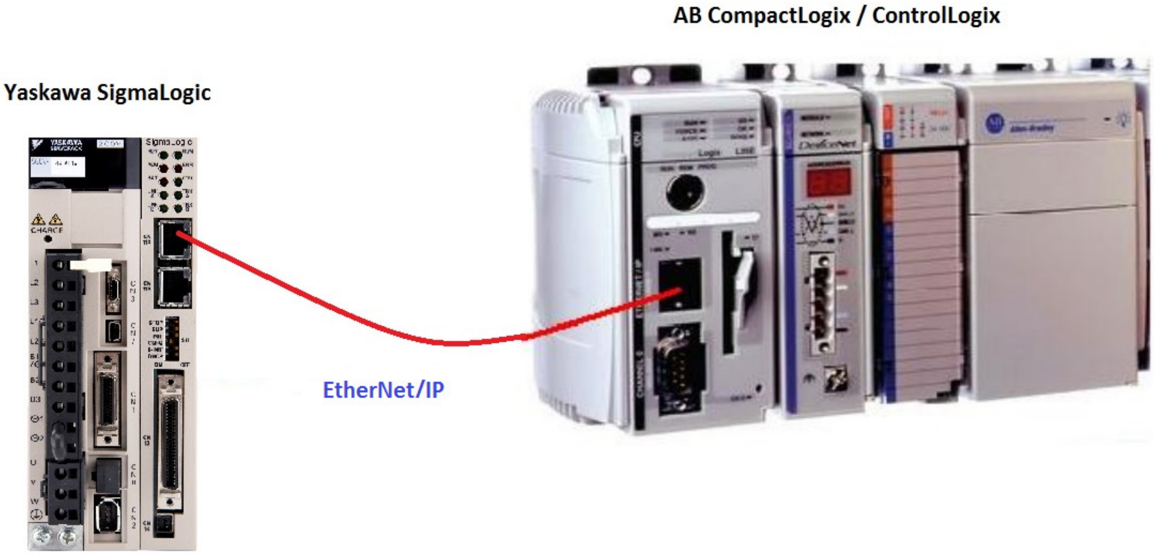


Figure 1: Required Components

Component	Product and Model Number
EIP Servo Indexer	SGDV-****E1A******B00 (SigmaLogic) or SGDV-****E*A******300 (MP2600)
Motor	Any Sigma-II, Sigma-III or Sigma-5 motor recognized by Sigma-5 amplifiers
PLC	CompactLogix, ControlLogix or SoftLogix with an EtherNet/IP Communications Adapter
Software	RSLogix v20 was used to create this code; v17 or higher works with SigmaLogic

2. Application Solution and Benefits

The example code uses each AOI provided by Yaskawa and shows a suggested way to use these AOIs in the context of an RSLogix ladder program. There are only a couple of basic rules when using the AOIs. The first rule is that each SigmaLogic used needs to have its own MCFG AOI. This defines the Axis structures and provides a huge amount of diagnostic information. The other rule is that most motion-related AOIs cannot be executed if another AOI is already in progress. The interlocking logic used in this example will prevent the user from violating this rule which would generate a FLT_RDY error in the AOI. The benefit to the RSLogix programmer is that this example code will help them avoid frustrating errors and alarms.

3. Initial Configuration

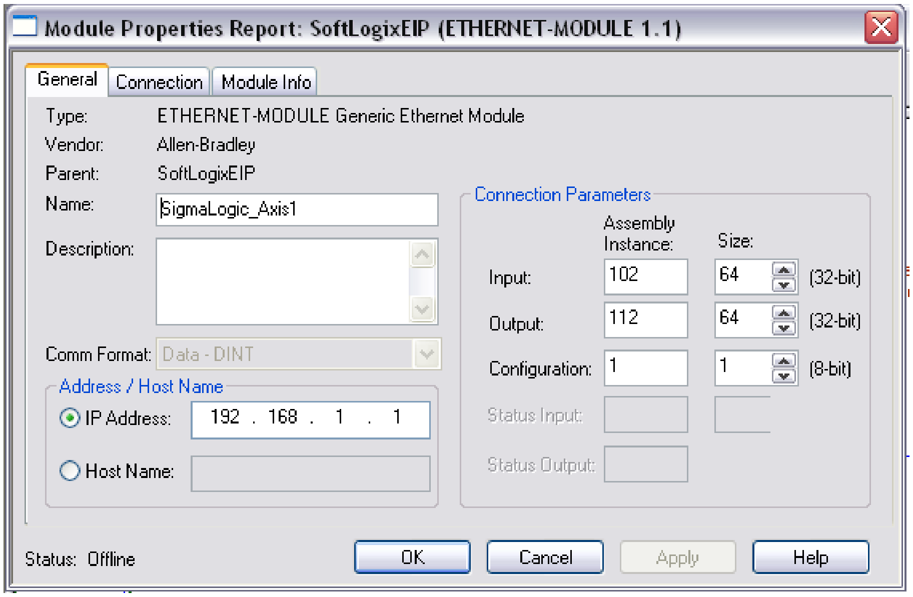
This application note assumes that the SigmaLogic has already been configured with LogicWorks and has been added to the RSLogix project. Yaskawa has produced several QuickStart videos available on Yaskawa's website and on YouTube. These QuickStart videos walk the user through setting up the SigmaLogic amplifier via LogicWorks, adding SigmaLogic to RSLogix and importing and using some of the popular AOIs. When searching these videos on Yaskawa's website, type "eLV.SigmaLogic" into the site search window. This will produce three QuickStart videos and any other eLearning Videos that Yaskawa has produced for SigmaLogic. The following link should bring up the search results:

https://www.yaskawa.com/pycsearch?keywords=eLV.SigmaLogic*&selCollection=Entire%20Site

These same videos are also published to Yaskawa's YouTube channel. Typing "SigmaLogic" into a YouTube search will bring up several results including a playlist of training videos. The following link should bring up the SigmaLogic training section:

https://www.youtube.com/watch?v=pkm3wVTe2-U&list=PLNAENIyEDCkwyKty6WKoekdBmELwBniQ9

From the RSLogix point of view, SigmaLogic is added as a Generic Ethernet Module with the following settings:



Module Properties Report: SoftLogixEIP (ETHERNET-MODULE 1.1)

Type: ETHERNET-MODULE Generic Ethernet Module

Vendor: Allen-Bradley

Parent: SoftLogixEIP

Name: SigmaLogic_Axis1

Description:

Comm Format: Data - DINT

Address / Host Name

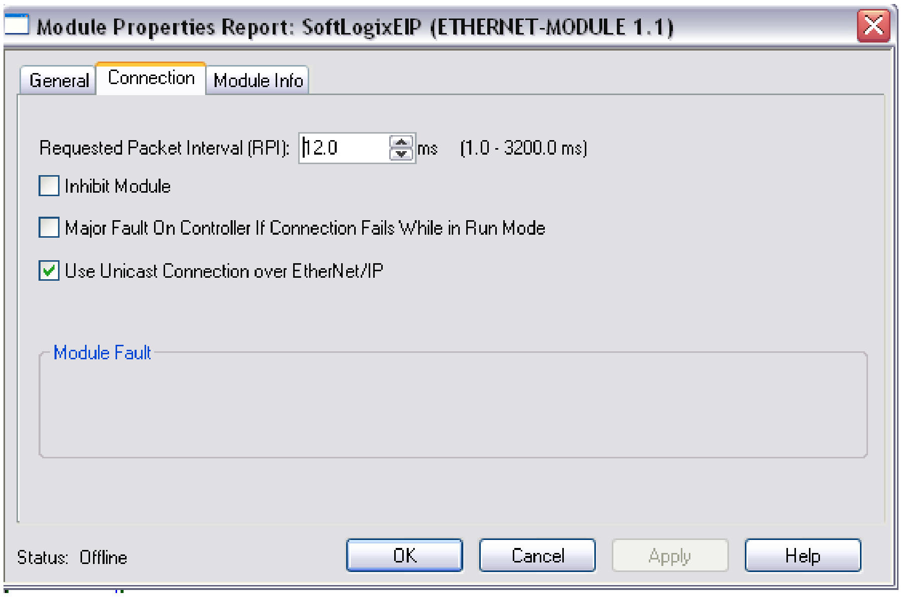
● IP Address: 192 . 168 . 1 . 1

○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	102	64	(32-bit)
Output:	112	64	(32-bit)
Configuration:	1	1	(8-bit)
Status Input:			
Status Output:			

Status: Offline



Module Properties Report: SoftLogixEIP (ETHERNET-MODULE 1.1)

Requested Packet Interval (RPI): 12.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

☑ Use Unicast Connection over EtherNet/IP

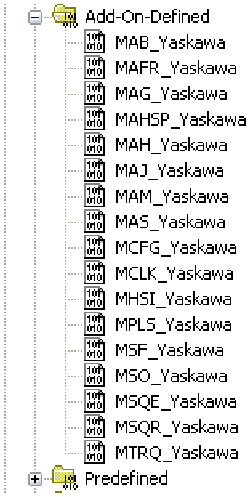
Module Fault

Status: Offline

All Add-On Instructions for RSLogix 5000 can be downloaded from Yaskawa's website at:

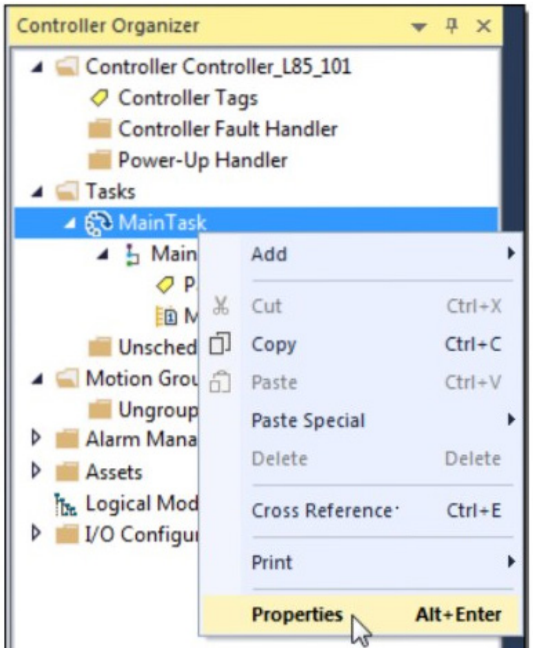
www.yaskawa.com/sigmalogic

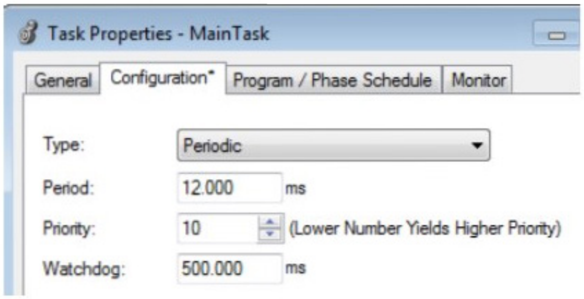
This will be a zip file containing all available AOIs and example code. QuickStart video #2 (Connection) guides the user through importing the AOIs into RSLogix. Once imported, they will show up in the Add-On-Defined Folder in the Controller Organizer.



- Add-On-Defined
  - MAB_Yaskawa
  - MAFR_Yaskawa
  - MAG_Yaskawa
  - MAHSP_Yaskawa
  - MAH_Yaskawa
  - MAJ_Yaskawa
  - MAM_Yaskawa
  - MAS_Yaskawa
  - MCFG_Yaskawa
  - MCLK_Yaskawa
  - MHSI_Yaskawa
  - MPLS_Yaskawa
  - MSF_Yaskawa
  - MSO_Yaskawa
  - MSQE_Yaskawa
  - MSQR_Yaskawa
  - MTRQ_Yaskawa
- Predefined

It is recommended to run the AOI's in a Periodic Task that is a multiple of 12ms so that it matches the RPI of EIP communications.





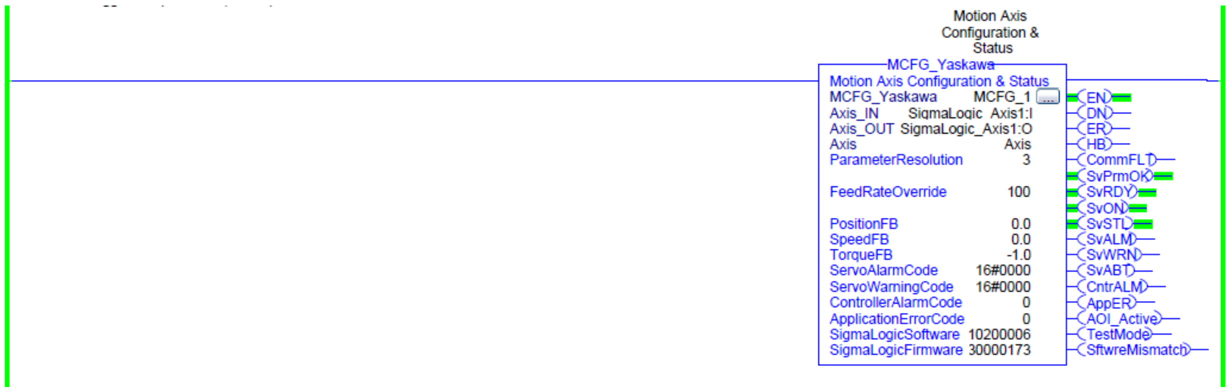
Type: Periodic

Period: 12.000 ms

Priority: 10 (Lower Number Yields Higher Priority)

Watchdog: 500.000 ms

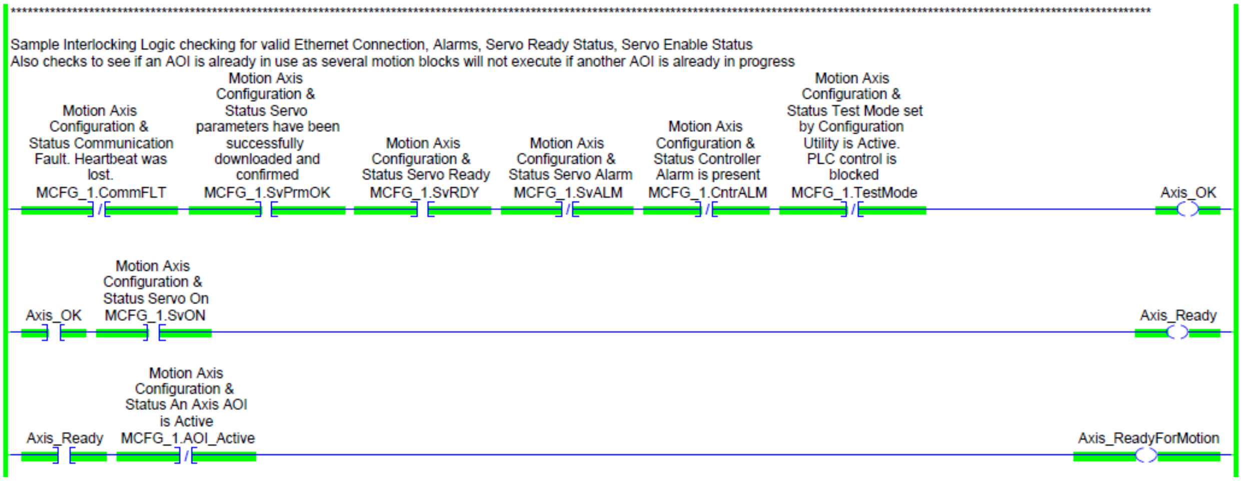
4. MCFG_Yaskawa (Configuration & Status)



```
                Motion Axis
              Configuration &
                   Status
 ---------------MCFG_Yaskawa---------------
 Motion Axis Configuration & Status
 MCFG_Yaskawa           MCFG_1 ...        -(EN)-
 Axis_IN       SigmaLogic  Axis1:I        -(DN)-
 Axis_OUT  SigmaLogic_Axis1:O             -(ER)-
 Axis                     Axis            -(HB)-
 ParameterResolution         3            -(CommFLT)-
                                          -(SvPrmOK)-
 FeedRateOverride          100            -(SvRDY)-
                                          -(SvON)-
 PositionFB                0.0            -(SvSTL)-
 SpeedFB                   0.0            -(SvALM)-
 TorqueFB                 -1.0            -(SvWRN)-
 ServoAlarmCode       16#0000             -(SvABT)-
 ServoWarningCode     16#0000             -(CntrALM)-
 ControllerAlarmCode         0            -(AppER)-
 ApplicationErrorCode        0            -(AOI_Active)-
 SigmaLogicSoftware 10200006              -(TestMode)-
 SigmaLogicFirmware 30000173              -(SftwreMismatch)-
```

MCFG_Yaskawa must be included in the RSLogix program in order to use SigmaLogic and should be always be enabled. The Axis_IN and Axis_OUT parameters link the AOI to the physical generic Ethernet module that was configured. The Axis parameter creates a User-Defined DataType called Yaskawa_EIP_Servo. The Axis structure is then used in all subsequent AOIs. For more information regarding the information available in the Axis structure, please refer to Appendix A&B in this document. There is a wealth of useful information that exists in this structure and is updated automatically as long as the MCFG_Yaskawa AOI is enabled.

5. Interlocks



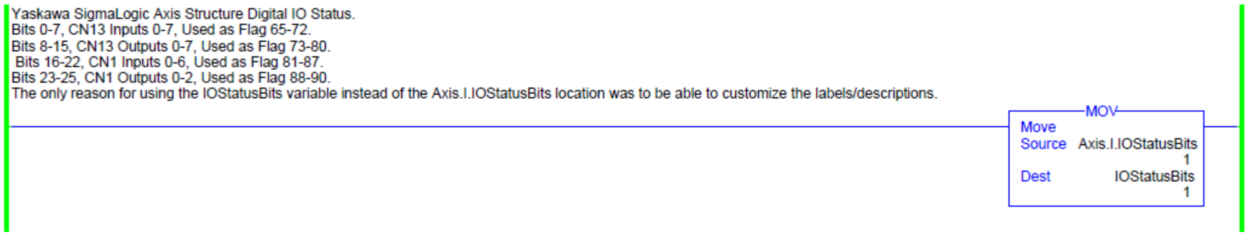
****

Sample Interlocking Logic checking for valid Ethernet Connection, Alarms, Servo Ready Status, Servo Enable Status
Also checks to see if an AOI is already in use as several motion blocks will not execute if another AOI is already in progress

Motion Axis Configuration & Status Communication Fault. Heartbeat was lost.
MCFG_1.CommFLT
Motion Axis Configuration & Status Servo parameters have been successfully downloaded and confirmed
MCFG_1.SvPrmOK
Motion Axis Configuration & Status Servo Ready
MCFG_1.SvRDY
Motion Axis Configuration & Status Servo Alarm
MCFG_1.SvALM
Motion Axis Configuration & Status Controller Alarm is present
MCFG_1.CntrALM
Motion Axis Configuration & Status Test Mode set by Configuration Utility is Active. PLC control is blocked
MCFG_1.TestMode
Axis_OK
Axis_OK
Motion Axis Configuration & Status Servo On
MCFG_1.SvON
Axis_Ready
Axis_Ready
Motion Axis Configuration & Status An Axis AOI is Active
MCFG_1.AOI_Active
Axis_ReadyForMotion

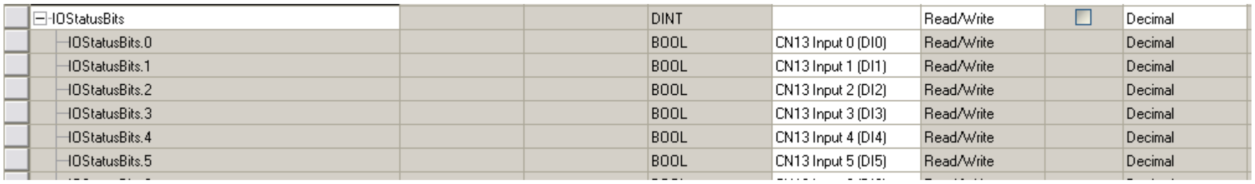
This section of logic was written to provide three Boolean variables for use further down in the code. Axis_OK will be TRUE if the SigmaLogic axis has good communication, the amplifier has power and no alarms. Axis_Ready further checks the servo enable status and will be TRUE when the servo motor is enabled. Axis_ReadyForMotion takes it one step further and makes sure that no AOI is already in progress. Most of the motion-related AOIs cannot be executed if another AOI is active so Axis_ReadyForMotion is used often to ensure that the conditions are acceptable to enable a new AOI.

6. SigmaLogic Input/Output Status

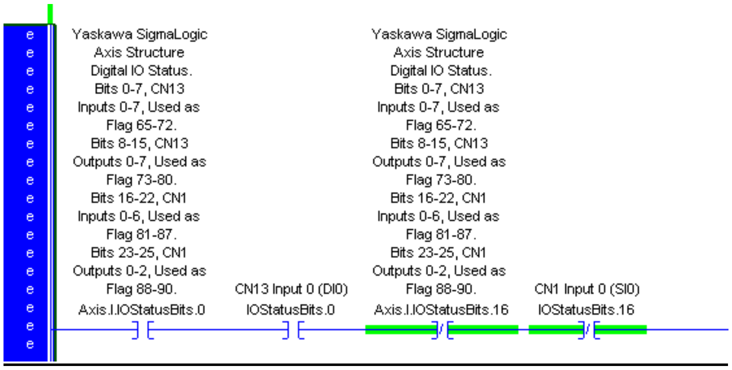


Yaskawa SigmaLogic Axis Structure Digital IO Status.
Bits 0-7, CN13 Inputs 0-7, Used as Flag 65-72.
Bits 8-15, CN13 Outputs 0-7, Used as Flag 73-80.
Bits 16-22, CN1 Inputs 0-6, Used as Flag 81-87.
Bits 23-25, CN1 Outputs 0-2, Used as Flag 88-90.
The only reason for using the IOStatusBits variable instead of the Axis.I.IOStatusBits location was to be able to customize the labels/descriptions.

This part of the code is completely optional but can provide a good way to customize the descriptions of any Inputs or Outputs used with SigmaLogic. The status of all of the Digital Input and Output points available on the SigmaLogic controller (CN13) and the amplifier (CN1) are stored in Axis.I.IOStatusBits. This rung simply copies this status into a DINT variable called IOStatusBits. The description for each bit of IOStatusBits can now be customized for display in ladder.



⊟-IOStatusBits			DINT		Read/Write	□	Decimal
IOStatusBits.0			BOOL	CN13 Input 0 (DI0)	Read/Write		Decimal
IOStatusBits.1			BOOL	CN13 Input 1 (DI1)	Read/Write		Decimal
IOStatusBits.2			BOOL	CN13 Input 2 (DI2)	Read/Write		Decimal
IOStatusBits.3			BOOL	CN13 Input 3 (DI3)	Read/Write		Decimal
IOStatusBits.4			BOOL	CN13 Input 4 (DI4)	Read/Write		Decimal
IOStatusBits.5			BOOL	CN13 Input 5 (DI5)	Read/Write		Decimal



Yaskawa SigmaLogic Axis Structure Digital IO Status. Bits 0-7, CN13 Inputs 0-7, Used as Flag 65-72. Bits 8-15, CN13 Outputs 0-7, Used as Flag 73-80. Bits 16-22, CN1 Inputs 0-6, Used as Flag 81-87. Bits 23-25, CN1 Outputs 0-2, Used as Flag 88-90.
Axis.I.IOStatusBits.0

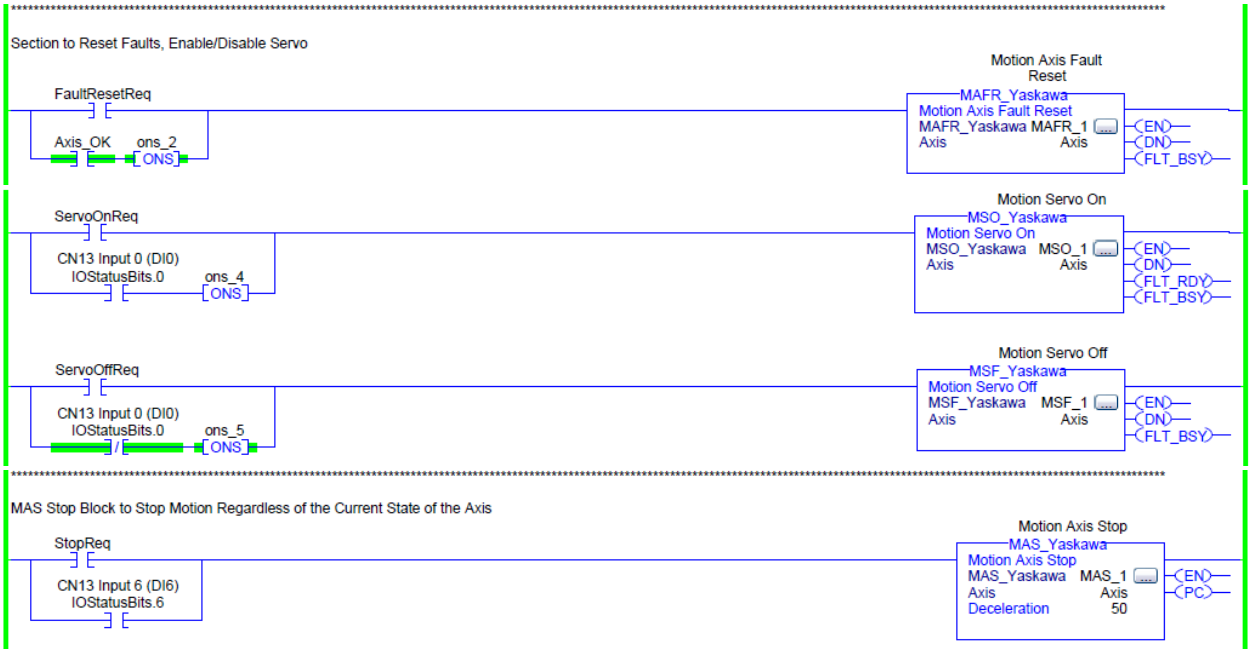
CN13 Input 0 (DI0)
IOStatusBits.0

Yaskawa SigmaLogic Axis Structure Digital IO Status. Bits 0-7, CN13 Inputs 0-7, Used as Flag 65-72. Bits 8-15, CN13 Outputs 0-7, Used as Flag 73-80. Bits 16-22, CN1 Inputs 0-6, Used as Flag 81-87. Bits 23-25, CN1 Outputs 0-2, Used as Flag 88-90.
Axis.I.IOStatusBits.16

CN1 Input 0 (SI0)
IOStatusBits.16

The picture above shows the differences in descriptions between Axis.I.IOStatus and IOStatus bits even though they display the same status.

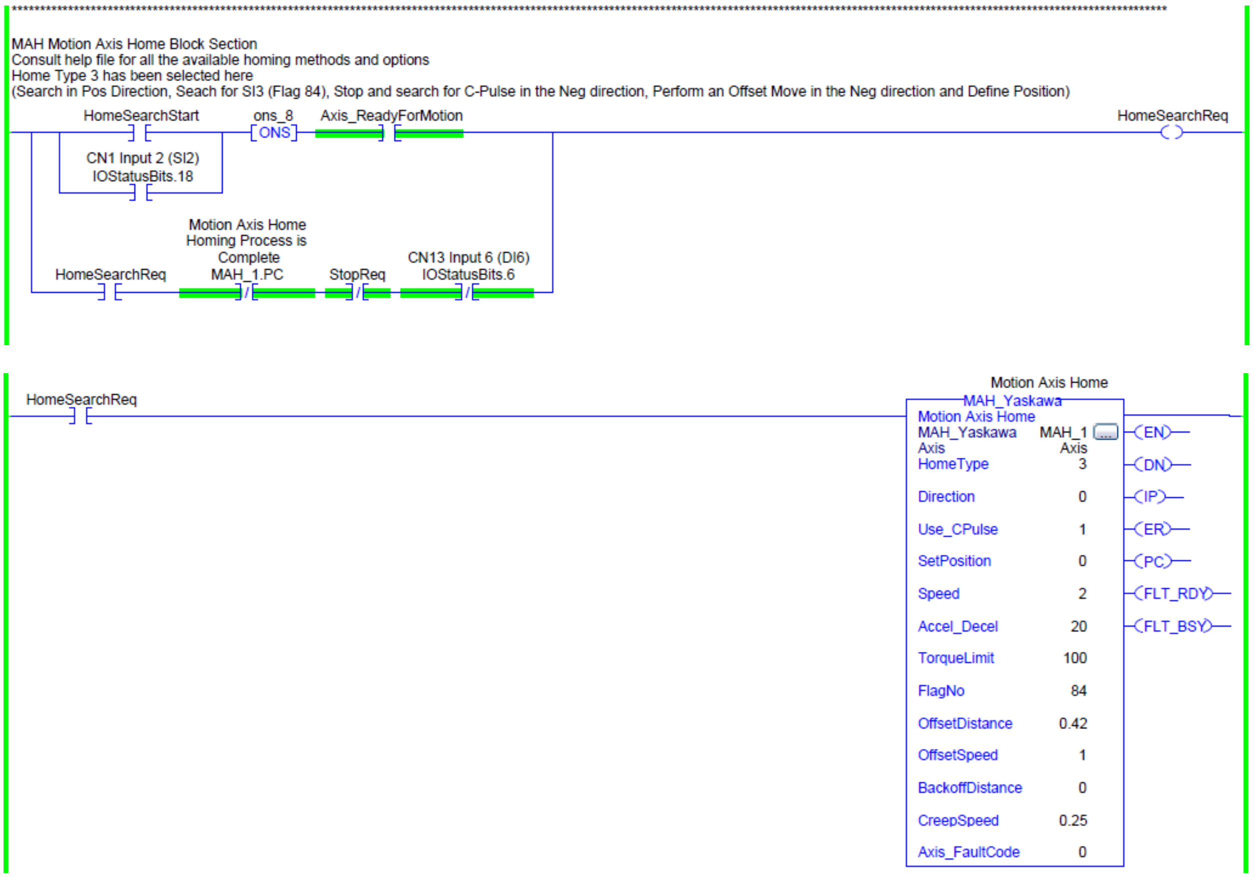
7. MAFR_Yaskawa (Fault Reset), MSO_Yaskawa (Servo On), MSF_Yaskawa (Servo Off), MAS_Yaskawa (Stop)


Section to Reset Faults, Enable/Disable Servo

MAS Stop Block to Stop Motion Regardless of the Current State of the Axis

The Fault Reset (MAFR_Yaskawa), Servo On (MSO_Yaskawa), Servo Off (MSF_Yaskawa) and Stop (MAS_Yaskawa) AOIs are pretty straightforward and do not require much in the way of interlocking. The Stop and Fault Reset can all be executed even if other AOIs are in progress.

8. MAH_Yaskawa (Homing)



```
****************************************************************************************************
MAH Motion Axis Home Block Section
Consult help file for all the available homing methods and options
Home Type 3 has been selected here
(Search in Pos Direction, Seach for SI3 (Flag 84), Stop and search for C-Pulse in the Neg direction, Perform an Offset Move in the Neg direction and Define Position)
```

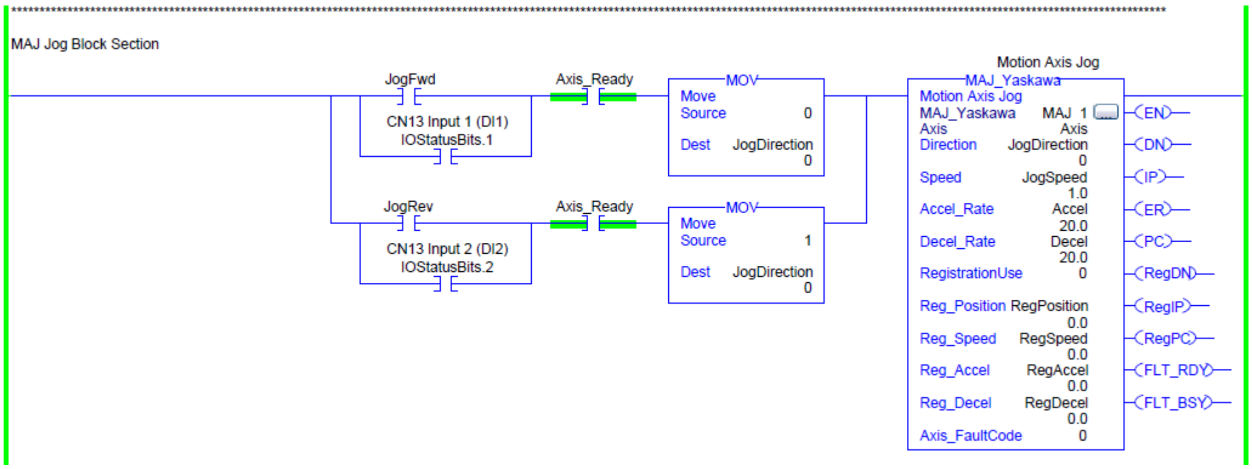
Motion Axis Home MAH_Yaskawa	
Motion Axis Home MAH_Yaskawa Axis	MAH_1 Axis
HomeType	3
Direction	0
Use_CPulse	1
SetPosition	0
Speed	2
Accel_Decel	20
TorqueLimit	100
FlagNo	84
OffsetDistance	0.42
OffsetSpeed	1
BackoffDistance	0
CreepSpeed	0.25
Axis_FaultCode	0

To start the homing search, the code is looking for the rising edge of either the HomeSearchStart bit or Input SI2 from the amplifier I/O. As long as Axis_ReadyForMotion is also true, the HomeSearchReq bit will be latched in. It will hold the enable on to the MAH AOI. This will stay latched until the home search is complete (PC) or a stop has been issued. This type of latched enable signal is used for most of the motion AOIs.

The MAH_Yaskawa AOI has many different options and configurations which define the type of home search being performed. The following table explains the different AOI inputs and resulting home search routines:

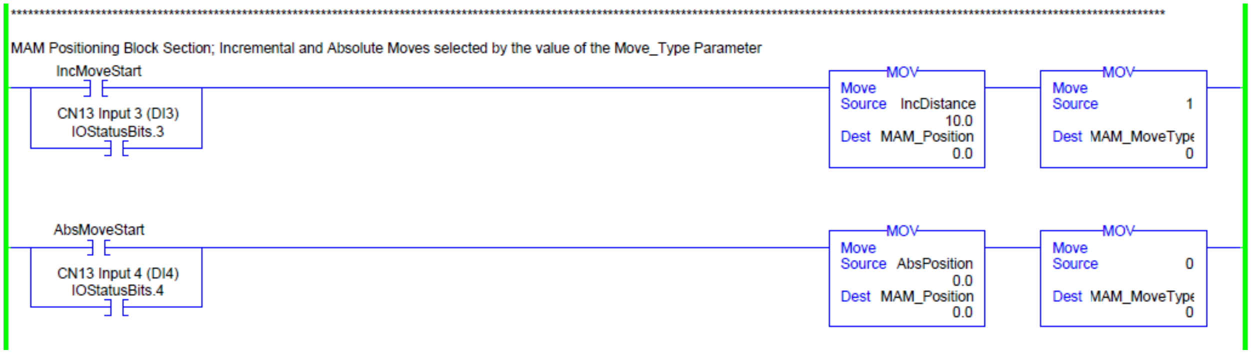
AOI Inputs			
HomeType	**Direction**	**Use_Cpulse**	**Description**
0	N/A	N/A	Set Position Directly
1	0	N/A	Home in Positive Direction to Hard Stop
1	1	N/A	Home in Negative Direction to Hard Stop
2	0	0	Home in Positive Direction to Limit Switch
2	1	0	Home in Negative Direction to Limit Switch
2	0	1	Home in Positive Direction to Limit Switch with C-Pulse
2	1	1	Home in Negative Direction to Limit Switch with C-Pulse
3	0	0	Home in Positive Direction to Flag
3	1	0	Home in Negative Direction to Flag
3	0	1	Home in Positive Direction to Flag with C-Pulse
3	1	1	Home in Negative Direction to Flag with C-Pulse

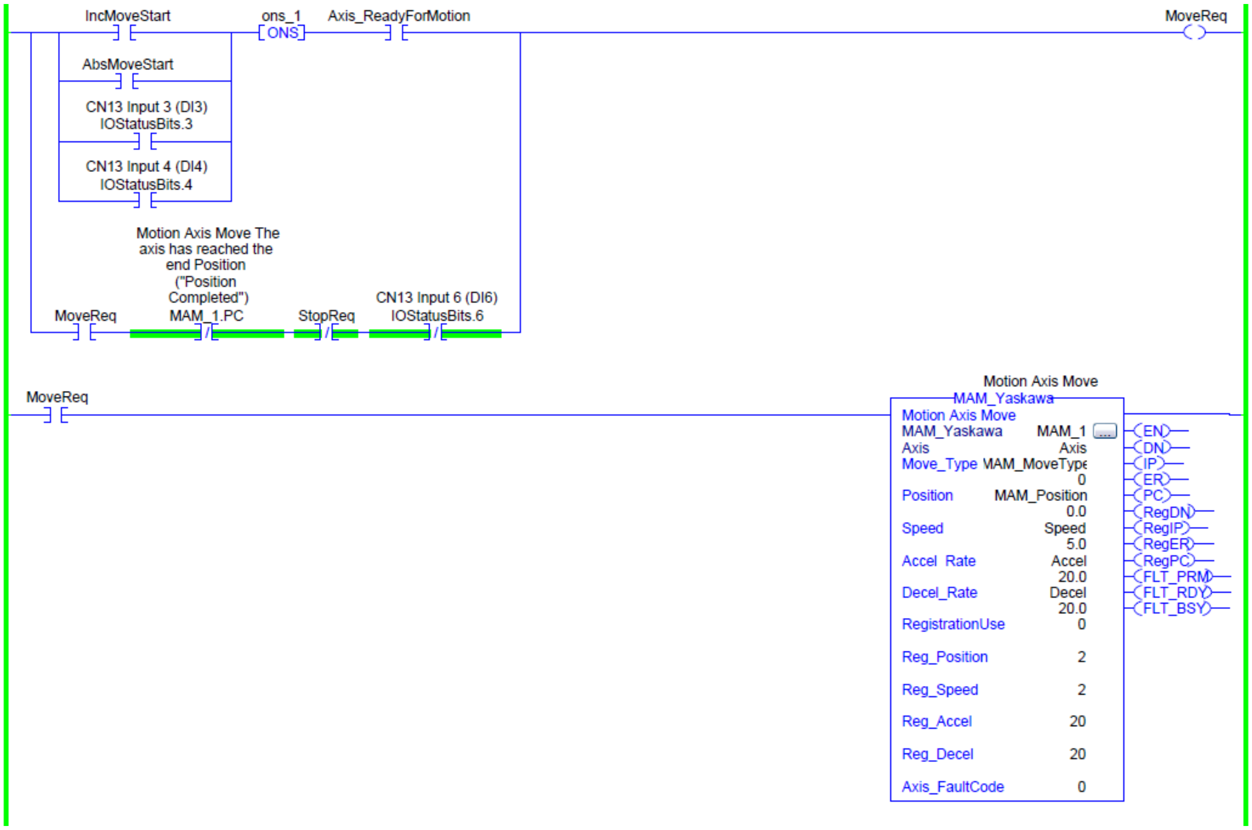
9. MAJ_Yaskawa (Jog, Jog w/ Registration)



The MAJ AOI jogs the motor forward or reverse depending on the Direction input. As long as the enable is active, the motor will be jogging. When the enable turns off, the motor decelerates at the Decel_Rate to a stop. If RegistrationUse is set to true, then the jog move will be interrupted and SigmaLogic performs a relative move when the registration latch input (Amplifier Input SI4) is activated. After a successful registration move has been completed, the enable must by cycled to start jogging once again. If the registration latch input is not activated, the motor will continue to jog.

10. MAM_Yaskawa (Move)

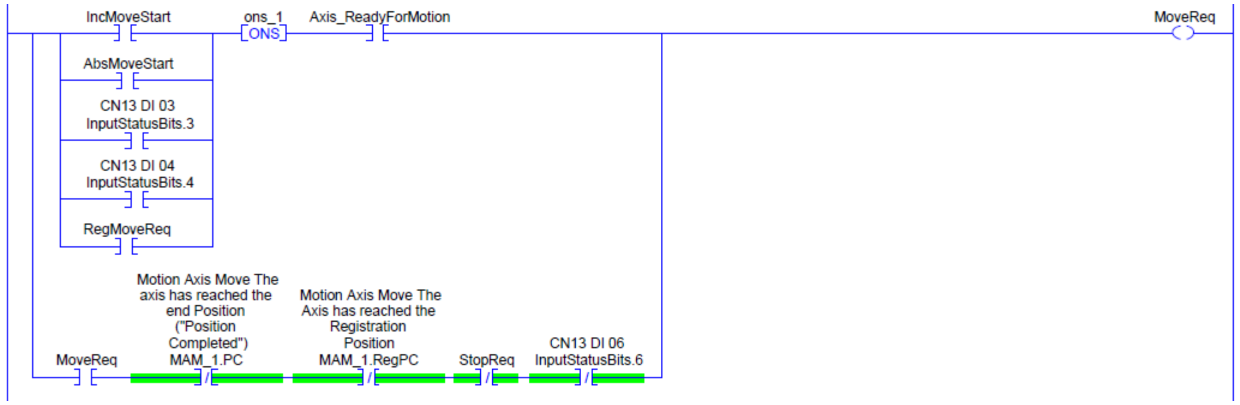




The MAM AOI performs either an incremental index move or a move to an absolute position. The Move_Type input dictates which type of move is commanded. From the code above, the Position and Move_Type inputs are loaded based on whether an incremental or absolute move is commanded. If Axis_ReadyForMotion is true, a rising edge of either

IncMoveStart, AbsMoveStart, Input 3 or Input 4 will latch in the MoveReq bit. This bit hold on the enable for the MAM AOI until the move is complete (PC) or a stop has been issued.

If RegistrationUse is set to true, then the move will be interrupted and SigmaLogic performs a relative move when the registration latch input (Amplifier Input SI4) is activated. If a registration move is triggered, then the RegIP output will be true and the RegPC output will be true once the registration move has completed. To use the same latching logic for the MAM enable when registration is used, the RegPC signal should also be added to logic as shown below.

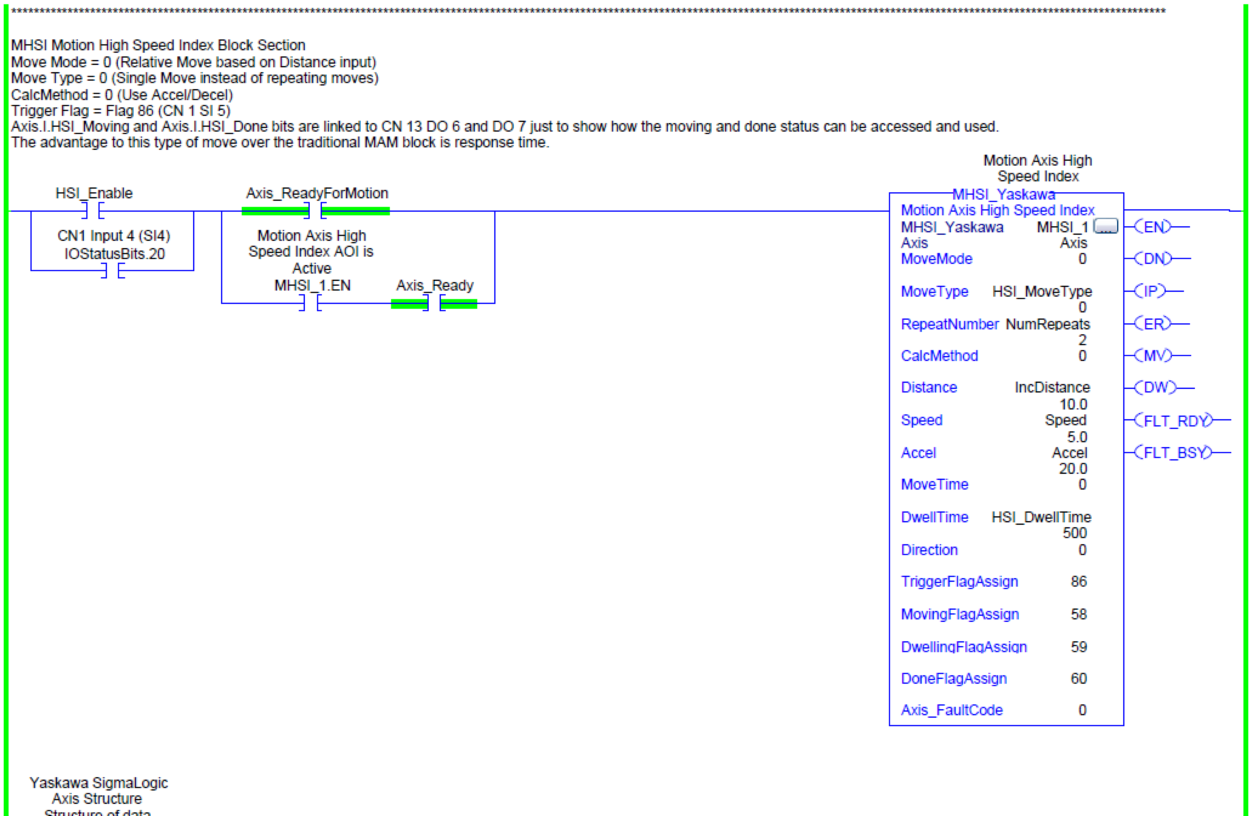


If the registration latch input is not activated, the motor will complete the original incremental or absolute index move and the PC output will turn on.

MAM Type	Direction	Description
0	N/A	Linear Absolute
1	Positive / Negative (-)	Incremental Relative
2	Shortest Path	Rotary Absolute
3	Positive	Rotary Absolute
4	Negative	Rotary Absolute

11. MHSI_Yaskawa (High Speed Index)

The MHSI AOI is used primarily when the delays of the Ethernet/IP communication cycle could impact the desired performance. The MHSI function reacts to a physical input to start the move instead of a command over EIP which represents the fastest way to trigger motion. Additionally, if the profile is a repeating profile with a move and a dwell, the MSHI AOI would give the user the fastest response and save programming effort creating logic with MAM, TON and counters.



```
*******************************************************************************************************************************************************
MHSI Motion High Speed Index Block Section
Move Mode = 0 (Relative Move based on Distance input)
Move Type = 0 (Single Move instead of repeating moves)
CalcMethod = 0 (Use Accel/Decel)
Trigger Flag = Flag 86 (CN 1 SI 5)
Axis.I.HSI_Moving and Axis.I.HSI_Done bits are linked to CN 13 DO 6 and DO 7 just to show how the moving and done status can be accessed and used.
The advantage to this type of move over the traditional MAM block is response time.
```

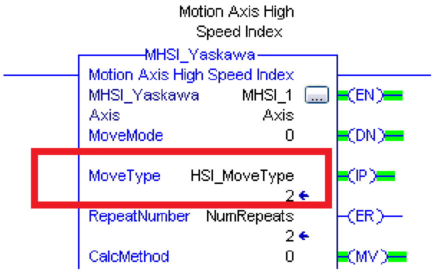
Motion Axis High Speed Index
MHSI_Yaskawa
Motion Axis High Speed Index

MHSI_Yaskawa	MHSI_1	
Axis	Axis	
MoveMode	0	
MoveType	HSI_MoveType 0	
RepeatNumber	NumRepeats 2	
CalcMethod	0	
Distance	IncDistance 10.0	
Speed	Speed 5.0	
Accel	Accel 20.0	
MoveTime	0	
DwellTime	HSI_DwellTime 500	
Direction	0	
TriggerFlagAssign	86	
MovingFlagAssign	58	
DwellingFlagAssign	59	
DoneFlagAssign	60	
Axis_FaultCode	0	

The code above uses a signal (HSI_Enable or SI4) to enable the MHSI AOI. Using MHSI is a two-step process. It must be enabled for the trigger flag to start the index. There are many ways to adjust the move performed based on the inputs MoveMode, MoveType and CalcMethod.

The embedded help file for the MHSI_Yaskawa AOI describes these parameters in detail. The example shown on the previous page performs a relative move of 10 revs when the rising edge of the trigger flag (86 or CN1 SI5) is activated. The trigger flag can be assigned to any other physical Input signal from CN1 or CN13 by modifying the flag number.

If the MHSI parameter MoveType changes to a value of 2, then the trigger flag would start a repeating move for two cycles. A relative move of 10 revs followed by a dwell of 500ms would be repeated twice. For any repeating moves to run, the trigger flag must stay ON. If the trigger flag pulses, then the MHSI will perform one move and one dwell before finishing.



Motion Axis High Speed Index

MHSI_Yaskawa

Motion Axis High Speed Index

MHSI_Yaskawa MHSI_1

Axis Axis

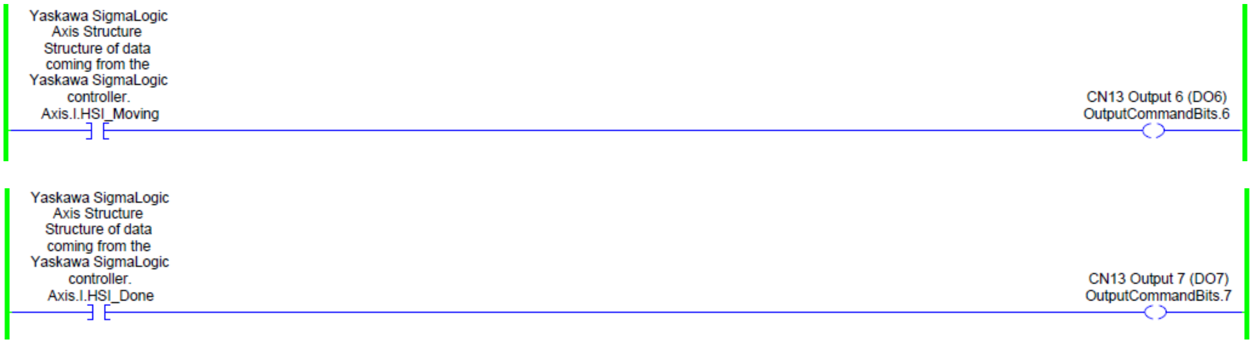
MoveMode 0

MoveType HSI_MoveType 2

RepeatNumber NumRepeats 2

CalcMethod 0

There are also two additional rungs used with this MHSI AOI example. These show how a programmer might choose to use the AOI status that is updated automatically and stored in the Axis structure. Each AOI has similar status information in the Axis structure. The code below turns on Output 6 when the MHSI index is in progress and turns ON Output 7 when the MHSI index is complete.



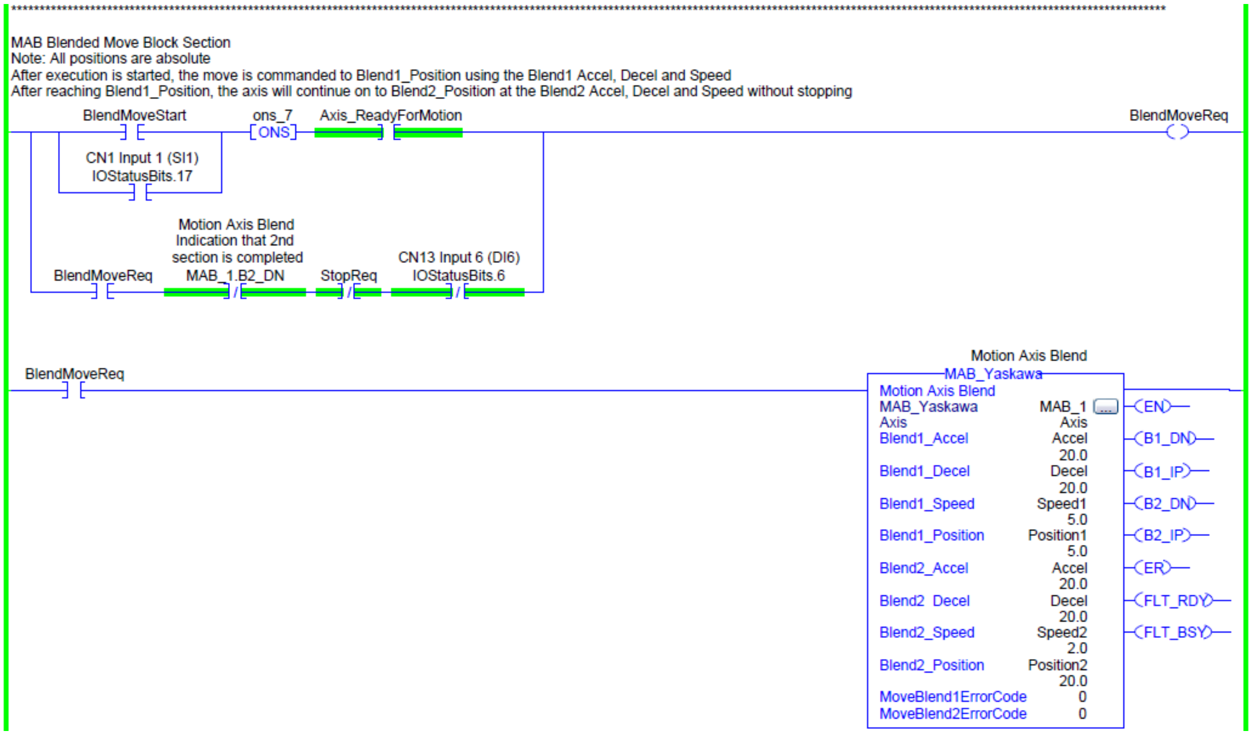
Yaskawa SigmaLogic Axis Structure
Structure of data coming from the Yaskawa SigmaLogic controller.
Axis.I.HSI_Moving

CN13 Output 6 (DO6)
OutputCommandBits.6

Yaskawa SigmaLogic Axis Structure
Structure of data coming from the Yaskawa SigmaLogic controller.
Axis.I.HSI_Done

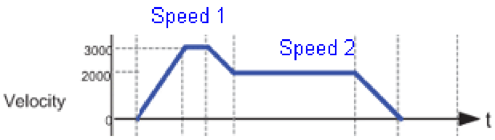
CN13 Output 7 (DO7)
OutputCommandBits.7

12. MAB_Yaskawa (Motion Axis Blend)



```
**************************************************************************************************************
MAB Blended Move Block Section
Note: All positions are absolute
After execution is started, the move is commanded to Blend1_Position using the Blend1 Accel, Decel and Speed
After reaching Blend1_Position, the axis will continue on to Blend2_Position at the Blend2 Accel, Decel and Speed without stopping
```

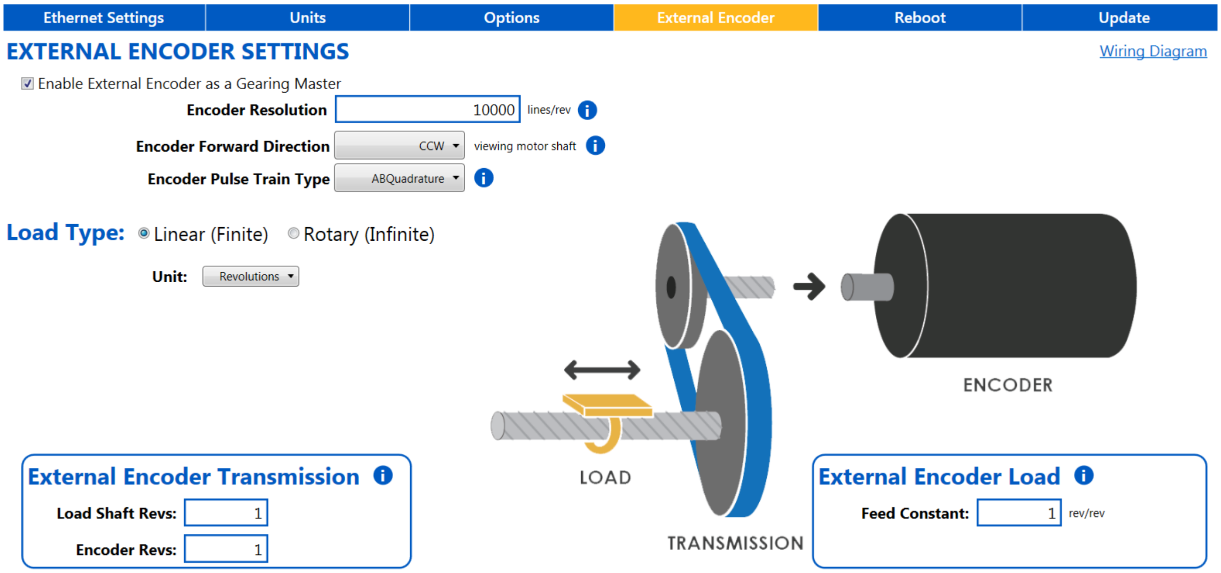
The MAB_Yaskawa AOI performs a two-stage absolute index. This is typically used when a move profile requires different speeds at the beginning and the end of the move. The position inputs to the AOI are absolute positions so the motor position when the MAB AOI is enabled will dictate motor direction.



The picture above shows a sample move profile where the motor moves to Position1 at Speed1 and then decelerates down to Speed2 to finish moving to the final end position (Position2).

13. MAG_Yaskawa (Electronic Gearing)

For the MAG_Yaskawa AOI to work properly, the external encoder must be set up first with LogicWorks. To avoid any errors in the MAG AOI, make sure that the “Enable External Encoder as a Gearing Master” is checked. Below is an example setup:



EXTERNAL ENCODER SETTINGS

☑ Enable External Encoder as a Gearing Master

Encoder Resolution: 10000 lines/rev

Encoder Forward Direction: CCW viewing motor shaft

Encoder Pulse Train Type: ABQuadrature

Load Type: ● Linear (Finite) ○ Rotary (Infinite)

Unit: Revolutions

External Encoder Transmission

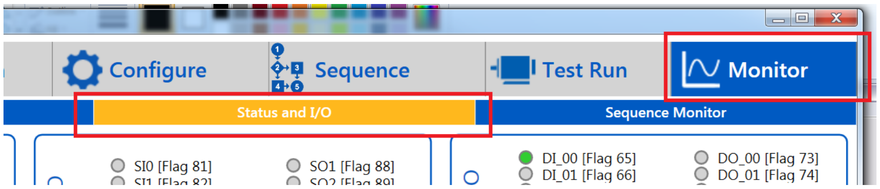
Load Shaft Revs: 1

Encoder Revs: 1

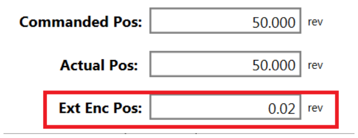
External Encoder Load

Feed Constant: 1 rev/rev

It is also recommended to check the feedback and scaling by going to the Monitor -> Status and I/O section.



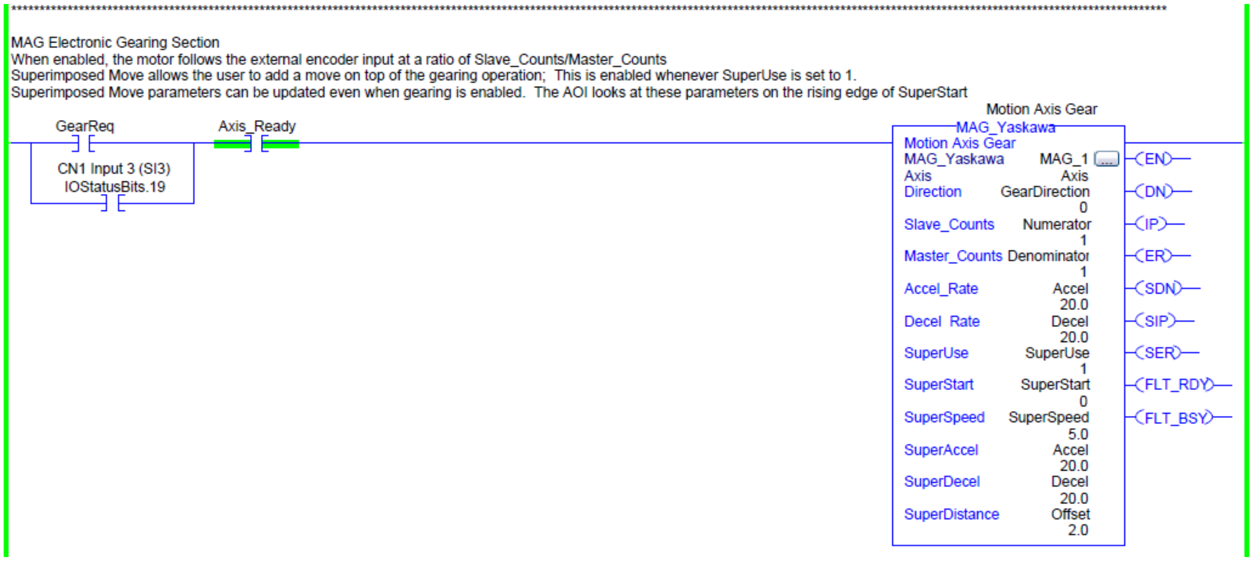
In the bottom left corner, it is possible to see the current external encoder position



Commanded Pos: 50.000 rev

Actual Pos: 50.000 rev

Ext Enc Pos: 0.02 rev



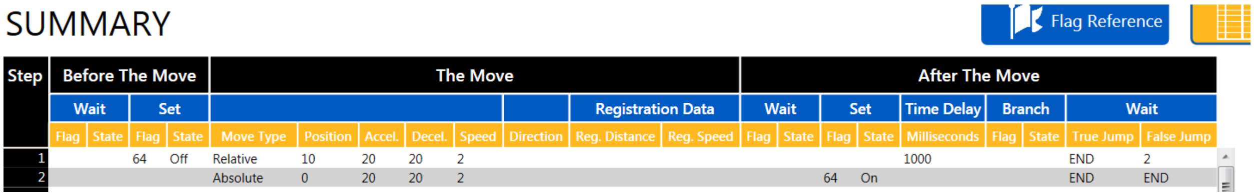
```
MAG Electronic Gearing Section
When enabled, the motor follows the external encoder input at a ratio of Slave_Counts/Master_Counts
Superimposed Move allows the user to add a move on top of the gearing operation;  This is enabled whenever SuperUse is set to 1.
Superimposed Move parameters can be updated even when gearing is enabled.  The AOI looks at these parameters on the rising edge of SuperStart
```

```
Motion Axis Gear
MAG_Yaskawa
Motion Axis Gear
MAG_Yaskawa         MAG_1
Axis                 Axis
Direction       GearDirection
                        0
Slave_Counts      Numerator
                        1
Master_Counts   Denominator
                        1
Accel_Rate            Accel
                     20.0
Decel  Rate           Decel
                     20.0
SuperUse           SuperUse
                        1
SuperStart       SuperStart
                        0
SuperSpeed       SuperSpeed
                      5.0
SuperAccel            Accel
                     20.0
SuperDecel            Decel
                     20.0
SuperDistance        Offset
                      2.0
```

When the MAG_Yaskawa AOI is enabled, the SigmaLogic axis follows the external encoder wired into CN13. While the encoder resolution and scaling is part of the LogicWorks setup shown on the previous page, the gearing ratio is adjustable from RSLogix via the Slave_Counts and Master_Counts inputs. MAG also allows for a superimposed move to be commanded on top of the gearing relationship. This is typically used for phase shifting, registration corrections or other offsets. When the enable to MAG is turned off, the motor will decelerate at the Decel_Rate to a stop.

14. MSQR & MSQE (Sequence Run and Edit)

The MSQR_Yaskawa and MSQE_Yaskawa AOIs are used to execute and modify pre-programmed sequences in SigmaLogic. Those sequences must be created initially in LogicWorks. The picture below shows a simple two-step sequence:

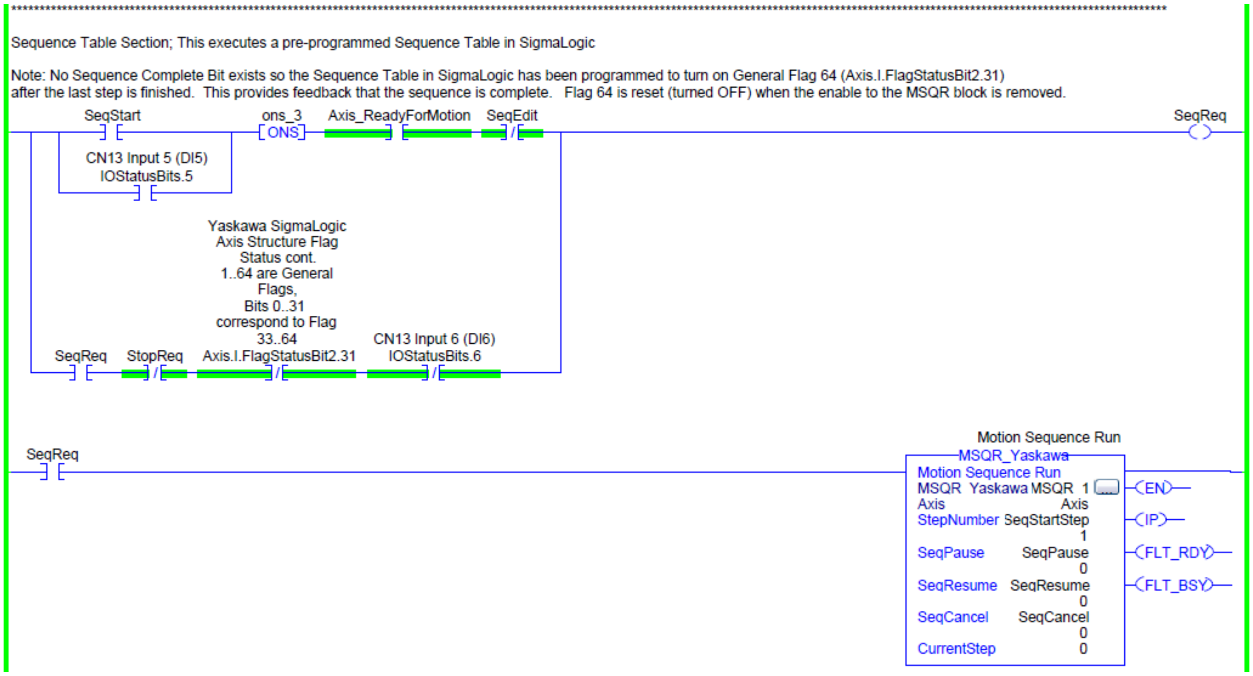


SUMMARY

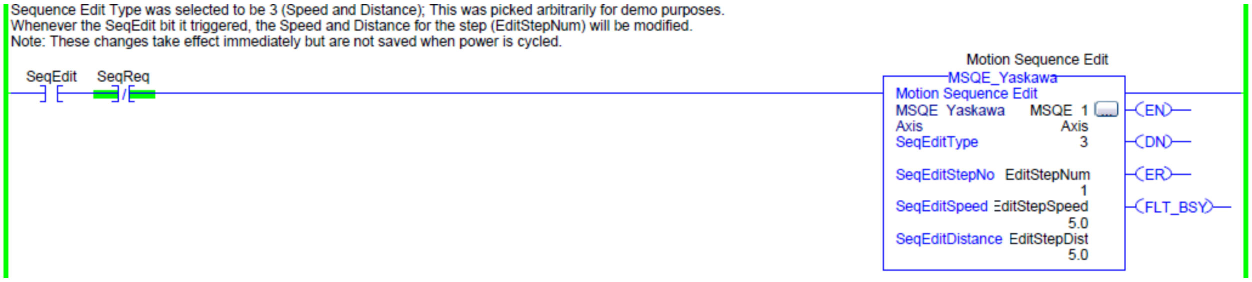
Step	Before The Move				The Move								After The Move								
	Wait		Set								Registration Data		Wait		Set		Time Delay	Branch		Wait	
	Flag	State	Flag	State	Move Type	Position	Accel.	Decel.	Speed	Direction	Reg. Distance	Reg. Speed	Flag	State	Flag	State	Milliseconds	Flag	State	True Jump	False Jump
1			64	Off	Relative	10	20	20	2								1000			END	2
2					Absolute	0	20	20	2						64	On				END	END

When the sequence is started on Step 1, the motor will go 10 revs in the positive direction, wait 1000ms and then return to the zero position. In this example, Flag 64 acts as a sequence complete bit since there is no other bit that would indicate that the sequence has finished. There are 64 user-defined flags for this type of handshaking and Flag 64 was chosen aribitrarily.

The example code on the next page shows one way to execute the sequence starting at Step 1 using the MSQR AOI. Once the rising edge of SeqStart or DI5 is detected, the SeqReq bit is latched in. This will keep the enable to MSQR on until Flag 64 (Axis.I.FlagStatusBit2.31) turns on or the move is aborted via Stop. There is an additional contact that will not allow the sequence to run if a sequence edit is in progress.



```
****************************************************************************************************************************

Sequence Table Section; This executes a pre-programmed Sequence Table in SigmaLogic

Note: No Sequence Complete Bit exists so the Sequence Table in SigmaLogic has been programmed to turn on General Flag 64 (Axis.I.FlagStatusBit2.31)
after the last step is finished.  This provides feedback that the sequence is complete.   Flag 64 is reset (turned OFF) when the enable to the MSQR block is removed.
```

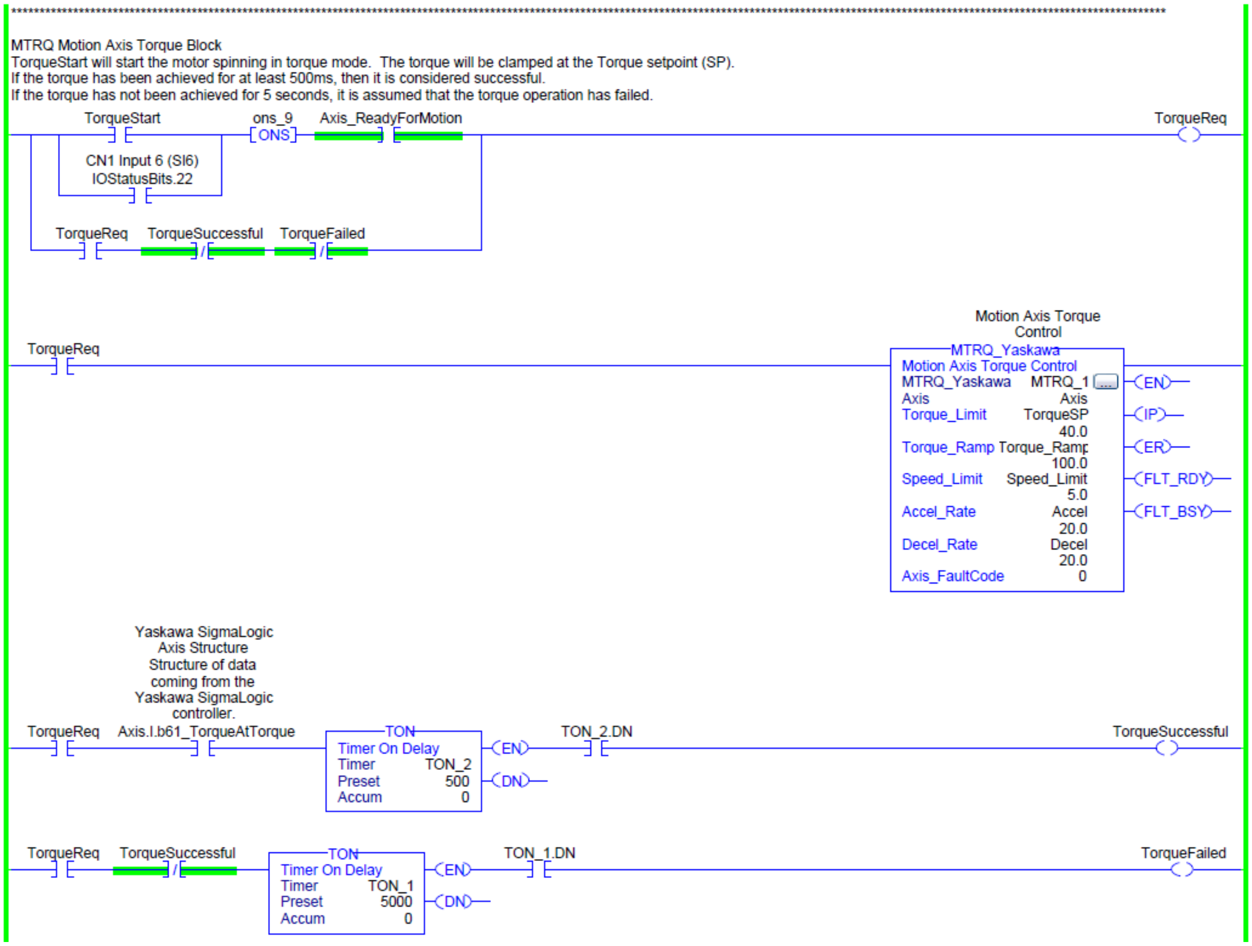


```
Sequence Edit Type was selected to be 3 (Speed and Distance); This was picked arbitrarily for demo purposes.
Whenever the SeqEdit bit it triggered, the Speed and Distance for the step (EditStepNum) will be modified.
Note: These changes take effect immediately but are not saved when power is cycled.
```

The MSQE_Yaskawa AOI provide a way to modify the Speed and Position of an individual step in the sequence table. The code above shows that the speed for Step 1 will be changed to 5 rev/s and the position for Step 1 will be changed to 5 revs if the SeqEdit bit turns on. There is an additional contact preventing the edit from occurring if a sequence is currently running. An important note is that these edits are not retained in SigmaLogic when power is cycled. When SigmaLogic boots up, the sequence table reverts back to the values that were last sent from LogicWorks.

15. MTRQ_Yaskawa (Torque Control)

The following code shows an example of using MTRQ for a demo capping application:

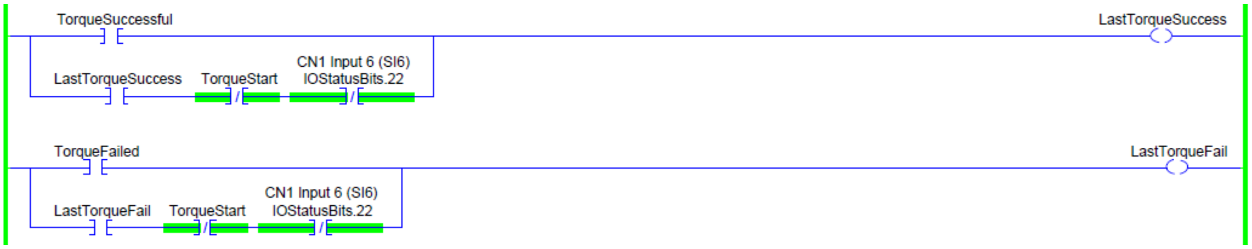


```
**************************************************************************************************************
MTRQ Motion Axis Torque Block
TorqueStart will start the motor spinning in torque mode.  The torque will be clamped at the Torque setpoint (SP).
If the torque has been achieved for at least 500ms, then it is considered successful.
If the torque has not been achieved for 5 seconds, it is assumed that the torque operation has failed.
```

```
MTRQ_Yaskawa
Motion Axis Torque Control
MTRQ_Yaskawa      MTRQ_1
Axis                Axis
Torque_Limit    TorqueSP
                    40.0
Torque_Ramp  Torque_Ramp
                   100.0
Speed_Limit  Speed_Limit
                     5.0
Accel_Rate         Accel
                    20.0
Decel_Rate         Decel
                    20.0
Axis_FaultCode         0
```

```
TON
Timer On Delay
Timer      TON_2
Preset       500
Accum          0
```

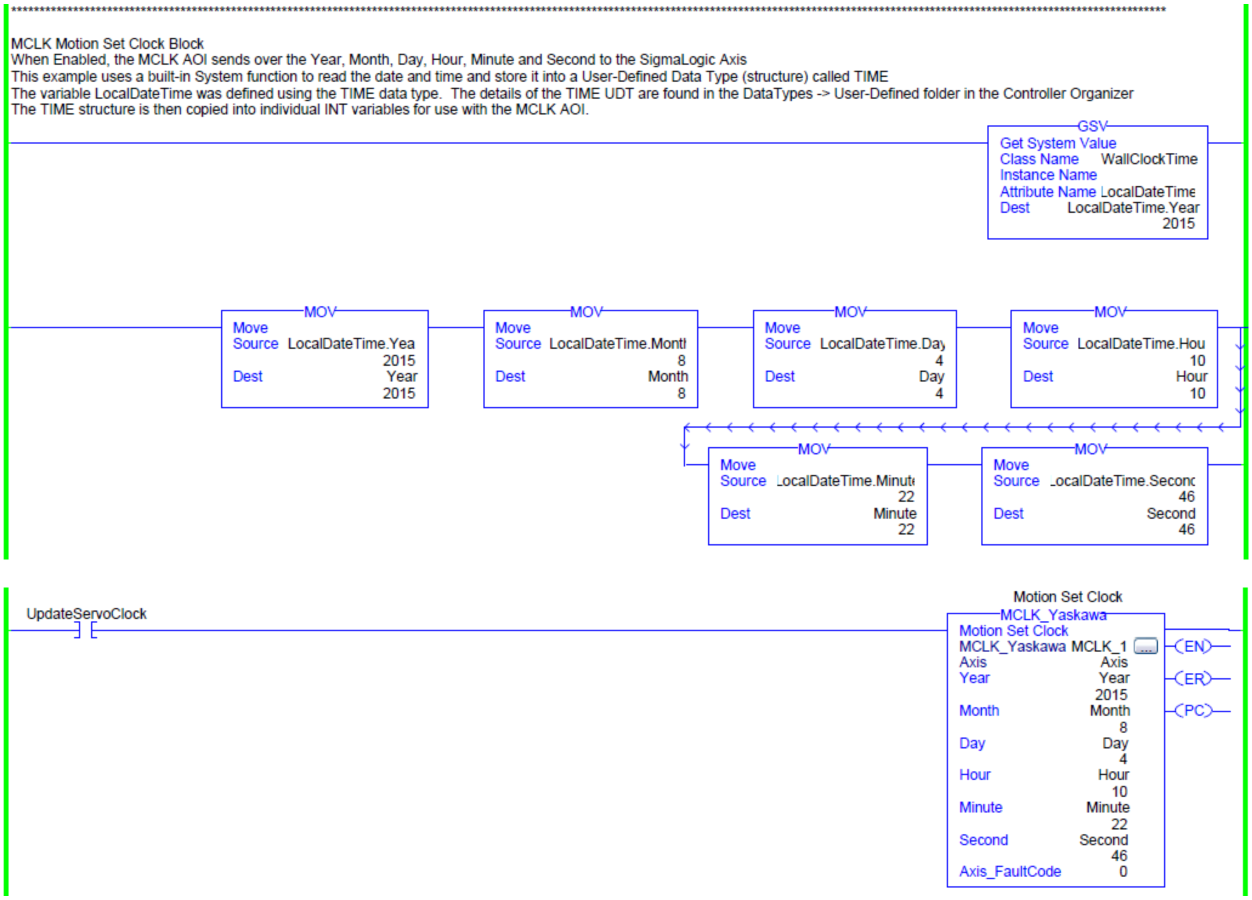
```
TON
Timer On Delay
Timer      TON_1
Preset      5000
Accum          0
```



The MTRQ section of code has been written to show how the MTRQ AOI might be used in an application. The concept is that the motor starts spinning at a commanded torque setpoint. If the actual torque reaches the torque setpoint for at least 500ms, then the operation is considered successful (TorqueSuccessful). If five seconds passes without this condition being met, then the operation is considered a failure (TorqueFailed). Either signal (TorqueSuccessful, TorqueFailed) will unlatch the TorqueReq signal and the motor will decelerate to a stop.

There are many ways to use the MTRQ_Yaskawa AOI, however. The important features are that when enabled, the MTRQ AOI puts the servo in torque mode. The torque is limited by the Torque_Limit input. The speed of the motor during the torque operation will be limited by the Speed_Limit input. The actual speed and torque of the motor will vary depending on the resistance of the load but will not be allowed to exceed the torque or speed limits.

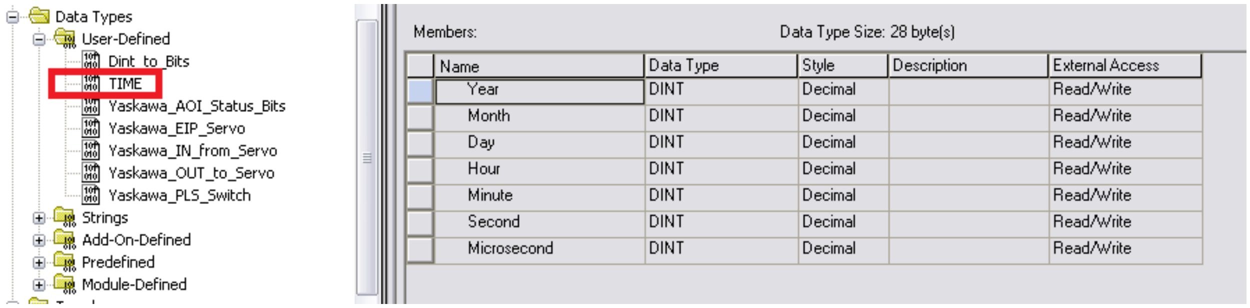
16. MCLK_Yaskawa (Set Clock)



MCLK Motion Set Clock Block
When Enabled, the MCLK AOI sends over the Year, Month, Day, Hour, Minute and Second to the SigmaLogic Axis
This example uses a built-in System function to read the date and time and store it into a User-Defined Data Type (structure) called TIME
The variable LocalDateTime was defined using the TIME data type. The details of the TIME UDT are found in the DataTypes -> User-Defined folder in the Controller Organizer
The TIME structure is then copied into individual INT variables for use with the MCLK AOI.

The MCLK_Yaskawa AOI was created to allow the SigmaLogic to be synchronized with the PLC clock. The benefit to using this AOI is so that the Alarm History in SigmaLogic matches any time stamping in the PLC. The PLC time and date is read from a System object called WallClockTime and stored into a structure called LocalDateTime. Then the individual elements from LocalDateTime are moved into the inputs of the MCLK AOI. When MCLK is enabled, the clock in SigmaLogic is updated. No power cycle is required on SigmaLogic.

The structure LocalDateTime uses a user-defined data type (UDT) called TIME that was created in the User-Defined folder in the Controller Organizer. Once created, the TIME structure contains the following elements:

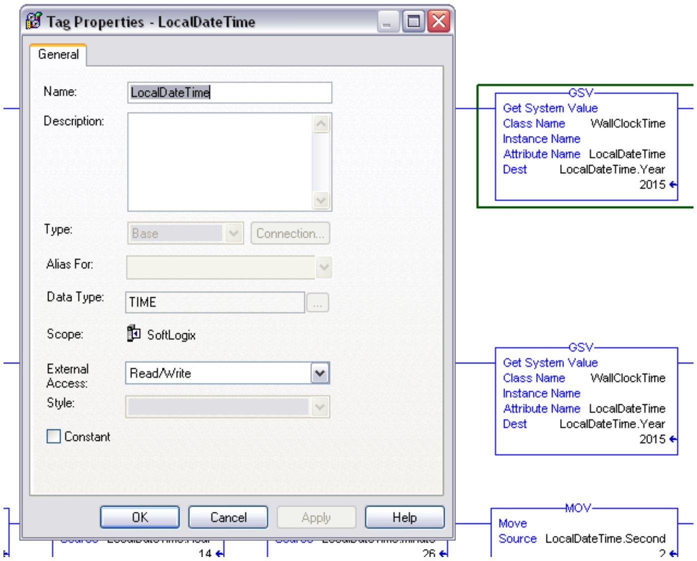


- Data Types
  - User-Defined
    - Dint_to_Bits
    - TIME
    - Yaskawa_AOI_Status_Bits
    - Yaskawa_EIP_Servo
    - Yaskawa_IN_from_Servo
    - Yaskawa_OUT_to_Servo
    - Yaskawa_PLS_Switch
  - Strings
  - Add-On-Defined
  - Predefined
  - Module-Defined

Members: Data Type Size: 28 byte(s)

Name	Data Type	Style	Description	External Access
Year	DINT	Decimal		Read/Write
Month	DINT	Decimal		Read/Write
Day	DINT	Decimal		Read/Write
Hour	DINT	Decimal		Read/Write
Minute	DINT	Decimal		Read/Write
Second	DINT	Decimal		Read/Write
Microsecond	DINT	Decimal		Read/Write

Here is the definition of the LocalDateTime structure:



Tag Properties - LocalDateTime

General

Name: LocalDateTime

Description:

Type: Base

Alias For:

Data Type: TIME

Scope: SoftLogix

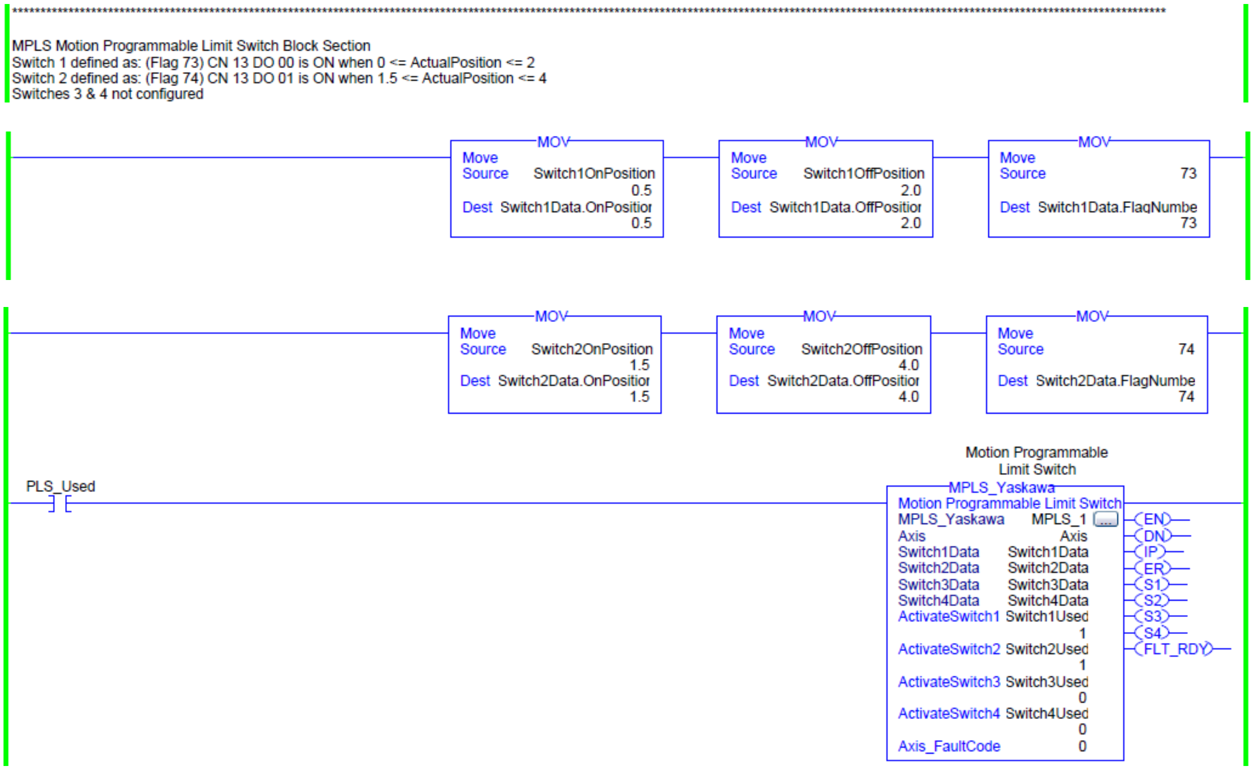
External Access: Read/Write

Style:

□ Constant

```
GSV
Get System Value
Class Name        WallClockTime
Instance Name
Attribute Name  LocalDateTime
Dest        LocalDateTime.Year
                          2015

GSV
Get System Value
Class Name        WallClockTime
Instance Name
Attribute Name  LocalDateTime
Dest        LocalDateTime.Year
                          2015

MOV
Move
Source  LocalDateTime.Second
                             2
```

17. MPLS_Yaskawa (Programmable Limit Switch Outputs)

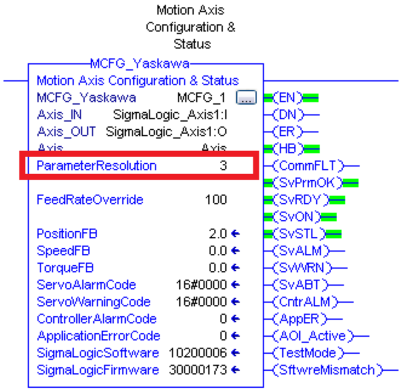


```
**************************************************************************************************
MPLS Motion Programmable Limit Switch Block Section
Switch 1 defined as: (Flag 73) CN 13 DO 00 is ON when 0 <= ActualPosition <= 2
Switch 2 defined as: (Flag 74) CN 13 DO 01 is ON when 1.5 <= ActualPosition <= 4
Switches 3 & 4 not configured

MOV  Move  Source Switch1OnPosition 0.5   Dest Switch1Data.OnPositior 0.5
MOV  Move  Source Switch1OffPosition 2.0  Dest Switch1Data.OffPositior 2.0
MOV  Move  Source 73                      Dest Switch1Data.FlagNumbe 73

MOV  Move  Source Switch2OnPosition 1.5   Dest Switch2Data.OnPositior 1.5
MOV  Move  Source Switch2OffPosition 4.0  Dest Switch2Data.OffPositior 4.0
MOV  Move  Source 74                      Dest Switch2Data.FlagNumbe 74

PLS_Used
   ] [
Motion Programmable
Limit Switch
MPLS_Yaskawa
Motion Programmable Limit Switch
MPLS_Yaskawa     MPLS_1 ...           -(EN)-
Axis             Axis                 -(DN)-
Switch1Data      Switch1Data          -(IP)-
Switch2Data      Switch2Data          -(ER)-
Switch3Data      Switch3Data          -(S1)-
Switch4Data      Switch4Data          -(S2)-
ActivateSwitch1  Switch1Used          -(S3)-
                 1                    -(S4)-
ActivateSwitch2  Switch2Used          -(FLT_RDY)-
                 1
ActivateSwitch3  Switch3Used
                 0
ActivateSwitch4  Switch4Used
                 0
Axis_FaultCode   0
```

The MPLS_Yaskawa AOI is used when outputs need to be triggered based on motor position. SigmaLogic can fire these position-based outputs faster than the PLC because the EIP communication cycle delays are cut out of the equation. The example above uses two of the available four PLS outputs. Switch1 is mapped to DO 00 (Flag 73) and Switch 2 is mapped to DO 01 (Flag 74). When MPLS is enabled, SigmaLogic will use the On Position and Off Position parameters of the SwitchData structure to trigger the outputs mapped by the Flag number. The On and Off Positions can be modified even while MPLS is enabled and those changes take effect immediately.

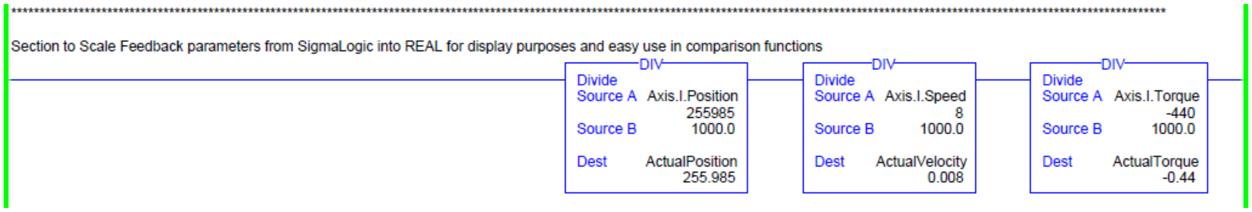
18. Scaling

Position, Speed and Torque feedback values are included in the Axis structure but the elements of that structure are DINT. The ParameterResolution input of the MCFG AOI defines the number of decimal points of resolution included in those DINT values.



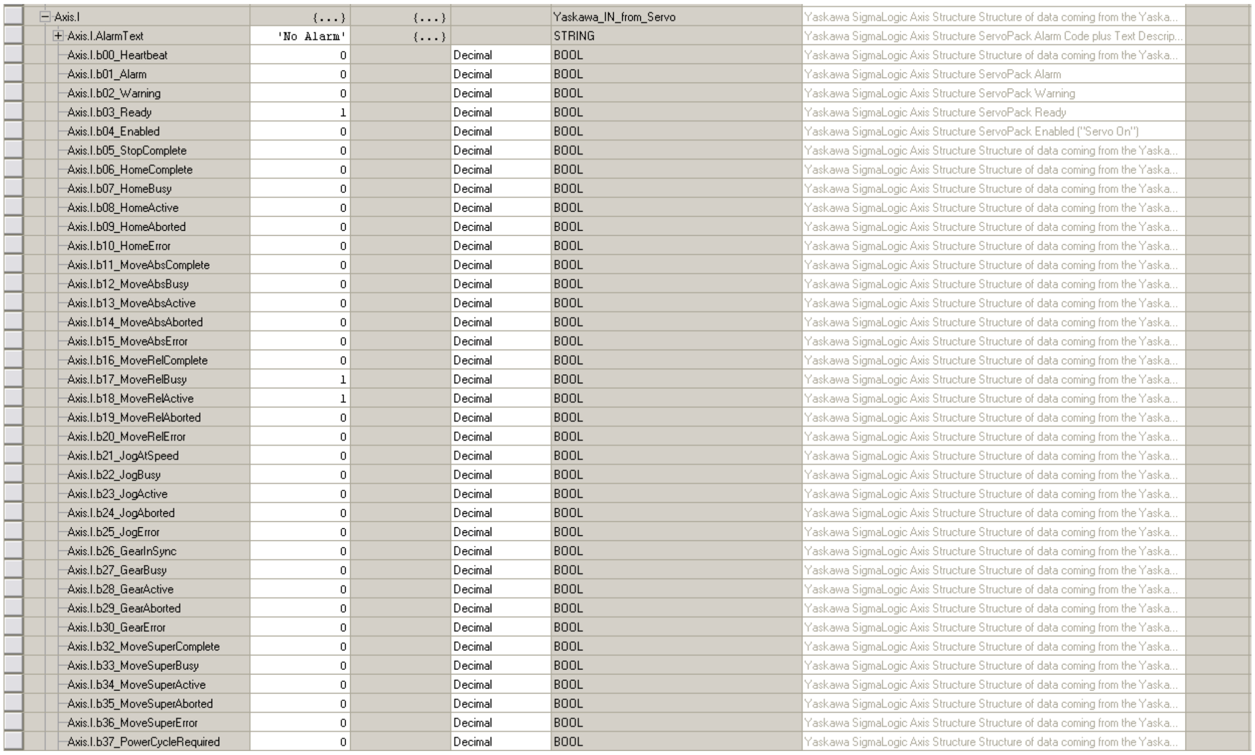
```
Motion Axis
Configuration &
Status
MCFG_Yaskawa
Motion Axis Configuration & Status
MCFG_Yaskawa          MCFG_1            (EN)
Axis_IN       SigmaLogic_Axis1:I        (DN)
Axis_OUT    SigmaLogic_Axis1:O          (ER)
Axis                           Axis     (HB)
ParameterResolution               3     (CommFLT)
                                        (SvPrmOK)
FeedRateOverride                100     (SvRDY)
                                        (SvON)
PositionFB                      2.0 ←   (SvSTL)
SpeedFB                         0.0 ←   (SvALM)
TorqueFB                        0.0 ←   (SvWRN)
ServoAlarmCode              16#0000 ←   (SvABT)
ServoWarningCode            16#0000 ←   (CntrALM)
ControllerAlarmCode               0 ←   (AppER)
ApplicationErrorCode              0 ←   (AOI_Active)
SigmaLogicSoftware         10200006 ←   (TestMode)
SigmaLogicFirmware         30000173 ←   (SftwreMismatch)
```

A clean way to take those DINT values and use them as REAL values in the program is to divide the DINT by the resolution and store into REAL variables like ActualPosition, ActualVelocity and ActualTorque.

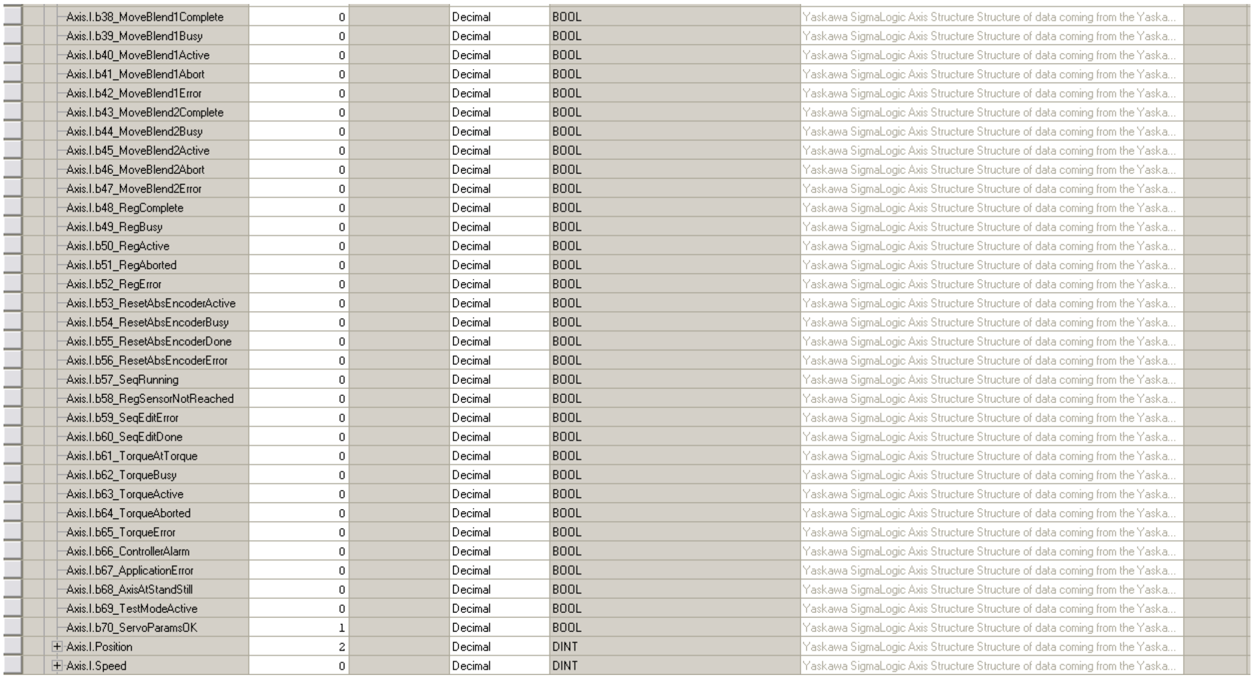


```
****************************************************************************************************************************************************************

Section to Scale Feedback parameters from SigmaLogic into REAL for display purposes and easy use in comparison functions

DIV                             DIV                             DIV
Divide                          Divide                          Divide
Source A   Axis.I.Position      Source A   Axis.I.Speed         Source A   Axis.I.Torque
                    255985                            8                          -440
Source B            1000.0      Source B         1000.0         Source B        1000.0

Dest        ActualPosition      Dest     ActualVelocity         Dest      ActualTorque
                   255.985                        0.008                          -0.44
```

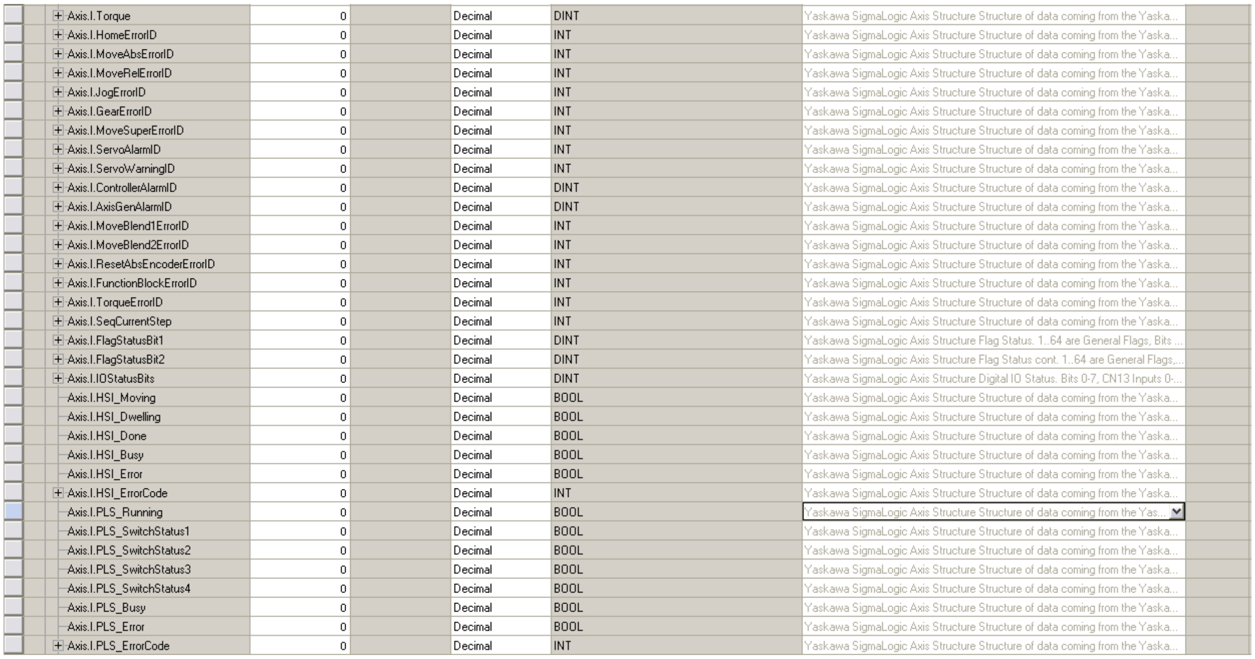
Appendix A – SigmaLogic Status Registers included in Yaskawa_IN_from_Servo



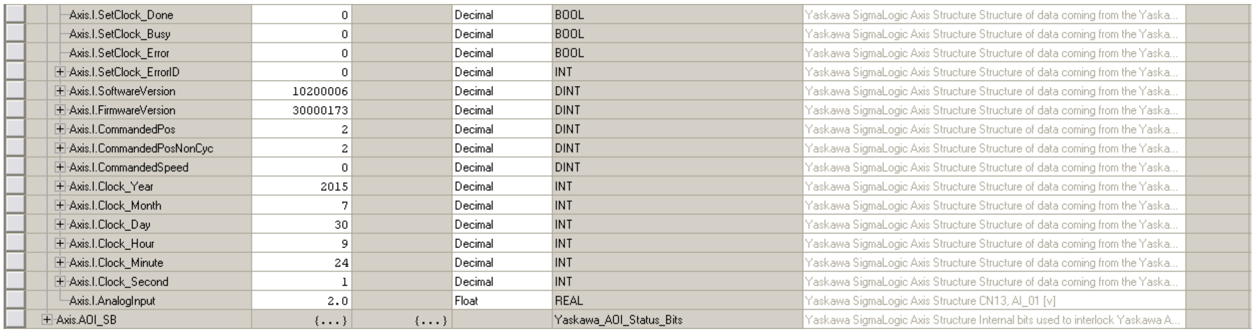
Axis.I	{...}	{...}		Yaskawa_IN_from_Servo	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.AlarmText	'No Alarm'	{...}		STRING	Yaskawa SigmaLogic Axis Structure ServoPack Alarm Code plus Text Descrip...
Axis.I.b00_Heartbeat	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b01_Alarm	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure ServoPack Alarm
Axis.I.b02_Warning	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure ServoPack Warning
Axis.I.b03_Ready	1		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure ServoPack Ready
Axis.I.b04_Enabled	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure ServoPack Enabled ("Servo On")
Axis.I.b05_StopComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b06_HomeComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b07_HomeBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b08_HomeActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b09_HomeAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b10_HomeError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b11_MoveAbsComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b12_MoveAbsBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b13_MoveAbsActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b14_MoveAbsAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b15_MoveAbsError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b16_MoveRelComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b17_MoveRelBusy	1		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b18_MoveRelActive	1		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b19_MoveRelAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b20_MoveRelError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b21_JogAtSpeed	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b22_JogBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b23_JogActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b24_JogAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b25_JogError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b26_GearInSync	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b27_GearBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b28_GearActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b29_GearAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b30_GearError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b32_MoveSuperComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b33_MoveSuperBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b34_MoveSuperActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b35_MoveSuperAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b36_MoveSuperError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b37_PowerCycleRequired	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...



Axis.I.b38_MoveBlend1Complete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b39_MoveBlend1Busy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b40_MoveBlend1Active	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b41_MoveBlend1Abort	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b42_MoveBlend1Error	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b43_MoveBlend2Complete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b44_MoveBlend2Busy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b45_MoveBlend2Active	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b46_MoveBlend2Abort	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b47_MoveBlend2Error	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b48_RegComplete	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b49_RegBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b50_RegActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b51_RegAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b52_RegError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b53_ResetAbsEncoderActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b54_ResetAbsEncoderBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b55_ResetAbsEncoderDone	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b56_ResetAbsEncoderError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b57_SeqRunning	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b58_RegSensorNotReached	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b59_SeqEditError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b60_SeqEditDone	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b61_TorqueAtTorque	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b62_TorqueBusy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b63_TorqueActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b64_TorqueAborted	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b65_TorqueError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b66_ControllerAlarm	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b67_ApplicationError	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b68_AxisAtStandStill	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b69_TestModeActive	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.b70_ServoParamsOK	1		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.Position	2		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.Speed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...

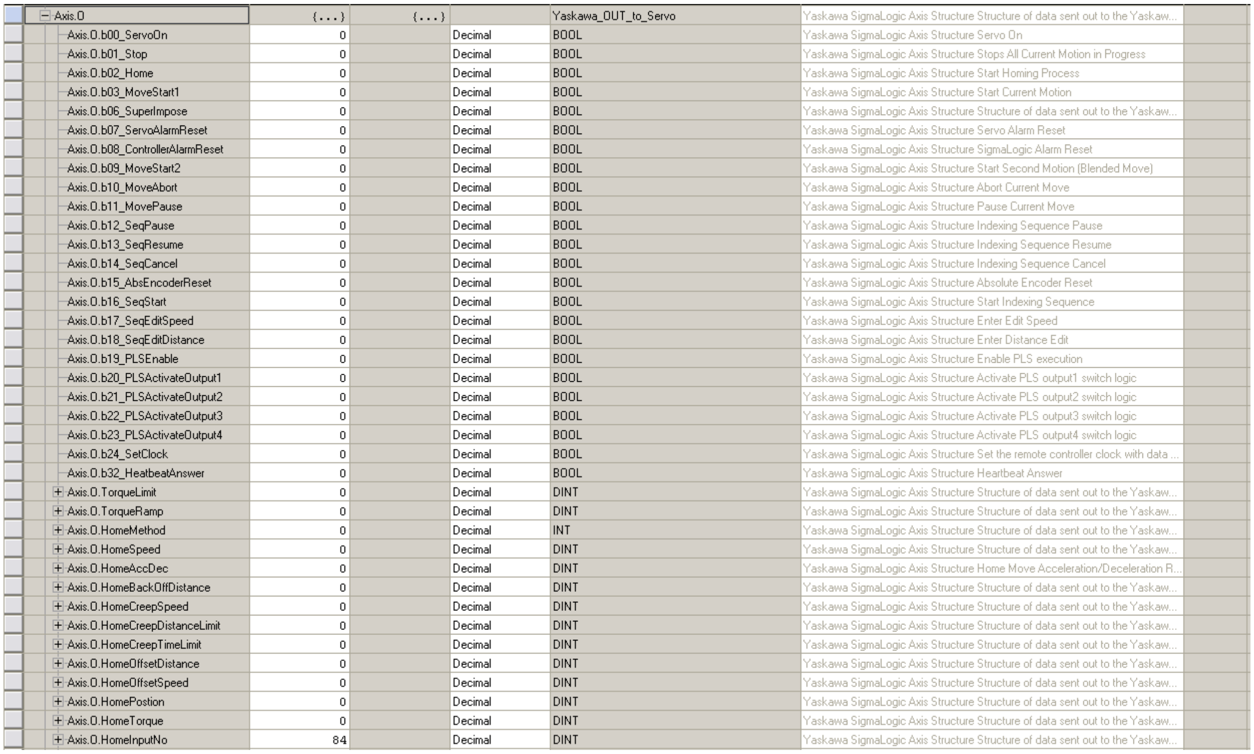


⊞ Axis.I.Torque	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.HomeErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.MoveAbsErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.MoveRelErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.JogErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.GearErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.MoveSuperErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.ServoAlarmID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.ServoWarningID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.ControllerAlarmID	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.AxisGenAlarmID	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.MoveBlend1ErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.MoveBlend2ErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.ResetAbsEncoderErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.FunctionBlockErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.TorqueErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.SeqCurrentStep	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.FlagStatusBit1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Flag Status. 1..64 are General Flags, Bits ...
⊞ Axis.I.FlagStatusBit2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Flag Status cont. 1..64 are General Flags,...
⊞ Axis.I.IOStatusBits	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Digital IO Status. Bits 0-7, CN13 Inputs 0-...
Axis.I.HSI_Moving	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.HSI_Dwelling	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.HSI_Done	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.HSI_Busy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.HSI_Error	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.HSI_ErrorCode	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_Running	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yas...
Axis.I.PLS_SwitchStatus1	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_SwitchStatus2	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_SwitchStatus3	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_SwitchStatus4	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_Busy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.PLS_Error	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
⊞ Axis.I.PLS_ErrorCode	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...

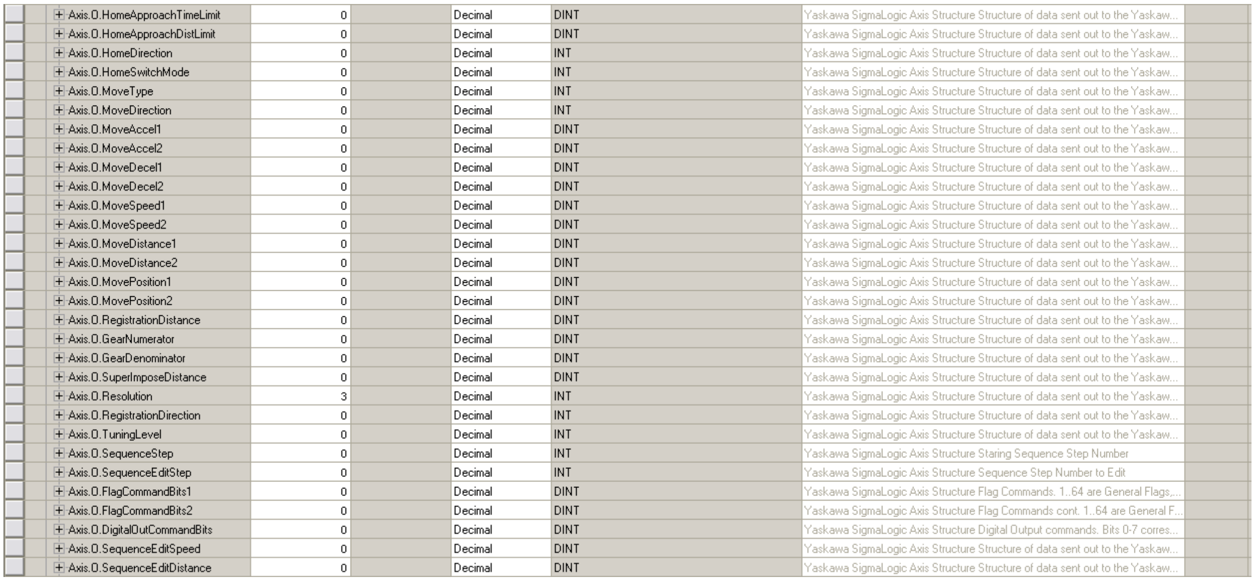


Axis.I.SetClock_Done	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.SetClock_Busy	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.SetClock_Error	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.SetClock_ErrorID	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.SoftwareVersion	10200006		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.FirmwareVersion	30000173		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.CommandedPos	2		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.CommandedPosNonCyc	2		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.CommandedSpeed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Year	2015		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Month	7		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Day	30		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Hour	9		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Minute	24		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
+ Axis.I.Clock_Second	1		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data coming from the Yaska...
Axis.I.AnalogInput	2.0		Float	REAL	Yaskawa SigmaLogic Axis Structure CN13, AI_01 [v]
+ Axis.AOI_SB	{...}	{...}		Yaskawa_AOI_Status_Bits	Yaskawa SigmaLogic Axis Structure Internal bits used to interlock Yaskawa A...

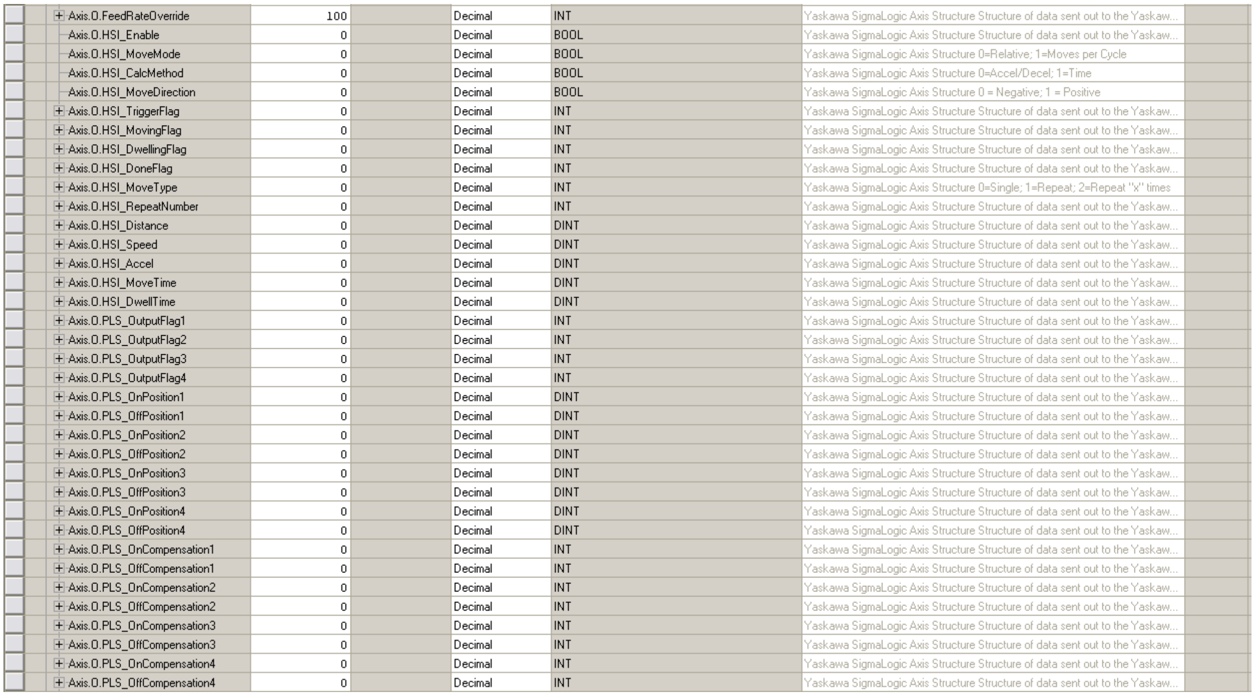
Appendix B – SigmaLogic Status Registers used in Yaskawa_OUT_to_Servo



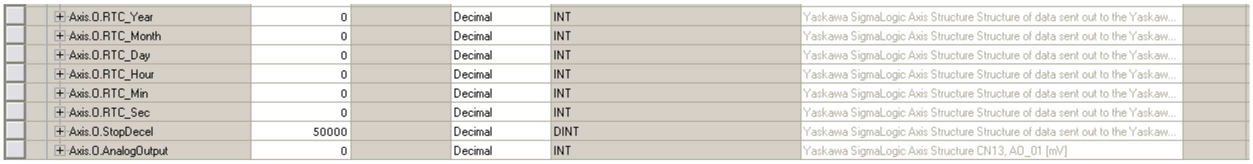
Axis.0	{...}	{...}		Yaskawa_OUT_to_Servo	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.b00_ServoOn	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Servo On
Axis.0.b01_Stop	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Stops All Current Motion in Progress
Axis.0.b02_Home	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Start Homing Process
Axis.0.b03_MoveStart1	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Start Current Motion
Axis.0.b06_SuperImpose	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.b07_ServoAlarmReset	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Servo Alarm Reset
Axis.0.b08_ControllerAlarmReset	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure SigmaLogic Alarm Reset
Axis.0.b09_MoveStart2	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Start Second Motion (Blended Move)
Axis.0.b10_MoveAbort	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Abort Current Move
Axis.0.b11_MovePause	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Pause Current Move
Axis.0.b12_SeqPause	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Indexing Sequence Pause
Axis.0.b13_SeqResume	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Indexing Sequence Resume
Axis.0.b14_SeqCancel	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Indexing Sequence Cancel
Axis.0.b15_AbsEncoderReset	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Absolute Encoder Reset
Axis.0.b16_SeqStart	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Start Indexing Sequence
Axis.0.b17_SeqEditSpeed	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Enter Edit Speed
Axis.0.b18_SeqEditDistance	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Enter Distance Edit
Axis.0.b19_PLSEnable	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Enable PLS execution
Axis.0.b20_PLSActivateOutput1	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Activate PLS output1 switch logic
Axis.0.b21_PLSActivateOutput2	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Activate PLS output2 switch logic
Axis.0.b22_PLSActivateOutput3	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Activate PLS output3 switch logic
Axis.0.b23_PLSActivateOutput4	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Activate PLS output4 switch logic
Axis.0.b24_SetClock	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Set the remote controller clock with data ...
Axis.0.b32_HeatbeatAnswer	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Heartbeat Answer
Axis.0.TorqueLimit	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.TorqueRamp	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeMethod	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeSpeed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeAccDec	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Home Move Acceleration/Deceleration R...
Axis.0.HomeBackOffDistance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeCreepSpeed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeCreepDistanceLimit	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeCreepTimeLimit	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeOffsetDistance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeOffsetSpeed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomePostion	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeTorque	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeInputNo	84		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...



Axis.0.HomeApproachTimeLimit	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeApproachDistLimit	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeDirection	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HomeSwitchMode	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveType	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveDirection	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveAccel1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveAccel2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveDecel1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveDecel2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveSpeed1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveSpeed2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveDistance1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MoveDistance2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MovePosition1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.MovePosition2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.RegistrationDistance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.GearNumerator	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.GearDenominator	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.SuperImposeDistance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.Resolution	3		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.RegistrationDirection	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.TuningLevel	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.SequenceStep	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Staring Sequence Step Number
Axis.0.SequenceEditStep	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Sequence Step Number to Edit
Axis.0.FlagCommandBits1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Flag Commands. 1..64 are General Flags,...
Axis.0.FlagCommandBits2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Flag Commands cont. 1..64 are General F...
Axis.0.DigitalOutCommandBits	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Digital Output commands. Bits 0-7 corres...
Axis.0.SequenceEditSpeed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.SequenceEditDistance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...



⊞ Axis.0.FeedRateOverride	100		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HSI_Enable	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
Axis.0.HSI_MoveMode	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure 0=Relative; 1=Moves per Cycle
Axis.0.HSI_CalcMethod	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure 0=Accel/Decel; 1=Time
Axis.0.HSI_MoveDirection	0		Decimal	BOOL	Yaskawa SigmaLogic Axis Structure 0 = Negative; 1 = Positive
⊞ Axis.0.HSI_TriggerFlag	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_MovingFlag	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_DwellingFlag	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_DoneFlag	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_MoveType	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure 0=Single; 1=Repeat; 2=Repeat "x" times
⊞ Axis.0.HSI_RepeatNumber	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_Distance	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_Speed	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_Accel	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_MoveTime	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.HSI_DwellTime	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OutputFlag1	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OutputFlag2	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OutputFlag3	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OutputFlag4	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnPosition1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffPosition1	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnPosition2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffPosition2	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnPosition3	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffPosition3	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnPosition4	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffPosition4	0		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnCompensation1	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffCompensation1	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnCompensation2	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffCompensation2	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnCompensation3	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffCompensation3	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OnCompensation4	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.PLS_OffCompensation4	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...



⊞ Axis.0.RTC_Year	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.RTC_Month	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.RTC_Day	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.RTC_Hour	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.RTC_Min	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.RTC_Sec	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.StopDecel	50000		Decimal	DINT	Yaskawa SigmaLogic Axis Structure Structure of data sent out to the Yaskaw...
⊞ Axis.0.AnalogOutput	0		Decimal	INT	Yaskawa SigmaLogic Axis Structure CN13, AO_01 [mV]