Quick Start Motion Application Software – QS_AIS_1

Quick Start Motion Application Software allows commissioning of 1 to 32 axes motion applications in a matter of hours. Basic motion control is transformed from a programming effort to an application exercise.

The foundation of the Quick Start package is the basic motion control application specific function block (ASFB). Two ASFB's address basic motion control, these are:

- QS_AIS_1 – Basic servo motion control ASFB
- QS_DIG_1 – Master axis feedback interface ASFB

One ASFB provides fault control and fault history, this is:

- QS_FLT_1 – Fault control and fault history ASFB

One ASFB provides integration with a Cimrex Operator Interface, this is:

- QS_C69_1 – Integration with Cimrex C69 HMI operating in portrait mode

These functions may be used with PiC, MMC or MMC for PC controls.

This document covers the functionality provided by QS_AIS_1.

QS_AIS_1 Overview

QS_AIS_1 provides a single-network solution to basic servo motion providing Drive Control, Manual Operation and Homing. Functions provide by QS_AIS_1 include:

Drive Control, including:

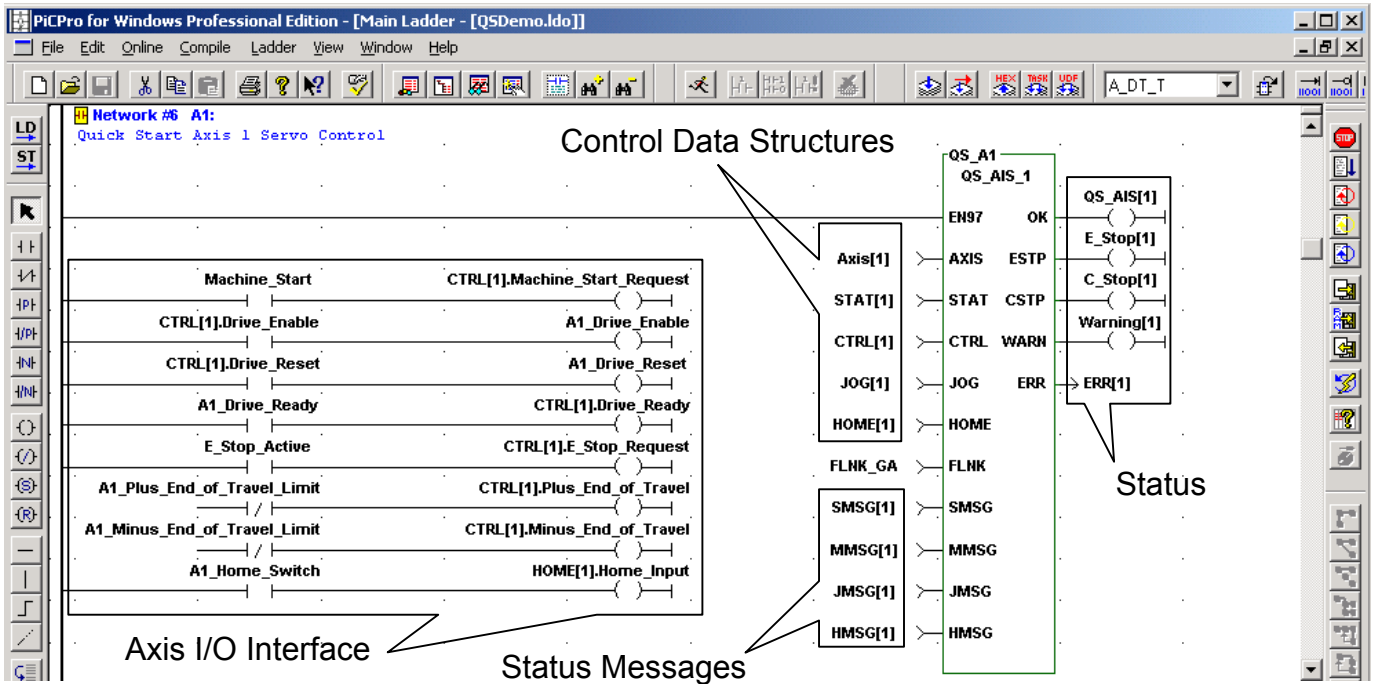
- Machine Start Logic
- Drive Enable and Reset Control
- Drive Ready Monitor
- Fault Detection and Status Reporting

Manual servo operation, including:

- Velocity Jog – jog at a rate
- Incremental Jog – jog a distance at a rate
- Handwheel Jog – follow a handwheel (feedback device) at a ratio

Home cycle, including:

- Home Reference Cycle
- Move to Park Position after Home Complete



Using QS_AIS_1 each axis is controlled by a single network in the main application program. Interface between your application and QS_AIS_1 using the Control Data Structures. Status Messages provide English-language status messages for all of the QS_AIS_1 operations. The Status outputs of QS_AIS_1 provide an overview of axis fault status. The Axis I/O Interface logic provides the interface between Control Data Structures and the I/O used for basic drive and motion control.

Control of the servo axis can be performed by additional application logic or operator interface manipulation of variables in the Control Data Structures.

The Control Data Structures include:

Structure	Description	Function
STAT	Status	Present Status Information
CTRL	Control	Control Drive Operation
JOG	Jog	Jog, Incremental Jog and Handwheel Jog Operation
HOME	Home	Home Cycle Operation
FLNK	Fault Link	Coordinate fault control in multiple axis applications

The Status Messages provide:

Variable	Description	Function
SMSG	Status	Description of operating status or fault if fault present
MMSG	Motion	Description of currently active move instruction
JMSG	Jog	Description of Jog Operation state
HMSG	Home	Description of Home Cycle Operation state

AXIS – Servo Axis Number

Axis specifies the servo axis number that this instance of QS_AIS_1 controls. Axis ranges from 1 to 32 and is converted to the servo setup axis number, 1 to 16 for axis 1 to 16 and 101 to 117 for axis 17 to 32, by QS_AIS_1. If Axis is out of range ERR, programming error will be 1007.

STAT Status Data Structure

Use the STAT, Status Data Structure to access status information describing the operation of this servo axis.

Name	Type	**Description**
STAT	Struct	Axis Status Data Structure
.Axis_Description	STRING[22]	I – Application specific description of this axis
.Initialization_Okay	BOOL	O - On when Axis Initialized
.Fault	BOOL	O - On when fault present
.Fault_Description	STRING[32]	O - Description of fault present on this axis
.E_Stop_Fault	BOOL	O - On when fault is E Stop fault
.C_Stop_Fault	BOOL	O - On when fault is C Stop fault
.Warning	BOOL	O - On when a warning condition is present
.Loop_Closed	BOOL	O - On when loop closed
.In_Position	BOOL	O - On when no move active and In Position
.Move_Active	BOOL	O - On when a move is active
.Active_Move_Queue_Number	USINT	O - Queue number for active move
.Next_Move_Queue_Is_Available	BOOL	O - On when another move can be queued
.Command_Position	DINT	O - Position axis is being commanded to move to
.Actual_Position	DINT	O - Position that axis is actually in
.Position_Error	DINT	O - Difference between commanded and actual
.Filter_Error	DINT	O - Difference due to position loop filter
.Statmsgn	INT	O – HMI status message number
.Movemsgn	INT	O – HMI move message number
.Jogmsgn	INT	O – HMI jog message number
.Homemsgn	INT	O – HMI home cycle message number
.Status_Move_Started	BOOL	O - On when iteration of a move has begun
.Status_Fast_Input_Occurred	BOOL	O - On when the fast input has been detected
.Status_Fast_Input_On	BOOL	O - On when the fast input is in the on state
.Status_Good_Mark_Detected	BOOL	O - On when REGIST has detected a good mark
.Status_Bad_Mark_Detected	BOOL	O - On when REGIST has detected a bad mark
.Status_DIST_Plus_TOLR_Exceeded	BOOL	O - On when REGIST distance exceeded
.Status_Fast_Input_Rising	BOOL	O - Indicates direction of last fast input transition
.Structure_Check_Constant	DINT	I – Verify structure is correct size – set to 12345
END_STRUCT		

STAT.Axis_Description - Input - Name of this axis. Appended to fault description to identify which axis caused the fault.
STAT.Initialization_Okay – Output - ON when axis is present in the system and has initialized properly - OFF if this axis number was not included in the application specific servo setup function or if the STRTSERV, servo system start up, function was not called or failed due to an error. See Initializing the Servo System for detailed information.
STAT.Fault – Output - ON when an E_Stop fault, C_Stop fault or Warning condition is present See the SMSG status message for a text description of all possible fault conditions. - OFF when no faults are present
STAT.Fault_Description – Output - Text message describing fault. See SMSG messages for descriptions.
STAT.E_Stop_Fault – Output - ON when an Emergency Stop fault is present
STAT.C_Stop_Fault – Output - ON when a Controlled Stop Fault is present
STAT.Warning – Output - ON when a Warning condition is present
STAT.Loop_Closed – Output - ON when servo loop is closed - OFF when servo loop is open
STAT.In_Position – Output - ON when no move is active and the next move queue is empty and the axis actual position is within the in-position band specified in servo setup. - OFF when a move is active or a move is in the next move queue or the axis position is outside of the in-position bandwidth specified in servo setup
STAT.Move_Active – Output - ON when a move is executing. See the MMSG status message for a text description of the currently executing move
STAT.Active_Move_Queue_Number – Output - A number from 1 to 255 indicating queue number of the currently active move. The queue number is used by the application to perform control operations specific to the currently executing move such as aborting it, changing velocity, etc
STAT.Next_Move_Queue_Is_Available – Output - ON when another move can be called and placed in the “next move” queue. Placing a move in the “next queue” allows moves to be blended together with no time gaps between moves.
STAT.Command_Position – Output - Indicates the position this axis is being commanded to move to.
STAT.Actual_Position – Output - Indicates the present position.

STAT.Position_Error – Output - Indicates the difference between the commanded position and the actual position.
STAT.Filter_Error – Output - Indicates the lag between the commanded position and the actual position due to the slow velocity filter
STAT.Statmsgn – Output - Indicates the message number for HMI status message display. See the SMSG table later in this document for a list of all Status messages.
STAT.Movemsgn – Output - Indicates the message number for HMI move message display. See the MMSG table later in this document for a list of all Status messages.
STAT.Jogmsgn – Output - Indicates the message number for HMI jog operation message display. See the JMSG table later in this document for a list of all Status messages.
STAT.Homemsgn – Output - Indicates the message number for HMI home cycle message display. See the HMSG table later in this document for a list of all Status messages.
STAT.Status_Move_Started – Output - Indicates that the queued move has begun iterating. Use to see that move queued with FAST_QUE has been triggered and is executing. See STATUSSV function for more information
STAT.Status_Fast_Input_Occurred – Output - Indicates that the Fast Input is in the ON state. See STATUSSV function for more information
STAT.Status_Fast_Input_On – Output - Indicates that the Fast Input is in the ON state. See STATUSSV function for more information.
STAT.Status_Good_Mark_Detected – Output - Indicates that REGIST has detected a valid registration mark. See STATUSSV and REGIST functions for more information
STAT.Status_Bad_Mark_Detected – Output - Indicates that REGIST has detected an invalid registration mark. See STATUSSV and REGIST functions for more information
STAT.Status_DIST_Plus_TOLR_Exceeded – Output - Indicates that REGIST has seen the axis travel a distance in excess of DIST + TOLR since the last good mark. See STATUSSV and REGIST functions for more information
STAT.Status_Fast_Input_Rising – Output - Indicates the direction of the last fast input transition. See STATUSSV function for more information. See STATUSSV and REGIST functions for more information
STAT.Structure_Check_Constant – Input - Must be set to an initial value of 12345. Checked by QS_AIS_1 on first scan, if not 12345 then ERR will be set to 1001, ESTP will be energized, and QS_AIS_1 will exit without executing.

CTRL Control Data Stucture

Use CTRL, the Control Data Structure to interface with and operate the servo amplifier. It includes logic for drive power-on sequencing, drive and axis I/O control and monitoring, as well as basic operations like opening and closing the position loop.

Name	Type	Description
CTRL	Struct	Axis Control Data Structure
.Machine_Start_Request	BOOL	I – One-shot to clear faults and close servo loop
.E_Stop_Request	BOOL	I – One-shot to create E Stop Fault
.C_Stop_Request	BOOL	I – One-shot to create C Stop Fault
.Drive_Enable	BOOL	O – On when drive should be enabled
.Drive_Reset	BOOL	O – On for one scan to reset drive fault
.Drive_Ready	BOOL	I – Monitor Drive Ready status
.Plus_End_of_Travel	BOOL	I – Monitor End of Travel Plus Limit Switch
.Minus_End_of_Travel	BOOL	I – Monitor End of Travel Minus Limit Switch
.Wait_for_Drive_Ready_Delay	TIME	I – Machine Start wait for Drive Ready time delay
.Abort_Move	BOOL	I – Abort Specifed Move Request
.Move_to_Abort	USINT	I – Queue number of move to abort
.Abort_All_Moves	BOOL	I – Abort all active and queued moves
.Structure_Check_Constant	DINT	I – Verify structure is correct size - set to 23456
END_STRUCT		

CTRL.Machine_Start_Request - Input
- One-shot to clear faults and initiate the machine start sequence. Energize to override Plus and Minus Endlimit faults to allow Jog off of endlimits. See Machine Start Sequence for detailed explanation. The servo loop will be closed at the end of the Machine Start Sequence.

CTRL.E_Stop_Request - Input
– When OFF allows normal operation
- When ON generates 'E Stop Request' E Stop Fault

CTRL.C_Stop_Request - Input
– When OFF allows normal operation
- When ON generates 'C Stop Request' C Stop Fault

CTRL.Drive_Enable – Output
- Connect to the application Drive Enable output. Intended to energize drive power.
- Set by Machine Start Sequence. Remains ON until E Stop Fault.

CTRL.Drive_Reset – Output
- Connect to the application Drive Reset output. Intended to clear drive faults.
– Pulsed for one scan at beginning of Machine Start Sequence if Drive Ready is OFF.

CTRL.Drive_Ready – Input
- Connect to the Drive Ready input that indicates the drive is not faulted and is ready to operate.
- Must be ON to allow machine start sequence to complete and close the position loop. If it turns OFF the 'Drive Not Ready' E Stop Fault will occur.

CTRL.Plus_End_of_Travel – Input - Connect to the Plus End of Travel limit switch for linear travel servo axes - When ON will cause an 'Plus End of Travel' E Stop Fault. Can be overridden to jog off the end limit by energizing CTRL.Machine_Start_Request.
CTRL.Minus_End_of_Travel – Input - Connect to the Minus End of Travel limit switch for linear travel servo axes - When ON will cause a 'Minus End of Travel' E Stop Fault. Can be overridden to jog off the end limit by energizing CTRL.Machine_Start_Request.
CTRL.Wait_for_Drive_Ready_Delay - Input - Defines time delay between the beginning of the machine start sequence and when the servo loop will be closed. The delay needs to be long enough to insure that the machine power on sequence can complete, drive faults can be cleared by drive reset and drive ready is on.
CTRL.Abort_Move – Input - One-shot to abort the move specified by the CTRL.Move_to_Abort move queue number. See Working with Move Queue Numbers for more information.
CTRL.Move_to_Abort - Input - Defines the queue number of the move to abort by one-shotting CTRL.Abort_Move
CTRL.Abort_All_Moves - Input - One-shot to abort both the active and the queued move. See Working with Move Queue Numbers for more information.
CTRL.Structure_Check_Constant – Input - Must be set to an initial value of 23456. Checked by QS_AIS_1 on first scan, if not 23456 then ERR will be set to 1002, ESTP will be energized, and QS_AIS_1 will exit without executing.

JOG Data Structure

Use the JOG Data Structure to control of three modes of manual axis operation. Jog Plus and Jog Minus provide manual jogging at an application specified rate, Jog Incremental provides incremental distance moves using an application specified rate and distance and Handwheel Jog allows following a master axis (typically connected to a Handwheel) at an application specified ratio.

Name	Type	Description
JOG	Struct	Axis Jog Control Data Structure
.Plus	BOOL	I - Energize to Jog Plus
.Minus	BOOL	I - Energize to Jog Minus
.Rate	UDINT	I - Specify Jog Rate
.Jog_Active	BOOL	O - On when Jog active
.Incremental_Plus	BOOL	I - One-shot to start Incremental Jog Plus
.Incremental_Minus	BOOL	I - One-shot to start Incremental Jog Minus
.Incremental_Abort	BOOL	I - One-shot to abort Incremental Jog
.Incremental_Rate	DINT	I - Specify Incremental Jog Move Rate
.Incremental_Distance	DINT	I - Specify Incremental Jog Distance
.Incremental_Jog_Active	BOOL	O - On when Incremental Jog active
.Handwheel_Mode	BOOL	I - Energize to Enable Handwheel Jog Mode
.Handwheel_Axis_Number	USINT	I - Specify Handwheel Jog Master Axis
.Handwheel_Master_Distance	DINT	I - Specify Handwheel Jog Master Distance
.Handwheel_Slave_Distance	DINT	I - Specify Handwheel Jog Slave Distance
.Handwheel_Mode_Active	BOOL	O - On when Handwheel Jog Active
.Structure_Check_Constant	DINT	I - Used to check structure size - set to 34567
END_STRUCT		

Jog.Plus – Input
- Energize to move axis in plus direction at rate specified by JOG.Jog_Rate. If CTRL.Plus_End_of_Travel is ON motion will not occur.
- De-energize to stop

Jog.Minus – Input
- Energize to move axis in minus direction at rate specified by JOG.Jog_Rate. If CTRL.Minus_End_of_Travel is ON motion will not occur.
- De-energize to stop

Jog.Rate – Input
- Specify rate for Jog.Plus or Jog.Minus

Jog.Jog_Active – Output
- ON when JOG.Plus or JOG.Minus operation is active

Jog.Incremental_Plus – Input
- One-shot to initiate incremental distance move in plus direction with distance specified by JOG.Incremental_Distance at the rate specified by JOG.Incremental_Rate

Jog.Incremental_Minus – Input
- One-shot to initiate incremental distance move in minus direction with distance specified by JOG.Incremental_Distance at the rate specified by JOG.Incremental_Rate

Jog. Incremental_Abort – Input
- One-shot to abort a Jog Incremental Plus or Minus move before it completes.

Jog.Incremental_Rate – Input
- Specifies the rate the Jog Incremental Plus or Minus move will be made at.

Jog.Incremental_Distance – Input
- Specifies the distance to be moved by Jog Incremental Plus or Minus moves

Jog.Incremental_Jog_Active – Output
- ON when Jog Incremental Plus or Minus move is active

Jog.Handwheel_Mode – Input
- Energize to.select Handwheel Mode, De-energize to exit Handwheel Mode

Jog.Handwheel_Axis_Number – Input
- Specify the digitized axis number that the Handwheel is interfaced to.

Jog.Handwheel_Master_Distance – Input
- Use with JOG.Handwheel_Slave_Distance to specify the ratio of handwheel motion to axis motion. For example:

JOG.Handwheel_Master_Distance	JOG.Handwheel_Slave_Distance	Handwheel to Axis Ratio
1	1	1:1
10	1	10:1
100	500	1:5

Jog.Handwheel_Slave_Distance – Input
- Use with JOG.Handwheel_Master_Distance to specify the ratio of handwheel motion to axis motion. See JOG.Handwheel_Master_Distance for explanation.

Jog.Handwheel_Mode_Active – Output
- ON when Jog Handwheel Mode is Active

Jog.Structure_Check_Constant – Input
- Must be set to an initial value of 34567. Checked by QS_AIS_1 on first scan, if not 34567 then ERR will be set to 1003, ESTP will be energized, and QS_AIS_1 will exit without executing.

HOME Data Structure

Use the HOME Data Structure to control axis homing.

Name	Type	Description
HOME	Struct	Axis Home Cycle Data Structure
.Start	BOOL	I – One-shot to start home cycle
.Abort	BOOL	I – One-shot to abort home cycle
.Active	BOOL	O – On while home cycle active
.Complete	BOOL	O – On when home cycle successfully completed
.In_Plus_Direction	BOOL	I – Energize to Home in plus direction
.Rate	UDINT	I – Home cycle rate
.Home_to_Scanned_Input	BOOL	I – Energize for Home Cycle to Ladder Input
.Home_Input	BOOL	I – Home Cycle Input Switch
.Home_to_Fast_Input	BOOL	I – Energize for Home Cycle to Fast Input
.Ignore_Index_or_Null	BOOL	I – Energize to ignore feedback index or null
.Distance_from_Index_or_Null	DINT	O – Indicates switch distance from index or null
.Dimension_to_Assign_to_Home	DINT	I – Specify Dimension to assign to Home
.Auto_Backoff_Home_Switch	BOOL	I – Energize to backoff if on home swtich
.Move_to_Park_After_Home	BOOL	I – Energize to move to Park after Home
.Park_Position	DINT	I – Postion to move to after Home
.Park_Rate	UDINT	I – Rate to move to Park after Home
.Structure_Check_Constant	DINT	I - Used to check structure size - set to 45678
END_STRUCT		

HOME.Start – Input - One-shot to start Home Cycle.
HOME.Abort – Input - One-shot to abort Home Cycle
HOME.Active – Output - ON when Home Cycle is active
HOME.Complete – Output - ON when Home Cycle successfully comepletes
HOME.In_Plus_Direction Input - Energize to specify Home Cycle in plus direction
HOME.Rate – Input - Specify velocity to move at during Home Cycle.
HOME. Home_to_Scanned_Input – Input - Energize if Home switch is wired to control input that is connected to Home.Home_Input
HOME. Home_Input – Input - Home switch when performing Home to Scanned Input
HOME. Home_to_Fast_Input – Input - Energize if Home switch is wired to the Fast Input associated with the control hardware feedback channel for this axis
HOME.Ignore_Index_or_Null – Input - Energize to ignore the encoder index or resolver null when homing. De-energize to ?

HOME.Distance_from_Index_or_Null – Output - Indicates the distance away from the encoder index or resolver null the feedback device was when the Home switch tripped. For repeatable Homing using encoder feedback this value should be ?
HOME.Dimension_to_Assign_Home – Input - Specify the position value to assign to Home when the Home Cycle completes successfully.
HOME.Auto_Backoff_Home_Switch – Input - Energize to allow automatic motion in the direction opposite of that specified by Home.In_Plus_Direction if when the Home Cycle is started the Home Switch is ON.
HOME.Move_to_Park_After_Home – Input - Energize to initiate a move to the position specified by HOME.Park_Position at the rate specified by the home park rate after the Home Cycle completes successfully.
HOME.Park_Position – Input - Specify Position to move to after successful Home Cycle if HOME.Move_to_Park_After_Home is ON
HOME.Park_Rate – Input - Specify velocity to move to park position after successful Home Cycle if HOME.Move_to_Park_After_Home is ON
HOME.Structure_Check_Constant – Input - Must be set to an initial value of 45678. Checked by QS_AIS_1 on first scan, if not 45678 then ERR will be set to 1004, ESTP will be energized and QS_AIS_1 will exit without executing.

FAULT LINK Data Structure

The Fault Link Data Structure is used in multi-axis applications to automatically link the fault status of a group of axes together. No user programming of any variables in the FLNK structure is required. Simply place the same structure, e.g. FLNK_GA, at the input to every axis' QS_AIS_1 ASFB and also into the FLNK input of the QS_FLT_1, fault manager ASFB.

When linked, if any axis in the group enters a C_Stop condition then all axes in the group will be placed in a C_Stop condition. Likewise, if any axis in the group enters an E_Stop condition then all axes in the group will be placed in an E_Stop condition. QS_FLT_1, the fault manager, will time stamp and log the first fault that occurred so it can be reported and corrected.

Multiple groups of axes can be programmed with independent fault control by using a separate fault link data structure. For example, create a new fault link structure, FLNK_GB, and use a separate instance of the fault control manager, QS_FLT_1. See the documentation for QS_FLT_1 for more information.

Name	Type	**Description**
FLNK	Struct	Coordinate fault control in multiple axis applications
.Axis	USINT	I/O – Axis number of first axis to fault
.E_Stop	BOOL	I/O – E Stop Fault present
.C_Stop	BOOL	I/O – C Stop Fault present
.Warning	BOOL	I/O – Warning condition present
.AMSG	STRING[22]	O – fault axis for first fault
.FMSG	STRING[32]	O – fault message for first fault
.FMSGn	INT	O – fault message number for first fault
.Structure_Check_Constant	DINT	I – Verify structure is correct size - set to 67891
END_STRUCT		

FLNK.variables
- Used by QS_AIS_1, QS_DIG_1 and QS_FLT_1 to coordinate faults. Must not be used by the application.

FLNK.Structure_Check_Constant – Input
- Must be set to an initial value of 67891. Checked by QS_AIS_1 on first scan, if not 67891 then ERR will be set to 1006, ESTP will be energized, and QS_AIS_1 will exit without executing.

QS_AIS_1 Status Message Outputs

QS_AIS_1 provides four English-language status message outputs to provide an indication of the state of the servo system at a glance. The English-language text can be viewed using PiCPro's animation and view list. The same messages are presented on the Cimrex C69 but are drawn from the Cimrex message library. The Cimrex C69 can be configured with multiple libraries to support languages other than English. The status messages are:

Variable	Description	Function
SMSG	Status	Description of operating status or fault if fault present
MMSG	Motion	Description of currently active motion instruction
JMSG	Jog	Description of Jog Operation state
HMSG	Home	Description of Home Cycle Operation state

A detailed list of all of the possible messages for each status message variable follows.

SMSG - Description of operating status or fault if fault present

SMSG during Normal Operation		
Machine Start In Process	Normal Operation	Stat.Statmsgn = 350
Indicates that CTRL.Machine_Start_Request has been one-shot or is being energized starting the Machine Start Sequence. See Machine Start Sequence for more information.		
Loop Closed - No Fault	Normal Operation	Stat.Statmsgn = 351
Indicates that the Machine Start Sequence has completed, the position loop is closed and no fault is present.		
Loop Open - No Fault	Normal Operation	Stat.Statmsgn = 352
Indicates that the position loop has been opened by one-shotting CTRL.Open_Loop_Request.		
SMSG when E Stop Fault is present		
First Scan E Stop	E Stop Fault	Stat.Statmsgn = 360
Indicates that the control scan has been stopped and restarted (typically after a power cycle)		
Axis Not Initialized	E Stop Fault	Stat.Statmsgn = 361
Indicates an application programming error causing an E Stop Fault. Caused by this axis not being included in the servo setup function or an error occurring during servo system initialization. See Initializing the Servo System for detailed information.		
Programming Error	E Stop Fault	Stat.Statmsgn = 362
Indicates an application programming error. The last variable in each of the Data Structures input to QS_AIS_1 has a constant which is checked. If it is incorrect QS_AIS_1 cannot function and this error will be reported. The ERR output will indicate which structure is incorrect.		

Drive Not Ready	E Stop Fault	Stat.Statmsgn = 363
Indicates that the CTRL.Drive_Ready signal is not energized causing an E Stop Fault.		
E Stop Request	E Stop Fault	Stat.Statmsgn = 364
Indicates that the CTRL.E_Stop_Request input was energized causing an E Stop Fault.		
Loss of Feedback	E Stop Fault	Stat.Statmsgn = 365
Indicates that the control system detected a loss of the feedback signal for this axis causing an E Stop Fault.		
Excess Following Error	E Stop Fault	Stat.Statmsgn = 366
Indicates that the difference between the commanded position and the actual position exceeded the excess error limit specified in servo setup.		
Math Overflow Error	E Stop Fault	Stat.Statmsgn = 367
Indicates that the distance travelled in one servo update created a math overflow causing an E Stop Fault.		
User Set E Stop	E Stop Fault	Stat.Statmsgn = 368
Indicates that the application program generated an E Stop Fault using the E_Stop Function.		
Plus End of Travel	E Stop Fault	Stat.Statmsgn = 369
Indicates that CTRL.Plus_End_of_Travel was energized causing an E Stop Fault.		
Minus End of Travel	E Stop Fault	Stat.Statmsgn = 370
Indicates that CTRL.Minus_End_of_Travel was energized causing an E Stop Fault		
Fault Link E Stop Request	E Stop Fault	Stat.Statmsgn = 371
Indicates that the fault link structure, FLNK, indicated that another axis entered the E Stop state causing this axis to E Stop. See the QS_FLT_1 documentation to see how the FMSG output of QS_FLT_1 presents a time stamped message identifying which axis entered the E Stop state first and the fault that caused it to enter the E Stop state.		
SMSG when C Stop Fault is present		
C Stop Request	C Stop Fault	Stat.Statmsgn = 399
Indicates that the CTRL.C_Stop_Request input was energized causing a C Stop Fault.		
Plus Software Limit Exceeded	C Stop Fault	Stat.Statmsgn = 390
Indicates that the commanded position exceeded the plus servo software limit specified in servo setup causing a C Stop Fault.		
Minus Sofware Limit Exceeded	C Stop Fault	Stat.Statmsgn = 391
Indicates that the commanded position exceeded the minus servo software limit specified in servo setup causing a C Stop Fault.		
User Defined C Stop	C Stop Fault	Stat.Statmsgn = 392
Indicates that the application program generated a C Stop Fault using the C_Stop Function.		
Machine Reference Dim. Error	C Stop Fault	Stat.Statmsgn = 393
Indicates that dimension specified created a math overflow causing a C Stop Fault.		

Feedrate Error	C Stop Fault	Stat.Statmsgn = 394
Indicates that the programmed move feedrate exceeded the velocity limit specified in servo setup or created a math overflow causing a C Stop Fault.		
Distance/Position Dim. Error	C Stop Fault	Stat.Statmsgn = 395
Indicates that programmed distance/position created a math overflow causing a C Stop Fault.		
Part Ref. Dimension Error	C Stop Fault	Stat.Statmsgn = 396
Indicates that the dimension specified created a math overflow causing a C Stop Fault.		
Part Ref. While In Motion	C Stop Fault	Stat.Statmsgn = 397
Indicates that a Part_Ref function specifying this axis was executed while this axis was in motion causing a C Stop Fault.		
Fault Link C Stop Request	C Stop Fault	Stat.Statmsgn = 398
Indicates that the fault link structure, FLNK, indicated that another axis entered the C Stop state causing this axis to C Stop. See the QS_FLT_1 documentation to see how the FMSG output of QS_FLT_1 presents a time stamped message identifying which axis entered the C Stop state first and the fault that caused it to enter the C Stop state.		
SMSG when Warning is present		
M/Slave Master Start Overflow	Warning	Stat.Statmsgn = 420
Indicates that the lock on dimension exceeded 32-bits when converted to feedback units.		
Master Axis Not Available	Warning	Stat.Statmsgn = 421
Indicates that the master axis is not initialized or is programmed at a different interrupt rate or master and slave axis specified for the move instruction are the same axis number.		
Profile Not Found	Warning	Stat.Statmsgn = 422
Indicates that the profile number specified to RATIOPRO does not exist		
FAST Que Axis Out of Range	Warning	Stat.Statmsgn = 423
Indicates that the axis travelled more than 65,535 FU in the opposite direction of the value entered in the DIST input of the FAST_QUE function.		
Servo Timing Error	Warning	Stat.Statmsgn = 424
Indicates that the amount of processor time required to control all of the servo and digitizing axes at the interrupt rates specified in the servo setup function is excessive.		

MMSG - Description of currently active move instruction

No Move Active	Stat.Movemsgn = 440
Default state, indicates no move is currently active	
Position	Stat.Movemsgn = 441
Indicates a move initiated with a POSITION function is executing	
Distance	Stat.Movemsgn = 442
Indicates a move initiated with a DISTANCE function is executing	
Velocity	Stat.Movemsgn = 443
Indicates a move initiated with a VEL_STRT function is executing	
Reference	Stat.Movemsgn = 444
Indicates a move initiated with a FAST_REF or LAD_REF is executing	
RatioPro	Stat.Movemsgn = 445
Indicates a move initiated with a RATIOPRO function is executing	
RatioGear/Sync	Stat.Movemsgn = 446
Indicates a move initiated with a RATIO_GR or RATIOSYN function is executing	
RatioCam	Stat.Movemsgn = 447
Indicates a move initiated with a RATIOCAM function is executing	
RatioSlope	Stat.Movemsgn = 448
Indicates a move initiated with a RATIOSLP function is executing	
RatioReal	Stat.Movemsgn = 449
Indicates a move initiated with a RATIO_RL function is executing	
Move Aborted	Stat.Movemsgn = 450
Indicates that the previous move was terminated by one-shotting CTRL.Abort_Move	
All Moves Aborted	Stat.Movemsgn = 451
Indicates that the previously active move and any move in the queue was terminated by one-shotting CTRL.Abort_All_Moves	

JMSG - Description of Jog Operation State

Jog Plus Active	Stat.Jogmsgn = 460
Indicates that a jog move in the plus direction has been initiated by energizing JOG.Plus	
Jog Minus Active	Stat.Jogmsgn = 461
Indicates that a jog move in the minus direction has been initiated by energizing JOG.Minus	
Jog Terminated	Stat.Jogmsgn = 462
Indicates that a jog plus or minus move has been terminated by de-energizing JOG.Plus or JOG.Minus	
Jog Incremental Plus	StatJogmsgn = 463
Indicates that a distance move in the positive direction has been initiated by energizing JOG.Incremental_Plus	
Jog Incremental Minus	Stat.Jogmsgn = 464
Indicates that a distance move in the negative direction has beed initiated by energizing JOG.Incremental_Minus	
Jog Incremental Done	Stat.Jogmsgn = 465
Indicates that the axis has moved the distance specified by JOG.Incremental_Distance	
Jog Incremental Aborted	Stat.Jogmsgn = 466
Indicates that a Jog Incremental move was aborted by energizing JOG.Incremental_Abort	
Jog Handwheel Active	Stat.Jogmsgn = 467
Indicates that the jog handwheel mode has been initiated by energizing JOG.Handwheel_Mode	
Jog Handwheel Done	Stat.Jogmsgn = 468
Indicates that the jog handwheel mode has been exited by de-energizing JOG.Handwheel_Mode	
Can’t Jog-Fault Present	Stat.Jogmsgn = 469
Indicates that a jog operation was requested while a C Stop or E Stop fault was present	
Can’t Jog-Rate Zero	Stat.Jogmsgn = 470
Indicates that a Jog.Plus or Jog.Minus operation with Jog.Rate set to zero was requested. Or a Jog.Incremental_Plus or Jog.Incremental_Minus operation was requested with Jog.Incremental_Rate set to zero was requested.	
Can’t Jog-Move Active	Stat.Jogmsgn = 471
Indicates that any Jog operation was requested while there was a move already in the queue.	
Can’t Jog Plus-End of Travel	Stat.Jogmsgn = 472
Indicates that a Jog.Plus or Jog.Incremental_Plus was requested while CTRL.Plus_End_of_Travel was energized.	
Can’t Jog Minus-End of Travel	Stat.Jogmsgn = 473
Indicates that a Jog.Minus or Jog.Incremental_Minus was requested while CTRL. Minus_End_of_Travel was energized	
Can’t Jog-End of Travel	Stat.Jogmsgn = 474
Indicates that a Jog.Handwheel_Mode was requested while either CTRL.Plus_End_of_Travel or CTRL.Minus_End_of_Travel was energized	

HMSG - Description of Home Cycle Operation State

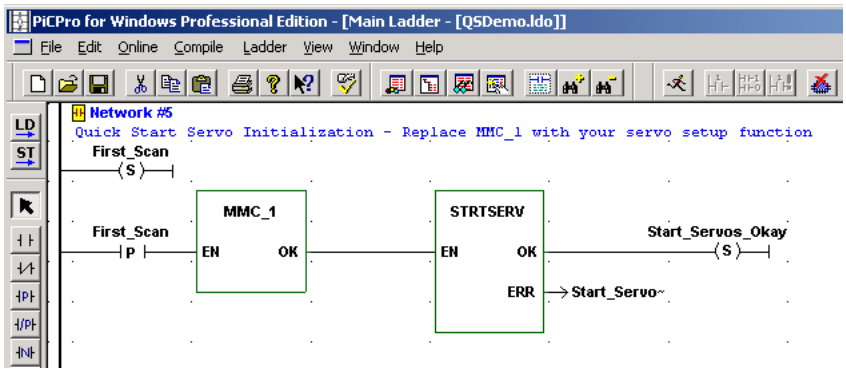
Autobackoff Home Switch	Stat.Homemsgn = 480
Indicates that when Home Cycle was activated the Home Switch was ON and Autobackoff was enabled and the axis is being moved in the opposite of the home direction at the home rate until the home switch is OFF.	
Moving to Home Switch	Stat.Homemsgn = 481
- Indicates that the axis is being moved toward the Home Switch in the home direction at the home rate.	
Moving to Park Position	Stat.Homemsgn = 482
Indicates that homing has completed and the axis is being moved to the park position.	
Home Successful	Stat.Homemsgn = 483
- Indicates that the home cycle has completed successfully. Is cleared at start of home cycle or if a Loss of Feedback fault is detected.	
Can't Home-Fault Present	Stat.Homemsgn = 484
- Indicates that an E Stop or C Stop fault was present when the request to start the home cycle was made.	
Can't Home-Move Active	Stat.Homemsgn = 485
- Indicates that a servo move was already active when the request to start the home cycle was made.	
Can't Home-End of Travel	Stat.Homemsgn = 486
- Indicates CTRL.Plus_End_of_Travel or CTRL.Minus_End_of_Travel is ON preventing the Home cycle from starting.	
Can't Home-On Home Switch	Stat.Homemsgn = 487
- Indicates that when the Home cycle was requested the Home Switch is ON and Home.Auto_Backoff_Home_Switch is OFF preventing the Home cycle from starting.	
Can't Move to Park-Rate Zero	Stat.Homemsgn = 488
- Indicates that Home.Move_to_Park_After_Home is ON and Home.Park_Rate is zero preventing the move to park position from being done.	
Home Aborted	Stat.Homemsgn = 489
- Indicates that the home cycle was aborted without completing successfully.	
Can't Home-Rate Zero	Stat.Homemsgn = 490
- Indicates that the home cycle could not be executed because the Home Rate is zero.	

QS_AIS_1 Outputs

Use the QS_AIS_1 ASFB outputs to see the status of this axis at a glance.

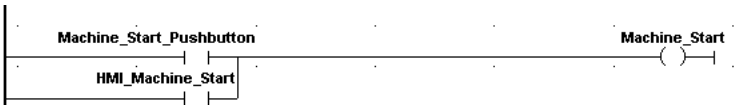
Name	Type	Description
OK	BOOL	ON when servo system intialized properly and all data structure size checks passed
ESTP	BOOL	ON when an E Stop fault is present for this axis
CSTP	BOOL	ON when an C Stop fault is present for this axis
WARN	BOOL	ON when an Warning condition is present for this axis
ERR	UINT	0 – All data structure sizes correct 1001 – STAT structure size incorrect 1002 – CTRL structure size incorrect 1003 – JOG structure size incorrect 1004 – HOME structure size incorrect 1005 – FLNK structure size incorrect 1006 – Axis Number out of range

Initializing the Servo System

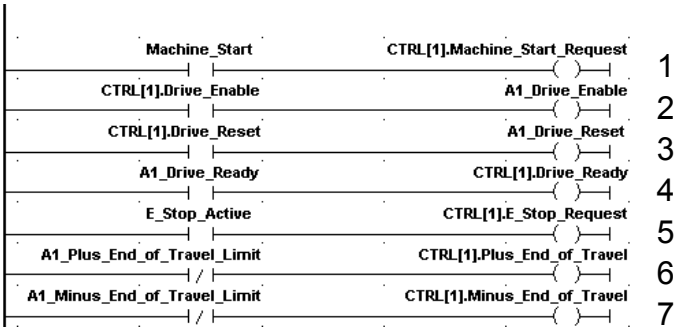


The servo system must be initialized before QS_AIS_1 is called for the first time. MMC_1 is the user defined servo setup function that defines the servo and digitizing axes in this application. STRTSERV is the standard start servo function that activates the real-time servo control software. If Start_Servos_Okay is not ON an error has occurred. See STRTSERV for definition of the error presented on ERR.

Machine Start Sequence



Machine_Start_Pushbutton must be mapped to the control input connected to the pushbutton used to clear faults, initiate machine power on and drive control.



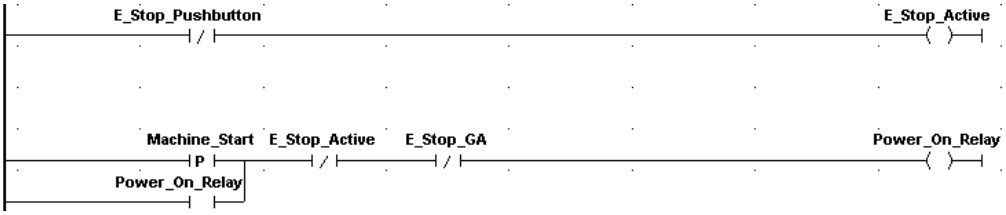
The control logic associated with each axis' QS_AIS_1 provides the interface between the control and the drive and application as described below.

A1_Drive_Enable(2), A1_Drive_Reset(3) and A1_Drive _Ready(4) should be mapped to the control I/O, typically an MMC Axis I/O connector, used to interface to the drive.

A1_Plus_End_of_Travel_Limit (6) and A1_Minus_End_of_Travel_Limit (7) should be mapped to the control input, typically an MMC Aux I/O connector, interfaced to this axis' end of travel limit switches. For rotary axis applications or applications not using end of travel limits they may be deleted.

Pulsing CTRL[1].Machine_Start_Request (1) initiates the machine start sequence. This sequence clears faults specific to the controlled axis, energizes CTRL[1].Drive_Reset (3) for one scan if CTRL[1].Drive_Ready (4) is not on then waits the time specified by CTRL[1].Wait_for_Drive_Ready_Delay (default time 2 seconds). At the end of the time delay faults will be cleared again and CTRL[1].Drive_Enable will be energized, the servo position loop will be closed and fault monitoring will be activated.

Energize, instead of pulsing, CTRL[1].Machine_Start_Request (1), to force QS_AIS_1 to not recognize the fault normally generated by the plus (6) and minus (7) end of travel limits. This allows the user to jog off an end limit switch.



E_Stop_Pushbutton (Emergency Stop) must be mapped to a control input as required by the machine safety directives being followed for this control application. Power_On_Relay must be mapped to a control output for control of the machine power on circuit and must be deenergized as required by the machine safety directives being followed for this control application. Proper adherence to machine safety directives is the responsibility of the user applying control.

Fault Control

QS_AIS_1 monitors the CTRL structure inputs, the servo software status and other axes fault status to provide fault control for the axis it is controlling.

QS_AIS_1 monitors three types of faults:

E Stop - E Stop faults will cause the servo position loop to open and the CTRL[1].Drive_Enable to turn off and will generate a Fault Link E Stop Request to all other axes integrated using FLNK and the QS_FLT_1 fault manager. If an E Stop condition is detected after either a Warning or C Stop condition has been detected the E Stop will take precedence.

C Stop - C Stop faults will cause the axis to decellerate to zero at the controlled stop decel rate specified by the applications servo setup function and will generate a Fault Link C Stop Request to all other axes integrated using FLNK and the QS_FLT_1 fault manager. If a C Stop condition is detected after a Warning condition has been detected the C Stop will take precedence.

Warning - Warning conditions have no effect on the operation of the servo task but are monitored and logged by the fault manager QS_FLT_1.

Faults are cleared by pulsing CTRL[1].Machine_Start_Request. The FLNK input is used to cause this axis to enter a fault condition if another axis' QS_AIS_1 function detects a fault. QS_FLT_1 monitors and logs the overall fault status.

See the SMSG section of this document for a comprehensive list of fault conditions monitored by QS_AIS_1 and the text messages used to present the fault status.

See the QS_FLT_1 document for information regarding the integration of faults in multi-axis systems and to learn how to integrate application specific E Stop and C Stop faults and Warning conditions.

Servo Axis Numbers and Axis Descriptions

The table below shows the relationship between the numbering of the servo axes used by QS_AIS_1, 1 to 32, and the control servo software, 1 to 16, and 101 to 116. It also shows the default text descriptions of the axes used in control text messages (STAT[axis].Description and used by the Cimrex C69 operator interface. The text descriptions of the axes can be changed to be specific to the application. Starting application specific descriptions with "A1 " to "A32 " will aid in troubleshooting. Also, the text descriptions are limited to 22 characters.

Control			Cimrex Message Library	
QS_AIS_1 Axis Number	Servo Software Axis Number	STAT[axis]. Description	Message Number	Text
1	1	Axis 1	1	Axis 1
2	2	Axis 2	2	Axis 2
3	3	Axis 3	3	Axis 3
4	4	Axis 4	4	Axis 4
5	5	Axis 5	5	Axis 5
6	6	Axis 6	6	Axis 6
7	7	Axis 7	7	Axis 7
8	8	Axis 8	8	Axis 8
9	9	Axis 9	9	Axis 9
10	10	Axis 10	10	Axis 10
11	11	Axis 11	11	Axis 11
12	12	Axis 12	12	Axis 12
13	13	Axis 13	13	Axis 13
14	14	Axis 14	14	Axis 14
15	15	Axis 15	15	Axis 15
16	16	Axis 16	16	Axis 16
17	101	Axis 17	17	Axis 17
18	102	Axis 18	18	Axis 18
19	103	Axis 19	19	Axis 19
20	104	Axis 20	20	Axis 20
21	105	Axis 21	21	Axis 21
22	106	Axis 22	22	Axis 22
23	107	Axis 23	23	Axis 23
24	108	Axis 24	24	Axis 24
25	109	Axis 25	25	Axis 25
26	110	Axis 26	26	Axis 26
27	111	Axis 27	27	Axis 27
28	112	Axis 28	28	Axis 28
29	113	Axis 29	29	Axis 29
30	114	Axis 30	30	Axis 30
31	115	Axis 31	31	Axis 31
32	116	Axis 32	32	Axis 32

Digitized (Feedback only) Axis Numbers and Descriptions

The table below shows the relationship between the numbering of the digitized axes used by QS_AIS_1, 1 to 32, and the control servo software, 49 to 80. It also shows the default text descriptions of the digitized axes used in control text messages DCTL[axis].Description and used by the Cimrex C69 operator interface. The text descriptions of digitized axes can be changed to be specific to the application. Starting application specific descriptions with "D1 " to "D32 " will aid in troubleshooting. Also, the text descriptions are limited to 22 characters.

Control			Cimrex Message Library	
QS_AIS_1 Digitized Axis	Servo Software Digitized Axis Number	DCTL[axis]. Description	Message Number	Text
1	49	Digitized 1	49	Digitized 1
2	50	Digitized 2	50	Digitized 2
3	51	Digitized 3	51	Digitized 3
4	52	Digitized 4	52	Digitized 4
5	53	Digitized 5	53	Digitized 5
6	54	Digitized 6	54	Digitized 6
7	55	Digitized 7	55	Digitized 7
8	56	Digitized 8	56	Digitized 8
9	57	Digitized 9	57	Digitized 9
10	58	Digitized 10	58	Digitized 10
11	59	Digitized 11	59	Digitized 11
12	60	Digitized 12	60	Digitized 12
13	61	Digitized 13	61	Digitized 13
14	62	Digitized 14	62	Digitized 14
15	63	Digitized 15	63	Digitized 15
16	64	Digitized 16	64	Digitized 16
17	65	Digitized 17	65	Digitized 17
18	66	Digitized 18	66	Digitized 18
19	67	Digitized 19	67	Digitized 19
20	68	Digitized 20	68	Digitized 20
21	69	Digitized 21	69	Digitized 21
22	70	Digitized 22	70	Digitized 22
23	71	Digitized 23	71	Digitized 23
24	72	Digitized 24	72	Digitized 24
25	73	Digitized 25	73	Digitized 25
26	74	Digitized 26	74	Digitized 26
27	75	Digitized 27	75	Digitized 27
28	76	Digitized 28	76	Digitized 28
29	77	Digitized 29	77	Digitized 29
30	78	Digitized 30	78	Digitized 30
31	79	Digitized 31	79	Digitized 31
32	80	Digitized 32	80	Digitized 32

Troubleshooting QS_AIS_1

In case of difficulties use PiCPro's animation and view capability to observe the operation of QS_AIS_1.

If the OK output of QS_AIS_1 is OFF there is a programming error. Check for:

Problem	Servo position loop won't close
QS_AIS_1 Output Conditions	OK = OFF ERR <> 0 SMSG = *Programming Error*
1 - Check ERR: ERR = 1001 – STAT structure size incorrect ERR = 1002 – CTRL structure size incorrect ERR = 1003 – JOG structure size incorrect ERR = 1004 – HOME structure size incorrect ERR = 1005 – FLNK structure size incorrect If an incorrect structure size error occurred make sure that the correct structure is programmed as the input to QS_AIS_1, that the structures array index is not greater than its array size in software declarations and that a revision changing the number of members in the structure has not occurred. ERR = 1006 – Axis Number out of range QS_AIS_1 supports axis numbers 1 to 32 and translates them to the servo software axis numbers 1 to 16 for AXIS 1 to 16 and 101 to 117 for AXIS 17 to 32.	
Problem	Servo position loop won't close
QS_AIS_1 Output Conditions	OK = OFF ERR = 0 SMSG = *Axis Not Initialized*
The servo axis specified by the QS_AIS_1 Axis input was not included in the servo setup function or an error occurred when the servo setup function was called. 1 - Check the servo setup function network to see if an error was reported. 2 - Open the servo setup function and make sure the axis is defined. 3 - Check the QS_AIS_1 axis input and make sure the axis number is correct.	

If the OK output of QS_AIS_1 is ON but the servo loop won't close. Check for:

Problem	Machine Start won't close the servo position loop
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *First Scan E Stop*
This is the normal state of SMSG after control power on or restart of the scan. SMSG never indicates *First Scan E Stop* after the first time CTRL.Machine_Start_Request is ON. 1 - To clear faults and start the close loop sequence CTRL.Machine_Start_Request must make a transition from OFF to ON. Check to make sure that it is not on all of the time. 2 - Check to make sure CTRL.Machine_Start goes on when the control input used to actuate it goes ON. 3 - Check the C69 fault history for First Scan E Stop faults - each one indicates that the control power has been powered on or the control scan has been stopped and restarted.	

Problem	On Machine Start SMSG immediately indicates *E Stop Request*
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *E Stop Request*
If CTRL.E_Stop_Request goes ON during the Machine Start sequence the Machine Start in Process is cancelled and QS_AIS_1 immediately enters the E Stop state. 1 - Check the logic driving CTRL.E_Stop_Request to determine why it is energizing.	
Problem	After Machine Start SMSG indicates *Machine Start in Process* forever or for a very long time.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *Machine Start in Process*
After Machine Start SMSG will indicate *Machine Start in Process* for the length of time specified by CTRL.Wait_For_Drive_Ready_Delay. At the end of that time Drive Enable will be energized and the servo position loop will close. The intent of the time delay is to allow enough time for the machine power on circuit to provide power to the servo amplifier and for the amplifier to be ready to control the motor. 1 - Check the value in CTRL.Wait_For_Drive_Ready_Delay and reduce it if necessary. The default value is two seconds.	
Problem	After Machine Start the servo position loop won't close
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Drive Not Ready*
This inidicates that CTRL.Drive_Ready was not on at the end of the machine start sequence or has gone off indicating a drive fault. 1 - Check that the drive is powered on and that drive faults are cleared. 2 - Check the plug-and-play axis interface cable between the control and drive. 3 - Check the logic driving CTRL.Drive_Ready. 4 - Check to see that the coil CTRL.Drive_Ready is energizing is mapped to an output in Software Declarations.	
Problem	Servo position loop closed but motor does not seem to be in servo lock
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *Loop Closed - No Fault*
1 - Check to see that the servo amplifier powered on. Check the Power On circuit. 2 - CTRL.Drive_Enable should be ON. Check to see that CTRL.Drive_Enable is connected to the control output used to enable the servo amplifier. Check to see that the coil CTRL.Drive_Ready is energizing is mapped to an output in Software Declarations. 3 - Make sure the servo amplifier is receiving the Drive Enable signal. 4 - Make sure the servo amplifier is configured for velocity mode operation. 5 - Check that the servo amplifier current limits are set to allow current output to the motor. 6 - If the servo amplifier is configured to allow external inputs to prevent motion are they ON. 7 - Check the plug-and-play axis interface cable between the control and drive. 8 - Check that the servo amplifier is receiving the analog velocity command from the control.	

If the OK output of QS_AIS_1 is ON and the servo loop closes but an E Stop fault occurs. Check for:

Problem	Servo position loop closes but fault occurs when motion attempted.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Excess Following Error*
If the axis was previously operating properly: 1 - Check for mechanical bind up of the servo axis. 2 - Check to see that the servo amplifier powered on. Check the Power On circuit. Check for blown fuses. 3 - Check the plug-and-play axis interface cable between the control and drive. 4 - Check that the servo amplifier is receiving the analog velocity command from the control. If this is a new startup and the axis was not previously operating properly: 1 - Use View Status 5 - Limits to check the Maximum Following Error limit. Is it zero? 2 - Use View Status 4 - Tuning to check Position Loop P Gain. Is it zero? 3 - Make sure the servo amplifier is configured for velocity mode operation. 4 - Check to make sure servo amplifier Velocity Loop gain settings are appropriate. 5 - Check that the servo amplifier current limits are set to allow current output to the motor. 6 - If the servo amplifier is configured to allow external inputs to prevent motion are they ON.	
Problem	Servo position loop closes but fault occurs
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *E Stop Request*
1 - Check the logic driving CTRL.E_Stop_Request to determine why it is energizing.	
Problem	Servo position loop closes but no motion occurs when motion attempted.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = OFF SMSG = *Loop Closed - No Fault* MMSG = *indicates type of move requested*
1 - Use Servo Setup to check that the Acceleration and Deceleration Ramp rates are not zero or so small that the axis velocity ramp up will take a very long time.	
Problem	Servo position loop closes but Loss of Feedback fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Loss of Feedback*
1 - Check the plug-and-play axis interface cable between the control and drive. 2 - Check cable routing of plug-and-play axis interface cable to make sure it is not near motor power cables or other electrical noise sources.	
Problem	Servo position loop closes but Math Overflow Error fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Math Overflow Error*
1 - See the E_Errors function description of the Overflow Error for a detailed description of the cause of this fault.	

Problem	Servo position loop closes but User Set E Stop fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *User Set E Stop*
1 - The application program is executing the E_STOP function. Use the QS_FLT_1 application specific E Stop capability instead of E_STOP to generate an E Stop fault with an application specific description.	
Problem	Servo position loop closes but Plus End of Travel fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Plus End of Travel*
1 - Check the logic driving CTRL.Plus_End_of_Travel, it turned on causing this fault. Note that energizing machine start will ignore the plus end of travel limit and QS_AIS_1 will allow jogging in the minus direction while the plus end of travel is on.	
Problem	Servo position loop closes but Minus End of Travel fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Minus End of Travel*
1 - Check the logic driving CTRL.Minus_End_of_Travel, it turned on causing this fault. Note that energizing machine start will ignore the minus end of travel limit and QS_AIS_1 will allow jogging in the plus direction while the minus end of travel is on.	
Problem	Servo position loop closes but Fault Link E Stop Request fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 ESTP = ON SMSG = *Fault Link E Stop Request*
1 - Check the FSTR output of QS_FLT_1, it will indicate which servo or digitized axis entered the E Stop Fault state first causing this axis to enter E Stop via a Fault Link E Stop Request.	
Problem	After the first fault occurs Machine Start won't clear faults.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 SMSG = *last fault that occurred*
1 - To clear faults and start the close loop sequence CTRL.Machine_Start_Request must make a transition from OFF to ON. Check to make sure that it is not on all of the time. 2 - Check to make sure CTRL.Machine_Start goes on when the control input used to actuate it goes ON.	

If the OK output of QS_AIS_1 is ON and the servo loop closes but a C Stop fault occurs. Check for:

Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *C Stop Request*
1 - Check the logic driving CTRL.C_Stop_Request to determine why it is energizing.	

Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Plus Software Endlimit Exceeded*
1 - Use View Status 5 - Limits to check Ignore Software Limits and the Soft Plus Limit. In servo setup Ignore Soft Limits till Referenced can be selected.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Minus Software Endlimit Exceeded*
1 - Use View Status 5 - Limits to check Ignore Software Limits and the Soft Minus Limit. In servo setup Ignore Soft Limits till Referenced can be selected.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *User Defined C Stop*
1 - The application program is executing the C_STOP function. Use the QS_FLT_1 application specific C Stop capability instead of C_STOP to generate an C Stop fault with an application specific description.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Machine Ref. Dim. Error*
1 - A calculation overflow occurred when scaling data, see C_Errors function for detailed explanation.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Feedrate Error*
1 - A calculation overflow occurred when scaling data or the specified feedrate exceeded the limit specified in the servo setup function, see C_Errors function for detailed explanation.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Distance/Position Dim. Error*
1 - A calculation overflow occurred when scaling data, see C_Errors function for detailed explanation.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Par Ref. Dimension Error*
1 - A calculation overflow occurred when scaling data, see C_Errors function for detailed explanation.	

Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Part Ref. While in Motion*
1 - The application program called the Part_Ref function while a move was in process. Correct the application program logic so Part_Ref is only called with no move active.	
Problem	Servo position loop closes but fault occurs.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 CSTP = ON SMSG = *Fault Link C Stop Request*
1 - Check the FSTR output of QS_FLT_1, it will indicate which servo or digitized axis entered the C Stop Fault state first causing this axis to enter C Stop via a Fault Link C Stop Request.	

If the OK output of QS_AIS_1 is ON and the servo loop closes but a Warning condition is reported. Check for:

Problem	Servo position loop closes but warning condition is reported.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 WARN = ON SMSG = *M/Slave Master Start Overflow*
1 - A calculation overflow occurred when scaling data, see P_Errors function for detailed explanation.	
Problem	Servo position loop closes but warning condition is reported.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 WARN = ON SMSG = *Master Axis Not Available*
1 - The master axis specified in a ratio move was not initialized in the servo setup function or was otherwise unavailable, see P_Errors function for detailed explanation.	
Problem	Servo position loop closes but warning condition is reported.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 WARN = ON SMSG = *Profile Not Found*
1 - The profile number specified in a RatioPro function was not valid.	
Problem	Servo position loop closes but warning condition is reported.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 WARN = ON SMSG = *FAST Que Axis Out of Range*
1 - The axis travelled more than 65,535 FU in the opposite direction of the value entered in the DIST input of the FAST_QUE function.	

Problem	Servo position loop closes but warning condition is reported.
QS_AIS_1 Output Conditions	OK = ON ERR = 0 WARN = ON SMSG = *Servo Timing Error*
1 - The amount of processor time required to control all of the servo and digitizing axes at the interrupt rates specified in the servo setup function is excessive. See the servo update rate specification table for the processor being used.	