Reference Manual

Allen-Bradley

Integrating the Mining, Mineral, and Cement Library (MMCL) into RSLogix 5000 Software

Important User Information

Solid-state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid-state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid-state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

	WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
	ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
	SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
	BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

Chapter 6

Motor Control Modules

Chapter 7

Process Control Modules

Chapter 8

Simulation Modules

Appendix A

Wiring Diagram Analog Input - PID Module - Actuator Analog Controlled

Appendix B

Introduction

The Mining, Mineral, and Cement Library (MMCL) provides the tools (software modules) to create highly standardized programs for mining, mineral, and cement plants. The aim is to simplify the application by employing a module technique that eases comprehension and minimizes errors. Each of the modules contains the following pieces.

Template display (Operator Station) for FactoryTalk View SE, the Human Machine Interface (HMI)

Add-On Instruction, or user-defined function block, for the ControlLogix controller

Both the template displays and the Add-On Instructions are designed to match and form a ready-to-use module that is configured by filling in the blanks on the configuration screen. Using the MMCL reduces time and costs for both programming and plant commissioning, and makes the application more maintainable. The modular process and motor control sequences are supported by the following modules for the HMI software and the controller.

Modules	Acronym
System Group Module	SysGrp
Control Group Module	CtrlGrp
Machine Group Module	MaGrp
Inter Process Communication	IPCom
Motor Modules for Normal, Reverse and Damper Starters	MotorN, MotorR, MotorD
Motor Overload Relay	E3p
Sub-System Module	SubSys
Valve Modules with 1 or 2 Coils	Valve1, Valve2
Digital Input Module Type 1..4	DigInp Type 1..4
Digital Input Module with Two Inputs	DigInp2
Analog Input and Analog Input Control Module	AnaInp, AnaInpC
Proportional Integral Derivative Module	PidMod
Actuator Module and Actuator Positioning Module	ActMod, ActPos
Digital Pulse Input Module	DigPulse

Users should have basic experience in writing PLC programs. We assume that users are familiar with ControlLogix data structures and instruction sets and with the FactoryTalk View SE software.

Control of a mining, mineral, and cement plant requires many repetitive functions, for example, belt conveyors with rope and drift switches, actuators with limit and torque switches, pumps with pressure and flow indication, etc. The MMCL standardizes these functions by applying a common style and method of programming. The goal is a user program that is independent of individual programmers.

The MMCL relates to Holcim's concept of overall plant automation and its type of process controls and visulation; however, it may be adapted to other concepts as needed. The MMCL provides the following functions:

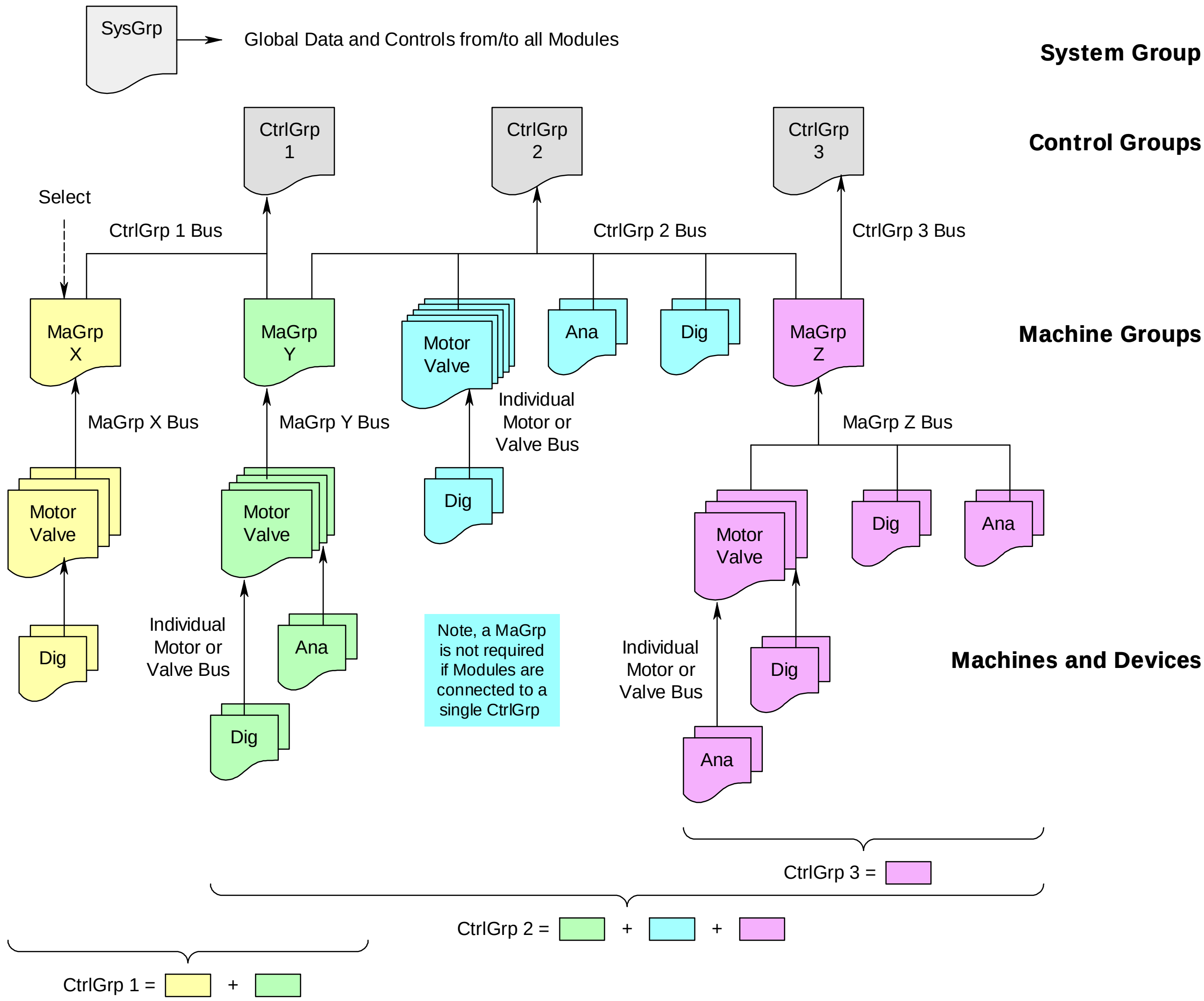
- Easy configuration with ready-to-use modules
- Structured programs for best maintainability
- Detailed alarming and alarm message handling
- Standardized motor and analog measurement control
- High level of comfort for both, operators and maintenance personnel

Concepts

Basic Structure and Module Interconnection

Similar to the plant equipment controlled, the application program can be split into functional Groups, Machines and Devices that are supported by ready-to-use software modules.

Modules are grouped by referencing a parent module that is specified by simply entering the tag name of its ParentBus. Standard information (for example, group control signals and alarms) is automatically transmitted from and to the master control group modules. The user program consists of mostly process interlocking controls rather than motor, valve, and analog controls that are normally difficult to maintain. The majority of the setup is done by filling in the blanks with types of modules (classes). A sample application program structure using standardized modules and bus interfaces (arrows) may look similar to the following flow chart.



The System Group (SysGrp) is programmed once only per controller, it receives common parameters from and provides common parameters to all modules. Control Groups (CtrlGrp), Machine Groups (MaGrp), and other modules interact as follows:

- CtrlGrp 1 is a parent (superior control) module to both, MaGrp X and MaGrp Y
- CtrlGrp 2 is a parent module to MaGrp Y, some directly controlled Motor/Valve + Dig/Ana modules, and MaGrp Z
- CtrlGrp 3 is a parent module to MaGrp Z
- MaGrp X may be selected/deselected and is controlled only one parent module, CtrlGrp 1
- MaGrp Y is controlled by both, CtrlGrp 1 and CtrlGrp 2
- MaGrp Z is controlled by both, CtrlGrp 2 and CtrlGrp 3

Control Philosophy

The control philosophy is reflected in the Human Machine Interface (HMI) of the control system. The idea is to indicate the necessary information required to run the plant, however, in case of a problem, alert the operator or manager with detailed information. On the other hand, if any task is requested, the system must react with meaningful information.

The goal of the control philosophy is for the operator or manager to not be flooded by unnecessary information so they can concentrate on plant performance and optimizing production. Consequently, the system will save costs, since it does not require experienced personnel or programming knowledge for normal trouble shooting: In case of a problem, the system indicates the prime cause clearly in detail in order to guide maintenance personnel to the faulty device.

For this reason, every module only releases the prime cause in case of an alarm and suppresses irrelevant information that would hamper trouble shooting. For example, it is not useful to indicate that a contactor has tripped, just because a safety rope on a belt conveyor was pulled, even though the contactor will trip to stop the drive. A contactor failure might lead to a call to an electrician, whereas the problem with the rope switch can be reset by other maintenance personnel.

Types of Devices and Messages

Each module can be setup to send Warning alarms only, or both Failure and Warning alarms as needed. Some modules may also send Diagnostic alarms.

- Failure alarms (red, severity=1) result from the malfunction of a device or a condition, causing a stoppage of the production process associated with a particular group.
- Warning alarms (yellow, severity=2) result from the malfunction of a device or a condition, which does not immediately cause a stoppage in the production line, but action should be taken to correct the fault.
- Diagnostic alarms (orange, severity=3) indicate, for example, that a limit switch failed or an input of a speed detector is bridged out (jumpered).

The modules are setup as Failure drives/devices and Warning drives/devices. An example of a Failure drive is a conveyor in a group of conveyors that are used for production. A Warning device may be a dust collector on a conveyor system that is not used for production. A level detector or an analog measurement may be considered a Failure device that stops production if the maximum or hihi level is reached. It may send a Warning alarm at high level, and it may release a Diagnostic alarm if a replacement or substitution value is entered for its input.

Each module can also release Status information that results from a change of state. It indicates, for example, that a valve is open or a group is running.

The MMCL automatically configures the following:

- Only the first alarm, or the prime cause, is released and sent to FactoryTalk View SE
- Alarms are stored until they are received/recorded and acknowledged by the HMI
- Warnings and Failures are collected and sent to the appropriate group modules.

Program Tasks

The controller program (RSLogix 5000 software) consists of two parts: the user software (application software) and the MMCL using Add-On Instructions.

User Software

The user software covers all application programs required to interlock devices and control the plant. The user calls the MMCL Add-On Instruction modules as CtrlGrp, MotorX, etc. within the programs. The User Software differs from controller to controller; it is documented individually per controller.

MMCL Software

The MMCL design provides scanning of all modules and ensures that the controller loading and cycle time are within the specified guidelines. Either continuous or periodic tasks can be used but they are not recommended to be used at the same time. To achieve better system performance, periodic task is a preference but certain rules should be followed when using multiple periodic tasks.

System Control is supported by the System Group Module (SysGrp) that takes care of master control and alarming such as Power Dip detection or communication supervision. The module includes an operator interface which is supplied as a ready-to-use HMI display.

Motor Control is supported by the following modules that include an operator interface which is supplied as a ready-to-use HMI display.

- The Control Group Module (CtrlGrp) provides control functions for the operator interface and control for groups of machines and devices. It accepts HMI commands, start-up and master interlocks and it supports alarm control and acknowledgement, local and remote selection, etc.
- The Machine Group Module (MaGrp) lets you split an entire Control Group into sub-groups that can be selected/deselected or combined with other Control Group Modules. It can be used to preselect and occupy machines, or share and switch machines on the fly.
- The Inter Process Communication Module (IPCom) is used for Inter Process Communication between two controllers.
- The Motor Control and Valve Modules (MotorN/R/D, SubSys, and Valve1/2) support all standard interlocking and supervision for starters of any type: Normal D.O.L starters, Forward/Reverse, Damper/Flap, Sub-Control Systems, etc., and Valves with one or two coils. The modules are linked with the Machine Group or Control Group Module and support alarm control by creating messages that can be acknowledged individually or by the Control Group Module.
- The Digital Input Module (DigInp and DigInp2) supervises input signals such as limit switches, level, speed detectors, etc. The module can be setup to directly take pulse signals off a speed detector. The module can be linked with a Motor Control and Valve Module for dynamic or static inputs, or directly with the Machine or Control Group Module for steady state inputs.
- The Digital Pulse Input Module (DigPulse) supervises the pulse signal of the same rotation devices such as Belt conveyers. This module takes care of differed conditions like start/stop or continuous running.

Process Control is supported by the following modules, all including an operator interface that is supplied by a ready-to-use HMI display.

- The Analog Input and Control Modules (AnaInp and AnaInpC) support scaling, filtering and supervising analog inputs. Values are conditioned and checked for underflow, overflow and limits in order to create alarms and digital outputs that may be used for controlling any discrete device.
- The Proportional Integral Derivative Module (PidMod) uses the controller standard PID instruction to control a process variable such as flow, pressure, temperature, or level. The module can be selected for manual, automatic, or external operation and is linked with a Machine or Control Group Module.
- The Actuator and Positioning Modules (ActMod and ActPos) are used to scale and set analog outputs and position digital starters. The module adjusts an actuator according to a setpoint that can be entered by the operator or taken from a PID module; it is linked with a Machine or Control Group Module.

User Program Considerations

The User Software covers all application programs required to interlock devices and control the plant. The user calls the MMCL Add-On Instructions modules as CtrlGrp, MotorX, etc. within the programs. The User Software differs from controller to controller; it is documented individually per controller.

Tag Naming Convention

- Use the Asset Code (AC) where applicable, especially for inputs, outputs and modules
- Definition:

  Upper case only = AC
  Example A, M1, BC, 512, 4C, PV

  Lower case or mixed = other signal/abbreviation
  Example: i, Inp, Ala, Pv

- Use 3-digit abbreviation where applicable (RSLogix supports 40 char max in Tag names)

123-456.78:90 = Asset Code (AC) character, delimiter and position

iGG#_MM#_D#_AAA = physical input (tag name, single element)

oGG#_MM#_D#_AAA = physical output (tag name, single element)

xGG#_MM#_D#_AAA = virtual input/output from/to subsystem (tag name, single element)

_GG#_MM#_D#.AAA = all other signals (tag name and member of tag)

| | | |

| | | Signal AAA if upper case = AC, else member of tag

| | Device D#

| Machine MM# (#=number 0..9 or A..Z)

Group GG#

Programming Rules

I/O modules can only be referenced/set once throughout the program. This is necessary because the I/O modules are updated asynchronously to the program scan. If the I/O module is applied at several locations, use the mapped tag within the program.

- Outputs can be programmed after the Add-On Instruction if they are not directly set by or entered in the Add-On Instructions. This prevents unpredictable starts at initialization. Note that the Add-On Instruction automatically clears all outputs at start-up of the controller.

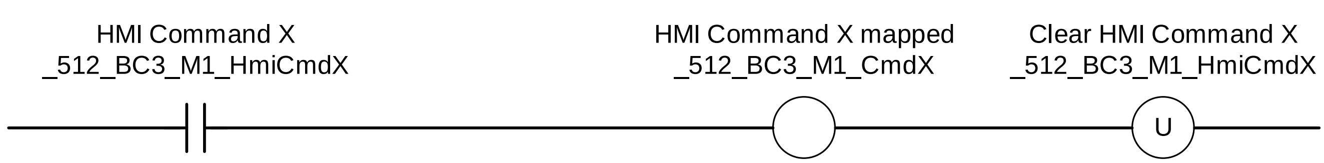
- For all other outputs set by the application program, you must reset the outputs at start-up. Note that the ControlLogix controller clears all COIL outputs at start-up.

- Unused module inputs should be fixed by programming unconditional rungs unless default values are specified for such inputs. Unused dynamic module inputs must be programmed by the user, for example, contactor feedback .R = output .D for the MotorN module.

- Timers must be run every program scan (this is especially true for timers in subroutines)

- HMI command inputs are treated as exceptions due to asynchronous input update. Commands are set by the HMI and cleared by the module after examining the information. Therefore the commands mapped to the modules cannot be used in the application program, use module status information instead.

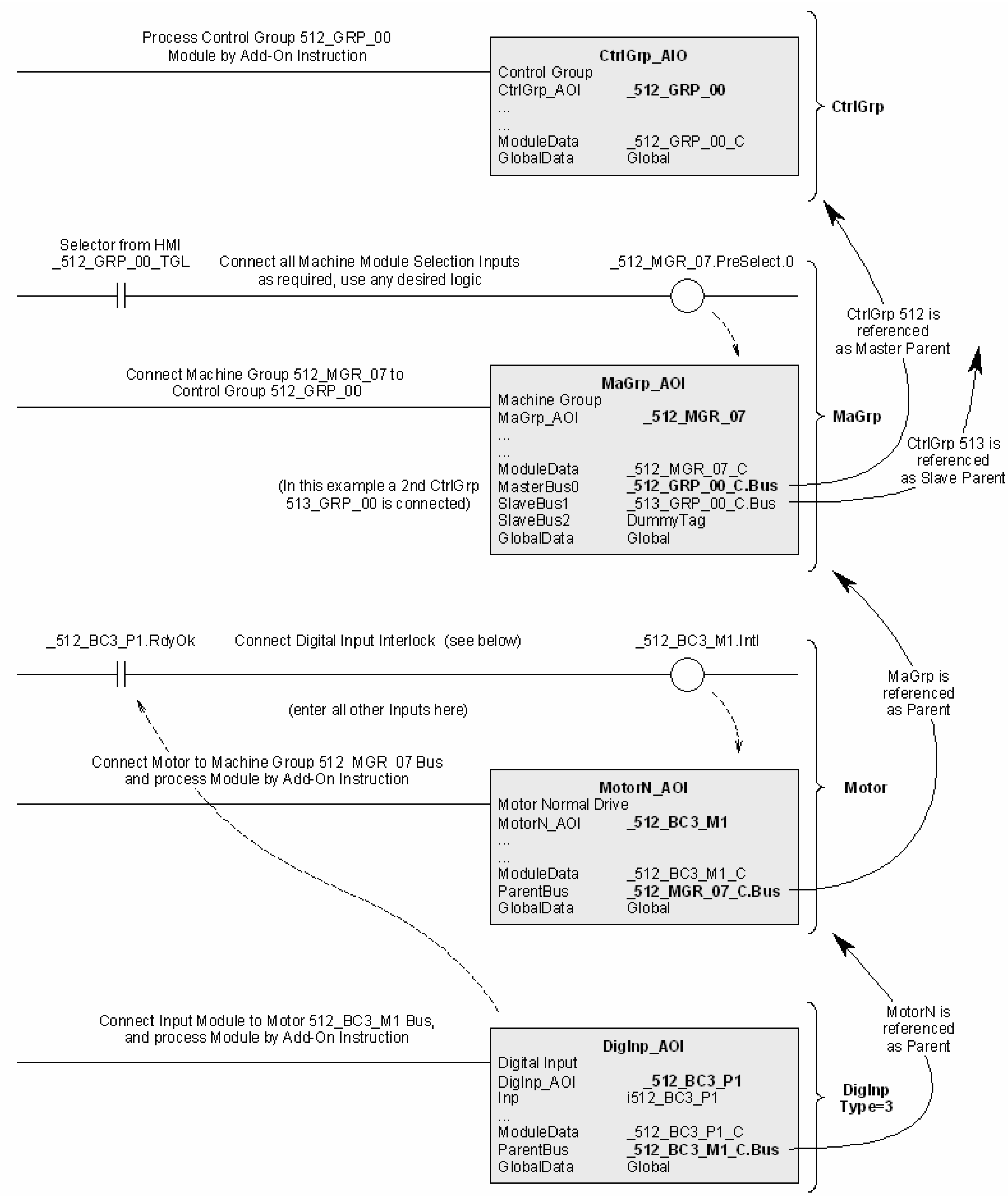
  To apply the same method, use separate HMI command inputs as required. Map the input to a second tag for further usage, and clear the HMI command after examination:



Bus Linking for Control Group, Machine Group and Motor with DigInp

Modules are linked by referencing a parent module that is specified by entering the tag name of its ParentBus. Standard information, for example, group control signals and alarms, is then automatically transmitted from and to the parent module.

As an example, a Digital Input for interlocking a motor is linked to its motor, the Motor is linked to its Machine Group and the Machine Group is linked to its Master and/or Slave Control Group:

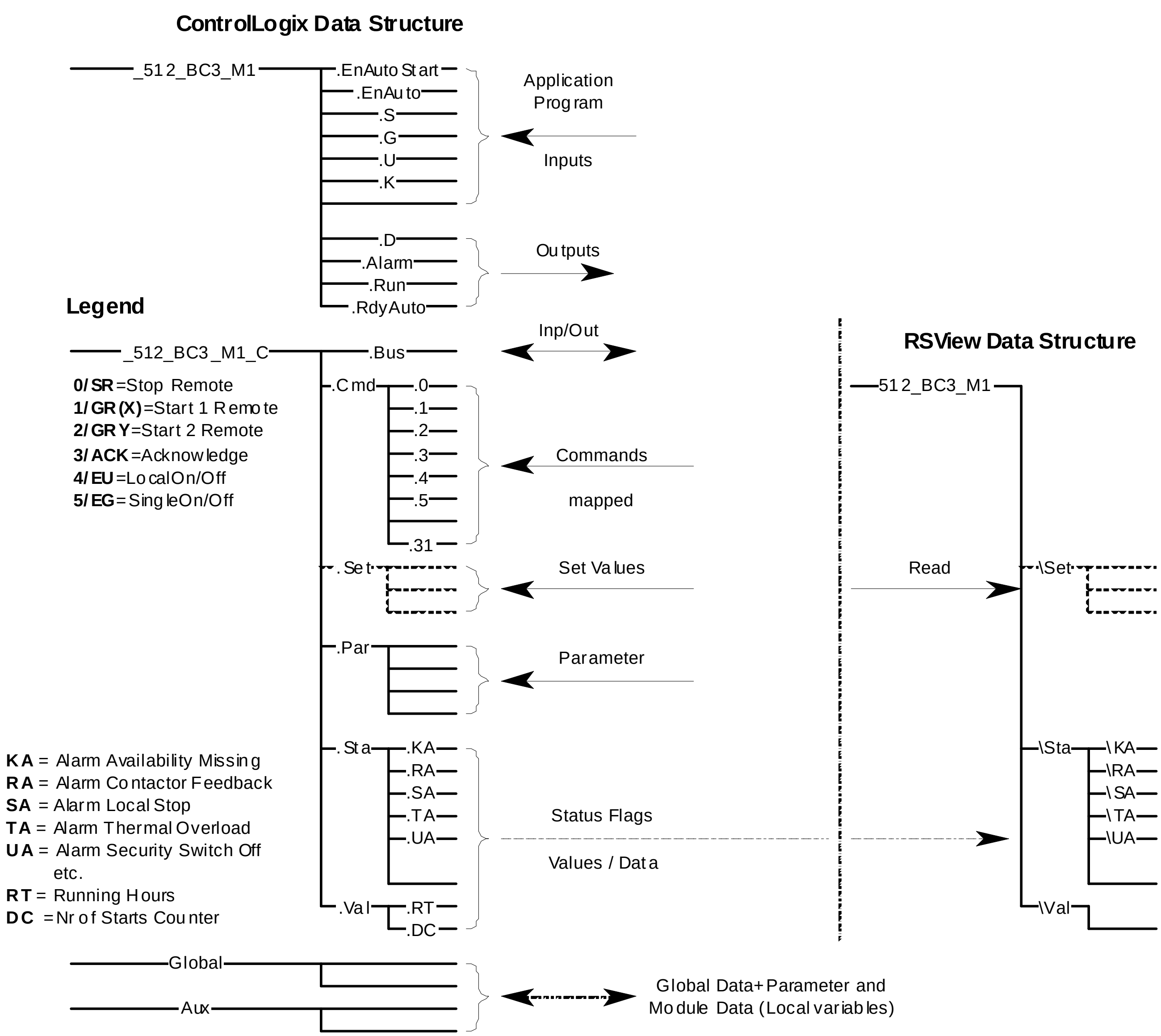


Notes:

Standard Software and Module Processing

Generic Module Data Structure - Interface to FactoryTalk View

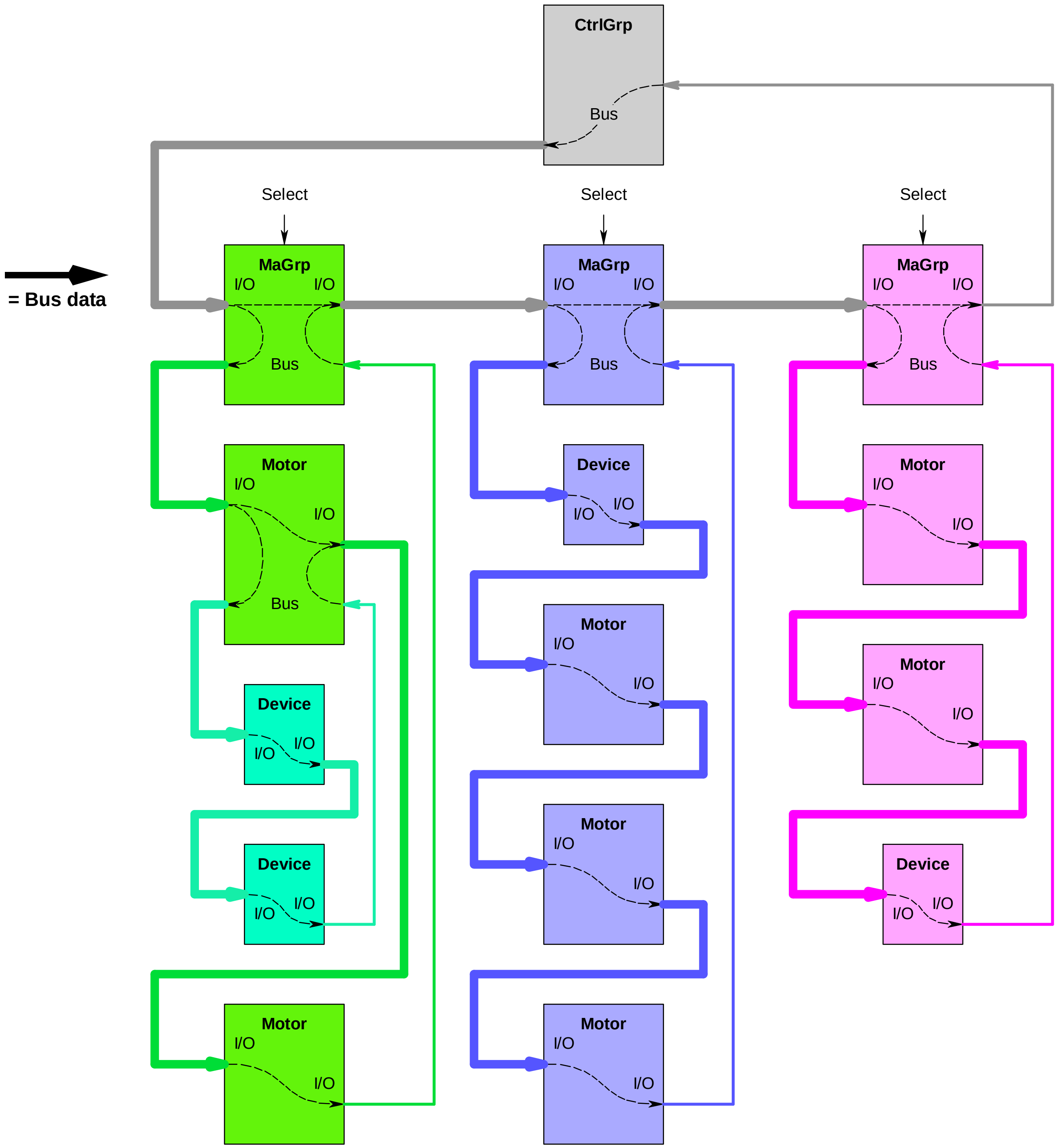
All modules are based on the following data structure.



Member Name	I/O	Required	Visible	Description	Type
{Module}.EnAutoStart	In		X	Enable Automatic Start	BOOL
{Module}.EnAuto	In		X	Enable Automatic Operation	BOOL
{Module}.S	In	X	X	Local Stop Input (0=Stop)	BOOL
{Module}.G	In	X	X	Local Start Input (Go)	BOOL
{Module}.U	In	X	X	Local Isolator / Safety SW (0=OFF)	BOOL
etc.					
{Module}.D	Out	X	X	Digital Output Contactor	BOOL
{Module}.Alarm	Out		X	Device Alarm (W or F)	BOOL
{Module}.Run	Out			Running in Any Mode	BOOL
{Module}.RdyAuto	Out		X	Ready Running in Auto Mode	BOOL
{Module}_C...	I/O	X	X	Module Control Data and Parameter	Struct
{Parent}_C.Bus	I/O	X	X	Parent Control Bus Interface	DINT
Global…	I/O	X	X	Global Data, Standard Parameter, Clock etc.	Struct
Aux…	Loc			Auxiliary Local Data, not accessible by user	Struct

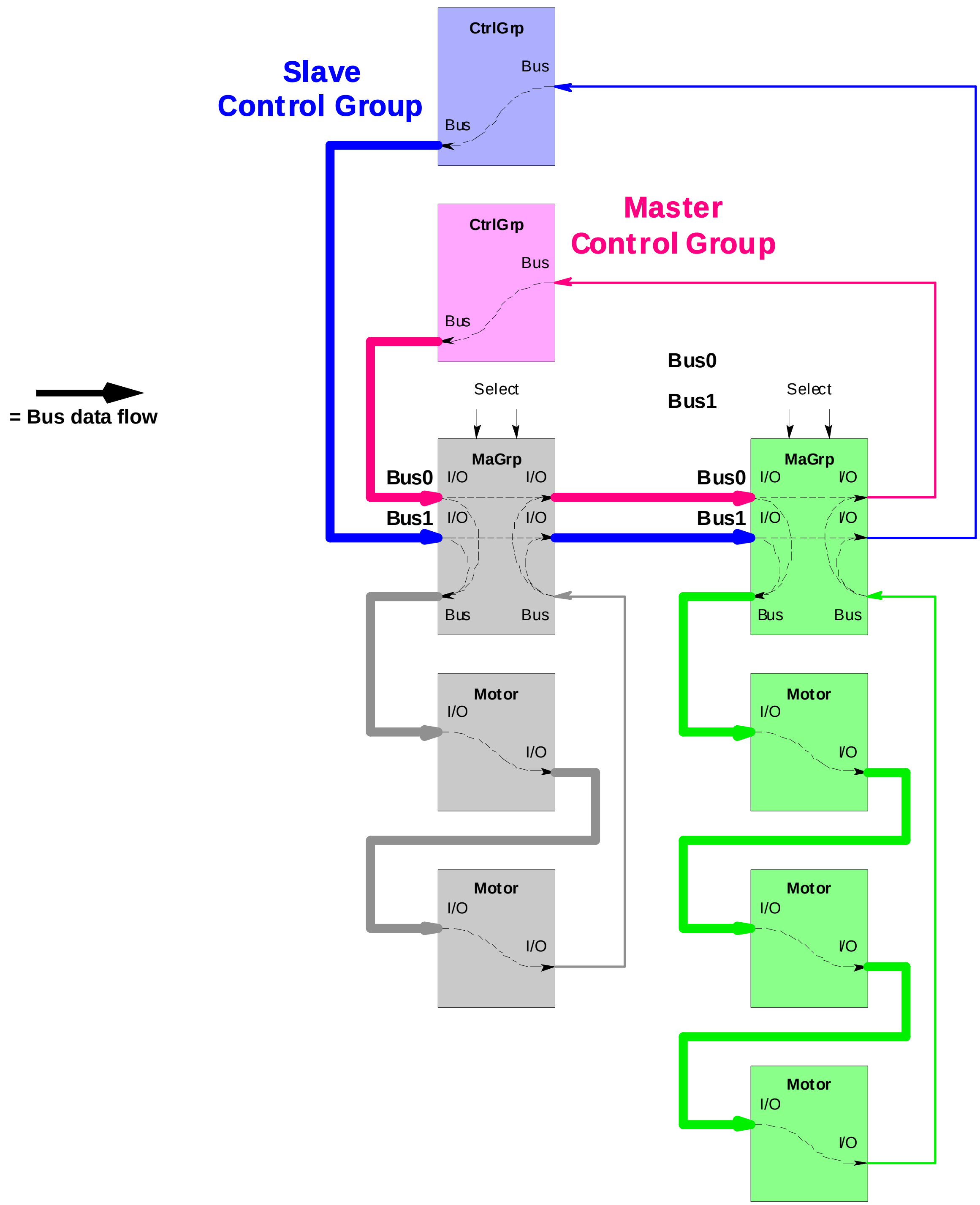
Module Bus Interface, Principle Dataflow

All modules are connected to a Control Group (CtrlGrp) by a Bus Interface, which is a 32-Bit shared memory, or DINT, for any desired group of devices. Every module reads from and writes to its parent bus in order to receive information from or send information to the parent module. The top module is always a CtrlGrp that has no other parent module.



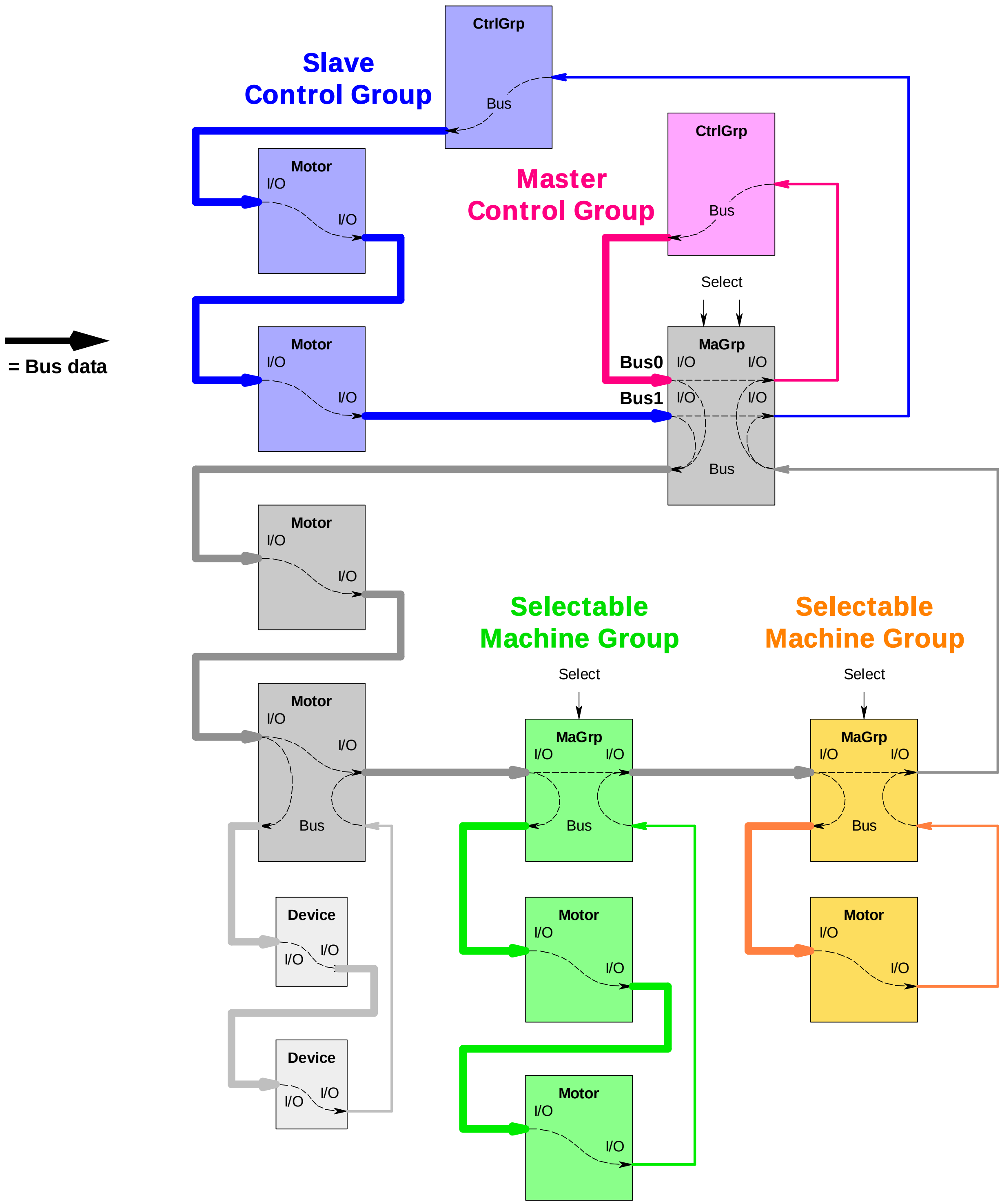
Shared Groups of Machines or Devices

A machine group is made up of associated machines or devices that perform a particular function. The Machine Group Module (MaGrp) can be selected or shared as part of a Master Control Group and/or multiple Slave Groups.



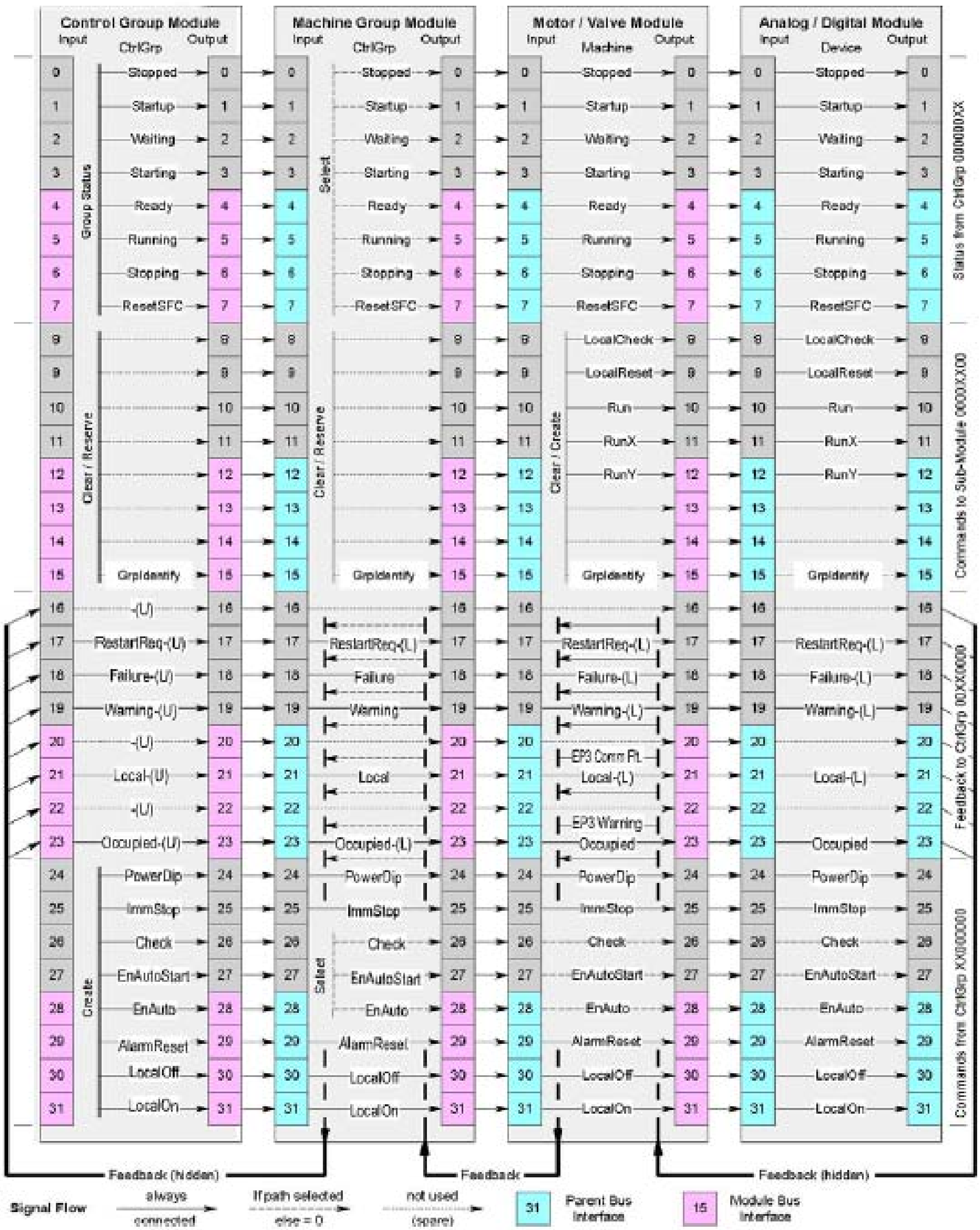
Combined Groups of Machines or Devices

Any combination of modules is allowed to form groups of devices that can be started and stopped by CtrlGrp modules or selected and deselected by MaGrp modules.



The 32-Bit Module Bus Interface Signals

All modules are of a particular group interconnected by the following 32-Bit Bus.

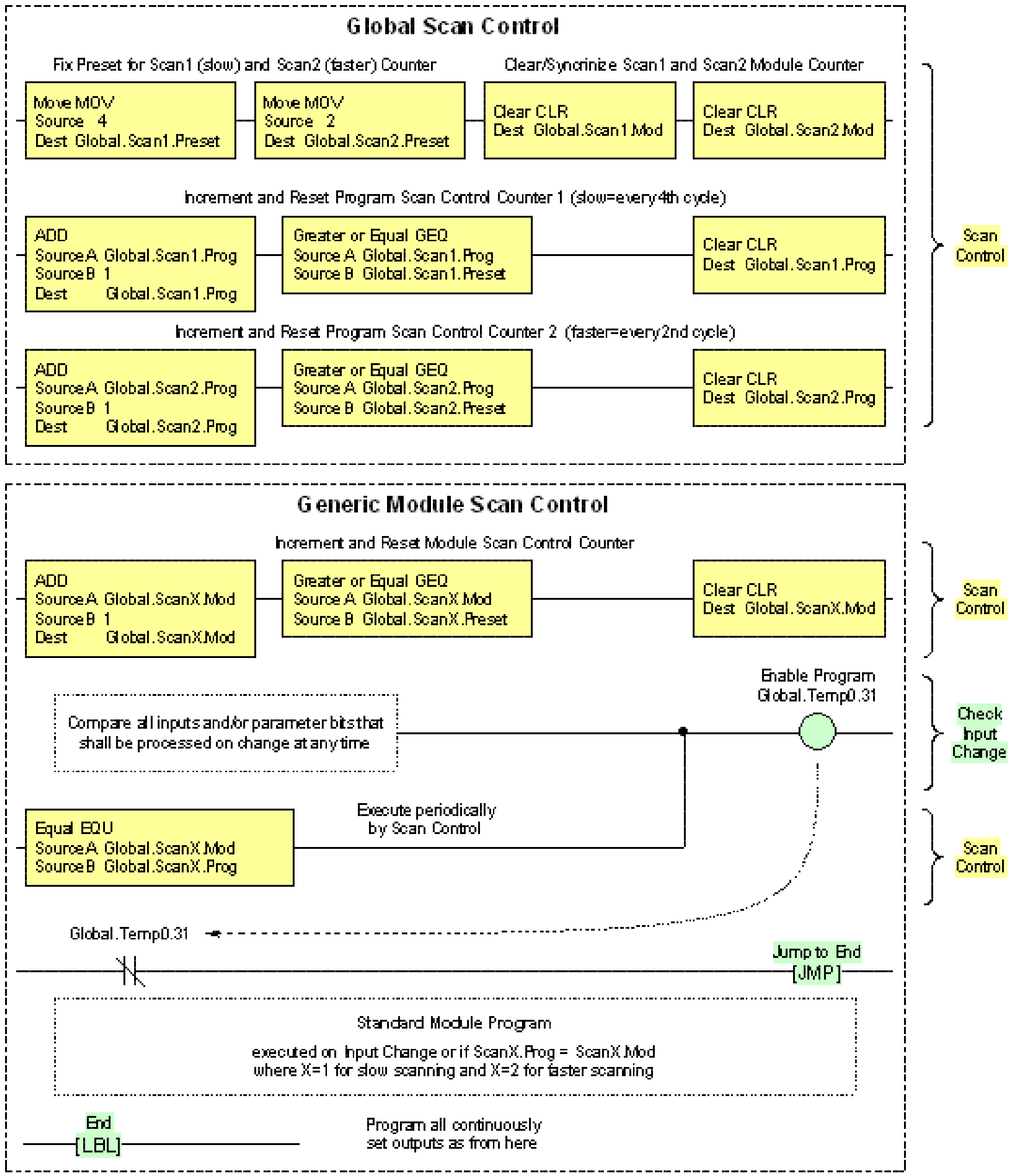


Module Bus Interface Data Exchange Overview

Module / Bus	Selection	Command Bits	Marshalling	Feedback Bits
Control Group Module **CtrlGrp**	Always	Mask=**0xFF0000FF** 31/30 LocalOn/Off 29 AlarmReset 28 EnAuto 27 EnAutoStart 26 Check 25 ImmStop 24 PowerDip 7..0 Group Status	Selected Bits used/cleared and created	Mask=**0x00FF0000** 23 Occupied 22 (reserve) 21 Local 20 (reserve) 19 Warning 18 Failure 17 RestartReq 16 (reserve)
Machine Group Master Bus 0 **MaGrp**	**Selected.0=OFF** (Always)	Mask=**0xE3000000** 31/30 LocalOn/Off 29 AlarmReset 25 ImmStop 24 PowerDip	All Master Bits moved to Return Bus 0	Mask=**0x002C0000** 21 Local 19 Warning 18 Failure
	Selected.0=ON	Mask=**0xFF0080FF** 31/30 LocalOn/Off 29 AlarmReset 28 EnAuto 27 EnAutoStart 26 Check 25 ImmStop 24 PowerDip 7..0 Group Status		Mask=**0x00FF0000** 23 Occupied 21 Local 19 Warning 18 Failure 17 RestartReq
Machine Group Slave Bus 1..x	**Selected.x=OFF** (Always)	Mask=**0x00000000** 31..0 (no commands)	All Slave Bits moved to Return Bus x	Mask=**0x00000000** 31..0 (no feedback)
	Selected.x=ON	Mask=**0x3F0080FF** 29 AlarmReset 28 EnAuto 27 EnAutoStart 26 Check 25 ImmStop 24 PowerDip 15 GrpIdentity 7..0 Group Status	Selec-ted Bits moved to Machine Bus Selec-ted Bits added to Group Buses	Mask=**0x000F0000** 19 Warning 18 Failure 17 RestartReq
Machine Group Interface	Always	Mask=**0xFF000000** 31..24 Group Cmd 7..0 Group Status		Mask=**0x00FF0000** 21 Local 19 Warning 18 Failure 17 RestartReq
Motor/Valve Sub System **Motor** **Valve**	Always	Mask=**0xFF00FFFF** 31..24 Group Cmd 12 Run direction Y 11 Run direction X 10 Run any direction 9 LocalReset 8 LocalCheck 7..0 Group Status	Bits added to Device Bus and/or set to Return Bus	Mask=**0x00FF0000** 21 Local 19 Warning 18 Failure 17 RestartReq
Device Module **DigInp** **AnaInp**	Always	Mask=**0xFF00FFFF** 31..24 Group Cmd 12..7 Machine Cmd 7..0 Group Status	Selected Bits set to Return Bus	Mask=**0x00FF0000** 21 Local 19 Warning 18 Failure 17 RestartReq

Module Processing and Generic Program Structures

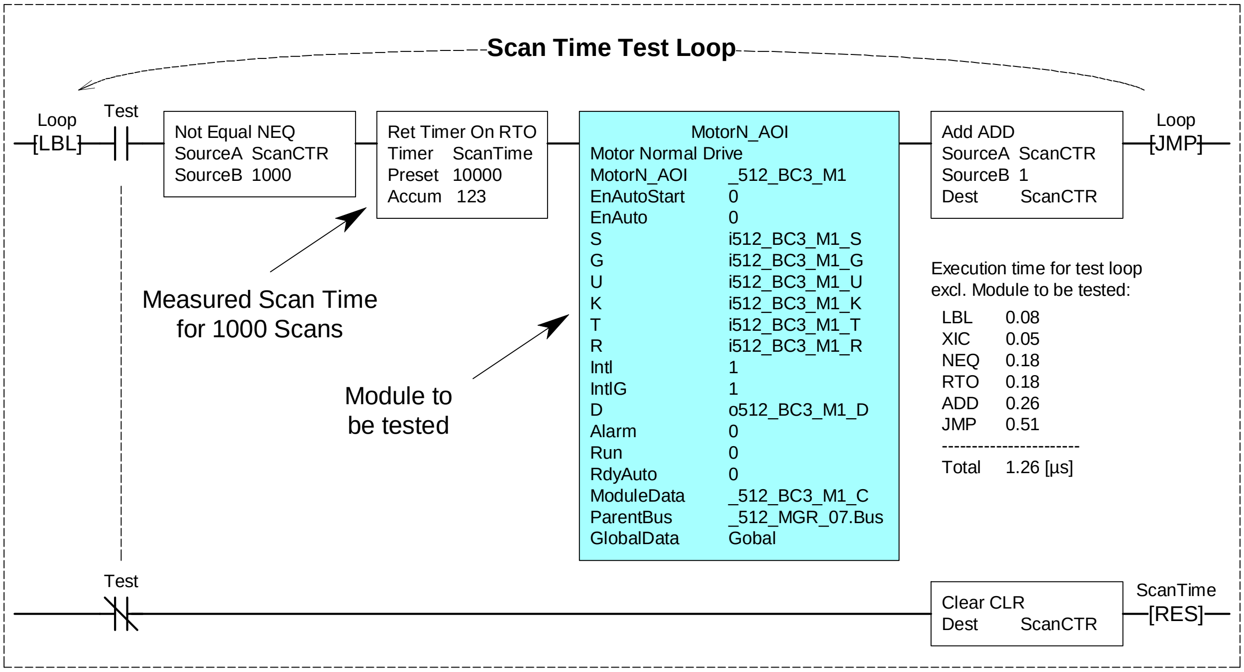
The standard modules do not need to be processed every program scan. This unburdens the controller and speeds up the basic cycle time. The user program can be clearly presented in one continuous or periodic task; in other words, it does not need to be scattered over different tasks. The method used for the standard software is to process the modules only at defined scan rates (Scan1 or Scan2) but react immediately on any input change.



- Use Global.Scan1 for a slow module scan rate and Global.Scan2 for a faster module scan rate
- To execute Module Program every Scan, omit both Scan Control and Check Input Change

Module Scan Time Measurement and Controller Cycle Time Estimation

All Module Scan Times are measured by the following test routine executed in a 500 ms Periodic Task in order to omit System Overhead Time by interrupts.



The Modules are scanned 1000 times; the processing time is accumulated by retentive timer ScanTime. For the benchmark the scan rates are every scan for Group Modules, every 2nd scan for Motor, Valve, and Digital Modules and every fourth scan for Analog Modules.

The table below shows the processing times in [µs] measured for MMCL Version 2007-07-30 with an 1756-L63 Processor running RSLogix 5000 Version 16.00.

Module Type / Processing	CtrlGrp	MaGrp	MotorN	MotorR	MotorD	E3p	SubSys	Valve1	Valve2	DigInp	AnaInp	AnaInpC	PidMod	ActMod	ActPos	Sum
No of Modules	20	40	150	10	20	0	0	50	20	790	264	66	30	20	20	1500
Never, AOI [µs]	n/a	n/a	21	24	28	n/a	22	21	23	13	14	14	15	15	n/a	
Every scan [µs]	63	38	90	124	154	25	101	100	117	42	95	148	167	192	30	
Every 2^{nd} scan [µs]	n/a	n/a	58	77	95	n/a	64	63	74	29	56	83	92	106	n/a	
Every 4^{th} scan [µs]	n/a	n/a	40	51	63	n/a	44	43	50	22	36	49	54	61	n/a	
Total Time per AOI Module Type [ms]	1.2	1.5	8.7	0.8	1.9	0	0.0	3.2	1.5	22.9	9.5	3.2	1.6	2.1	0.6	59

The Total Scan Time is calculated for a benchmark specified for a representative Number of Modules of each Module Type and a total of 1500

Modules. The Cycle Time includes additional time for application programs and system overhead that is estimated at 10% and 20%, respectively. Based on these assumptions, the Cycle Time for the 1500 Modules would be approximately 75 [ms][1] on a single controller and the worst case would be approximately 133 [ms][2].

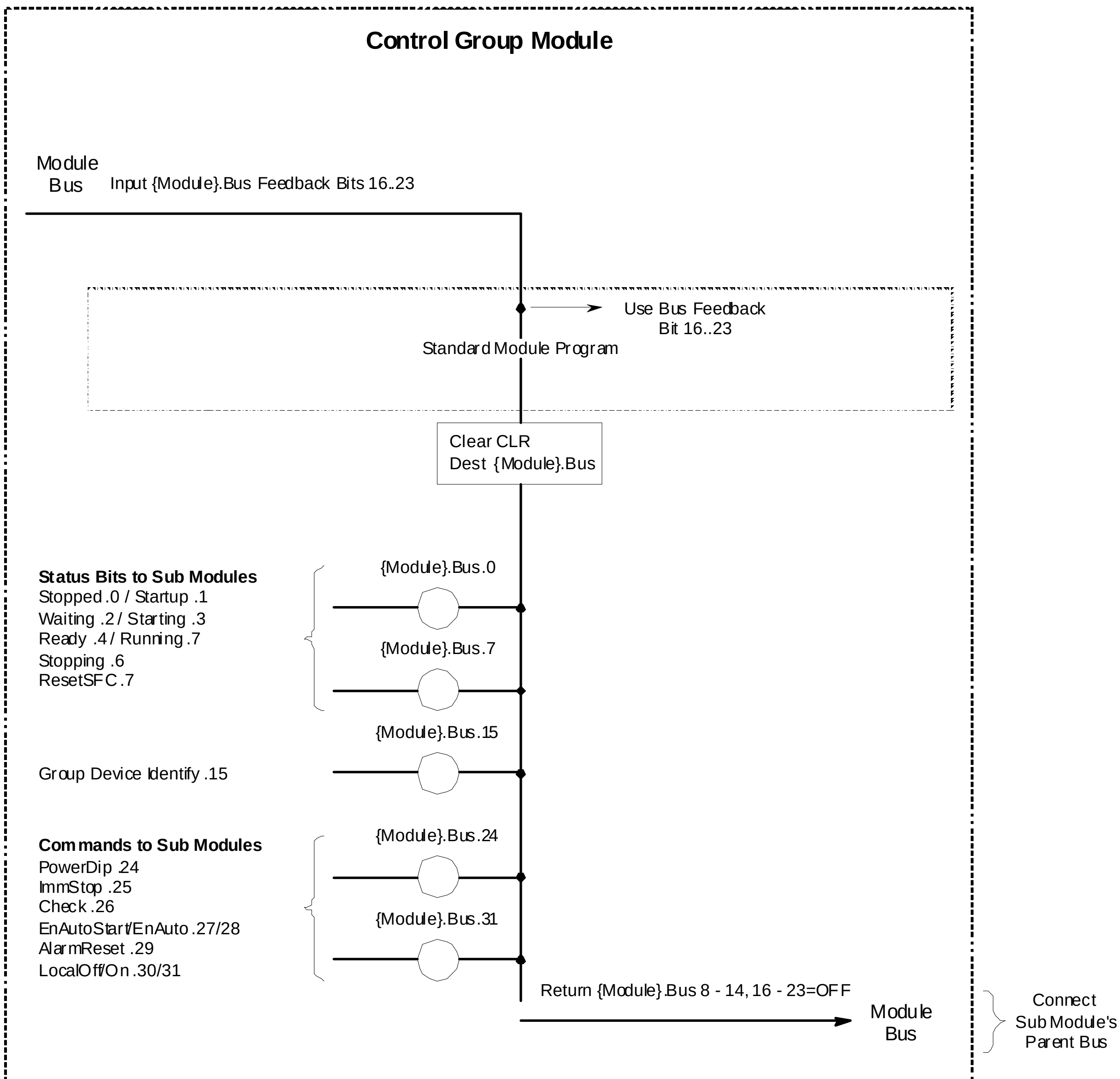
TIP For calculation of DigInp2 and DigPulse use the same numbers (time) as for DigInp.

(1) Modules are normally processed every *X*th cycle and also on every input change (for example, Scan Time also depends on actual input changes).

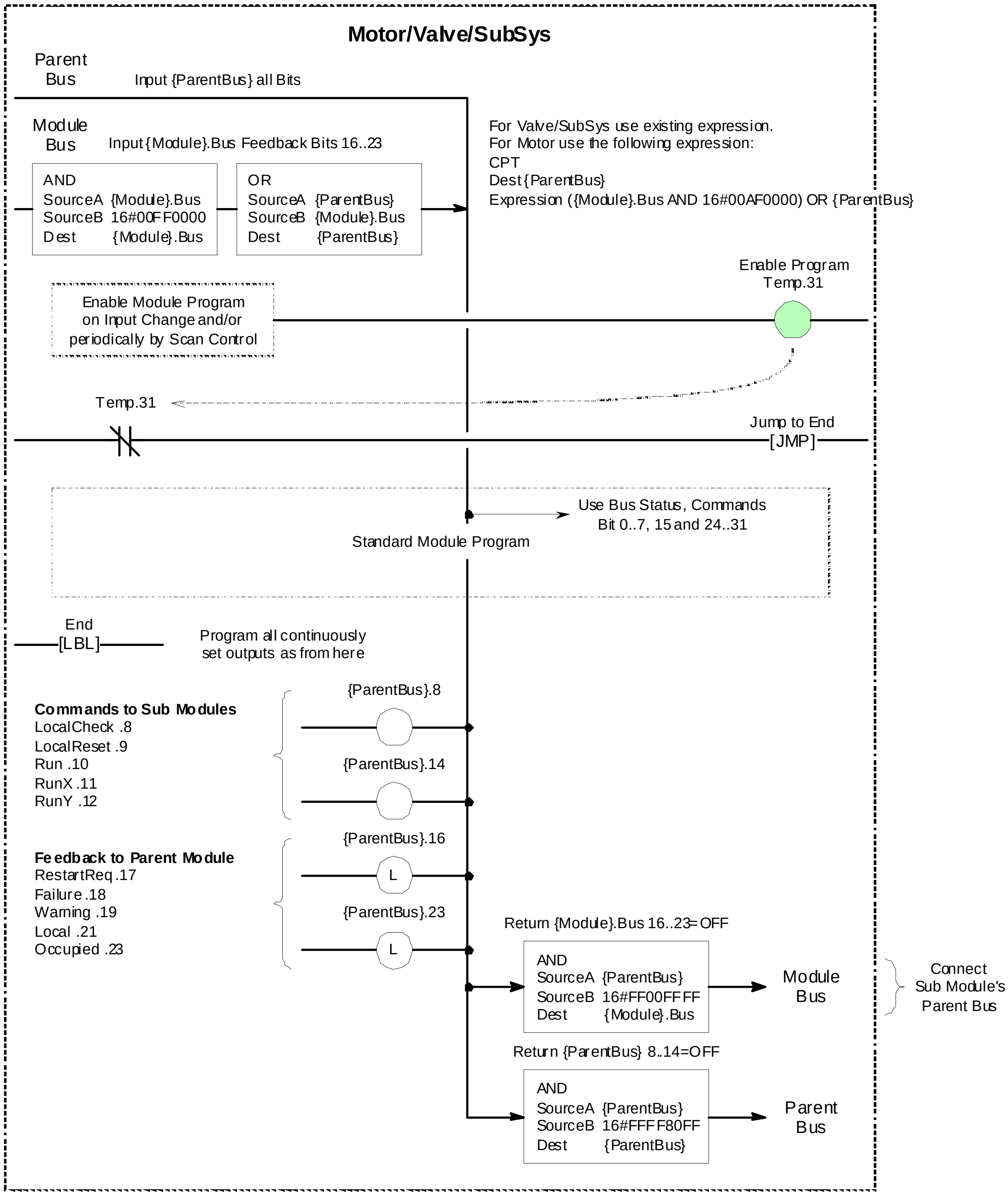
(2) Worst Case is calculated by choosing every scan for the scan rate. This is only valid at initialization or at immediate shutdown of all modules.

Generic Module Bus Signal Handling

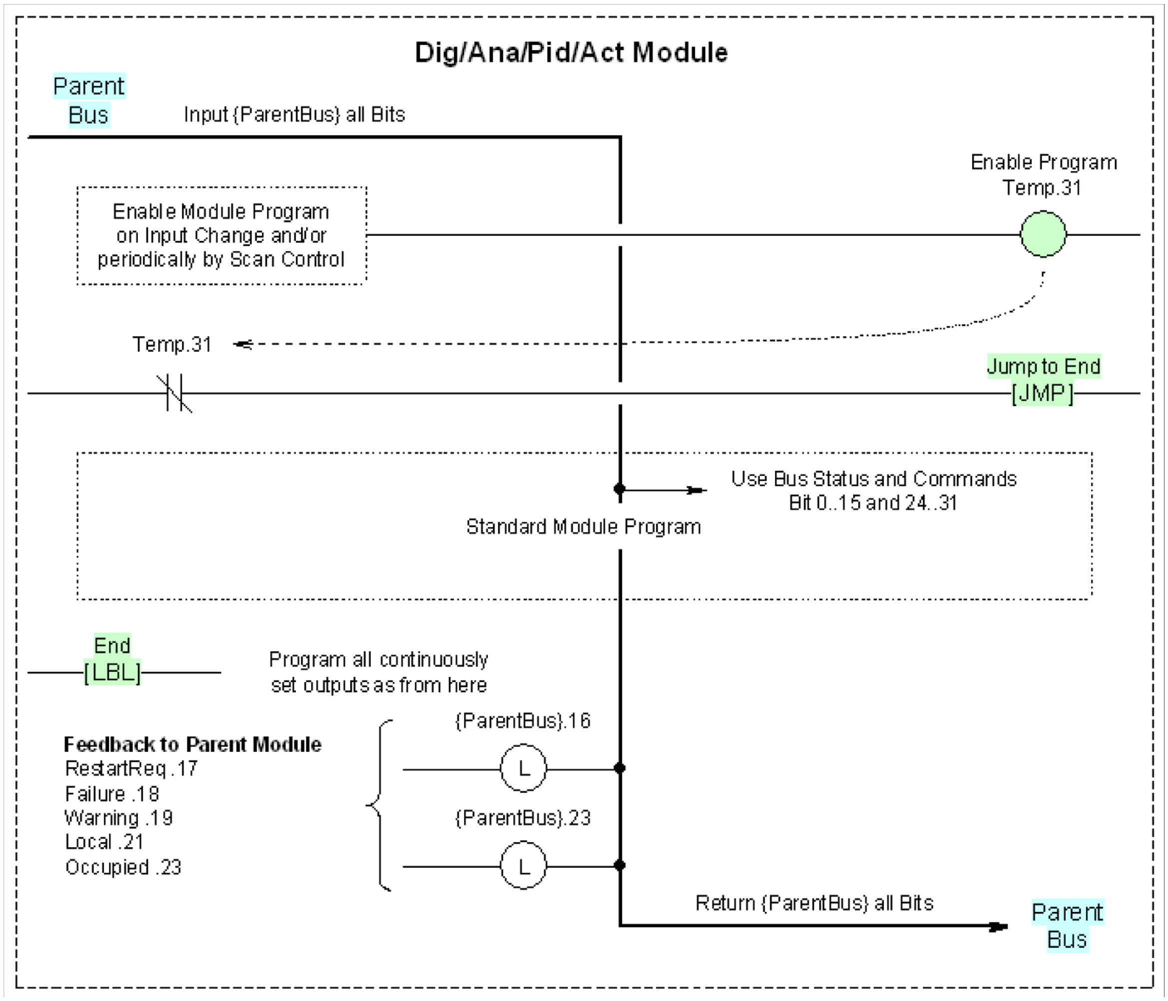
The Control Group Module (CtrlGrp) is parent to all other modules; it is not connected to a parent.



The Motor/Valve/SubSys Modules are connected to a parent module (CtrlGrp or MaGrp) and may be parent to their own sub-modules or devices



The DigInp/AnaInp/PidMod/ActMod Modules are connected to a parent (CtrlGrp, MaGrp or Motor/Valve/SubSys) and cannot be parent to other modules.



Notes:

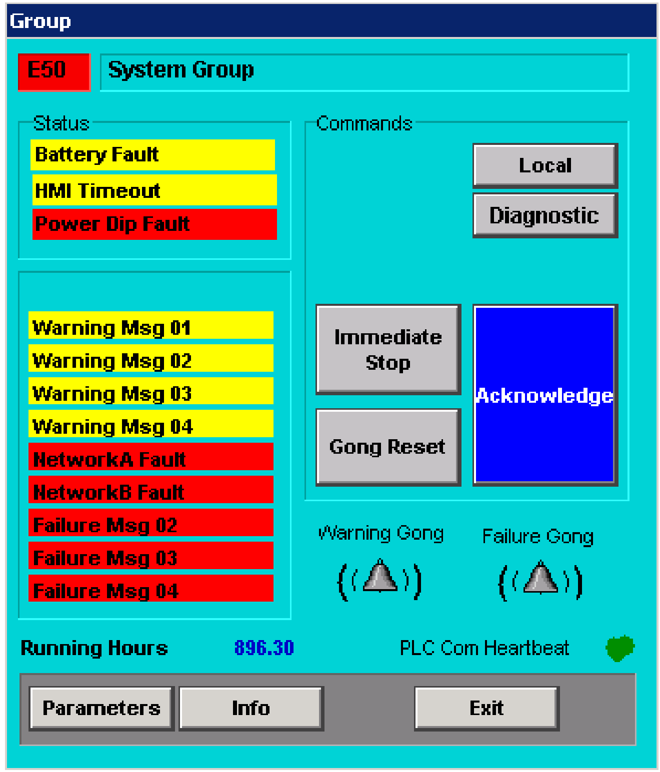
System Control Module

SysGrp - System Group Module

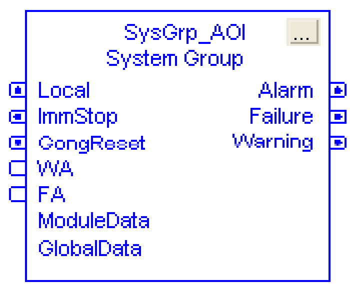
The System Group Module (SysGrp) is the master control module for an entire ControlLogix system. Typically, the module is programmed only once in the Main Routine. The module includes the following functions.

- Display template for FactoryTalk View SE
- System Group collective Failure and Warning indication
- ControlLogix controller supervision
- Communication check to HMI and bus interface for I/O link supervision by the user program
- Power dip detection, indication and time preset
- Emergency power priority selection
- User specific warnings and/or alarm (Failure or Warning) inputs
- Failure and Warning alarm gong control and time presets
- Start warning horn and flash outputs and time presets
- Running hour counter
- Controlling Apply Parameter global signal

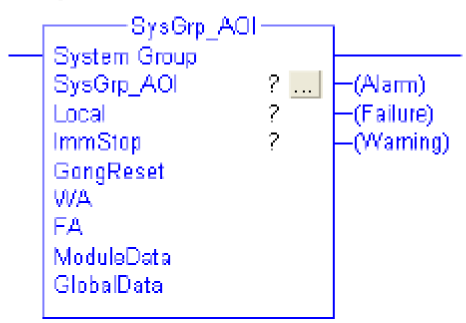
SysGrp Operator Interface



SysGrp Function Block

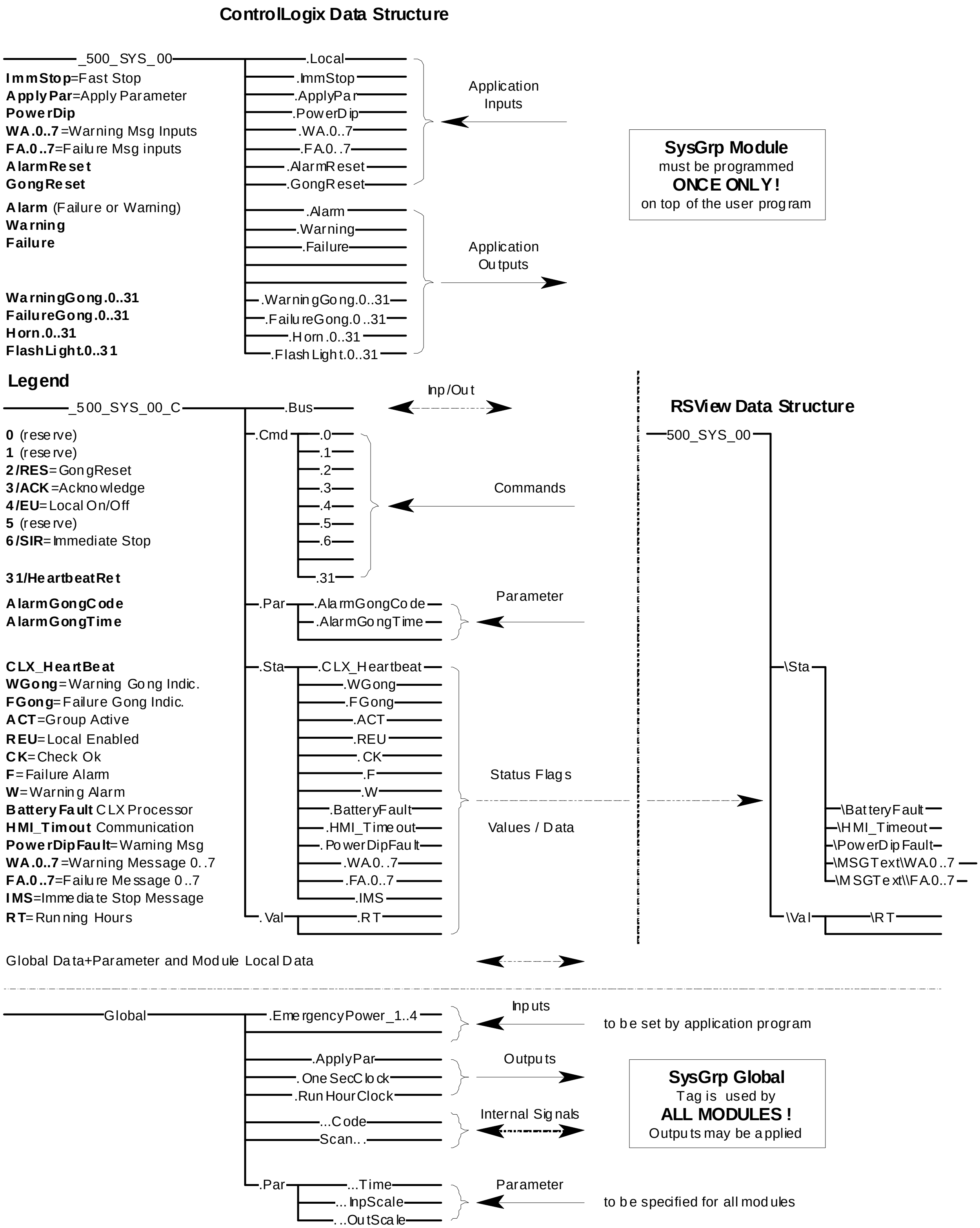


SysGrp Ladder



For module definitions and signal descriptions, see Appendix B page 143.

SysGroup Structure - Interface to FactoryTalk View



Notes:

Group Control Modules

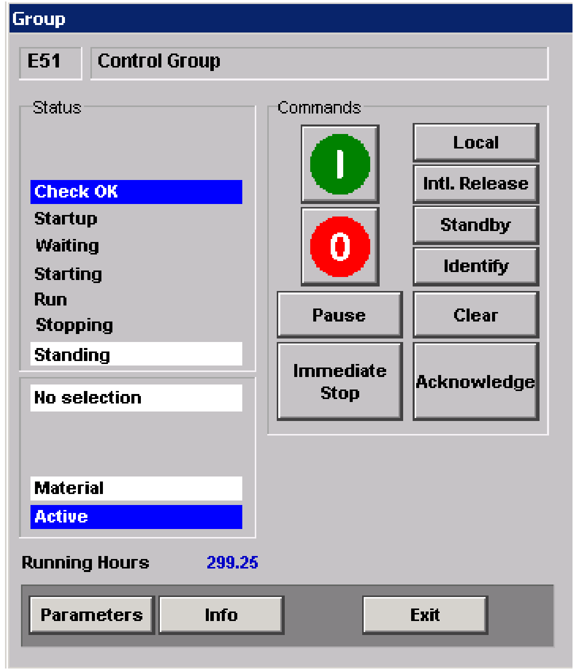
Group control is supported by the following modules that include an operator interface which is supplied as a ready-to-use HMI display.

CtrlGrp - Control Group Module

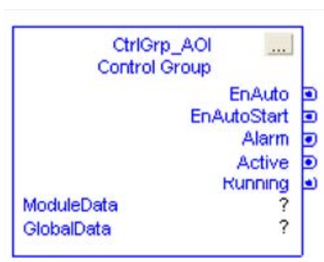
The Control Group Module (CtrlGrp) provides control functions for the operator interface and control for groups of machines and devices. It accepts HMI commands, start-up and master interlocks, and it supports alarm control and acknowledgement, local and remote selection, etc. The following are features of the CtrlGrp.

- Display template for FactoryTalk View SE
- Control Group collective Failure, Warning and sequence status indication
- Collective start/stop sequence control, immediate stop, local start enable, acknowledge
- Collective start-, stop- and immediate stop interlocks
- Release/disable interlock function (password protected)
- Power dip control and shutdown indication
- Enable automatic and enable auto start/restart outputs for module interlocking
- Running hour counter

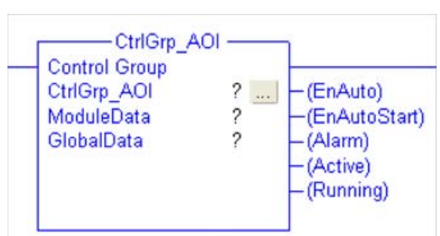
CtrlGrp Operator Interface



CtrlGrp Function Block

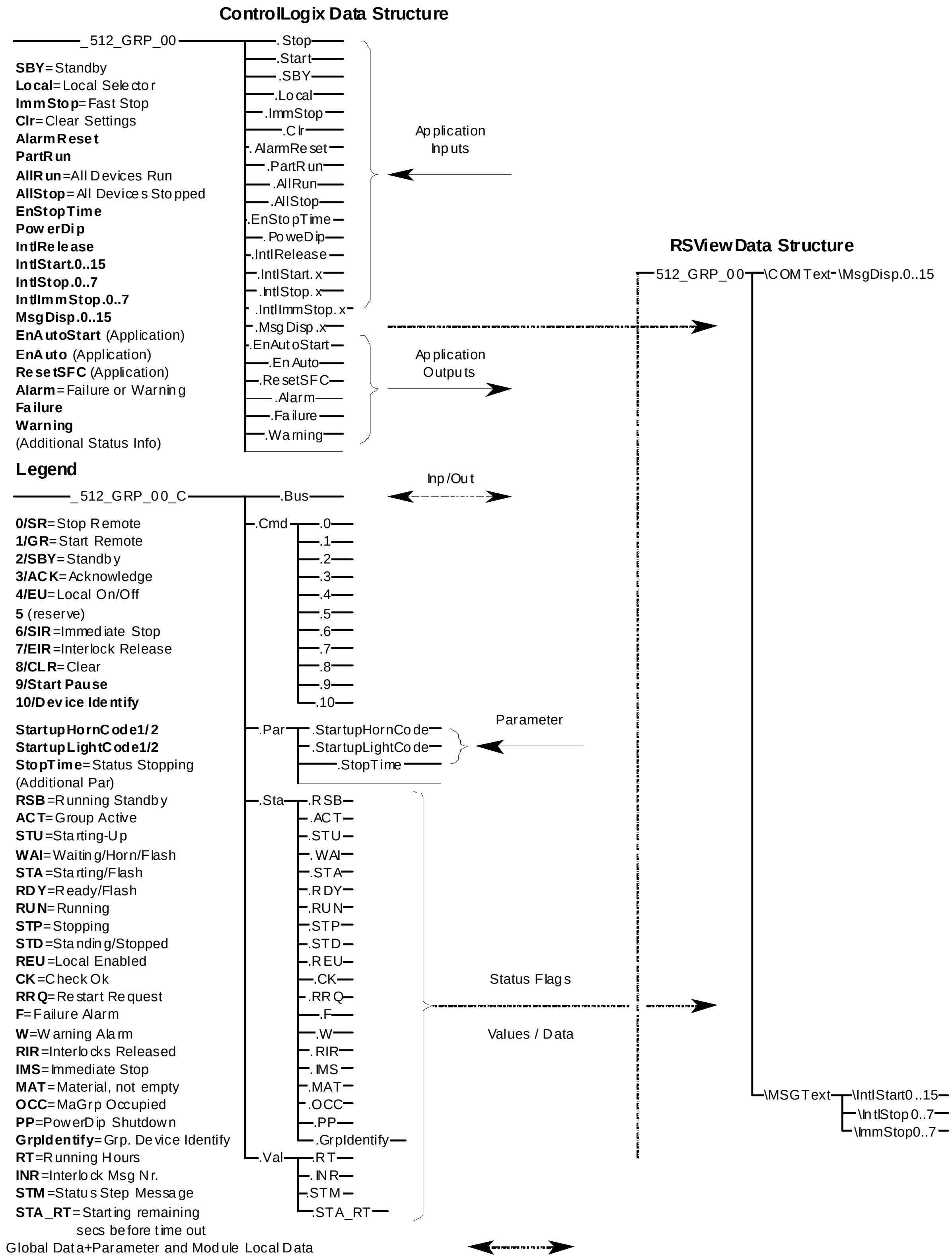


CtrlGrp Ladder



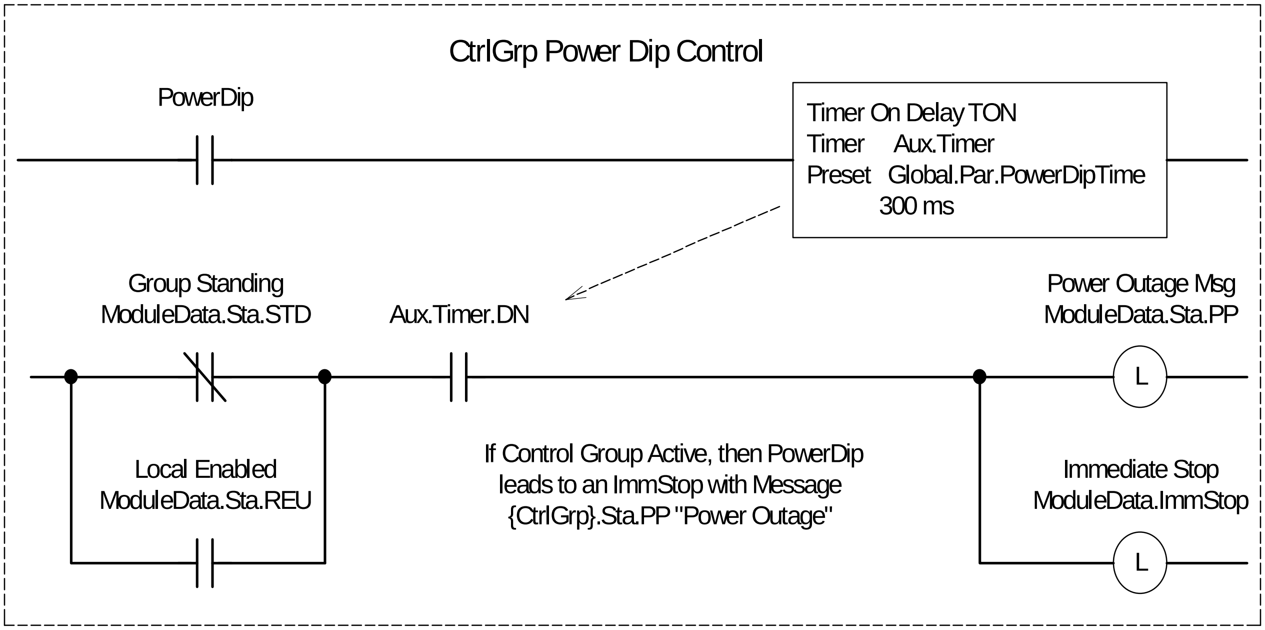
For module definitions and signal descriptions, see Appendix B, page 143.

CtrlGrp Structure - Interface to FactoryTalk View



CtrlGrp Power Dip Control

A power dip or power outage leads to a delayed of all connected modules by sending an Immediate Stop command through the Bus.



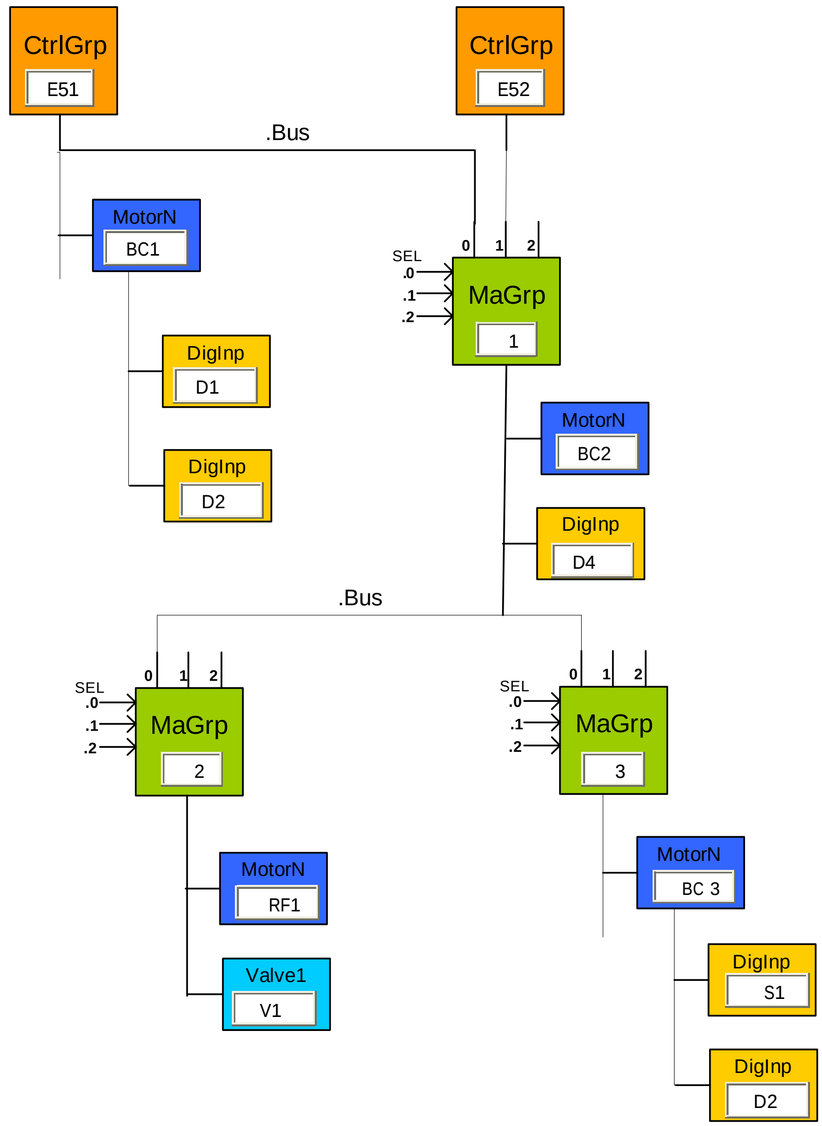
MaGrp - Machine Group Module

The Machine Group Module (MaGrp) allows for splitting an entire Control Group into sub-groups that can be selected, deselected or combined with other Control Group Modules. It can be used to preselect and occupy machines or share and switch machines on the fly. The following are features of MaGrp.

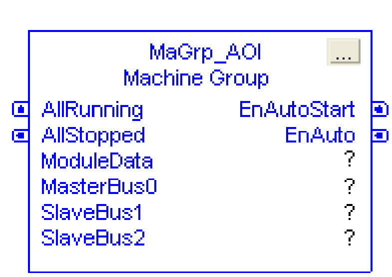
- Path preselection
- Selection indication
- Enable restart at new selection
- Enable switching without restart
- Share Machine Group
- Transmit selected Control Group signals from and to Machine Modules
- Enable automatic and enable auto start/restart outputs of selected control group(s) for module interlocking
- Parent bus connectors for one Master Control Group and X Slave Control Groups

Note, the Master CtrlGrp can access the modules for Local Control, Alarm Reset, Immediate Stop and Power Dip without selection. The Slave CtrlGrp can access the Modules only if selected.

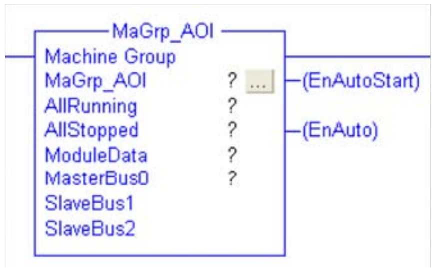
Grouping of Machines by MaGrp Bus Links



MaGrp Function Block

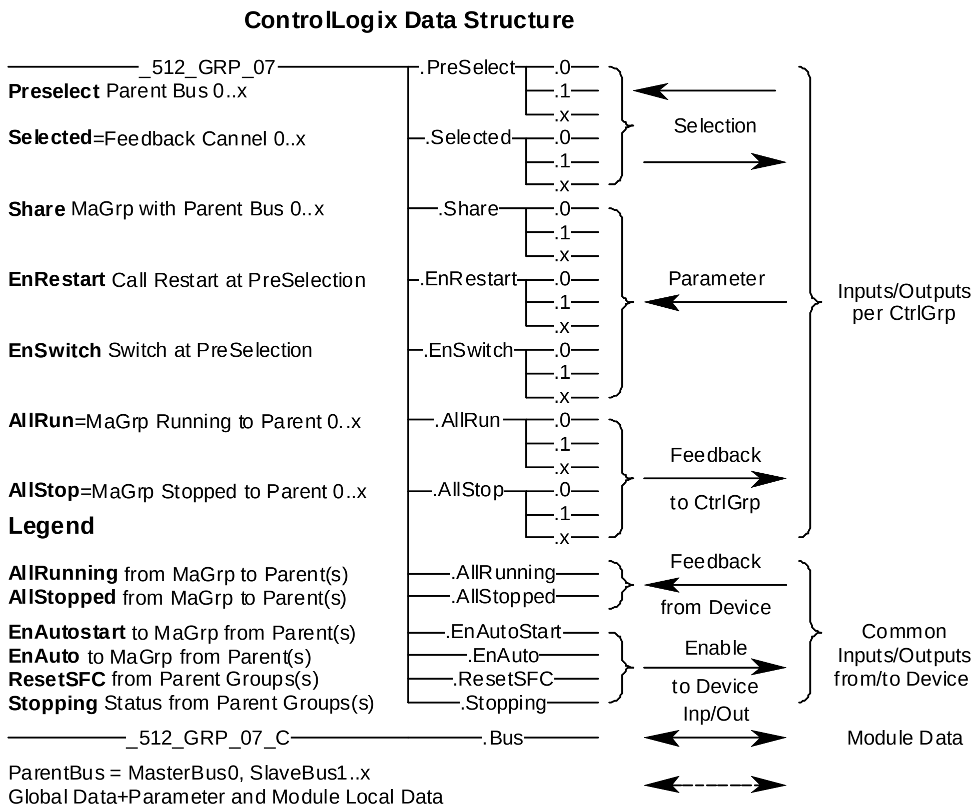


MaGrp Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

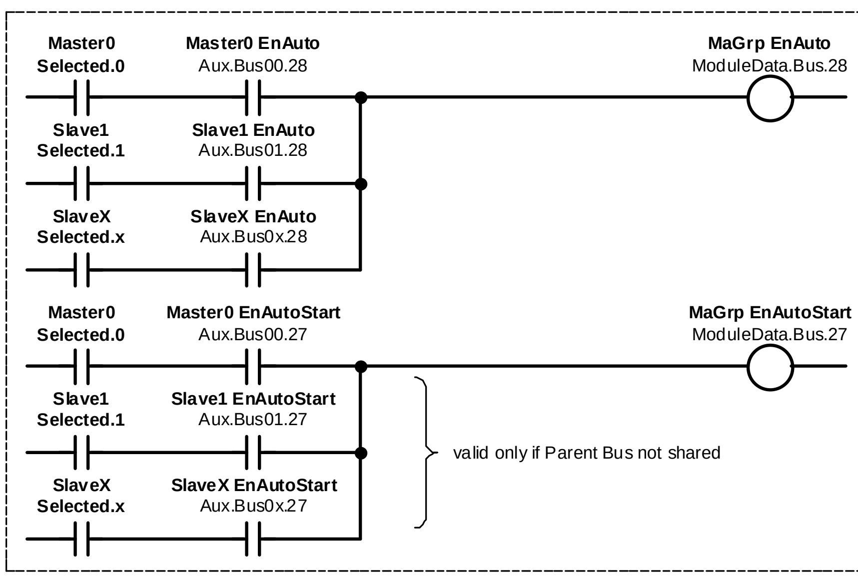
MaGrp Structure



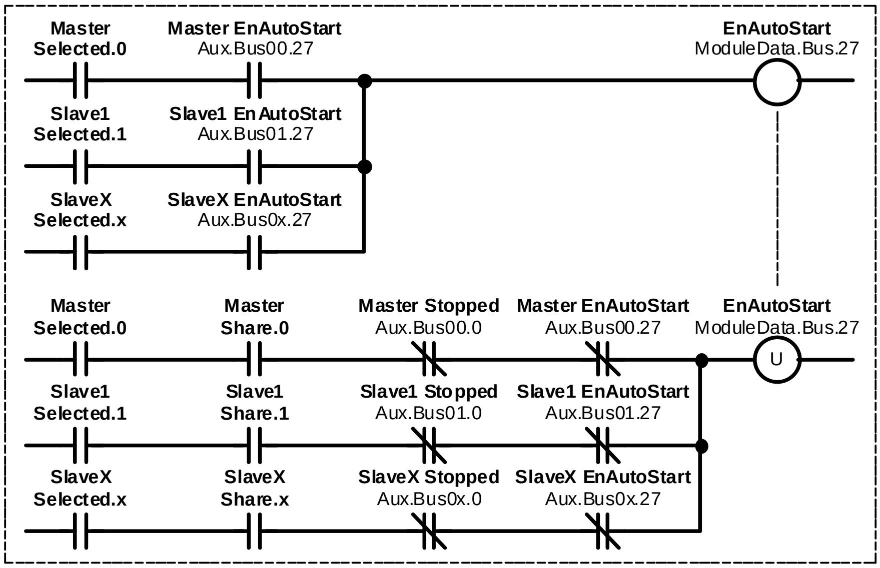
MaGrp Functions

The Machine Group Module performs the following main functions.

- Connects/disconnects a group of devices (Machine Group) to/from an active Control Group.
- Prevents new selection as long as a Control Group is active or not stopped normally (default)
- Automatically calls for restarting a Control Group at new selection if EnRestart.0..x is set, or enable the Machine Group without a restart request if EnSwitch.0..x is set
- Cares for exclusive use of a Machine Group by one Control Group (default), or allows for sharing a Machine Group by multiple Control Groups if Share.0..x is set
- Output Selected.0..x is set ON at startup of the Parent Group or if EnSwitch.0..x is set
- Signals all devices of the Machine Group for Immediate Stop and Local Control dependent on the current selection, the Master Group Local and Stop commands remain always active
- Checks devices of selected Machine Group on Failures/Warnings prior starting the Control Group
- Transmits common Failures/Warnings to the Master Group and to selected Slave Groups
- The MaGrp's EnAuto and EnAutoStart outputs are determined by the Selected.0..x input



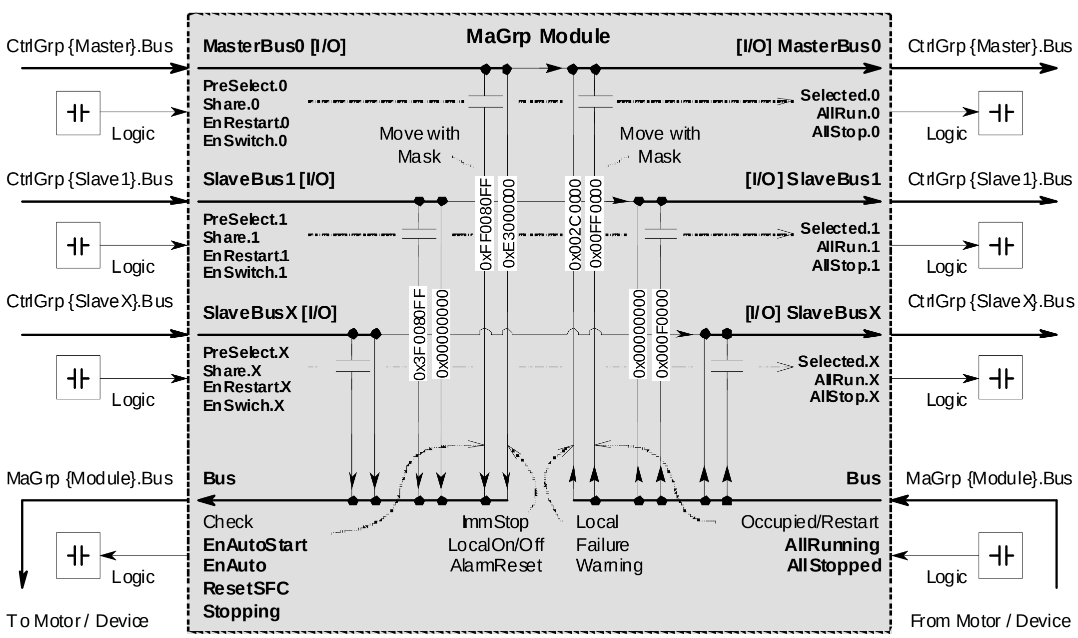
- If Share.0..x is enabled, the MaGrp's EnAutoStart output is determined by the shared Parent module; in other words, the output is only released if the EnAutoStart of ALL active Parent modules is ON



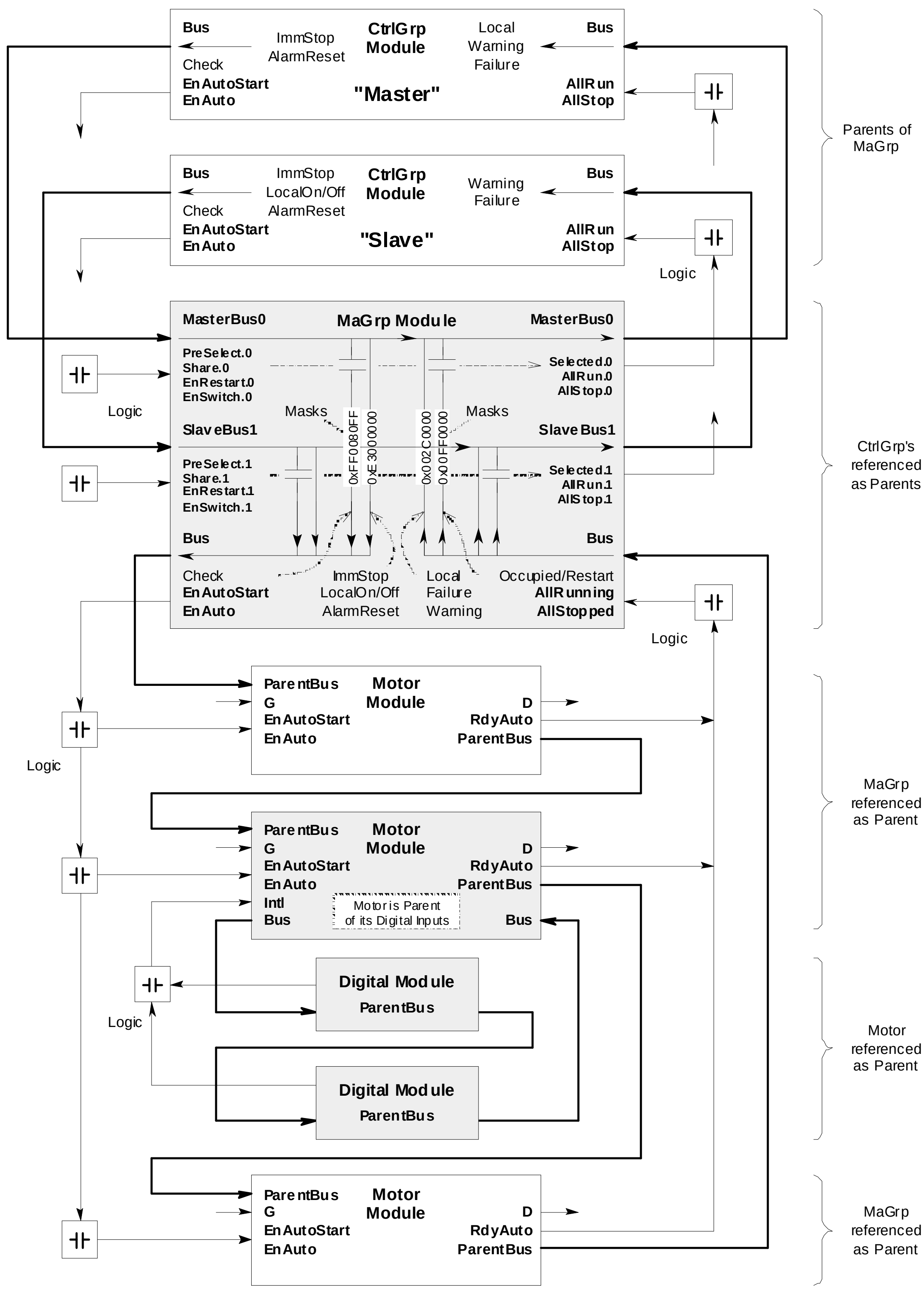
Used to do a restart if a Group is "shared", otherwise the 0->1 transition at EnAutoStart does

MaGrp Bus Signal Marshalling Functional Diagram

The diagram below shows how Bus information is transmitted through the MaGrp module depending on the Parent Selected.0..x. The masks specify which of the signals are transmitted.



Sample Wiring Diagram Bus Interface CtrlGrp, MaGrp, Motor, Digital



IPCom - Inter Process Communication Module

The Inter Process Communication Module (IPCom) is used for Inter Process Communication between two controllers.

With this module the communication is established and supervised to a remote controller.

The module's main function is BUS data distribution. For example, a CtrlGrp will share its Bus with another area (controller) to control many devices; the complete connection of this remote link can be realized with this IPCom module. At the same time we also transfer various user data, which can be allocated optionally and, for example, used for interlocks and signals, and transferred to other controllers.

The IPCom module uses Producer/Consumer networking protocol Common Industrial Protocol (CIP). After the programmer has created and configured a Produced/Consumed tag-structure, the IPCom module plugs in to this tag as the communication channel.

Communication monitoring is based on the exchange of a watchdog counter. If the received counter remains unchanged longer than the parameter Timeout time, then an error is indicated and user data is held at 0. If the parameter Par.HoldOutput = 1, then the user data remains on the last received status. The IPCom module is working only in connection with a second IPCom module. This pair must be configured as master and slave. The master module usually has to appear in the program which holds the CtrlGrp. The remote IPCom module is configured as slave and is normally called in another controller.

IMPORTANT

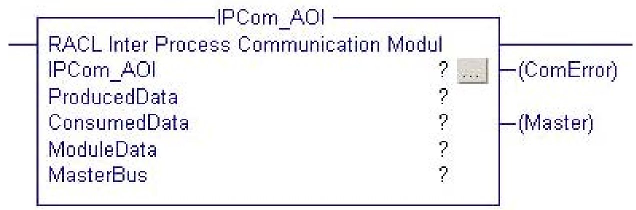
For each Bus communication, only one IPCom module can be configured as Master. However it is possible to define several IPCom modules as Slaves. That means that the CtrlGrp (master) will transfer its bus to several remote MaGrps or Devices.

In case of several IPCom-Slaves, the watchdog communication supervision on the IPCom-Master side, does not work because of multiple watchdog feedbacks.

The modules' BUS data can also be linked with the same functions as CtrlGrp or MaGrps.

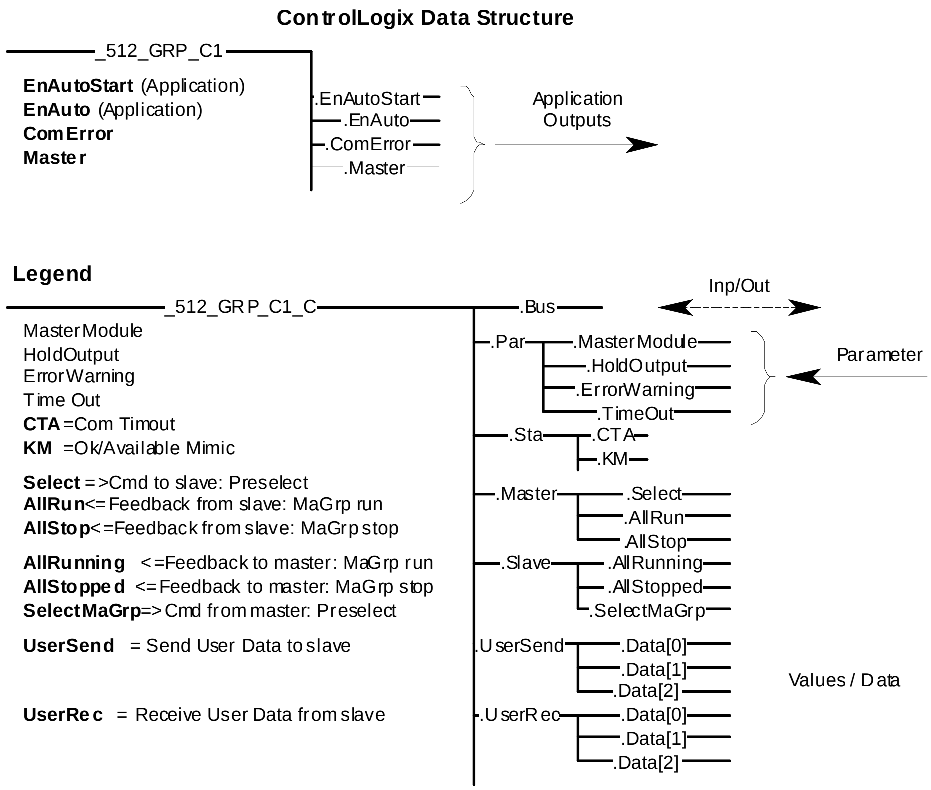
Note: The IPCom Module does not have an HMI Template.

IPCom Module Ladder



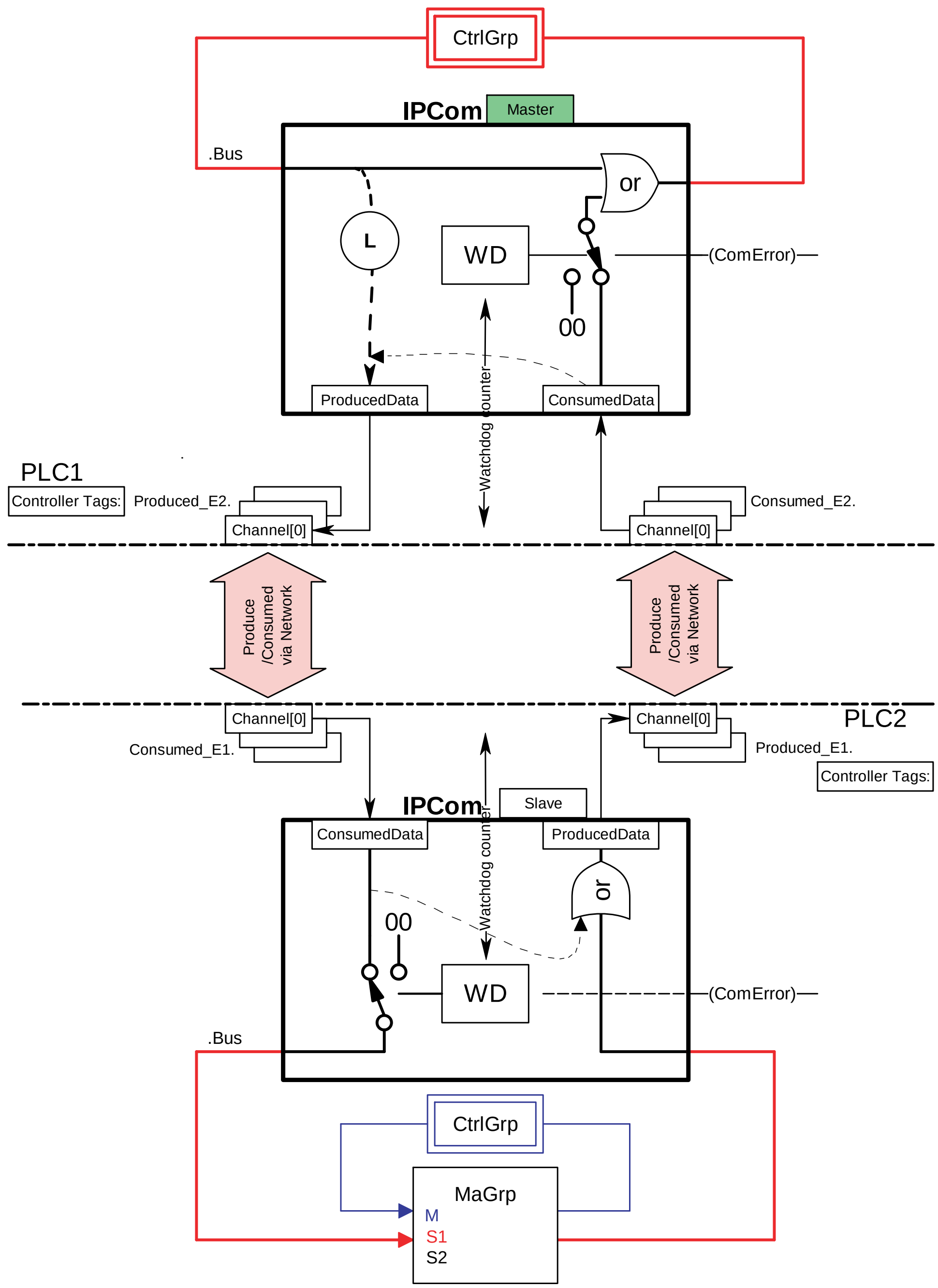
For module definitions and signal descriptions, see Appendix B, page 143.

IPCom Module Structure

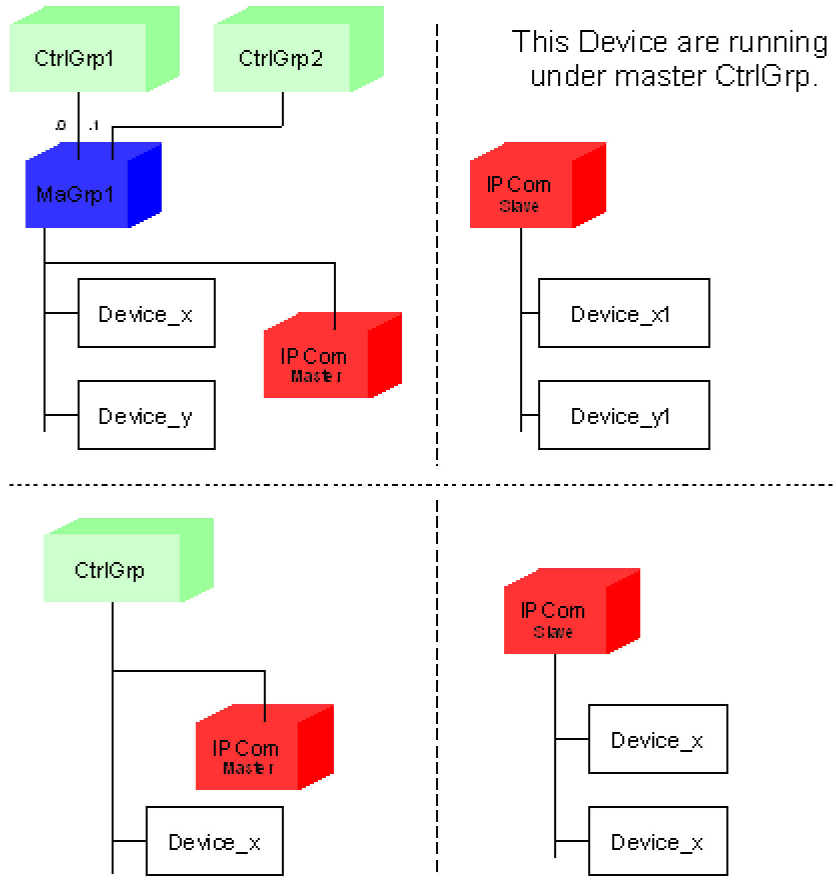


IPCom Bus Signal Marshalling Functions Diagram

The diagram below shows how the Bus is transmitted through the IPCom module and how the data is transmitted with Produced/Consumed functionality.



IPCom Module Design Example and Possible Connection

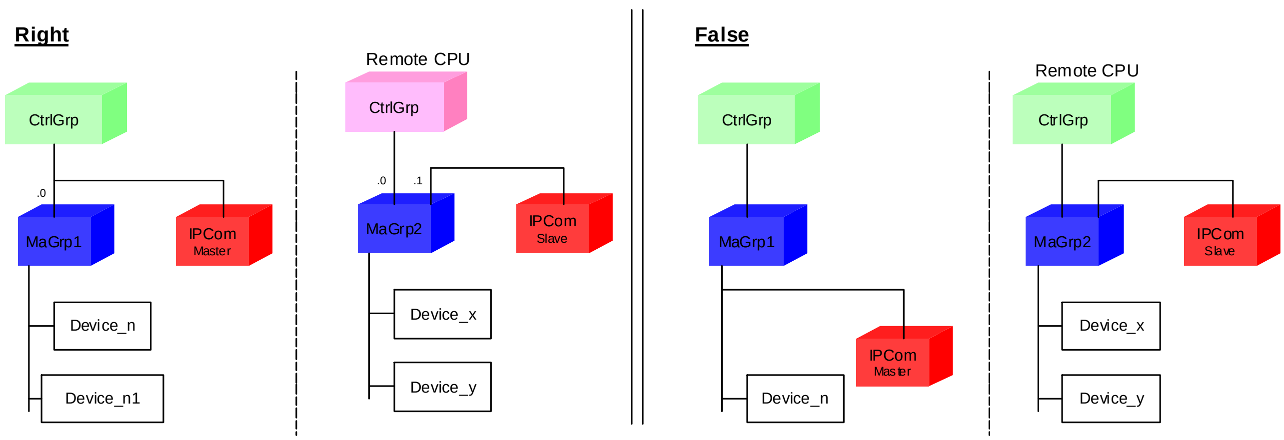


Exception of Bus Connection

In case of any occupied situation in the remote computer, you have to know following:

You can not connect an IPCom under a MaGrp. The IPCom has to be connected directly to the CtrlGrp. The reason for this exception is that a MaGrp blocks this occupied (not available) bus-bit.

If the remote MaGrp is already selected, the Module creates the occupied Signal when the MaGrp is selected a second time. The occupied signal is transmitted over the Bus to the corresponding CtrlGrp.



In this example, the occupied signal, which is generated by remote MaGrp2, does not function any longer.

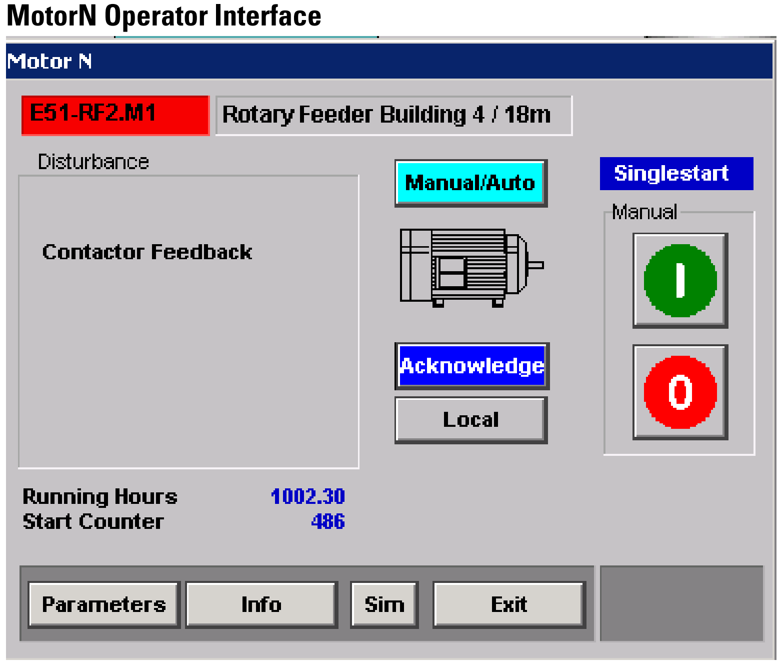
Motor Control Modules

MotorN - Motor Normal Starter Module

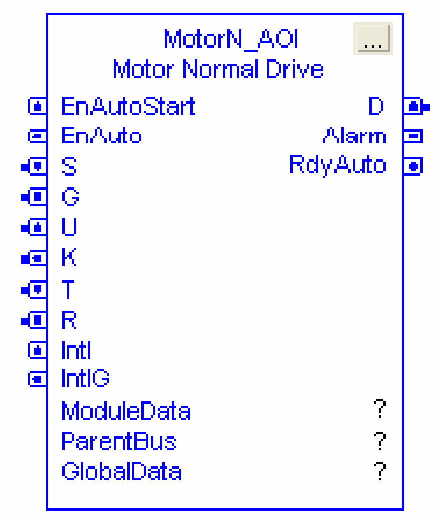
The Motor Normal Starter (MotorN), also known as Full Voltage Non-Reversing (FVNR), supports all standard interlocking and supervision for normal, across-the-line starters. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of MotorN.

- Display template for FactoryTalk View SE
- Failure or Warning starter
- Enable automatic operation and enable auto Start/restart
- Local or remote single start (password protected) including start-up Warning
- Detailed alarms for local isolator, MCC availability, thermal-OL, contactor feedback
- Safety alarm steady state interlock
- Machine protection interlock that may be overridden at local start
- Running hour counter and number of starts counter
- Parent bus link to Control Group or Machine Group module

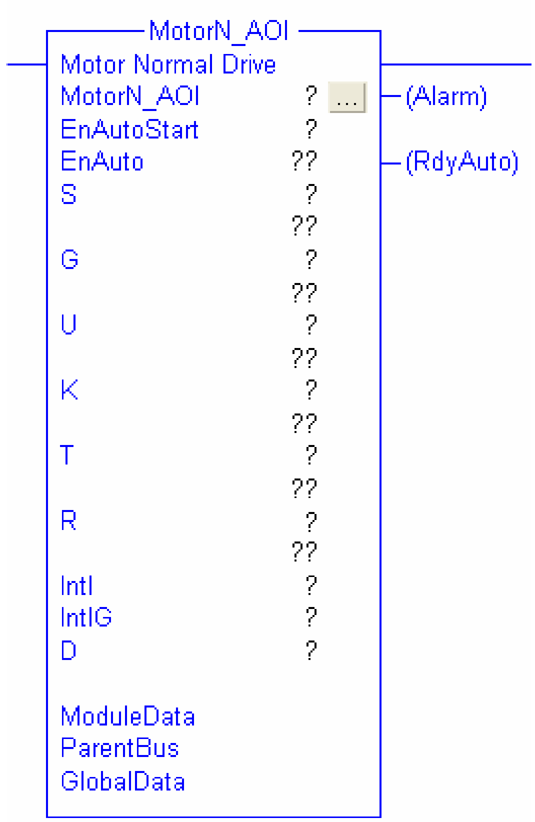
MotorN Operator Interface



MotorN Function Block

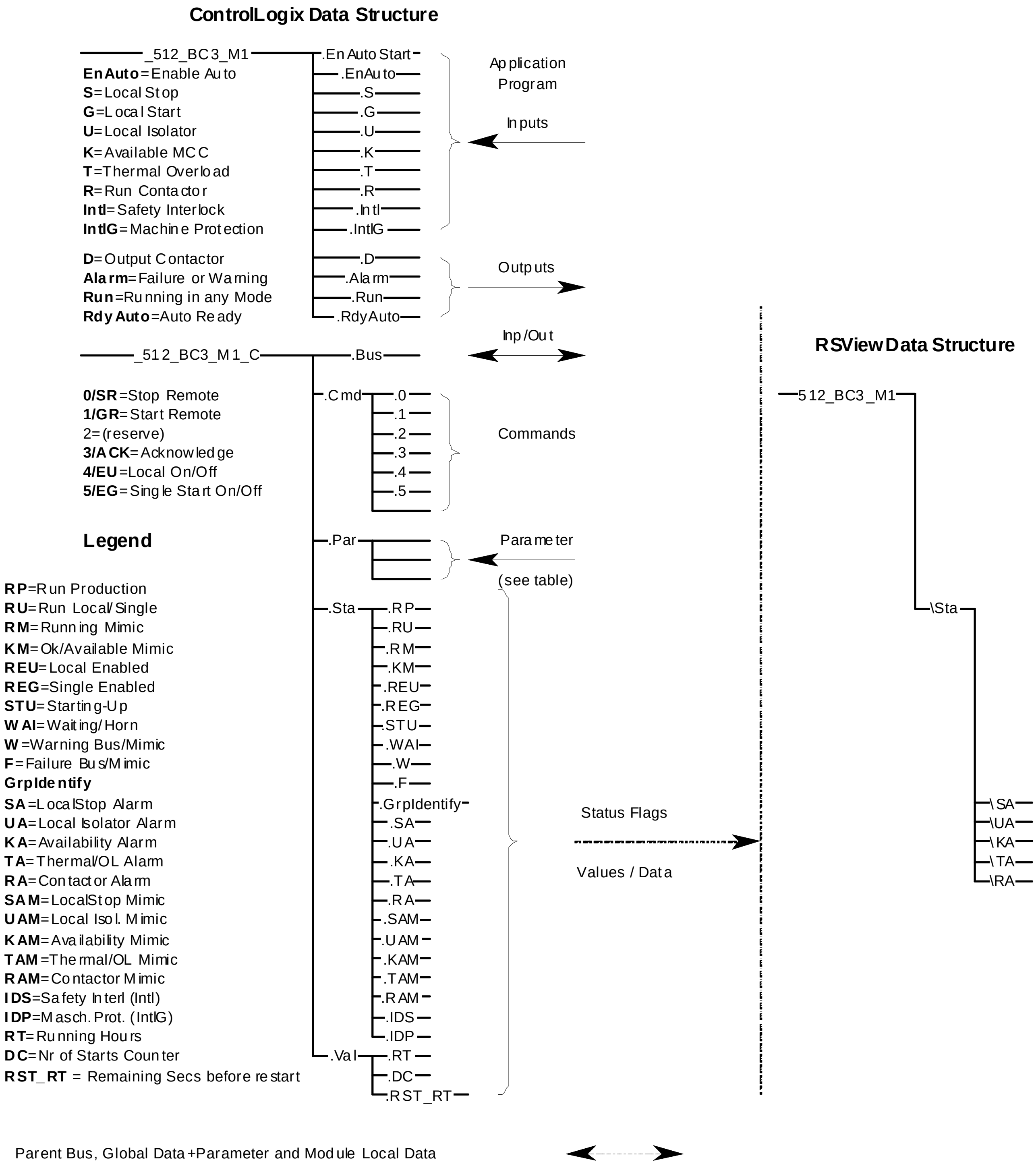


MotorN Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

MotorN Structure (Normal Starter) – Interface to FactoryTalk View

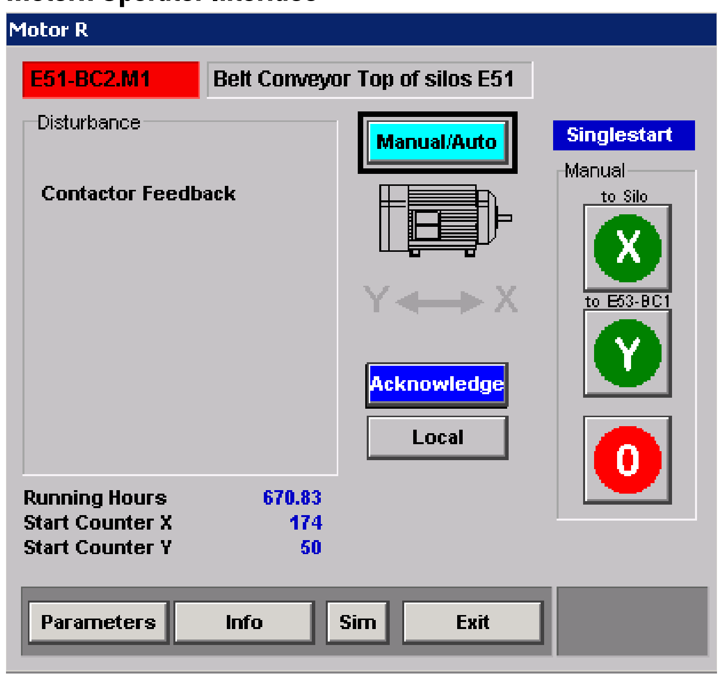


MotorR - Motor Forward/Reverse Starter Module

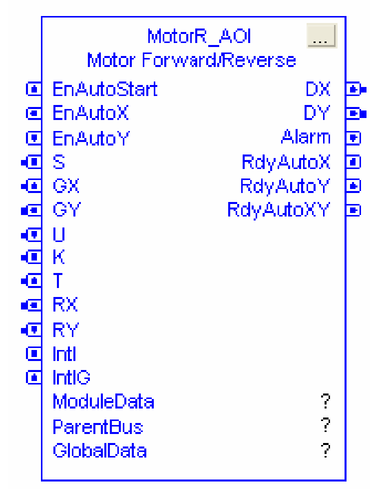
The Motor Forward/Reverse Starter (MotorR) supports all standard interlocking and supervision for normal reversing starters, also known as Full Voltage Reversing. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of MotorR.

- Display template for FactoryTalk View SE
- Failure or Warning starter
- Forward/reverse enable automatic operation and enable auto start/restart
- Forward/reverse local or remote single start (password protected) including start-up warning
- Detailed alarms for local isolator, MCC availability, thermal-OL, forward/reverse contactor feedback
- Safety alarm steady state interlock
- Machine protection interlock that may be overridden at local start
- Running hour counter and number of starts counter forward/reverse
- Parent bus link to Control Group or Machine Group Module

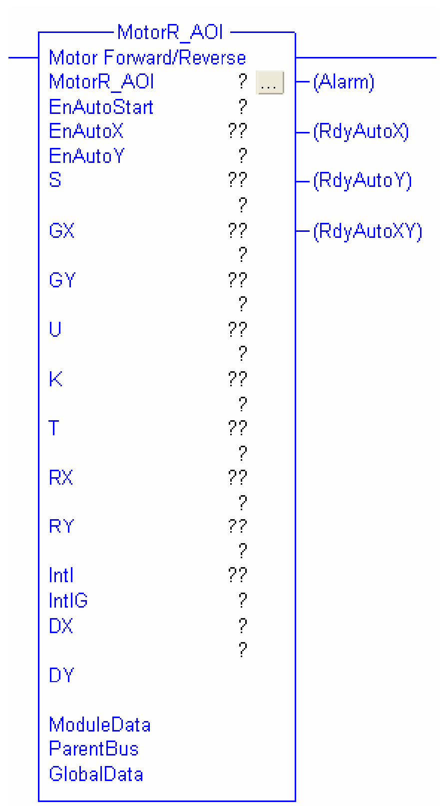
MotorR Operator Interface



MotorR Function Block

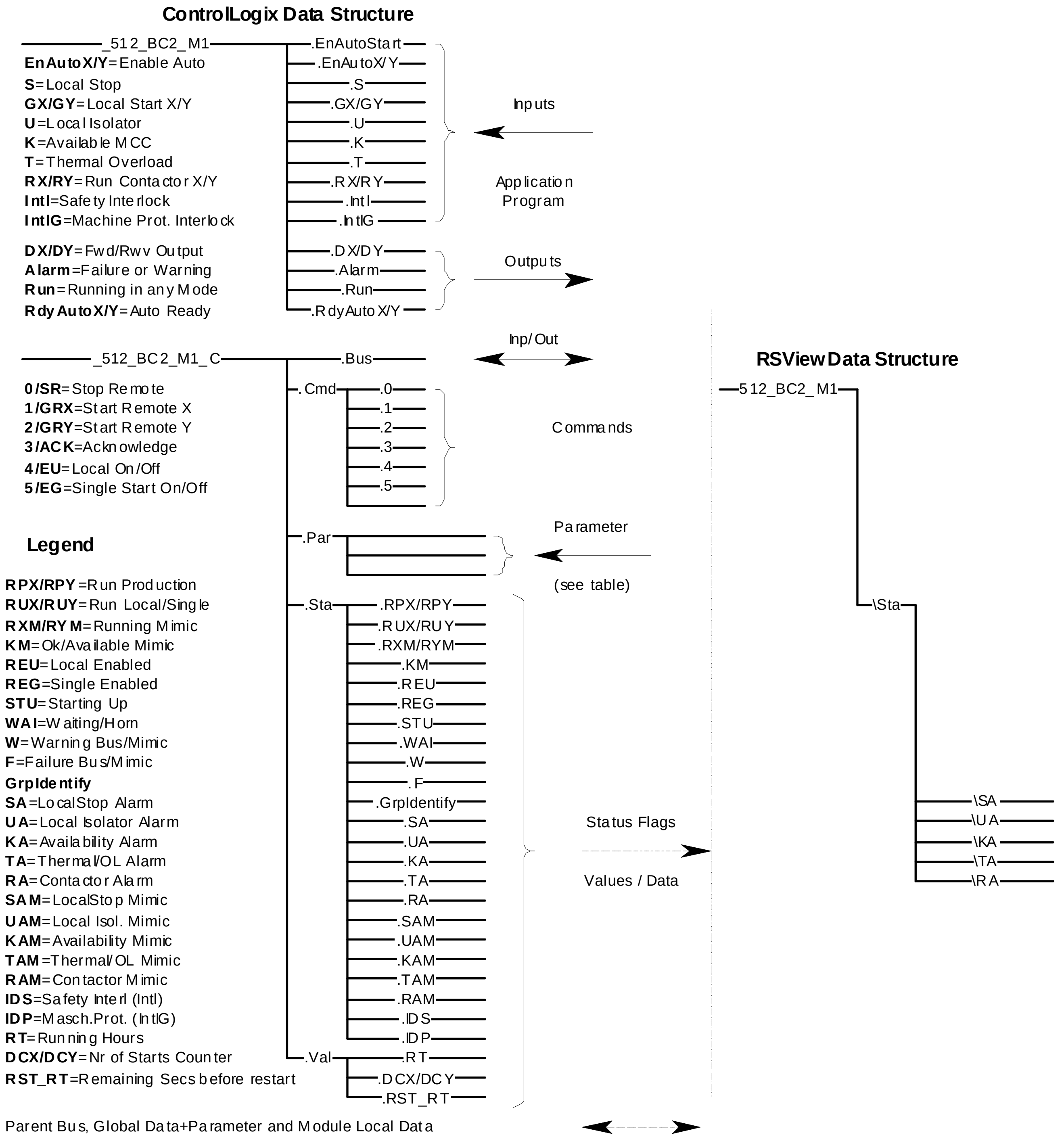


MotorR Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

MotorR Structure (Forward/Reverse Starter) – Interface to FactoryTalk View

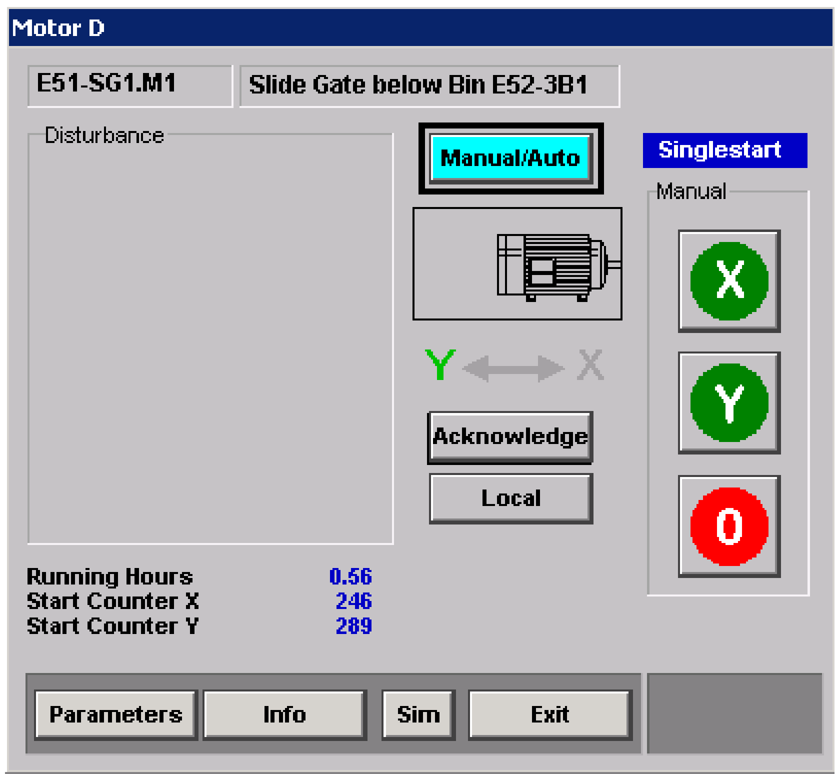


MotorD - Motor Damper/Flap Module

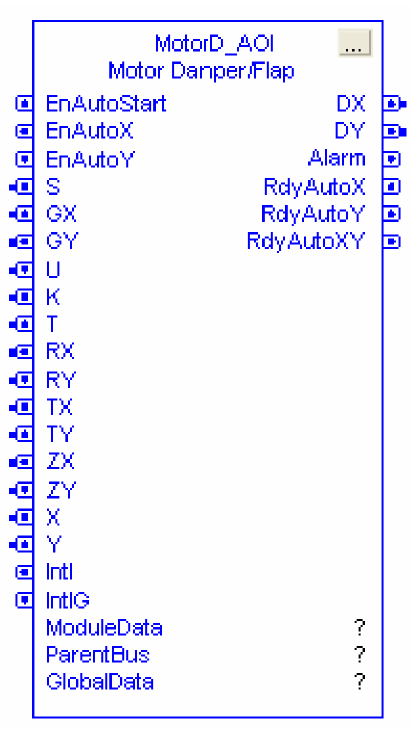
The Motor Damper/Flap (MotorD) supports all standard interlocking and supervision for motorized flaps, valves and dampers. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of MotorD.

- Display template for FactoryTalk View SE
- Failure or Warning starter
- Supervision forward/reverse for: Torque Switch, Control and Safety Limit Switch
- Parameter select safety position at shutdown
- All other features are the same as the MotorR module (see page 54)

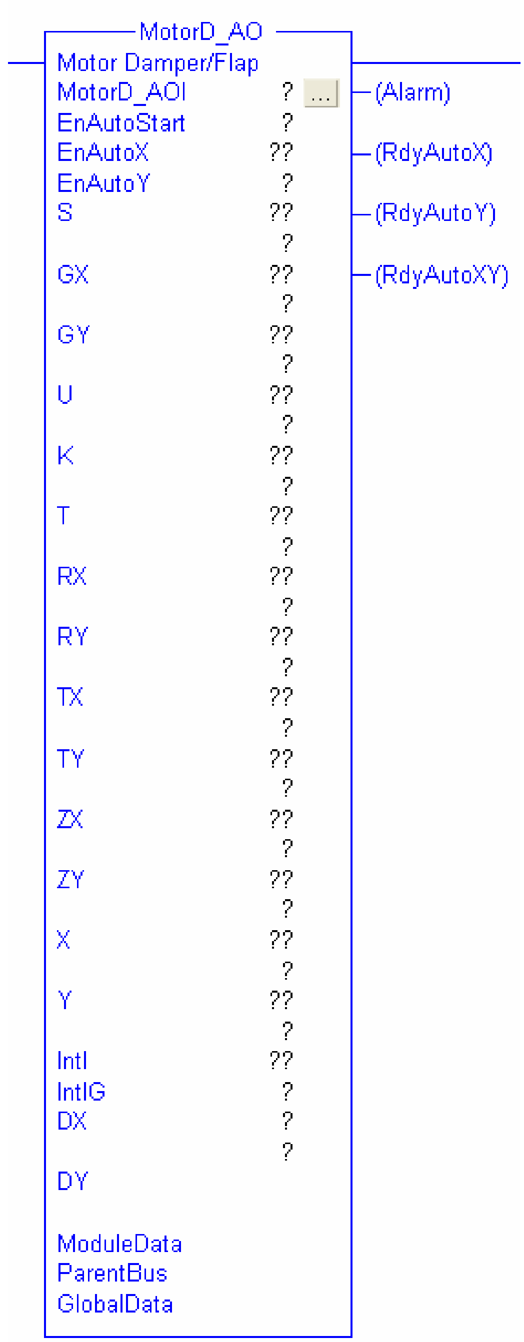
MotorD Operator Interface



MotorD Function Block

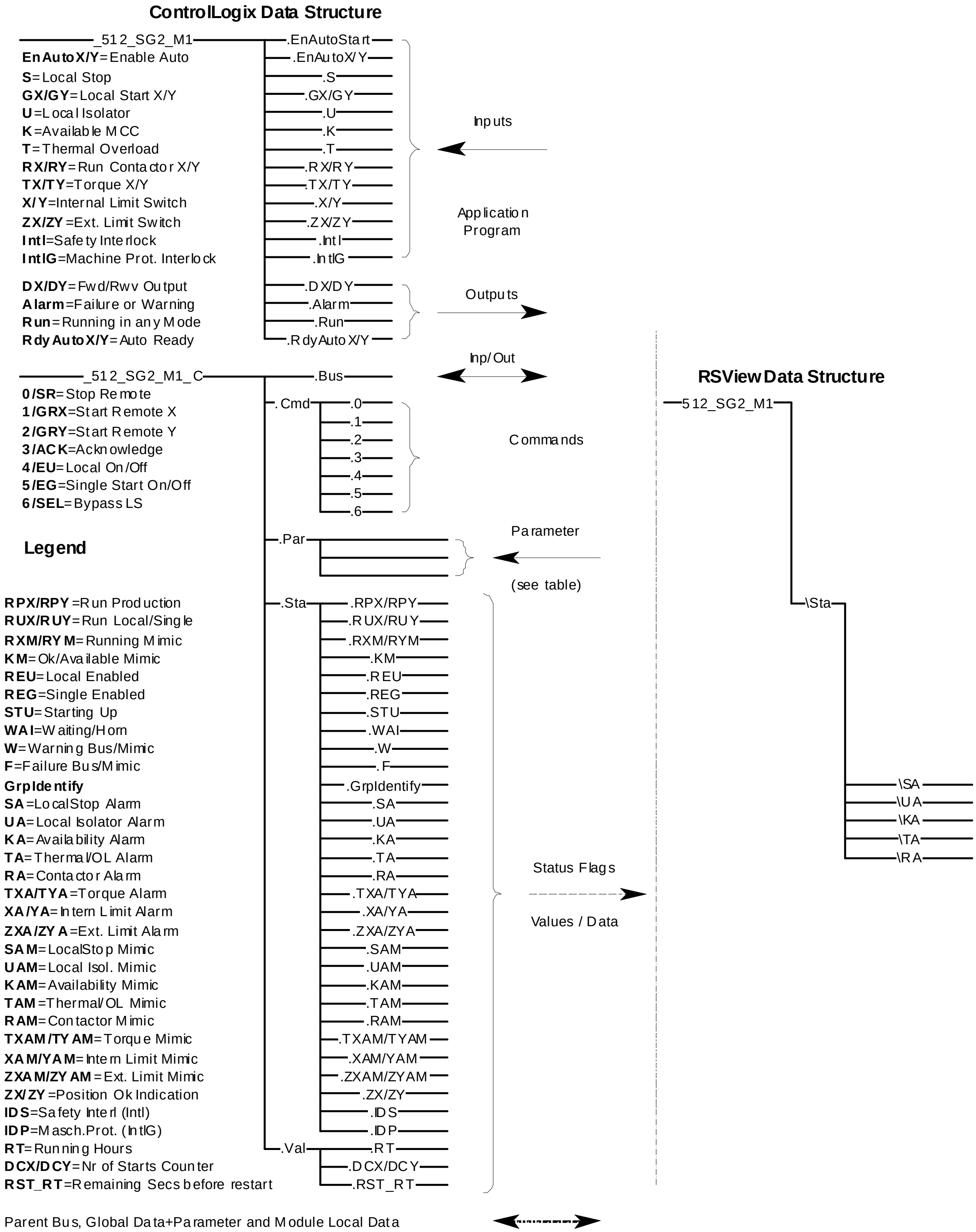


MotorD Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

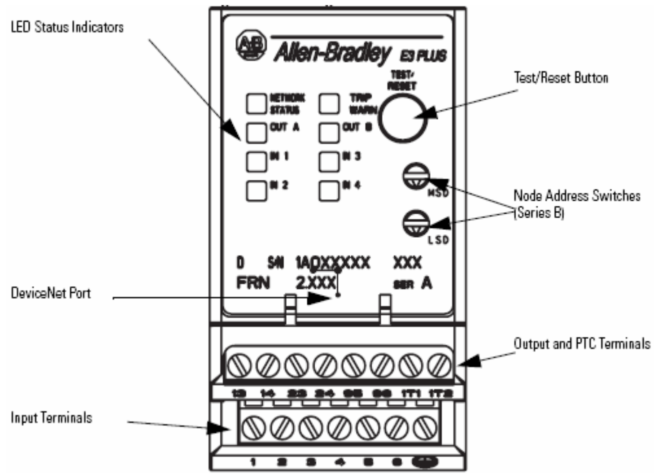
MotorD Structure (Damper/Flap) – Interface to FactoryTalk View



E3 - Motor Overload Relay Module

E3 Overload Relay

The E3 Overload Relay is a multi-function solid-state microprocessor-based electronic overload relay for the protection of squirrel-cage induction motors rated from 1 to 2,250 Amps. Two versions are available: the E3 and E3 Plus.



Protection and Warning Functions

The E3 Overload Relay provides the following protection and warning functions.

- Overload
- Phase loss (trip only)
- Stall (trip only)
- Jam
- Underload
- Current imbalance
- Ground fault (E3 Plus only)
- Thermistor (PTC) input (E3 Plus only)

E3p Module

The E3p module is an interface between E3Plus OverloadRelay and the MotorN, MotorD or MotorR modules. The E3p_AOI is always called directly after the Motor Module.

The Module input “ParentBus” connects to Motor_C.Bus.

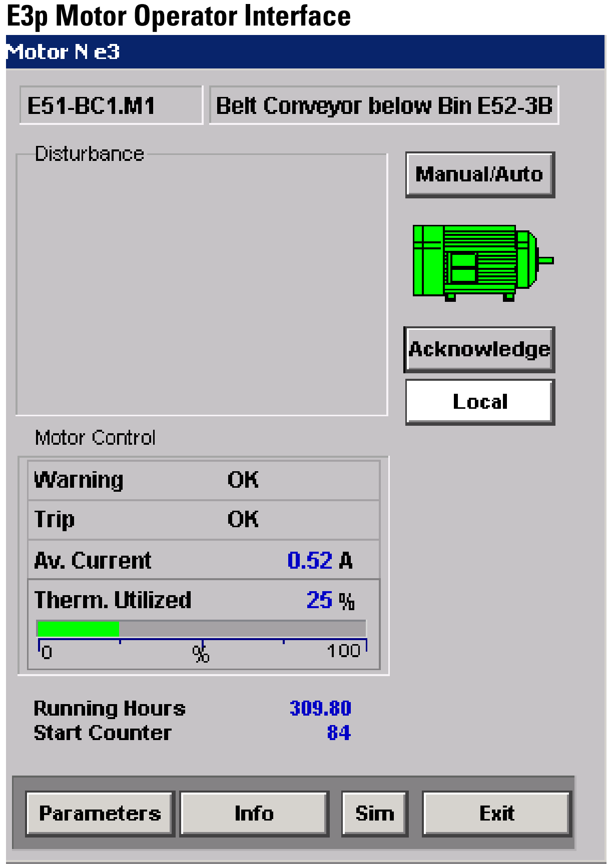
The following E3p module parameters are used in this module:

- P21= Device Status
- P14= Trip Status
- P4 = Average Current
- P9 = Therm Utilized

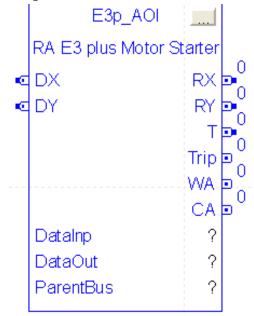
If an E3p Module is used for motor control then you have to call a specific HMI template. For example, the MotorN_E3_large template is called for a MotorN module. The following are features of E3p_AOI.

- No specific Operator Interface
- DeviceNet Interface Mapping
- Provide the Motor with MCC Signals such as R, T

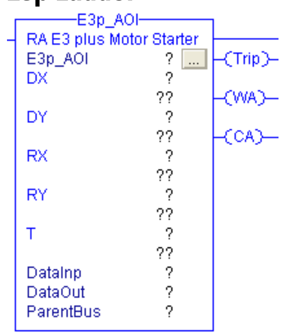
E3p Motor Operator Interface



E3p Function Block

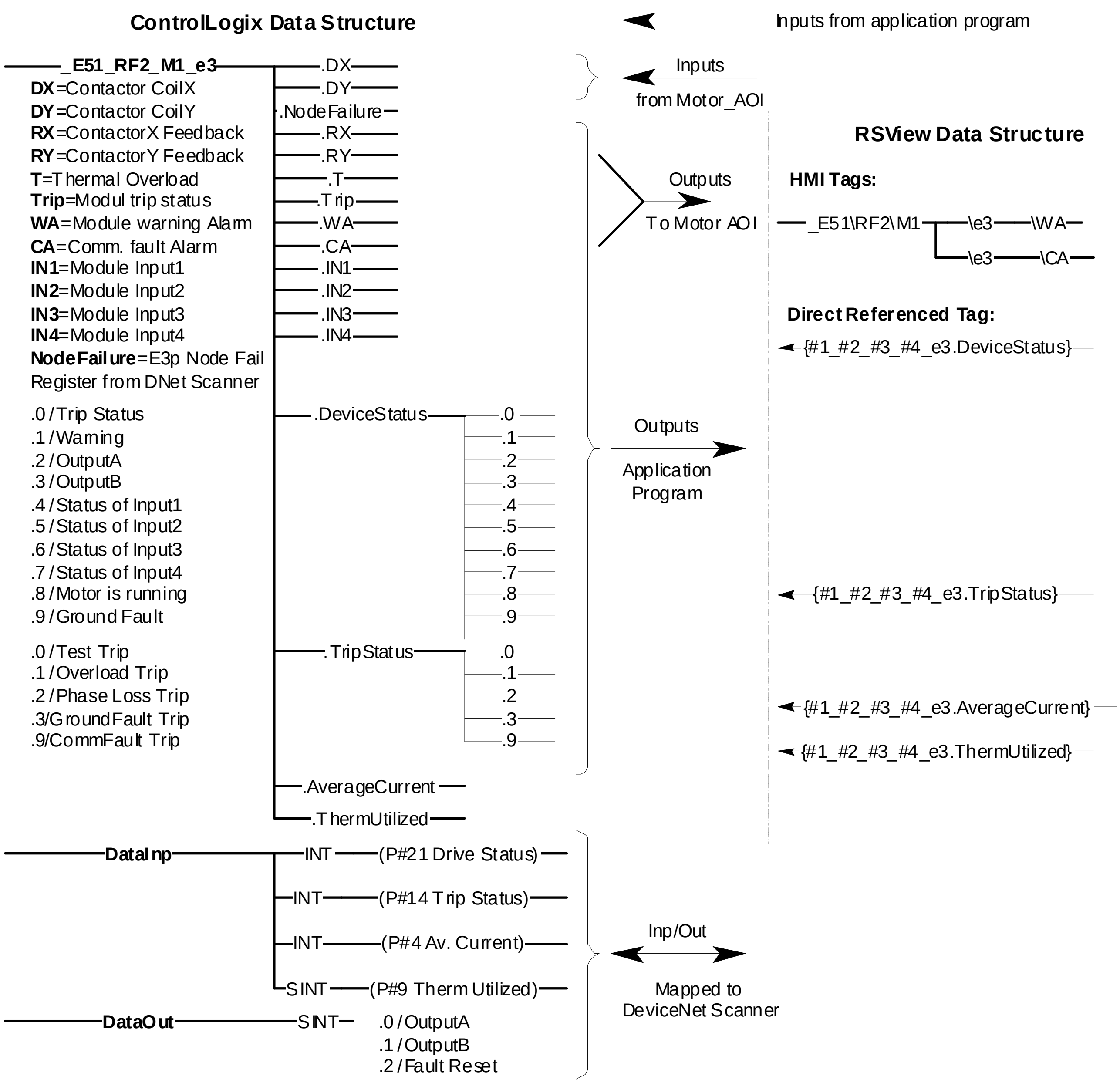


E3p Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

E3 Data Structure

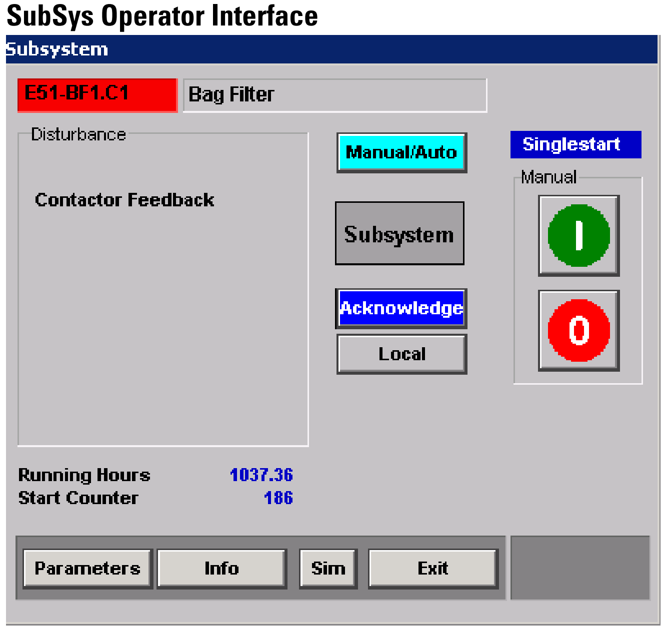


SubSys - Sub Control System Module

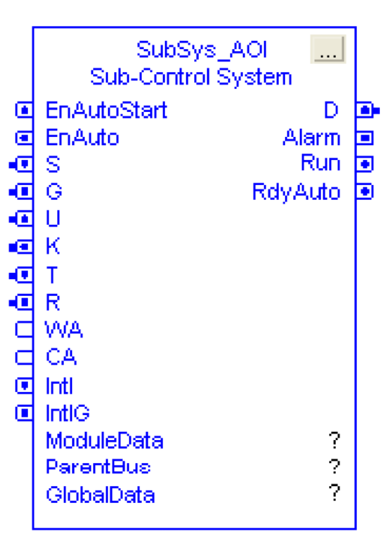
The Sub Control System (SubSys) supports all standard interlocking and supervision sub systems such as dust collectors, variable speed control panels, etc. This module is used for connecting non-MMCL controllers (PLC5, SLC500, 3rd party) to MMCL controllers. The module is similar to the MotorN, but includes additional unspecified alarm inputs. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of SubSys.

- Display template for FactoryTalk View SE
- Failure or Warning starter
- Enable automatic operation and enable auto start/restart
- Local or remote single start (password protected) including start-up Warning
- Detailed alarms for local isolator, MCC availability, thermal-OL, contactor feedback
- User specific warnings and/or alarm (Failure or Warning) inputs
- Safety alarm steady state interlock
- Machine protection interlock that may be overridden at local start
- Running hour counter and number of starts counter
- Parent bus link to Control Group or Machine Group Module

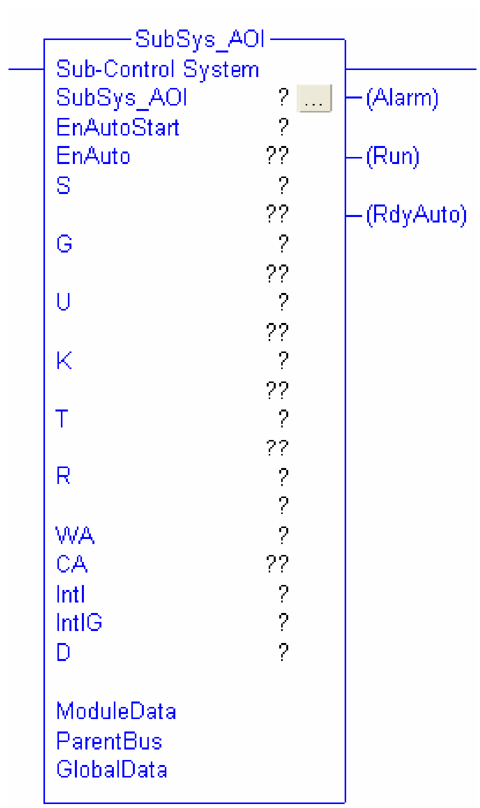
SubSys Operator Interface



SubSys Function Block

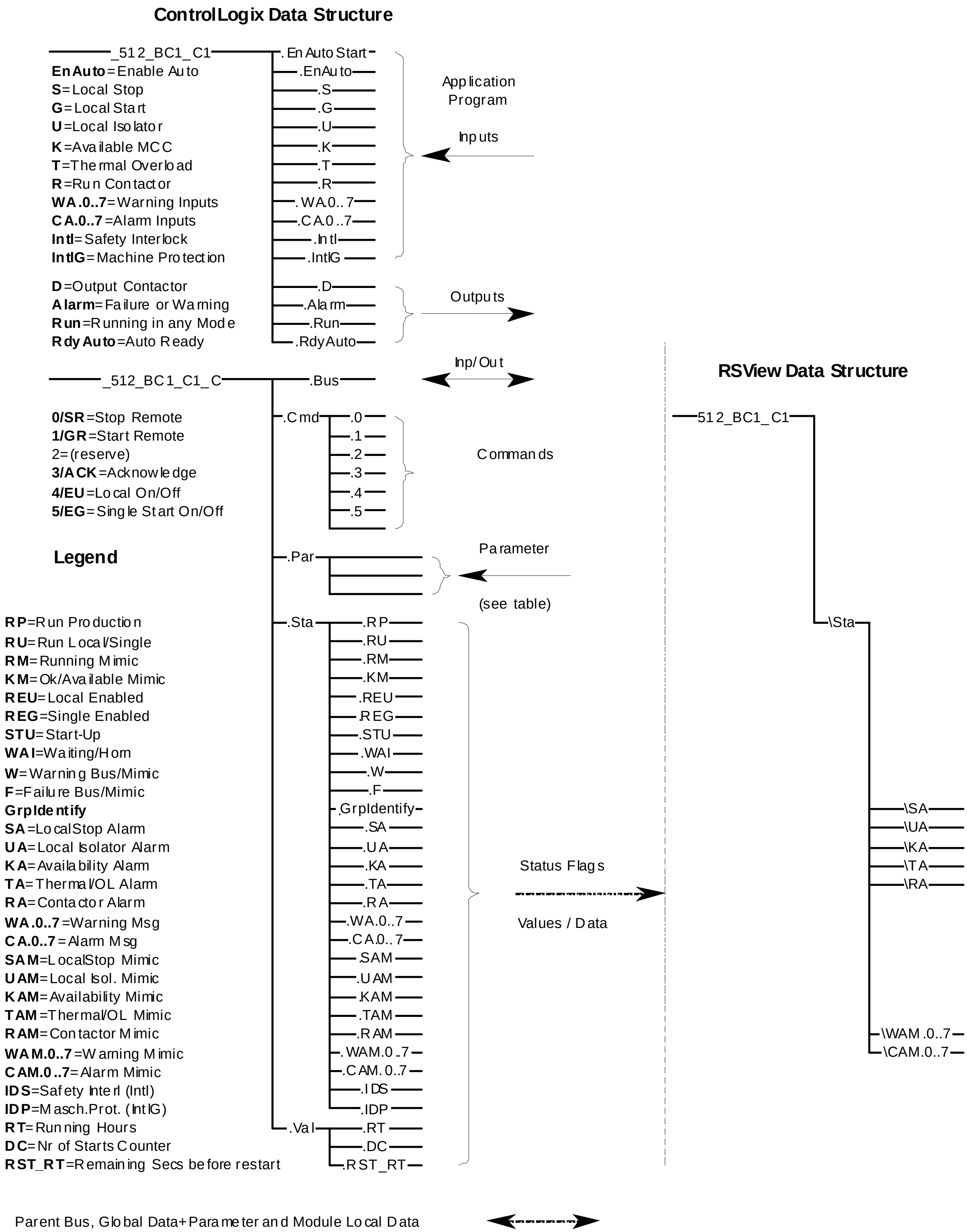


SubSys Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

SubSys Structure (Sub-System) – Interface to FactoryTalk View

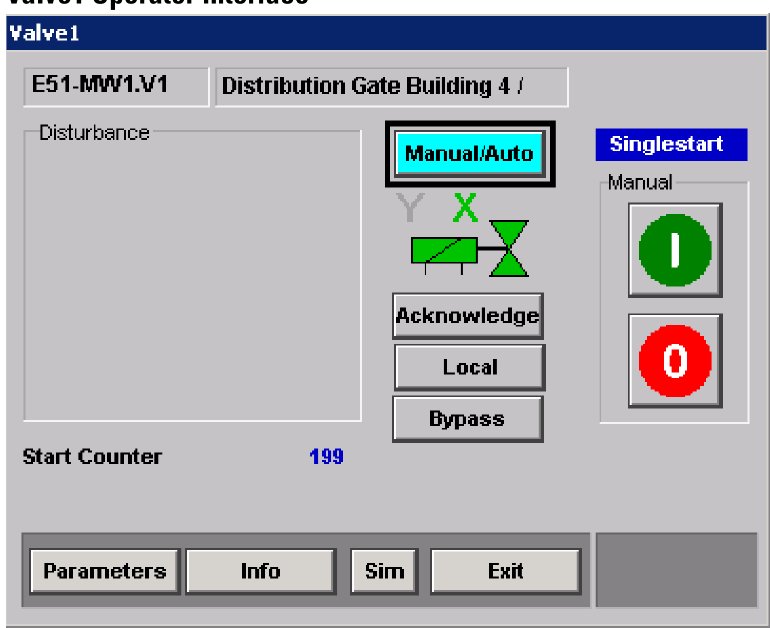


Valve1 - Valve with 1 Coil Module

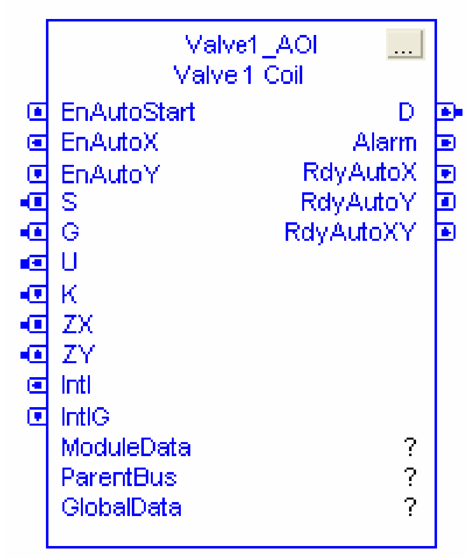
The Valve 1 Coil Module (Valve1) supports standard valves with one solenoid. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of Valve1.

- Display template for FactoryTalk View SE
- Failure or Warning Starter
- Enable automatic operation and enable auto start/restart
- Bypass Limit switch (password protected, internal simulation of limit switch)
- Local or remote single start (password protected) including start-up Warning
- Detailed alarms for local isolator, MCC availability and position limit switch feedback
- Safety alarm steady state interlock
- Machine protection interlock that may be overridden at local start
- Parameter hold output in case of an alarm
- Number of starts counter
- Parent bus link to Control Group or Machine Group Module
- Bypassing and simulating the limit switches

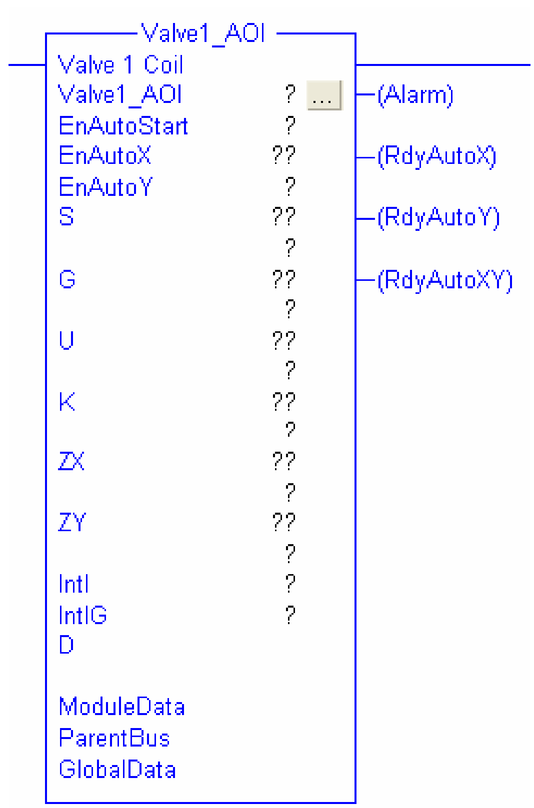
Valve1 Operator Interface



Valve1 Function Block

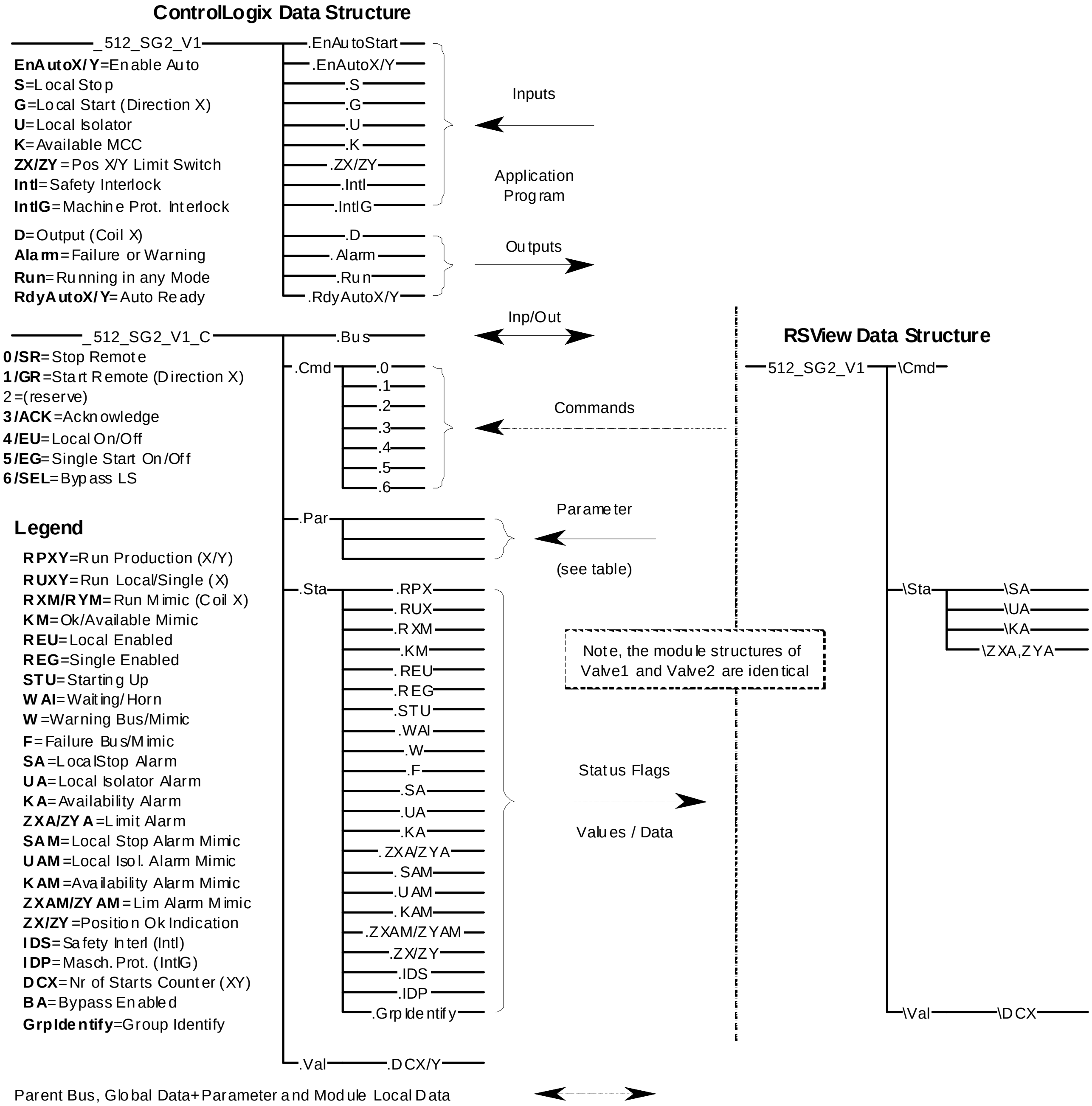


Valve1 Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

Valve1 Structure (One Coil) – Interface to FactoryTalk View

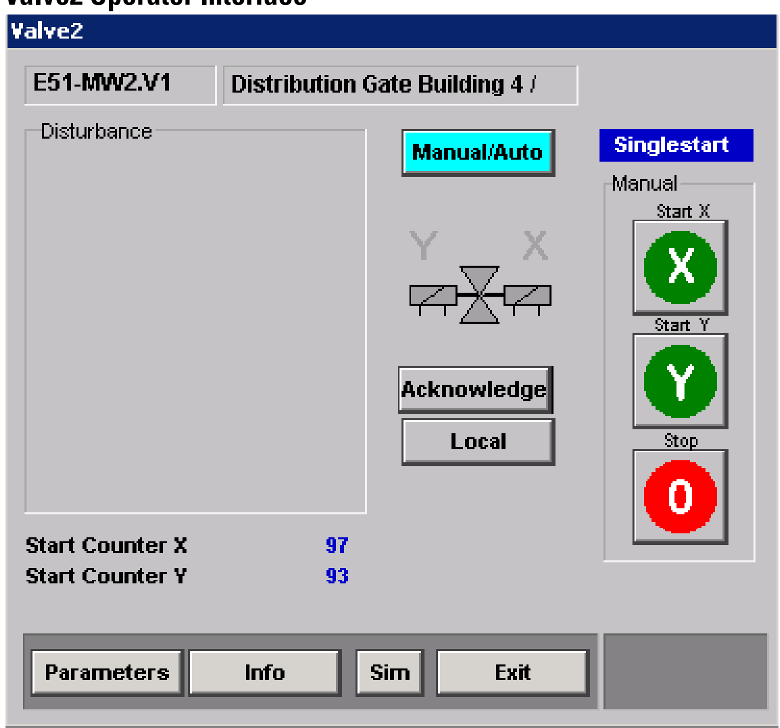


Valve2 - Valve with 2 Coils Modules

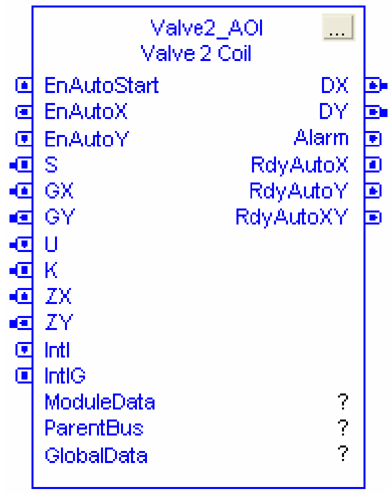
The Valve with 2 Coils Module (Valve2) supports standard valves with two solenoids. The module is linked with the Machine Group or Control Group Module and supports alarm control by creating messages that can be acknowledged individually or by the Control Group Module. The following are features of Valve2.

- Display template for FactoryTalk View SE
- Type Failure or Warning starter
- Open/close enable automatic operation and enable auto start/restart
- Open/close local or remote single start (password protected) including start-up Warning
- Bypass Limit switch (password protected, internal simulation of limit switch)
- Detailed alarms for local isolator, MCC availability and position limit switch feedback
- Safety alarm steady state interlock
- Machine protection interlock that may be overridden at local start
- Parameter set output OFF if in position, hold outputs in case of an alarm
- Number of starts counter open/close
- Parent bus link to Control Group or Machine Group Module
- Bypassing and simulating the limit switches

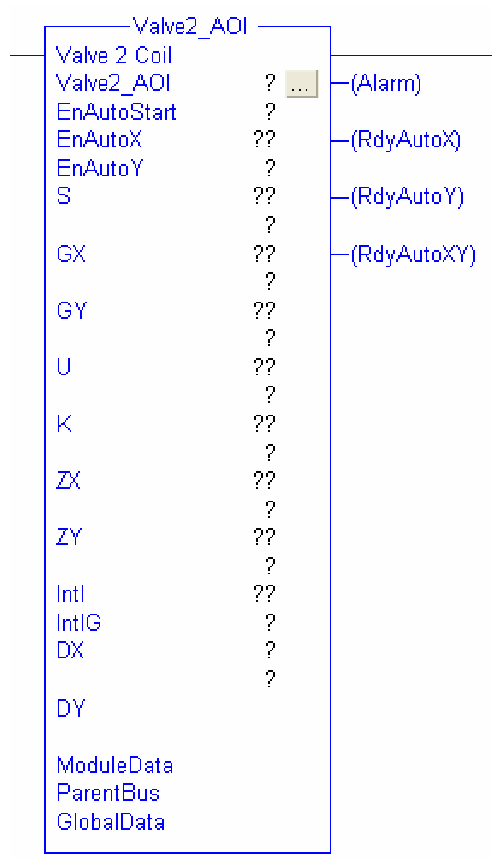
Valve2 Operator Interface



Valve2 Function Block

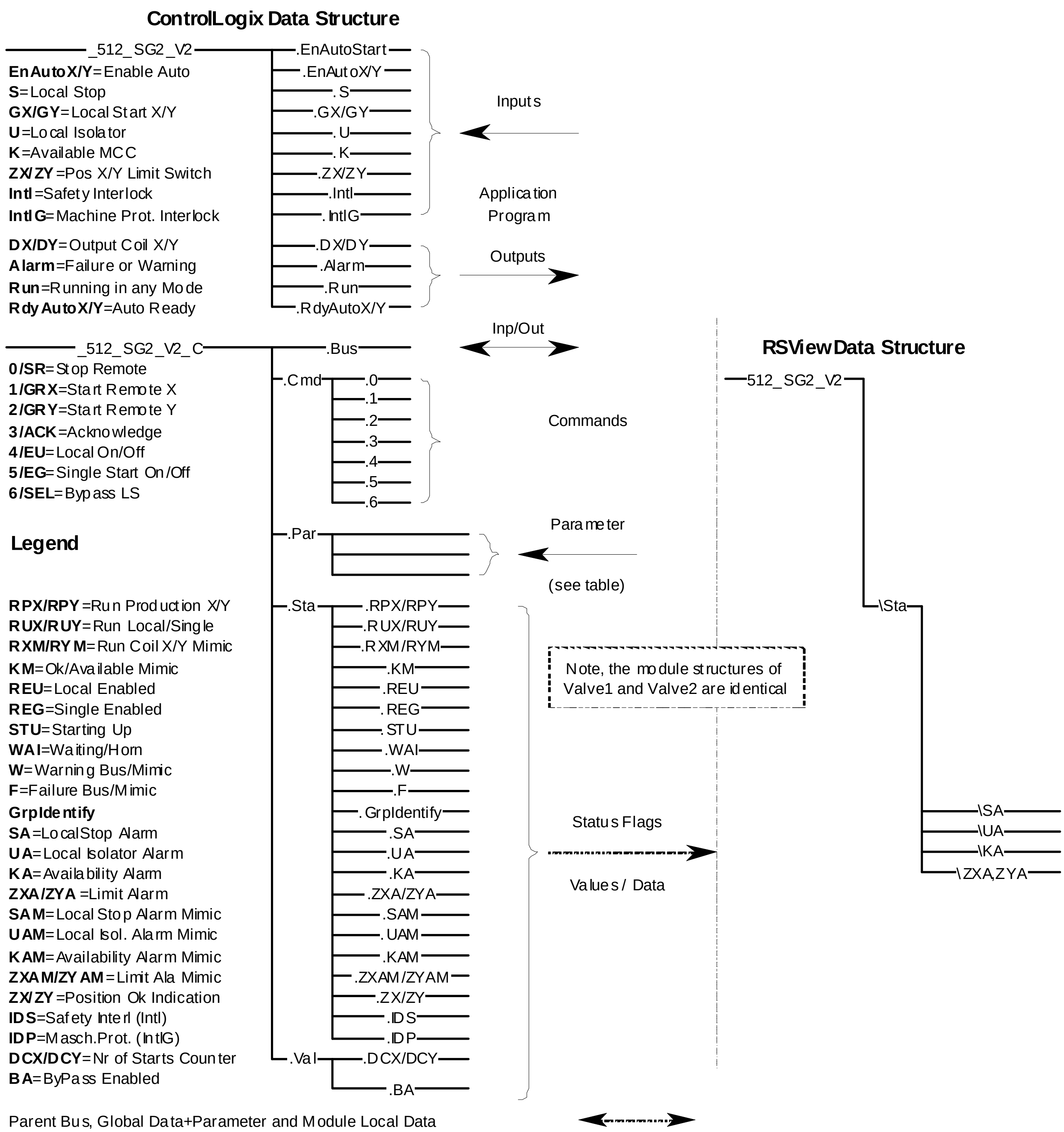


Valve2 Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

Valve2 Structure (Two Coils) – Interface to FactoryTalk View

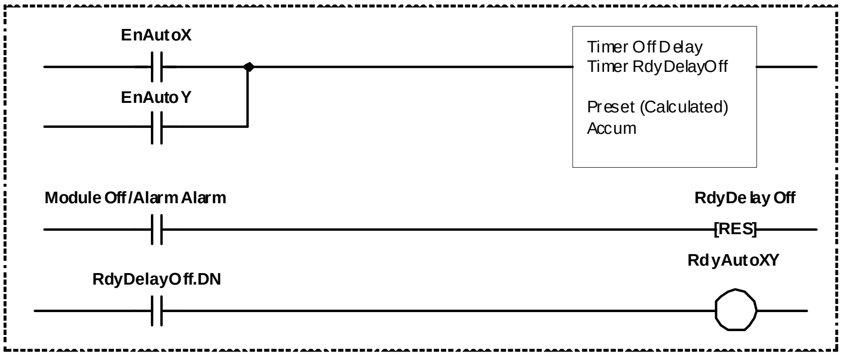


MotorN/D/R and SubSys Contractor Feedback

- The contactor feedback inputs R/RX/RY are directly used in HMI graphics by Sta.RM/RXM/RYM if ON; that is, if there is an alarm and a motor is still green, the contactor contact are weld on. This alarm is used to indicate a possible welded contact. An alarm Sta.RA is always released when the feedback does not correspond to the appropriate contactor output D/DX/DY.

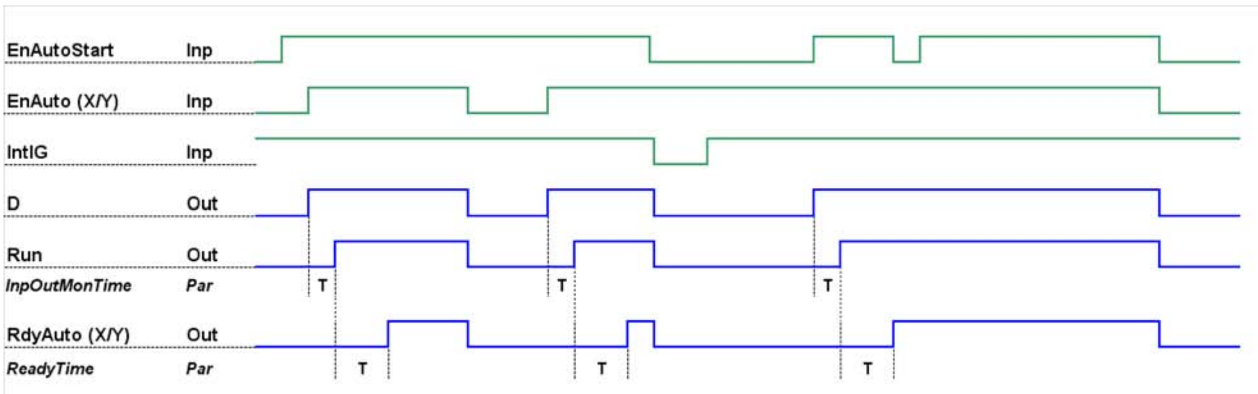
MotorD/R and Valve1/2 RdyAutoXY Signal Output

- All forward/reverse modules support an additional RdyAutoXY output that remains ON when reversing during automatic mode. Note that RdyAutoX and RdyAutoY are set to OFF when the motor is reversing or if the device is not in position.



- The RdyAutoXY output is a steady-state signal that may be used for interlocking if the device is allowed to stop or reverse on the fly in automatic mode.

Motor and Valve Automatic Start/Stop Timing Diagram

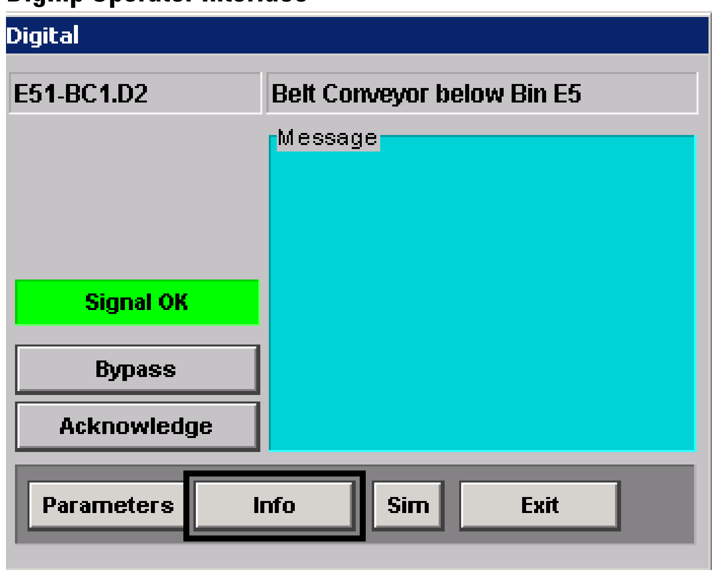


DigInp - Digital Input Module

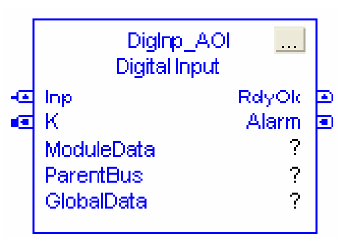
The Digital Input Module (DigInp) supervises input signals such as limit switches, level, speed detectors, etc. The module can be setup to directly take the pulse signals of a speed detector. The module can be linked with a Motor Control and Valve Module for dynamic or static inputs, or directly with the Machine or Control Group Module for steady-state inputs. The following are features of DigInp.

- Display template for FactoryTalk View SE
- Failure or Warning device
- Static, dynamic forward/reverse, dynamic or speed pulse supervision
- Bypass selector with diagnostic indication (password protected)
- Signal on indication (to remove bypass without the risk of an alarm)
- Parameter input signal debouncing
- Warning and/or alarm (Failure or Warning) timeout
- Ready OK output for parent module interlocking
- Parent bus link to Control Group, Machine Group or Motor/Valve/SubSys Module

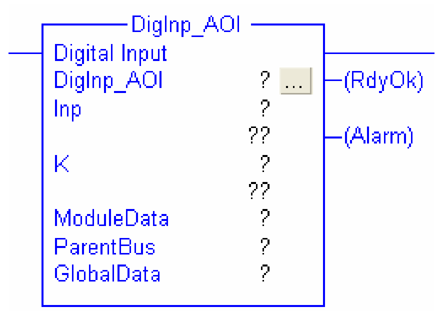
DigInp Operator Interface



DigInp Function Block

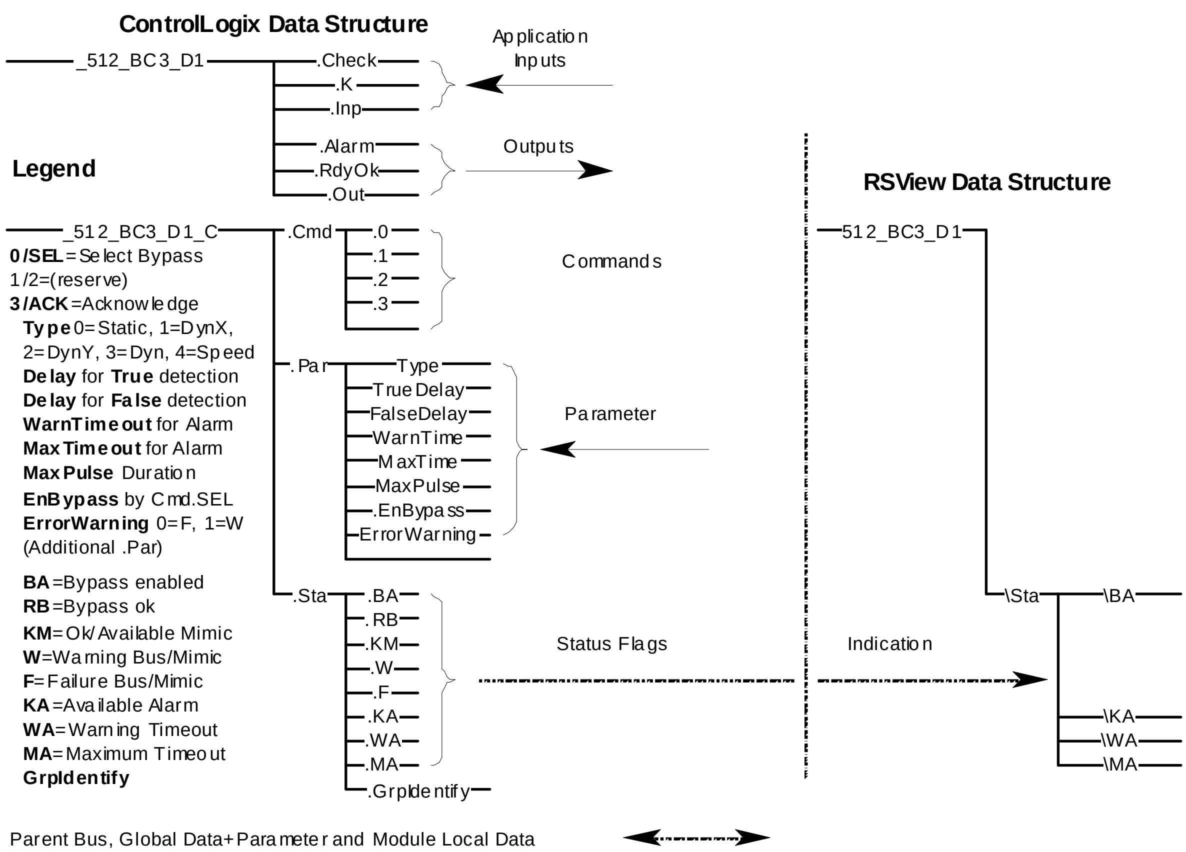


DigInp Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

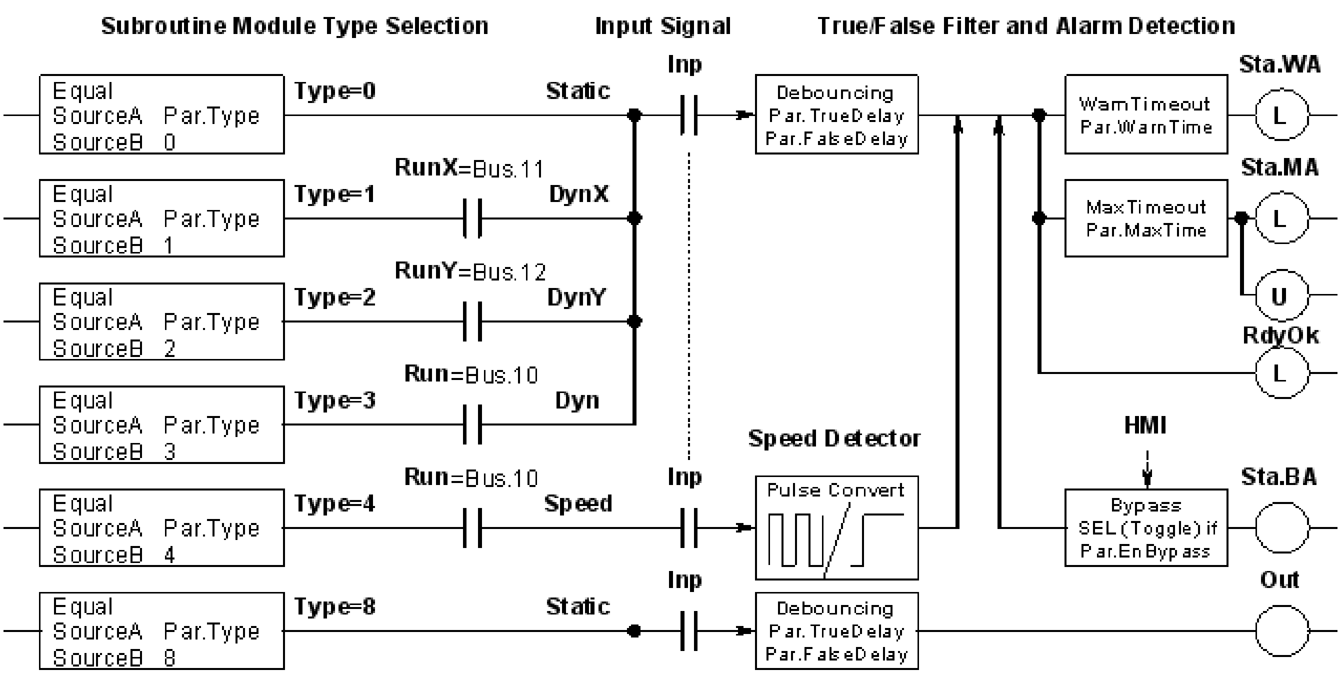
DigInp Structure – Interface to FactoryTalk View



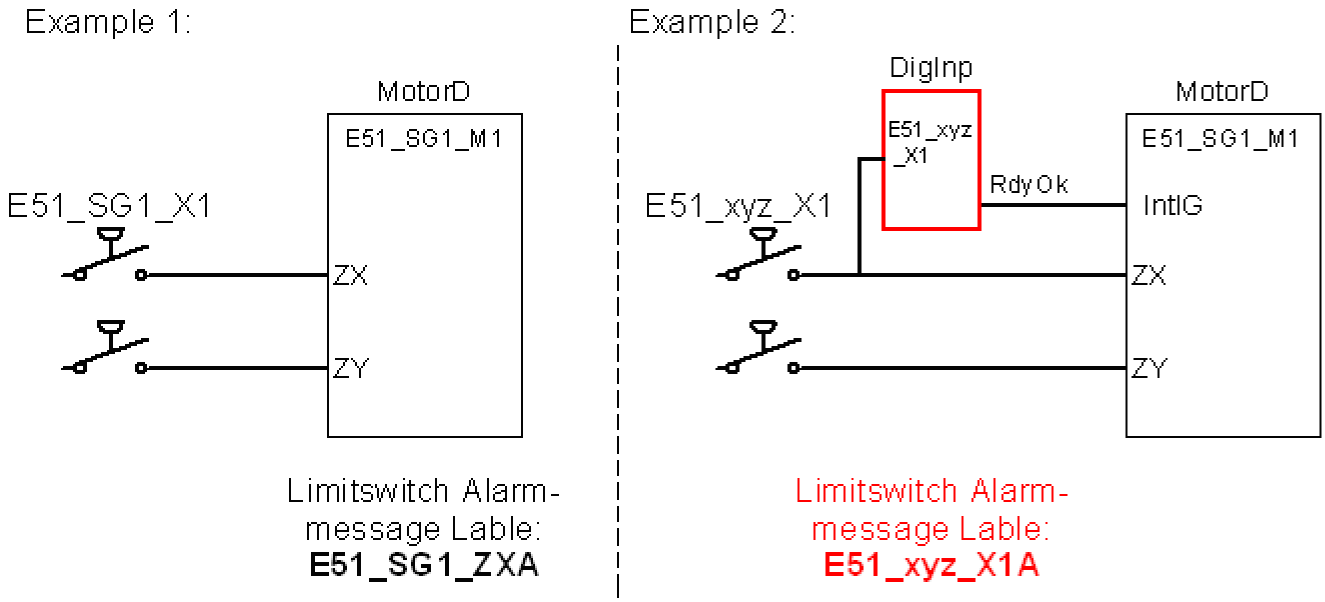
DigInp Functions

The Digital Input Module performs the following main functions.

- Input logic NC/NO selectable
- Type (8) Module without alarm, Output follows the filtered (debouncing) input only
- Supervision of inputs and interlocking for parent modules such as MotorX, SubSys and ValveX
- Four different types of inputs: Static, Dynamic X, Dynamic Y, Dynamic and Pulse (speed)
- Two alarm levels with separate Warning and Failure Messages
- Configurable bypass selection for the operator/engineer (password protected)
- On a MaxTimeout, the output is set at RdyOk=OFF until the input signal is ok again and the alarm is acknowledged. The availability indication stays at Sta.KM=OFF until the input is ok. Additionally, for dynamic inputs the Parent device must run to set Sta.KM=ON again.



Example: How To Use DigInp as Limitswitch

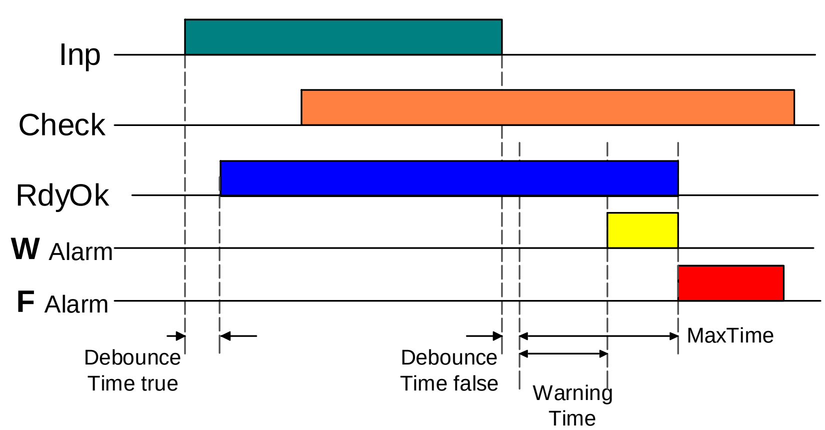


DigInp creates its one Alarm message in the HMI Alarm List (AC compliant).

If the limitswitch, for example, is not in the same location as the Motor then you can use a DigInp module to supervise this limitswitch.

Typical Signal Timing

Par.Type=0 **Static NC**



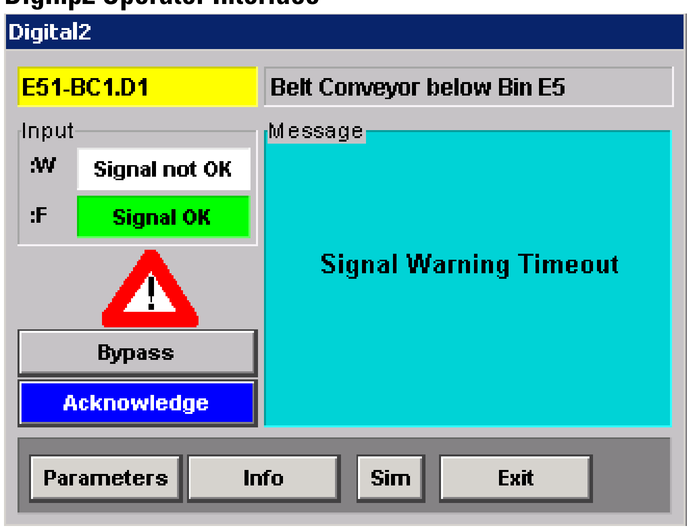
DigInp2 - Digital Input Module

The Digital Input Module (DigInp2) supervises input signals such as drift switch, etc. This module is used for drift switches or other devices which have two contacts wired, one for warning (level 1) and one for failure (level 2, over travel). The Input signal, InpW, will generate only a warning alarm. The DigInp2 can not be used as a speed detector (only DigInp), however the main functionality is nearly the same as the DigInp module.

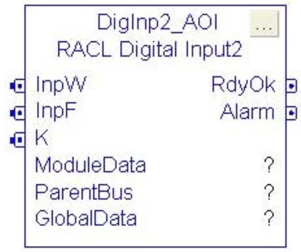
The module can be linked with a Motor Control and Valve Module for dynamic or static inputs, or directly with the Machine or Control Group Module for steady state inputs. The following are features of DigInp2.

- Display template for FactoryTalk View SE
- Failure or Warning device
- Static, dynamic forward/reverse, dynamic supervision and input logic NO/NC
- Bypass selector to bridge the InpF, with diagnostic indication (password protected)
- Signal on indication (to remove bypass without the risk of an alarm)
- Parameter input signal debouncing
- Independent Warning and Failure alarm input
- Ready OK output for parent module interlocking
- Parent bus link to Control Group, Machine Group or Motor/Valve/SubSys Module

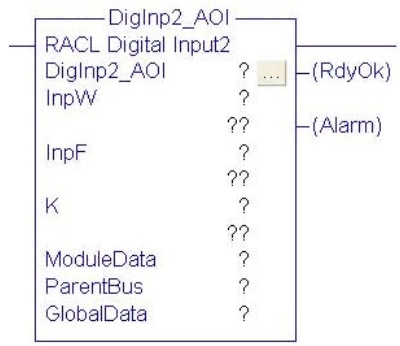
DigInp2 Operator Interface



DigInp2 Function Block



DigInp2 Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

DigPulse - Digital Pulse Module

The Digital Pulse Module is designed to supervise motion input signals coming directly from a sensor (pulse) or from an external speed detector device (maintain signals) on equipment that rotates, like a belt conveyer. The module takes care of the differed monitoring states, for example, when the motor is starting (Acc), running (Run), or stopping phase (Dec). These monitoring states use different adjustable supervision timers.

The percentage value of a Variable Speed Drive (VSD) Input is used as a factor by calculating the supervision timer in Acc and Run state.

The module can be set up to directly take the pulse signals of a speed detector or take a maintain signal form a signal evaluator.

The module considers the special situation with the variable accelerates of a drive and also in connection with variable speed during normal operation. frog, not sure about this sentence

The module can be linked with a Motor Control and Valve Module or directly with the Machine or Control Group Module.

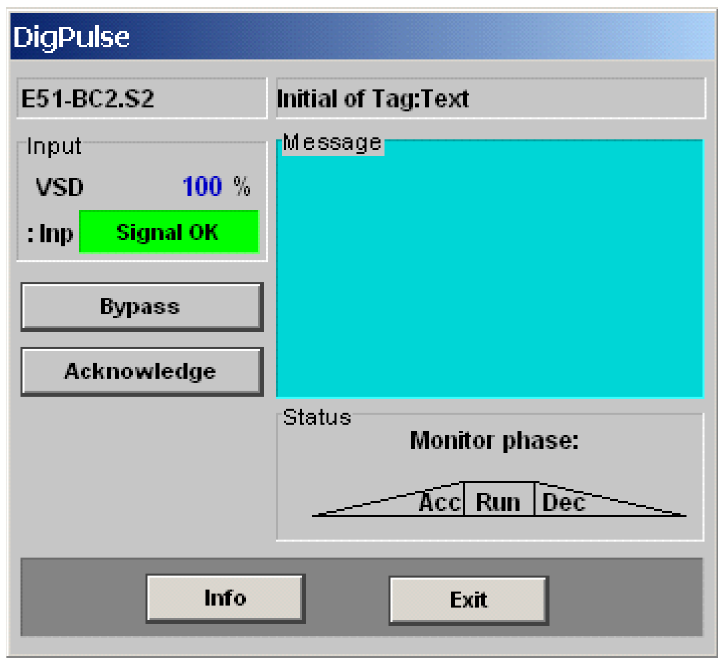
The following are features of DigPulse.

- Display template for FactoryTalk View SE
- Failure or Warning device
- Input type configuration for pulse supervision or maintain input signals (NC/NO)
- Bypass selector with diagnostic indication (password protected)
- Signal on indication (to remove bypass without the risk of an alarm)
- VSD Variable Speed Device Input [0-100%] influences the Alarm supervision
- Parameter for different start- and stop- Supervision Timer
- Warning and/or alarm (Failure or Warning) timeout
- Ready OK output for parent module interlocking
- Parent bus link to Control Group, Machine Group or Motor/Valve/SubSys Module

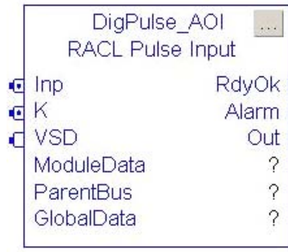
The module response for warning and failure alarms considers the motor running condition where there are 3 motor states: starting, running, and stopping/stopped. Because of the size of the equipment, the momentum (inertia) causes the equipment to coast for a while after stopping. With timer values Parameter for the warning/failure so that the system can be tuned for the coasting equipment.

An example of this is long belt that stretches like a rubber band. Because of this stretching, it takes a moment for the far end of the belt to move after the motor starts.

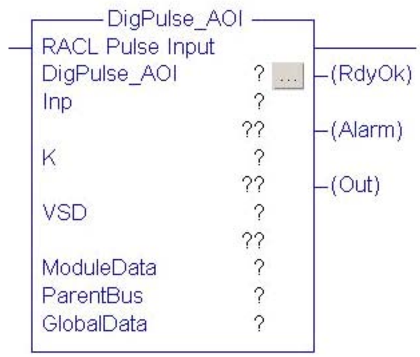
DigPulse Operator Interface



DigPulse Function Block

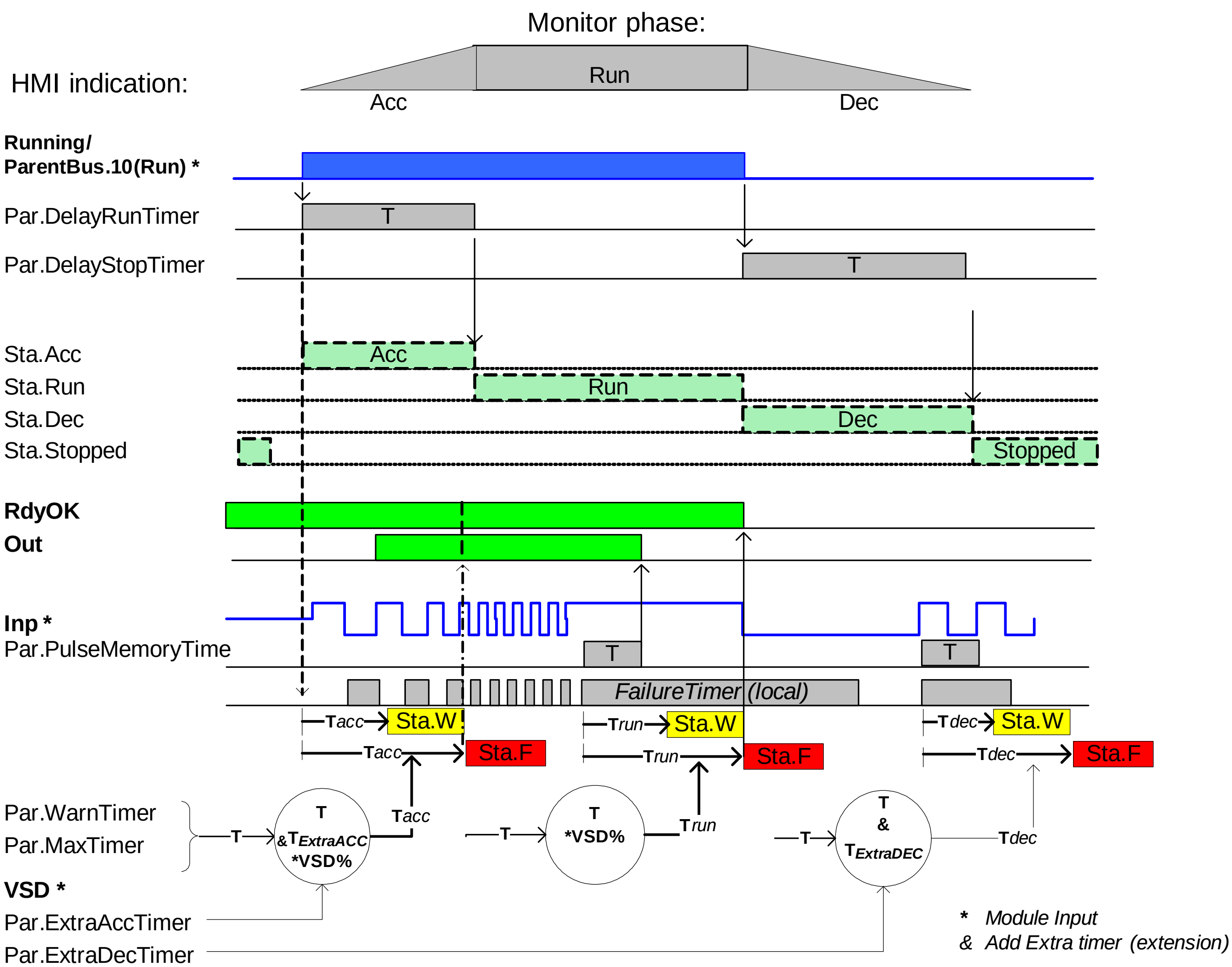


DigPulse Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

DigPulse Timing Diagram



Notes:

Process Control Modules

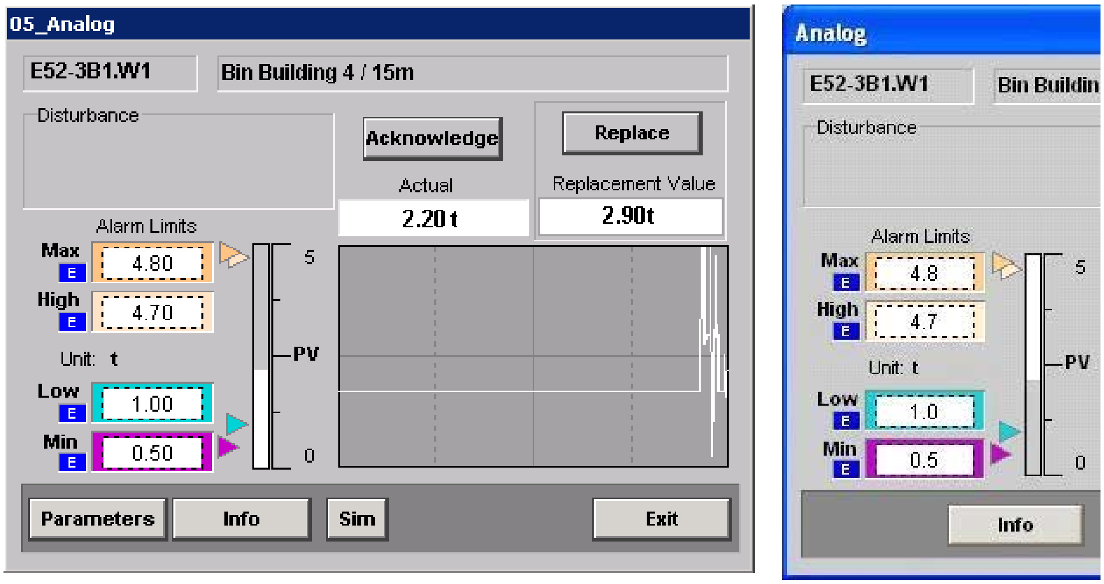
Process control is supported by the following modules that include an operator interface which is supplied as a ready-to-use HMI display.

AnaInp - Analog Input Module

The Analog Input Module (AnaInp) supports scaling, filtering and supervising of an analog input. Values are conditioned and checked for underflow, overflow, and limits. The following are features of AnaInp.

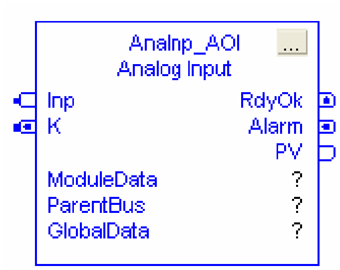
- Display template for FactoryTalk View SE
- Failure or Warning device
- Max/high/low/min level alarm enable and supervision type static or dynamic
- Replacement/bypass value with diagnostic indication (password protected)
- Auto bypass selection on error, actual value indication (remove bypass without alarm)
- Input scaling from global system group table for selected converter types
- Supports negative gradient scaling if max scale is set less than min scale
- Parameter sample rate and input filter max gradient = 1..100% of range per second
- Parameter alarm delay, alarm deadband and min/max clamping deadband
- Ready OK output for parent module interlocking
- Parent bus link to Control Group, Machine Group or Motor/Valve/SubSys Module

AnaInp Operator Interface

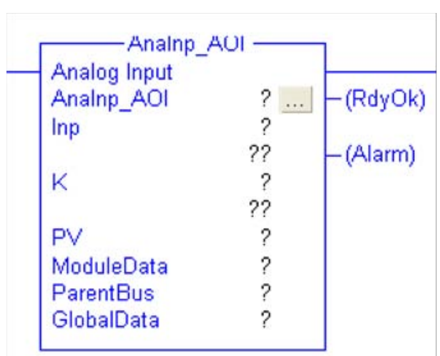


For Negative Gradient Scaling, the bar graph begins at the top

AnaInp Function Block

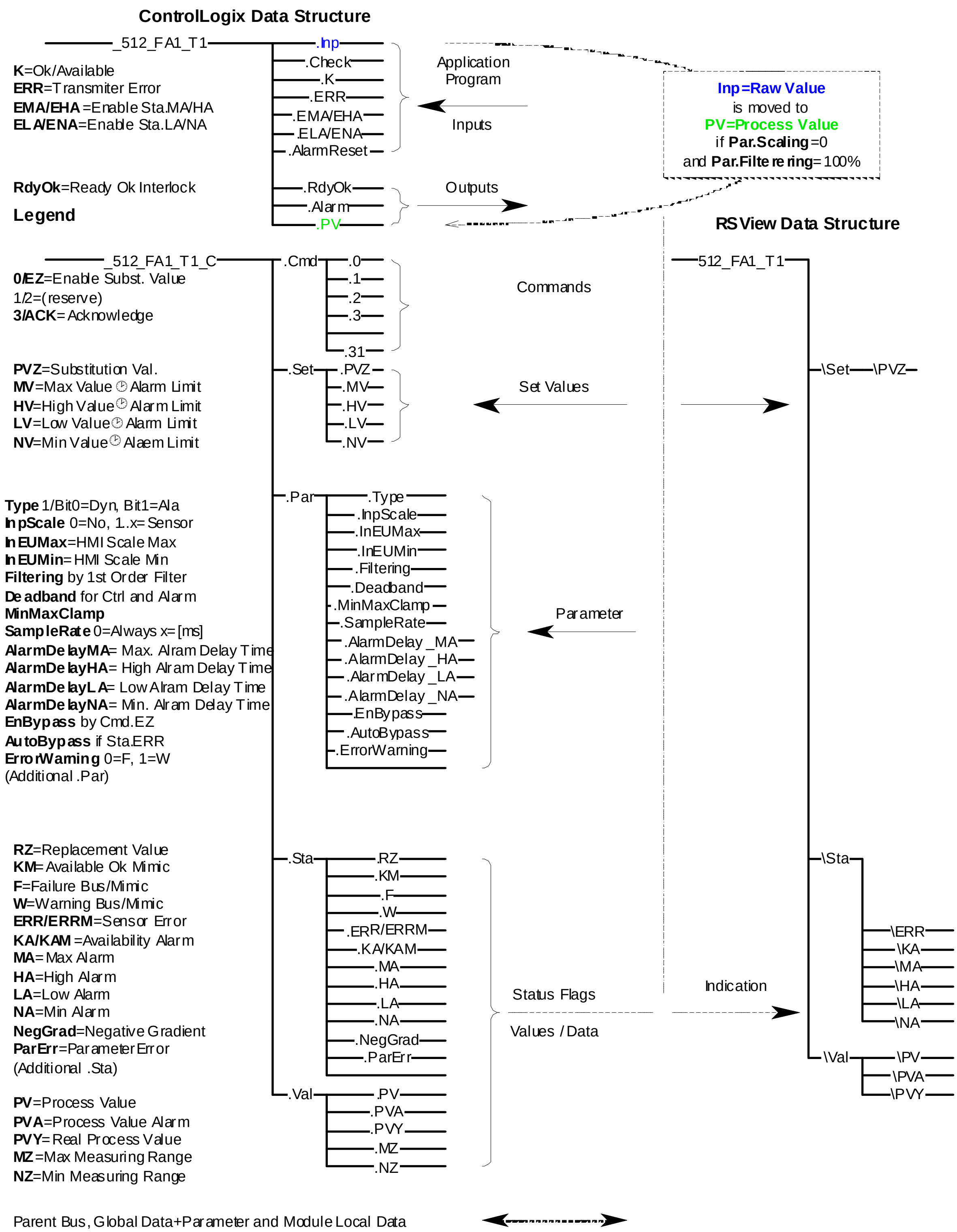


AnaInp Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

AnaInp Structure - Interface to FactoryTalk View

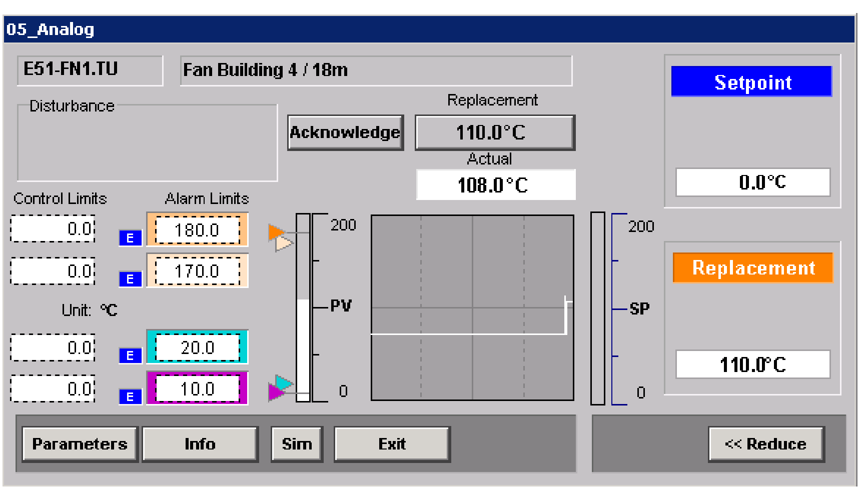


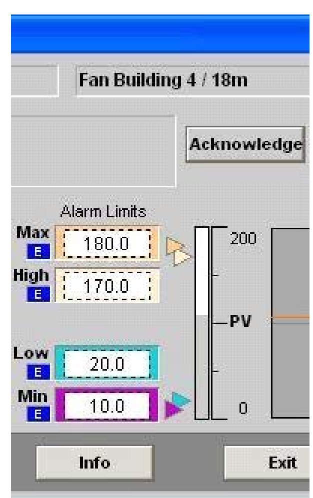
AnaInpC - Analog Input Control Module

The Analog Input Control Module (AnaInpC) supports scaling, filtering and supervising analog inputs. Values are conditioned and checked for underflow, overflow, and limits to create alarms. Also, four digital outputs may be used for controlling any discrete device. The module has a built-in Setpoint that can be used in conjunction with the input. The following are features of AnaInpC.

- Display template for FactoryTalk View SE
- Failure or Warning device
- Max/high/low/min level alarm enable and supervision type static or dynamic
- Max/high/low/min level control limit outputs type static or dynamic
- Replacement/bypass value with diagnostic indication (password protected)
- Auto bypass selection on error, actual value indication (remove bypass without alarm)
- Setpoint with automatic back-tracking if not enabled for the operator
- Input scaling from global system group table for selected converter types
- Supports negative gradient scaling if max scale is set less than min scale
- Parameter sample rate and input filter max gradient = 1..100% of range per second
- Connector for external filtering and smoothing the input signal
- Parameter alarm delay, alarm deadband and min/max clamping deadband
- Ready Ok output for parent module interlocking
- Parent bus link to Control Group, Machine Group or Motor/Valve/SubSys Module

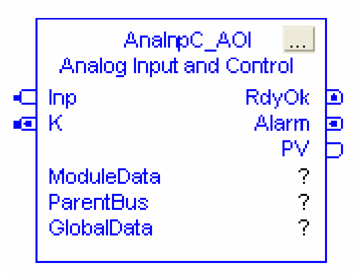
AnaInpC Operator Interface



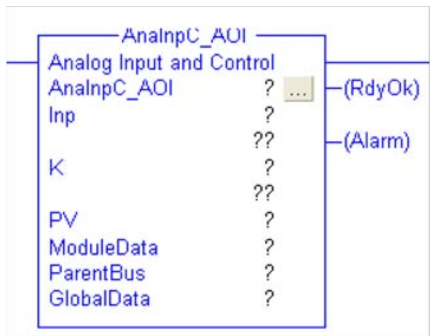


For Negative Gradient Scaling, the bar graph

AnalnpC Function Block

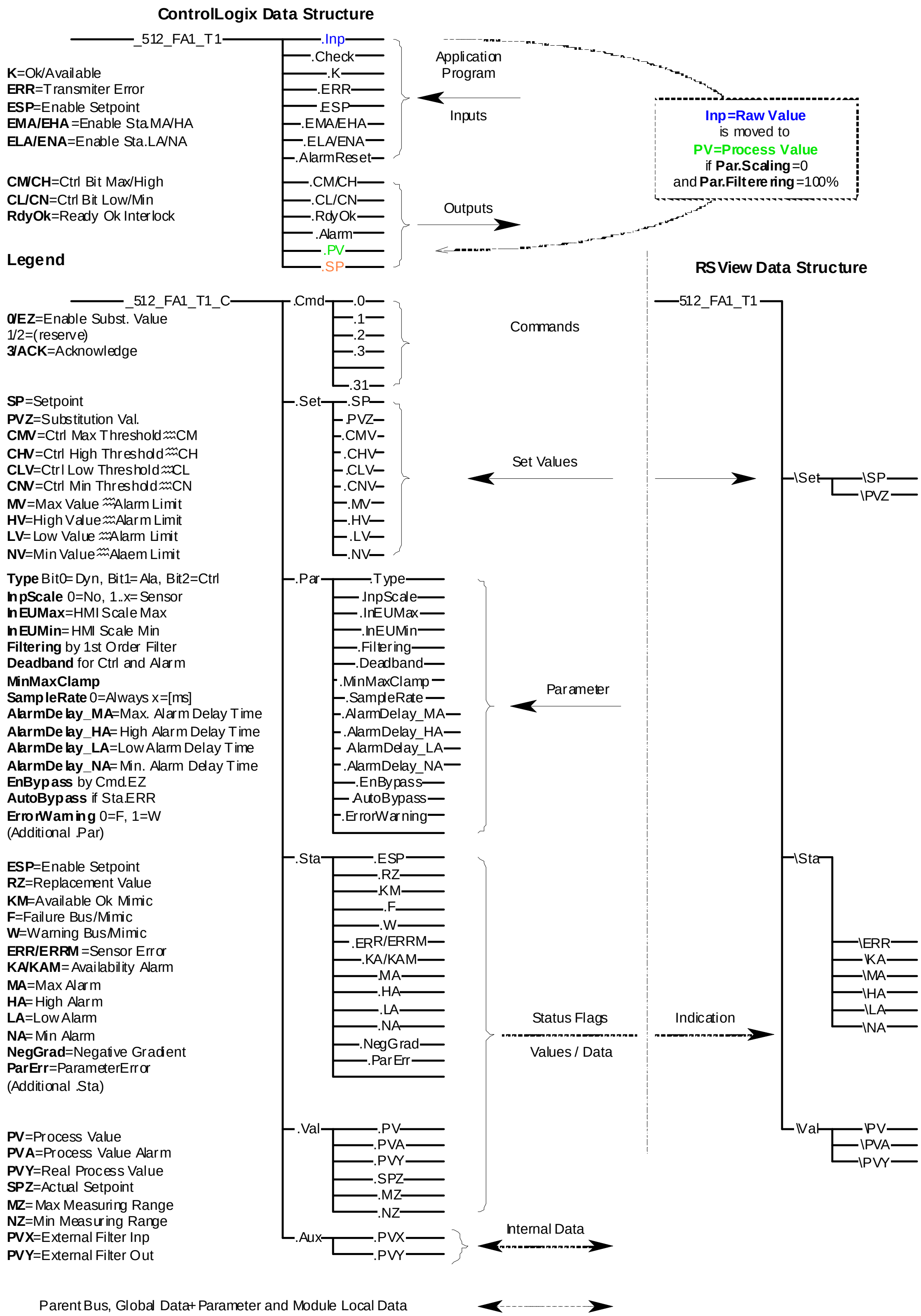


AnalnpC Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

AnaInpC Structure – Interface to FactoryTalk View



AnaInp and AnaInpC Process Variable Scaling

Raw input value scale, underflow, and overflow values are taken from Global.Par.AnaInpScale[x]; where x=1..9 is specified by Par.InpScale. If Par.InpScale=0 then the raw value is transmitted to the module's PV without scaling.

$$PV_{Scaled} = \frac{(\mathbf{In} - \mathbf{InRawMin}) * (\mathbf{Par.InEUMax} - \mathbf{Par.InEUMin})}{(\mathbf{InRawMax} - \mathbf{InRawMin})} + \mathbf{Par.InEUMin}$$

Where the Raw Scale InRawMax/Min is specified by Par.InpScale=x and taken from Global.Par.AnaInpScale[x].InRawMax/Min
(for x=1..9 Array Offset, see System Group Global Data)
the Scaled Range InEUMax/Min is specified by Par.InEUMax/Min

Note: InRawMax must be greater than InRawMin, and InEUMax must not be equal InEUMin else Parameter Error Warning Sta.ParErr is displayed and the module is not processed.

Positive Gradient -> if InEUMax > InEUMin then Sta.NegGrad=OFF and application PV MaxRange Val.MZ=InEUMax,
MinRange Val.NZ=InEUMin

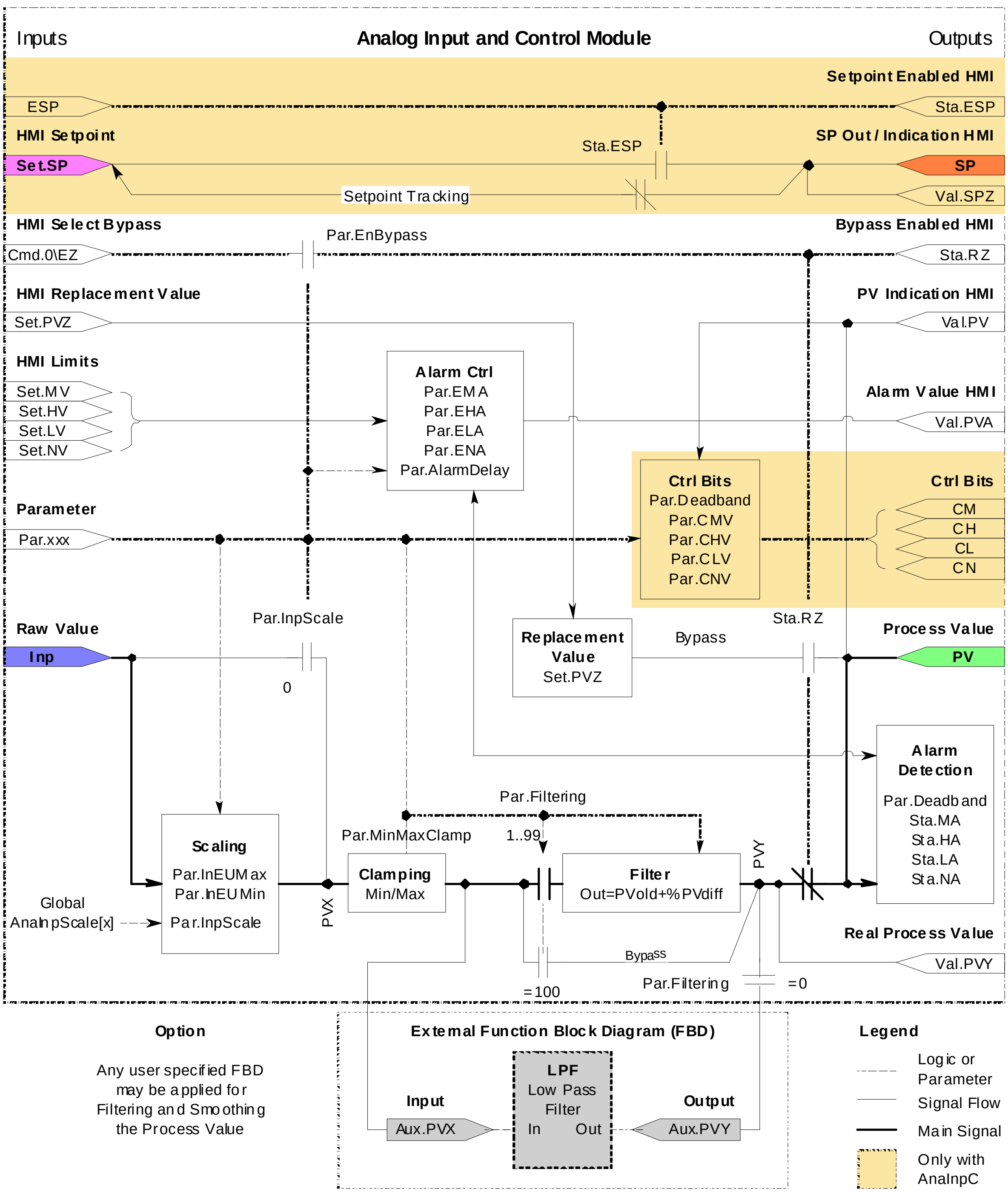
Negative Gradient -> if InEUMax < InEUMin then Sta.NegGrad=ON and application PV MaxRange Val.MZ=InEUMin,
MinRange Val.NZ=InEUMax

Internal Filter PVY = PVXold+Filtering*(PVX-PVXold)/100*SampleTime/s

Where PVXold is PVX of the previous scan, Filtering = 1..99 [%] (100% = Filter bypassed) and
SampleTime/s = actually measured SampleRate/1000 [ms/ms] (Sample Timer ACC Value)

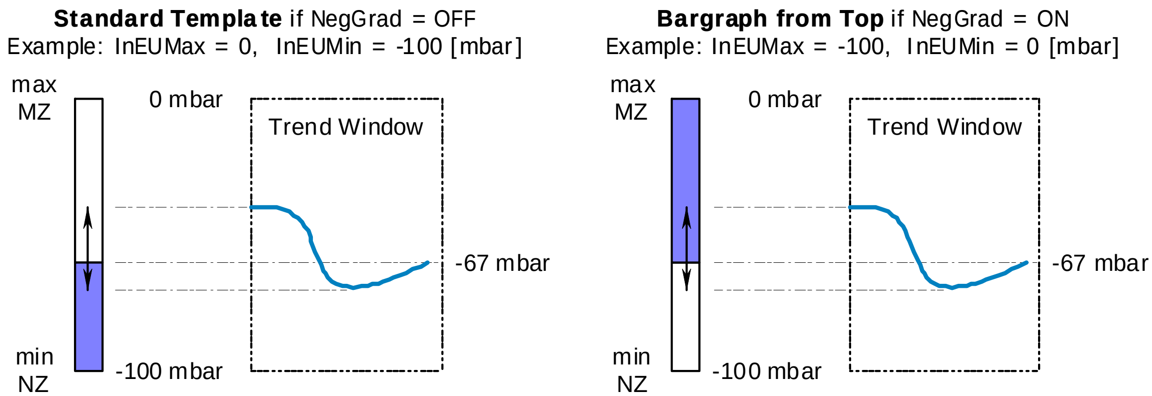
AnaInp and AnaInpC Functional Diagram, Scaling, Alarming and Filtering

The Setpoint SP is fed back (tracking) to the HMI input Set.SP unless enabled by Sta.ESP. Thus, if not enabled, the Setpoint SP can be overwritten by the user program.



AnaInp and AnaInpC Negative Gradient Scaling

The Analog Input Module supports Negative Gradient Scaling if the max scale input Par.InEUMax is set less than the min scale input Par.InEUMin. This is indicated by status flag Sta.NegGrad=ON, the flag can be used to invert the bar graph to display from the top down in the HMI display.



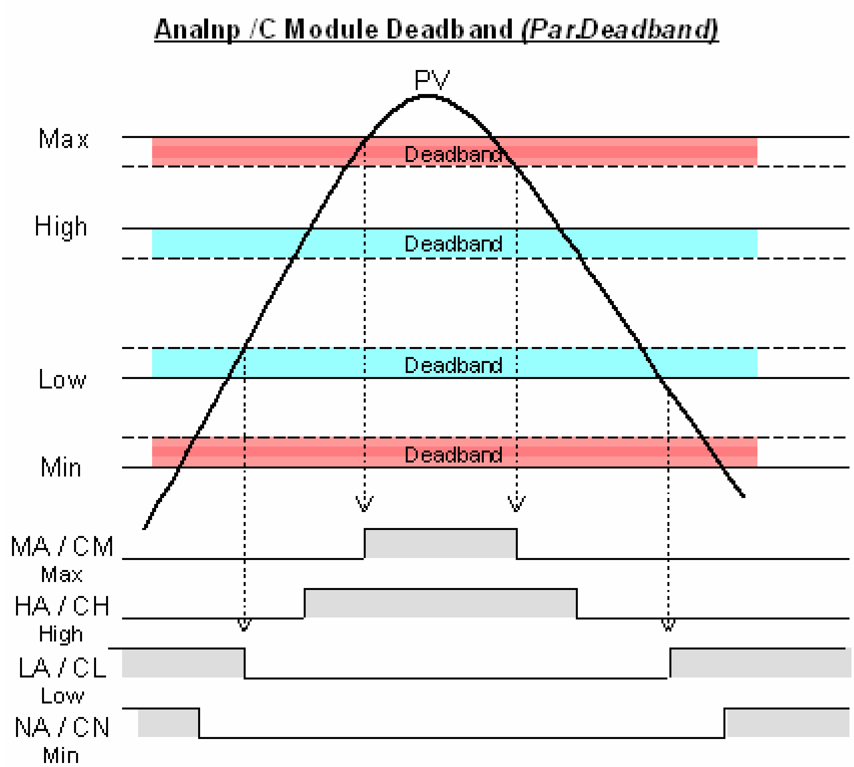
This method allows the indication of under pressure the same way as positive or negative pressures. For all indications, however, the max scale Val.MZ is set greater than the min scale Val.NZ; that is, if Sta.NegGrad = ON-
then Val.MZ = Par.InEUMin and Val.NZ = Par.InEUMax.

AnaInpC External Function Blocks for Input Signal Treatment

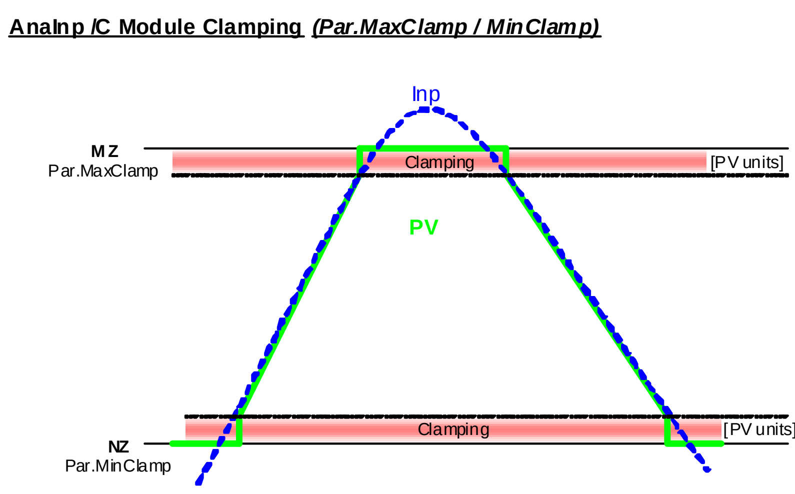
The following table shows a set of possible functions if Par.Filtering = 0 (option).

Function Name	Description (AnaInpC option only)
Low Pass Filter (LPF)	Filters input frequencies that are above the cutoff frequency
High Pass Filter (HPF)	Filters input frequencies that are below the cutoff frequency
Notch Filter (NTCH)	Filters input frequencies that are at the notch frequency
Maximum Capture (MAXC)	Finds the maximum signal in time
Minimum Capture (MINC)	Finds the minimum signal in time
Function Generator (FGEN)	Converts an input based on a piece-wise linear function
High/Low Limit (HLL)	Limits an analog input between two values
Lead-Lag (LDLG)	Provides a phase lead-lag compensation for an input signal
Second-Order Lead Lag (LDL2)	Filters with a pole pair and a zero pair
Moving Average (MAVE)	Calculates a time average value
Moving Standard Deviation (MSTD)	Calculates a moving standard deviation
Ramp/Soak (RMPS)	Provides for alternating ramp and soak periods
Totalizer (TOT)	Provides a time-scaled accumulation of an analog input value
Derivative (DERV)	Calculates the amount of change of a signal over time in per-second units
Enhanced Select (ESEL)	Selects one of as many as six inputs
Multiplexer (MUX)	Selects one of eight inputs
Rate Limiter (RLIM)	Limits the amount of change of a signal over time
Select (SEL)	Selects one of two inputs
Selected Negate (SNEG)	Selects between the input value and the negative of the input value
Selected Summer (SSUM)	Selects real inputs to be summed

Deadband Diagram



Clamping Diagram

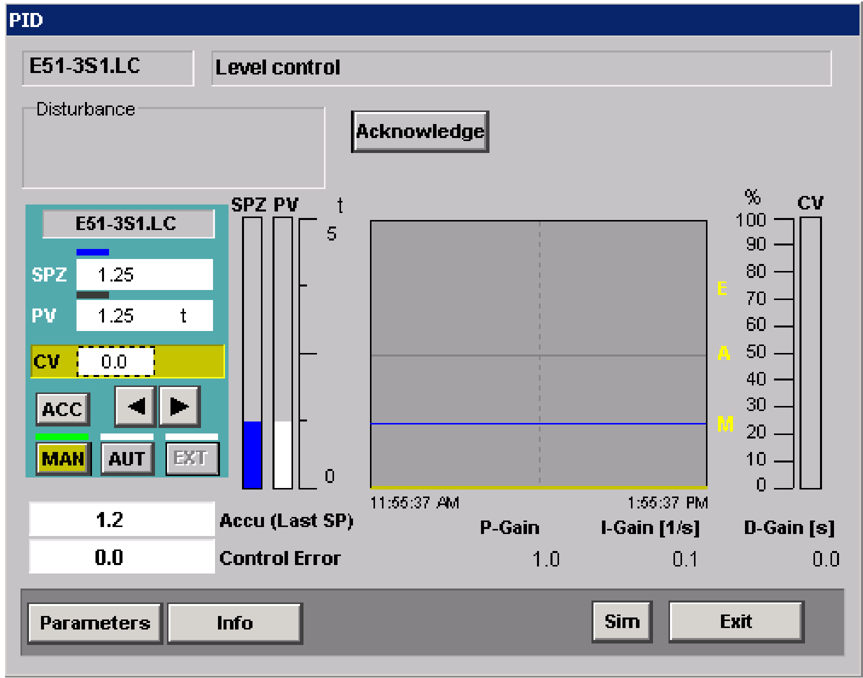


PidMod - Proportional Integral Derivative Module

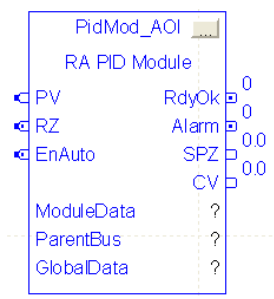
The Proportional Integral Derivative Module (PidMod) uses the RSLogix 5000 standard PID (non PIDE) instruction to control a process variable, such as flow, pressure, temperature, or level. The module can be selected for manual, automatic, and external operation, and is linked with a Machine or Control Group Module. The following are features of PidMod.

- Display template for FactoryTalk View SE
- Failure or Warning Device
- Use of RSLogix 5000 standard PID Instruction
- Setpoint with automatic Back-Tracking, if not enabled for the operator
- Accumulated Value (Memory) to re-instate last/old valid Setpoint in Auto Mode
- Control Output Manual setting with automatic Back-Tracking if not enabled for the operator
- Adjustable Proportional, Integral and Derivative Terms (Password Protected)
- Max/High/Low/Min Level Alarm Enable and Supervision
- Parameter Alarm Delay and Update Rate
- Ready Ok Output for Interlocking
- Parent Bus Link to Control Group or Machine Group Module

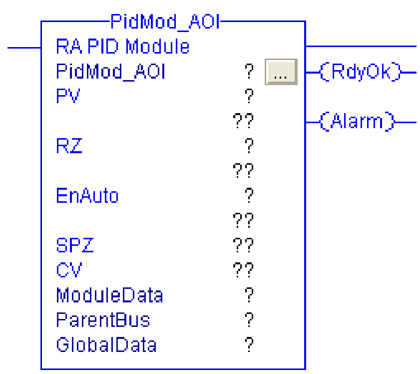
PidMod Operator Interface



PidMod Function Block

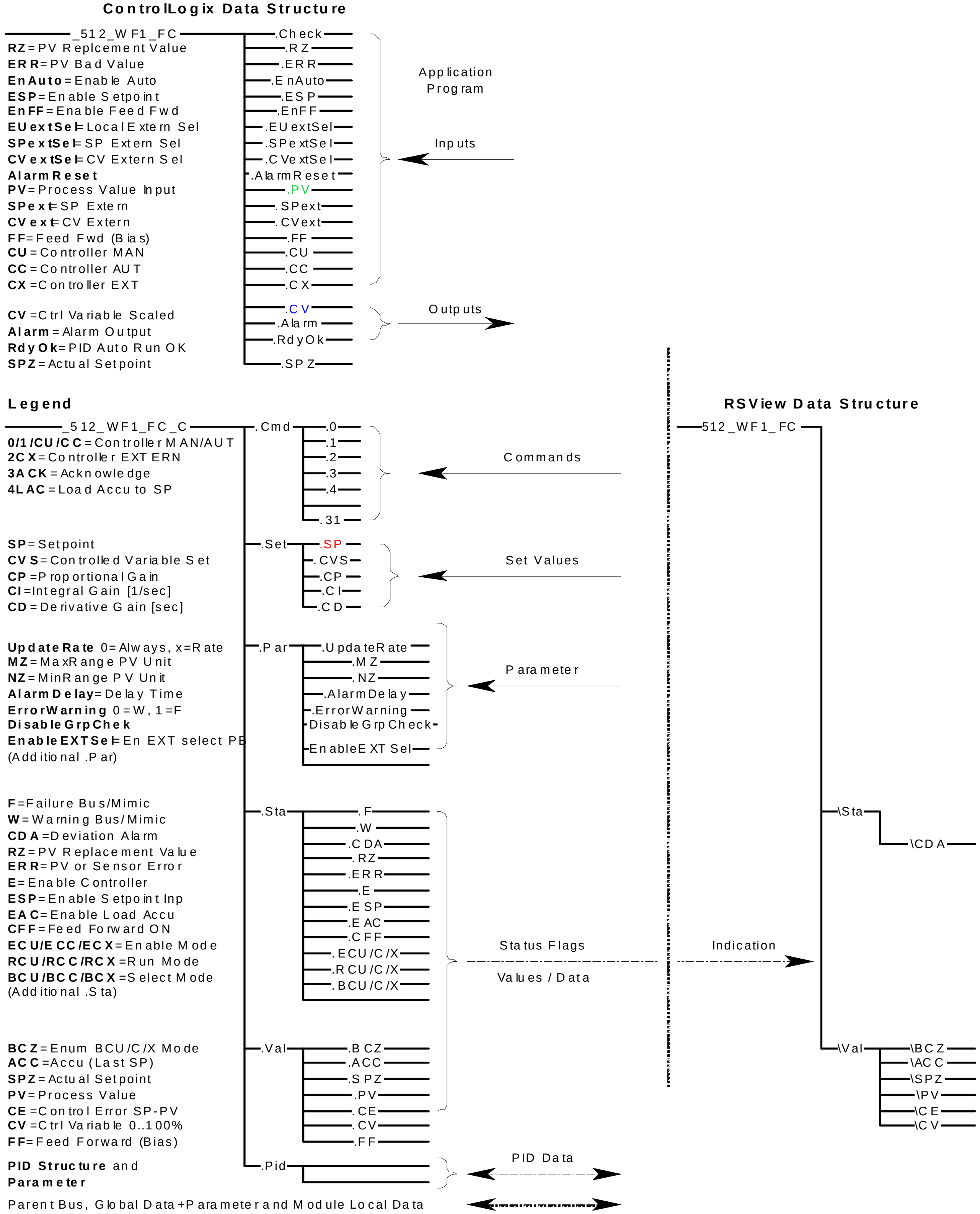


PidMod Ladder

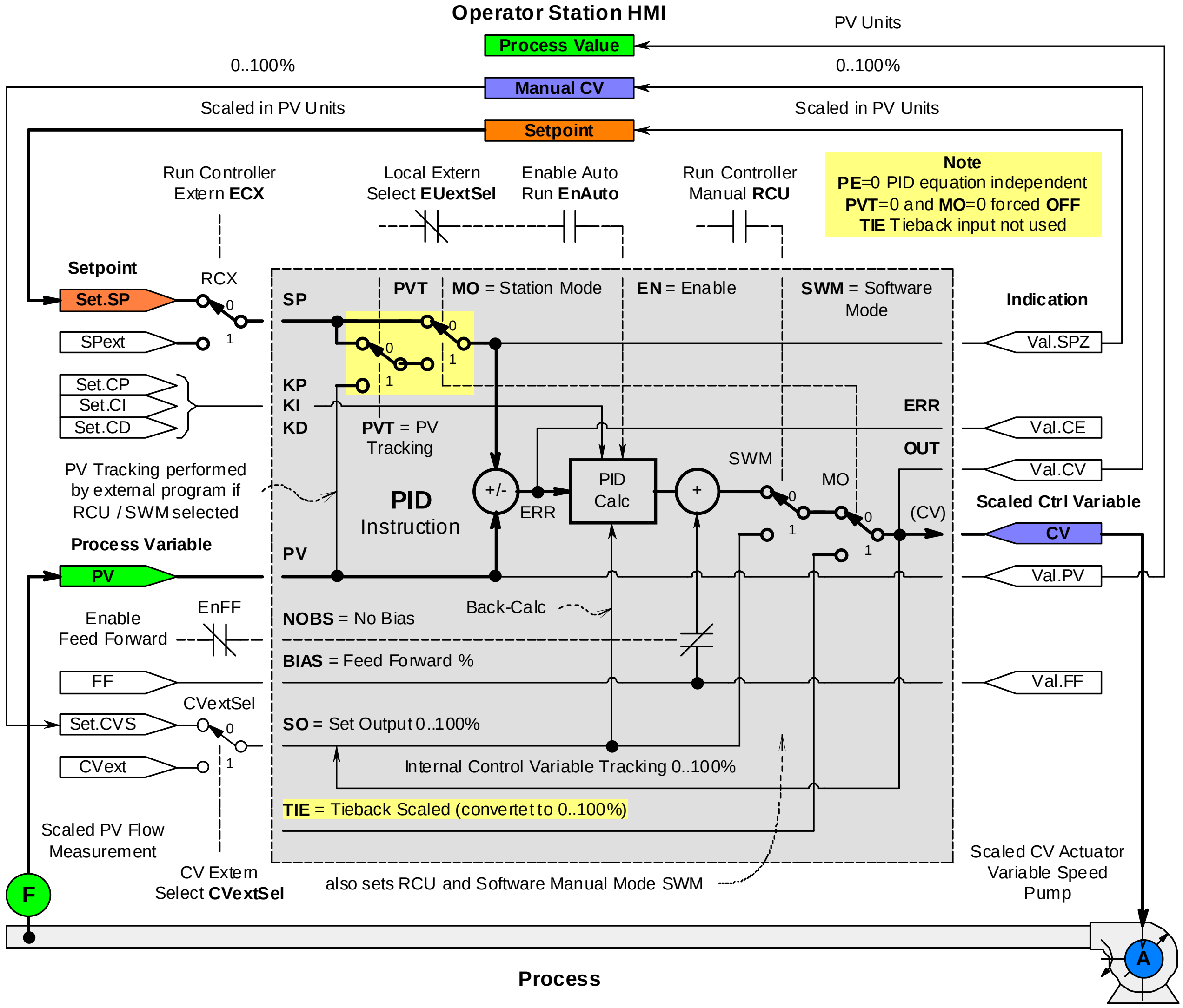


For module definitions and signal descriptions, see Appendix B, page 143.

PidMod Structure – Interface to FactoryTalk View

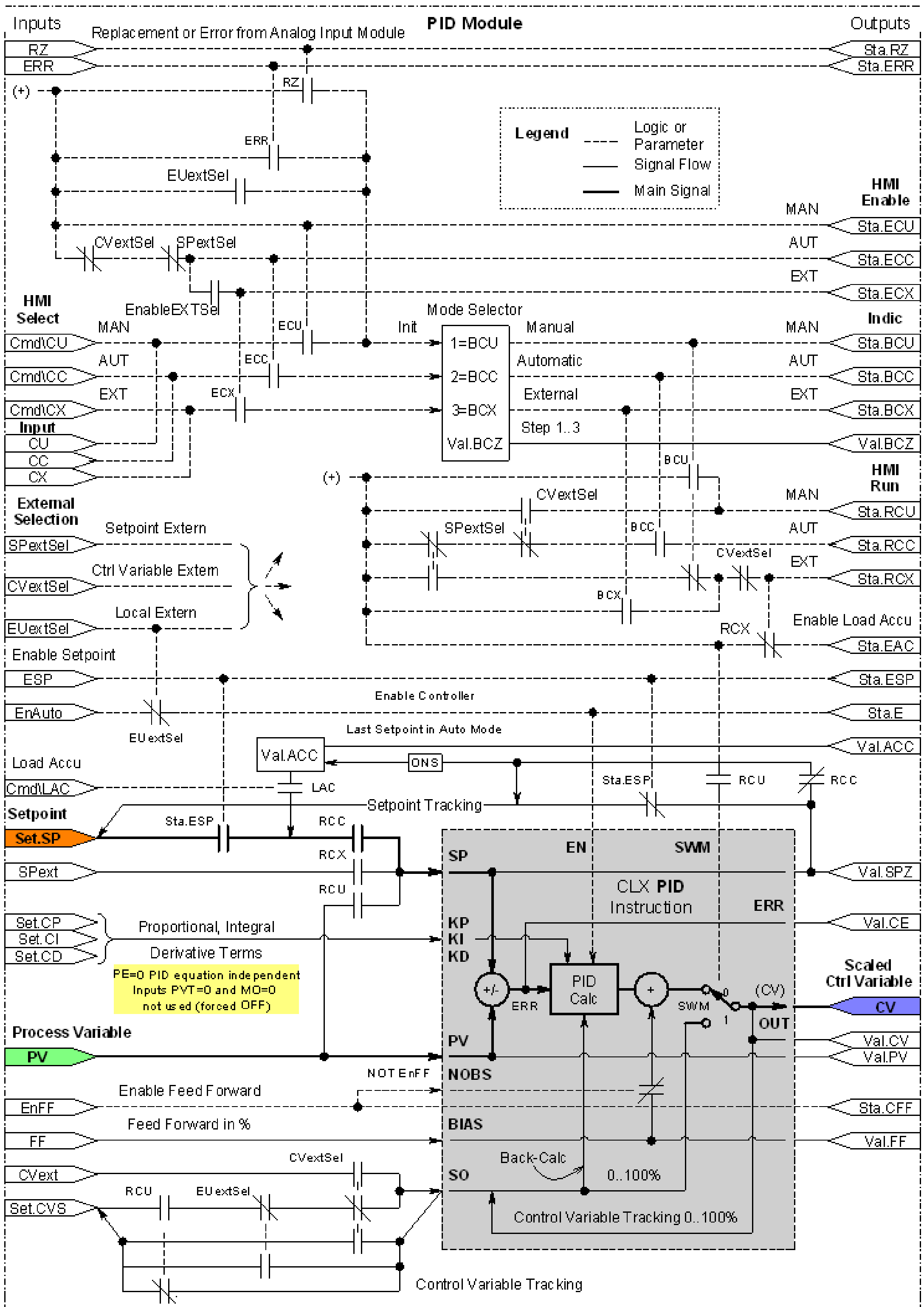


PidMod Principle Dataflow



PidMod Functional Diagram

Setpoint SP and controlled Output SO are fed back to the HMI input Set.SP and Set.CVS respectively, unless enabled by appropriate controller inputs (tracking).



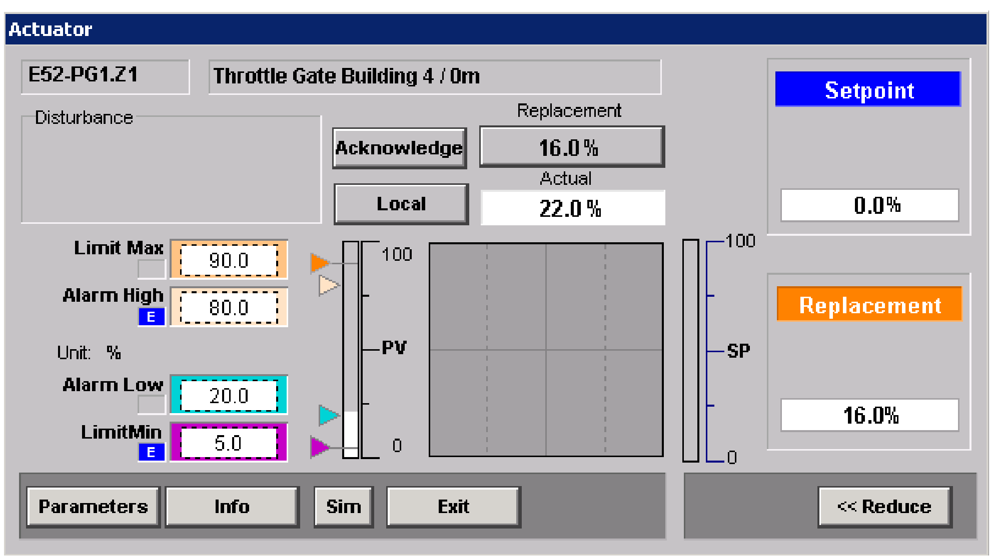
ActMod - Actuator Module

The Actuator Module (ActMod) is used to scale and set analog outputs and position analog starters. The module adjusts an actuator according to a setpoint that is entered by the operator or taken from a PID module; it is linked with a Machine or Control Group Module. Features of ActMod are:

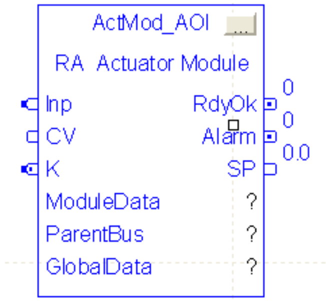
- Display template for FactoryTalk View SE
- Failure or Warning device
- High/low level alarm enable and supervision, max/min level control limits to stop actuator
- Replacement/bypass value with diagnostic indication (password protected)
- Auto bypass selection on error, actual value indication (remove bypass without alarm)
- Setpoint with automatic back-tracking if not enabled for the operator
- Remote setpoint to directly connect PID control variable
- Parameter ramp rate for local setpoint control
- Input scaling from global system group table for selected converter types
- Supports negative gradient scaling if max scale is set less than min scale (see AnaInp)
- Parameter sample rate and input filter max gradient = 1..100% of range per second
- Parameter alarm delay, alarm deadband and min/max clamping deadband
- Ready OK output for interlocking

Parent Bus Link to Control Group or Machine Group Module

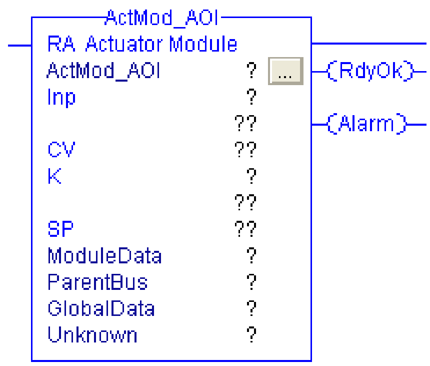
ActMod Operator Interface



ActMod Function Block

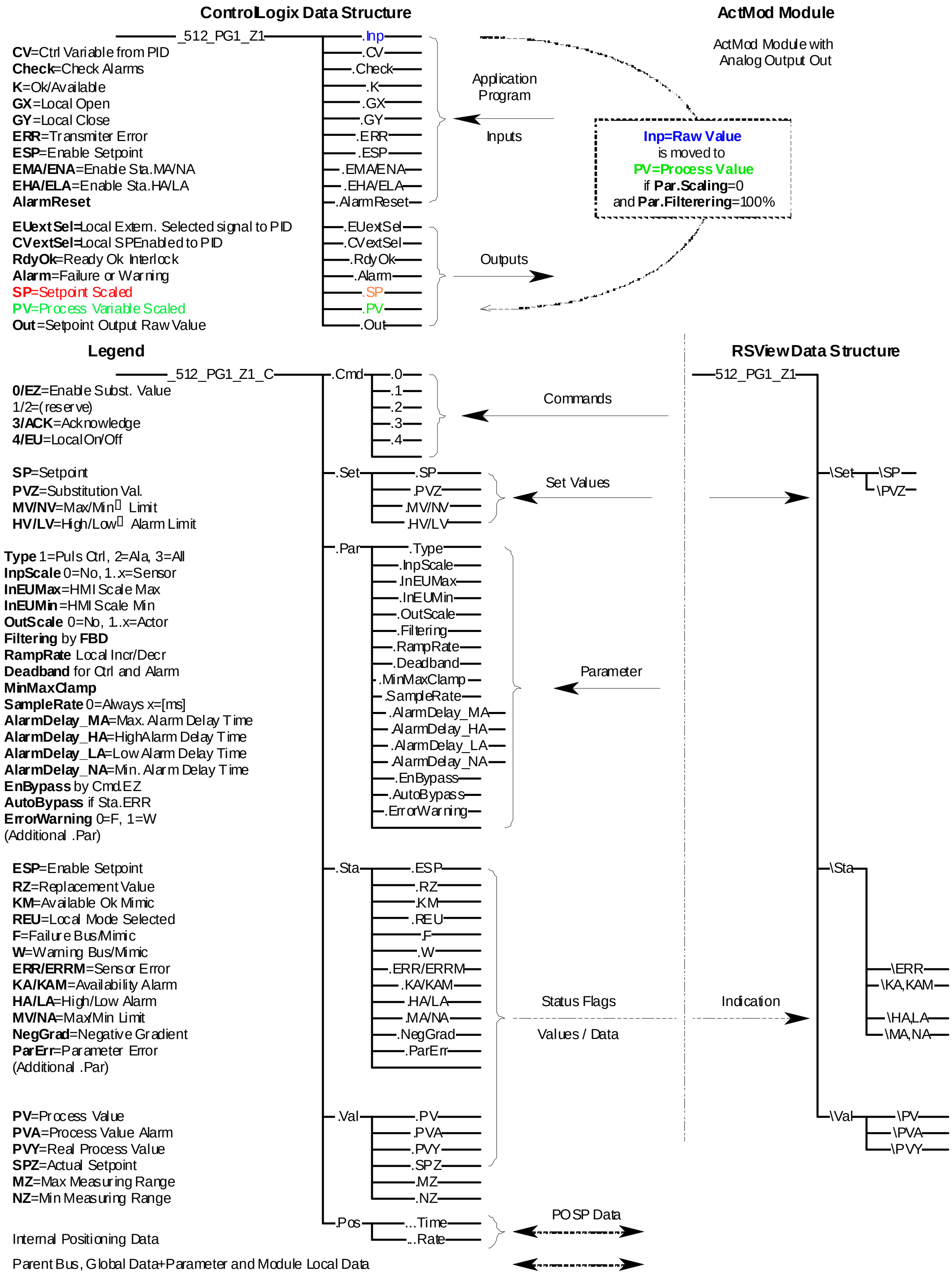


ActMod Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

ActMod Structure – Interface to FactoryTalk View



Process Variable and Setpoint Output Scaling

Raw input value scale and under/overflow values are taken from Global.Par.AnaInpScale[x] where x = 1..9 is specified by Par.InpScale. If Par.InpScale = 0 then the raw value is transmitted to the module's PV without scaling.

$$PV_{Scaled} = \frac{(In - InRawMin) * (Par.InEUMax - Par.InEUMin)}{(InRawMax - InRawMin)} + Par.InEUMin$$

Where the Raw Scale InRawMax/Min is specified by Par.InpScale = x and taken from Global.Par.AnaInpScale[x].InRawMax/Min
(for x = 1..9 Array Offset, see System Group Global Data)
the Scaled Range InEUMax/Min is specified by Par.InEUMax/Min

Note that InRawMax must be greater than InRawMin and InEUMax must not be equal InEUMin, else Parameter Error Warning Sta.ParErr is displayed and the MODULE IS NOT PROCESSED.

Positive Gradient → if InEUMax > InEUMin then Sta.NegGrad = OFF and application PV MaxRange Val.MZ = InEUMax,
MinRange Val.NZ = InEUMin

Negative Gradient → if InEUMax < InEUMin then Sta.NegGrad = ON and application PV MaxRange Val.MZ = InEUMin,
MinRange Val.NZ = InEUMax

Internal Filter PVY = PVX_{old}+Filtering*(PVX-PVX_{old})/100*SampleTime/s

Where PVX_{old} is PVX of previous scan, Filtering = 1..99 [%] (100% = Filter bypassed) and SampleTime/s = actually measured SampleRate/1000 [ms/ms] (Sample Timer ACC Value)

Raw output value scale is taken from Global.Par.AnaOutScale[x] where x = 1..9 is specified by Par.OutScale. If Par.OutScale = 0 then the module's SP is transmitted to the raw Out without scaling.

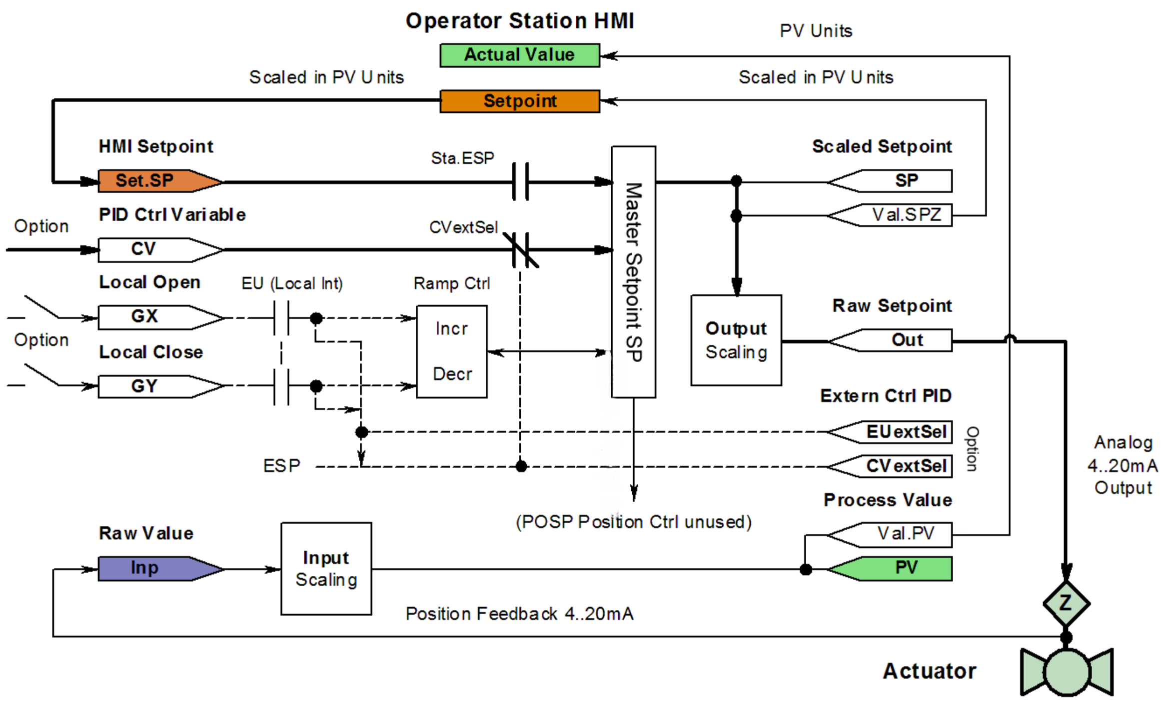
$$Out_{Raw} = \frac{(SP - Val.NZ) * (OutRawMax - OutRawMin)}{(Val.MZ - Val.NZ)} + OutRawMin$$

Where the Raw Scale OutRawMax/Min is specified by Par.OutScale = x and taken from Global.Par.AnaOutScale[x].OutRawMax/Min (for x = 1..9 Array Offset, see System Group Global Data) the Scaled Range Val.MZ/NZ is specified by Par.InEUMax/Min and depends on Positive or Negative Gradient

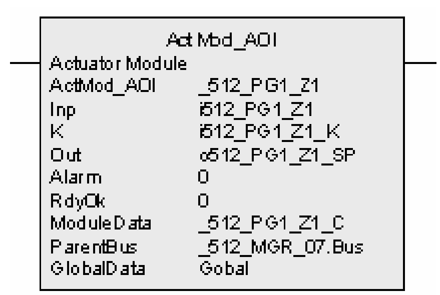
Note that OutRawMax must be greater than OutRawMin else, Parameter Error Warning Sta.ParErr is displayed and the module is not processed.

Actuator Module - Analog Output with Position Feedback

The ActMod uses its Analog Output Out with a raw range specified by Par.OutScale. The actual position feedback (option) may be transmitted to the Analog Input that is scaled by Par.InpScale. The following flow chart shows the ActMod principle dataflow for Analog Output:

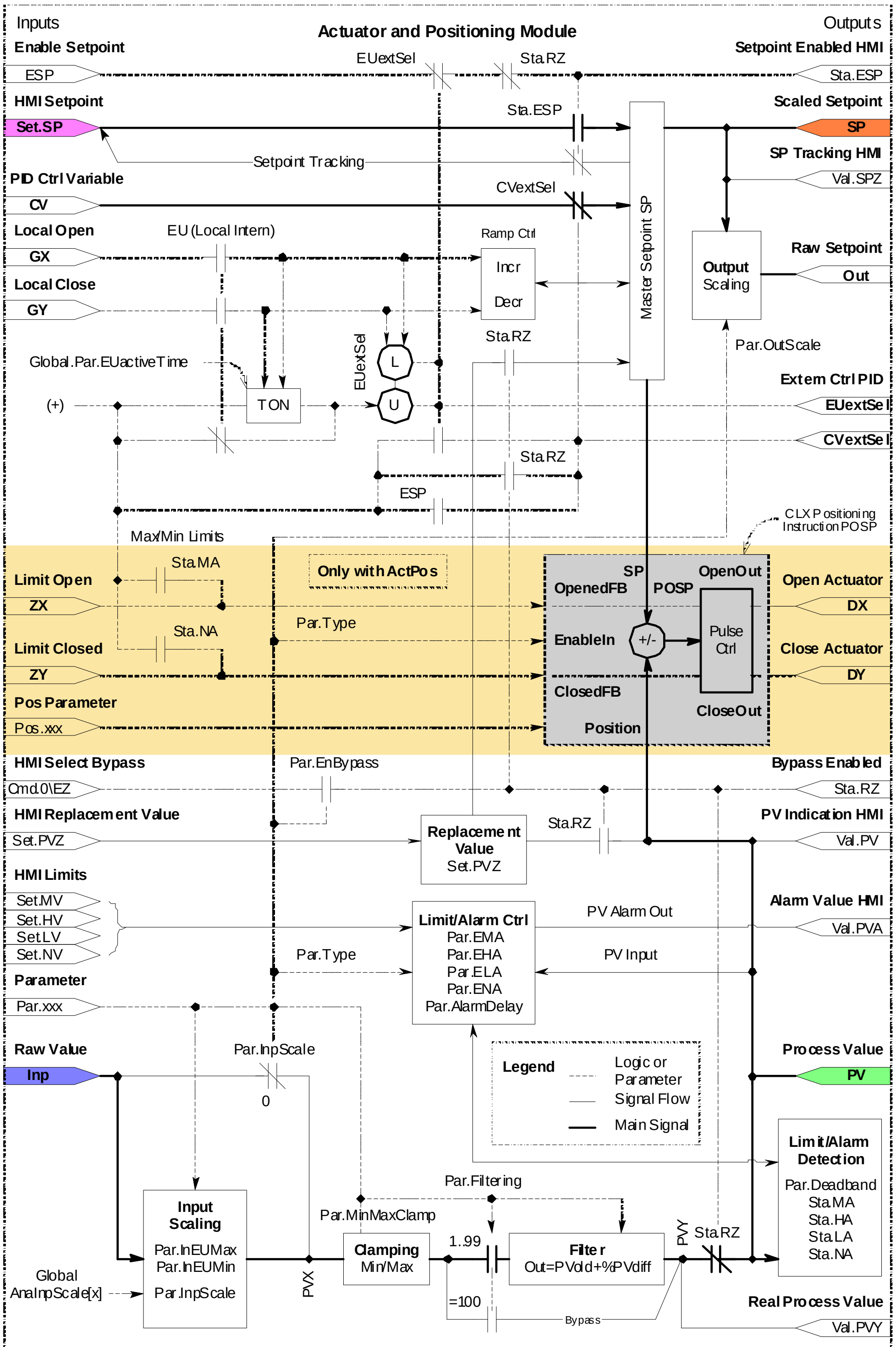


The application program for an Actuator with Analog Output may look as follows:



ActMod and ActPos Functional Diagram

The Setpoint SP is fed back (tracking) to the HMI input Set.SP unless enabled by Sta.ESP. Thus, if not enabled, the Setpoint SP can be overwritten by the user program.

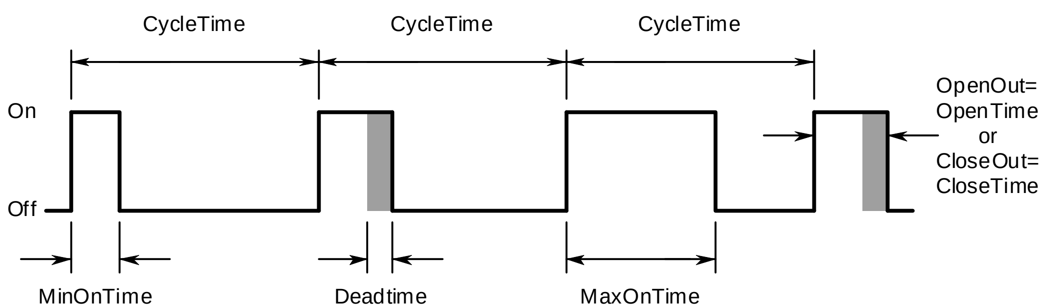


ActPos - Actuator Positioning

The Actuator Positioning Module (ActPos) is the subset to the ActMod that is used to position digital starters. The module uses the RSLogix 5000 standard POSP function block in order to adjust an actuator according to the setpoint of ActMod; it is linked only with ActMod. The following are features of ActPos.

- Directly connects to ActMod
- Uses RSLogix 5000 standard Positioning Pulse function block, POSP, with pulse outputs for damper MotorD
- Parameter CycleTime min/max ontime and deadtime for pulse shaping
- Hardware limit switch inputs for open/close interlocking

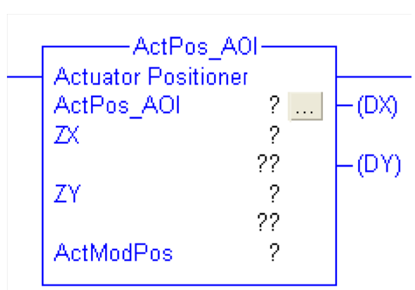
ActPos Pulse Shaping



ActPos Function Block

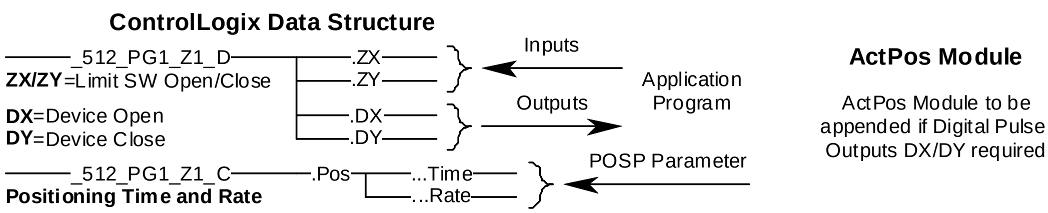
Name convention:
ActPos_AOI <TagName of *ActMod*>**_POS**
ActModPos <TagName of *ActMod*>**_C.POS**

ActPos Ladder



For module definitions and signal descriptions, see Appendix B, page 143.

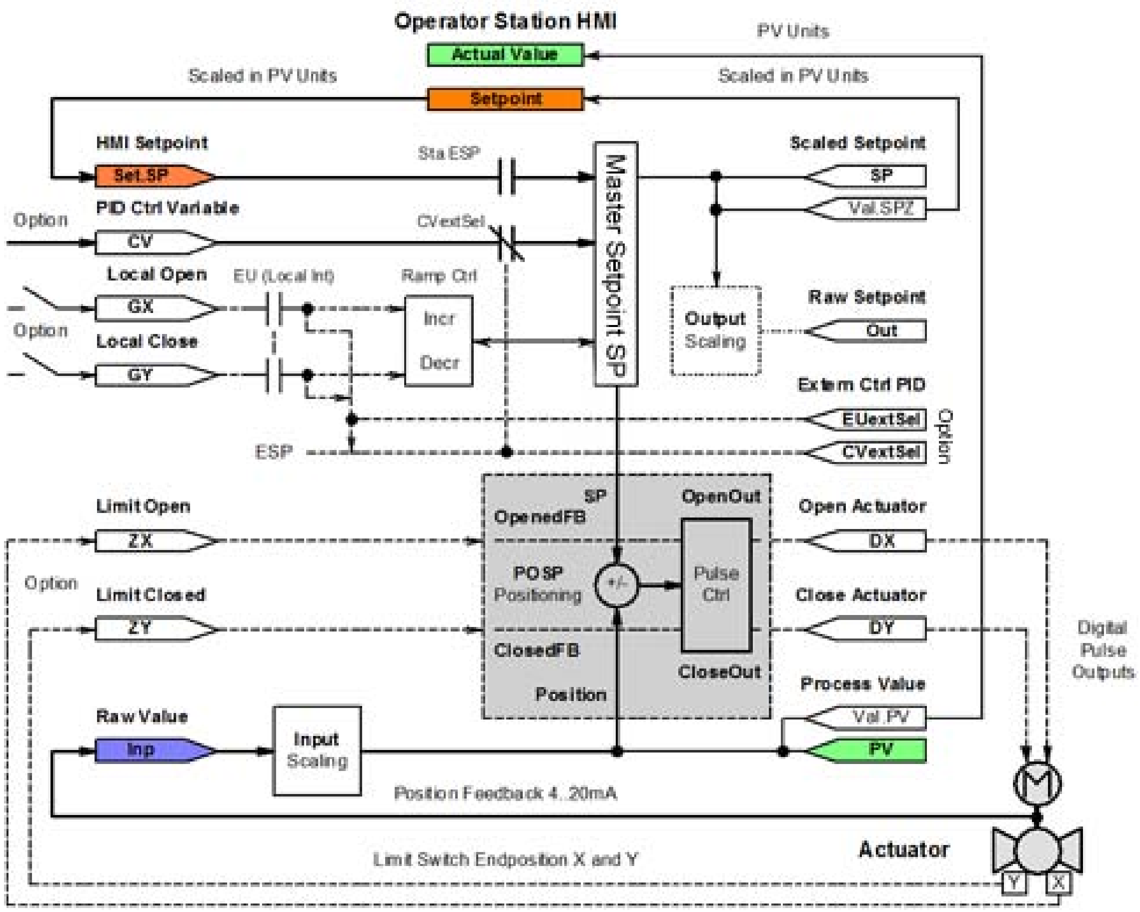
ActMod and ActPos Structure – Interface to FactoryTalk View



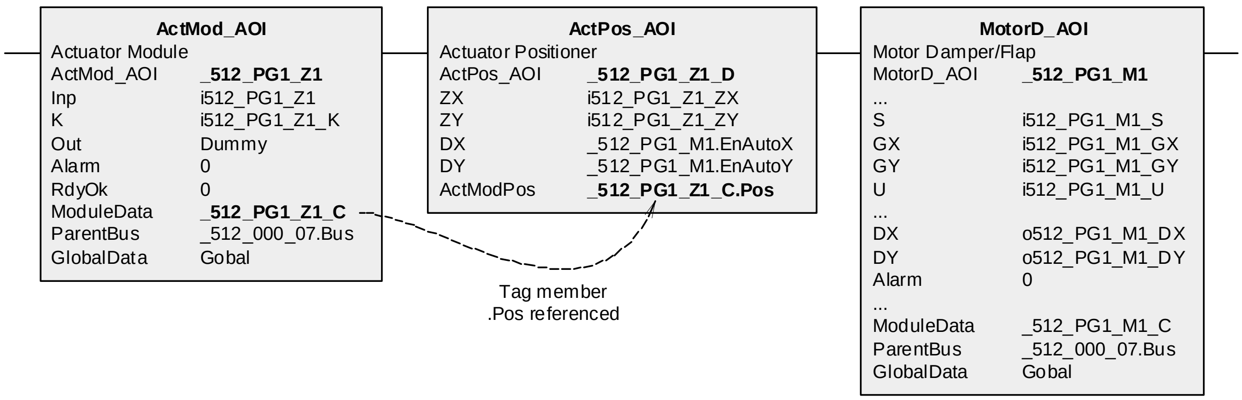
For the Functional Diagram refer to Actuator Module ActMod.

Actuator Positioner - Digital Pulse Outputs with Position Feedback

In this case an Actuator Positioner with Digital Pulse Outputs DX and DY is required in conjunction with the ActMod. The actual position feedback is transmitted to the Analog Input In and scaled by Par.InpScale in order to be compared with the Setpoint SP. The following flow chart shows the principle dataflow for ActMod and ActPos with Digital Pulse Outputs:



The application program for an Actuator with Digital Pulse Outputs may look like the following program.



Note: the ActMod is followed by an ActPos that controls the Pulse Outputs DX and DY. The ActMod Analog Out is not used and may be specified as a Dummy tag. The ActPos Outputs DX and DY may be directly transmitted to a positioning device or as shown, to a Damper MotorD Module unless additional interlocking is required.

Actuator Positioner - How the POSP Instruction uses the Internal Cycle Timer

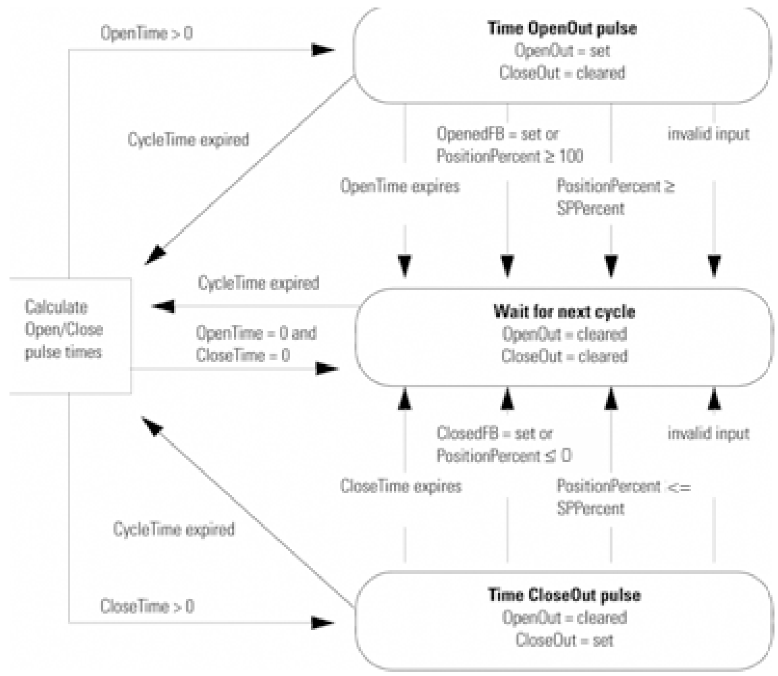
The POSP instruction uses the CycleTime parameter to determine how often to recalculate the duration of Open and Close output pulses. An internal timer is maintained and updated by the DeltaT parameter. DeltaT is the elapsed time since the instruction last executed. Whenever the internal timer equals or exceeds the programmed CycleTime (cycle time expires) the Open and Close outputs are recalculated.

You can change the CycleTime at any time.

If CycleTime = 0, the internal timer, OpenOut, and CloseOut are cleared.

Producing output pulses

The following diagram shows the three primary states of the POSP instruction.



Calculating open and close pulse times

OpenOut is pulsed whenever SP > Position feedback. When this occurs, the instruction sets CloseTime = 0 and the duration for which OpenOut is to be turned on is calculated as:

OpenTime = (SPPercent-PositionPercent)/OpenRate

- If OpenTime_{n-1} < CycleTime, then add Deadtime to OpenTime.
- If OpenTime > MaxOnTime, then limit to MaxOnTime.
- If OpenTime < MinOnTime, then set OpenTime = 0.

If any of the following conditions exist, OpenOut is not pulsed and OpenTime = 0.

- OpenFB is set or PositionPercent ≥ 100
- CycleTime = 0
- OpenRate = 0
- SPPercent is invalid

The CloseOut is pulsed whenever SP < Position feedback. When this occurs, the instruction sets OpenTime = 0 and the duration that CloseOut is turned on is calculated as:

CloseTime = (PositionPercent-SPPercent)/CloseRate

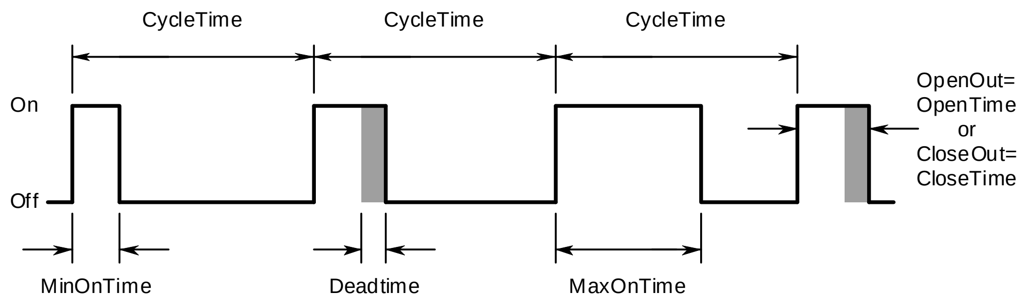
- If CloseTime_{n-1} < CycleTime, then add Deadtime to CloseTime.
- If CloseTime > MaxOnTime, then limit to MaxOnTime.
- If CloseTime < MinOnTime, then set CloseTime to 0.

If any of the following conditions exist, CloseOut will not be pulsed and CloseTime will be cleared.

- ClosedFB is set or PositionPercent ≤ 0
- CycleTime = 0
- CloseRate = 0
- SPPercent is invalid

OpenOut and CloseOut will not be pulsed if SPPercent equals PositionPercent. Both OpenTime and CloseTime will be cleared.

Legend


CycleTime
CycleTime
CycleTime
On
Off
OpenOut=
OpenTime
or
CloseOut=
CloseTime
MinOnTime
Deadtime
MaxOnTime

Notes:

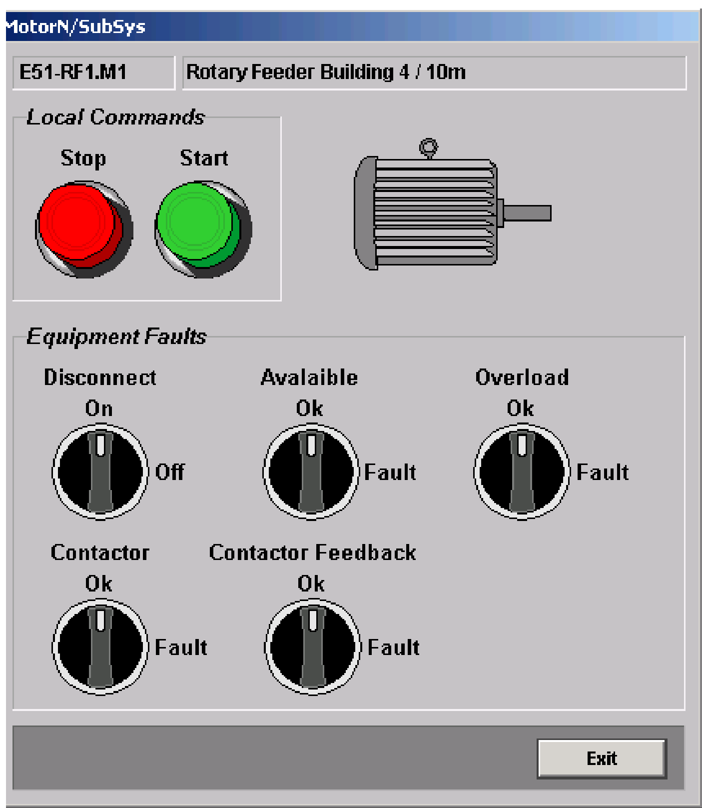
Simulation Modules

MotorN Sim - Motor Normal Drive Simulation

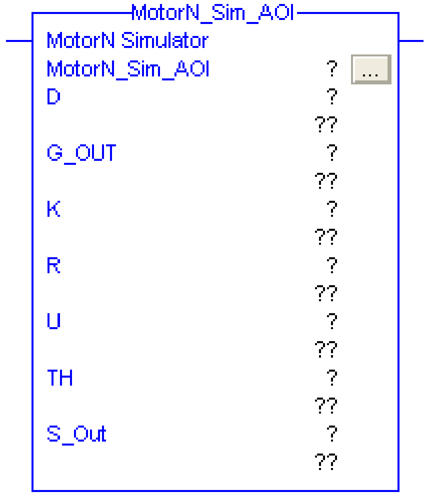
MotorN simulation block generates inputs for Motor Normal AOI block, it can replace real motor for code testing. Motor run feedback is generated when Contactor input D is set to 1. Motor can be also controlled from HMI faceplate by Start/Stop buttons which simulates Local buttons. All available failures for MotorN block can be simulated from HMI faceplate by using switches.

- The following are features of MotorN simulation:
- Display template for FactoryTalk View SE
- Contactor feedback simulation
- Local buttons simulation
- Failure generation for all available MotorN failures

Motor Normal Simulation Interface



Motor Normal Drive Simulation Ladder Block



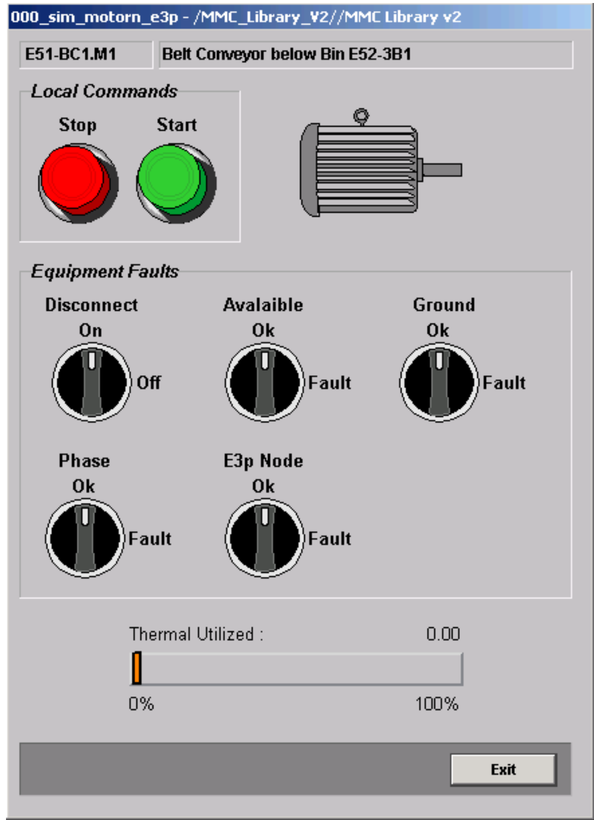
MotorNE3p SIM – Motor Normal Drive with E3p simulation

MotorNE3p simulation block generates inputs for Motor Normal Drive with E3p AOI block, it can replace real motor with E3p for code testing. Motor run feedback is generated when Contactor input D is set to 1. Motor can be also controlled from HMI faceplate by Start/Stop buttons which simulates Local buttons. E3p failures can be simulated from HMI faceplate by using switches. Thermal utilization can be set by slider. Thermal overload warning and failure is generated automatically if utilization is set to 85-99% for warning and 100% for failure.

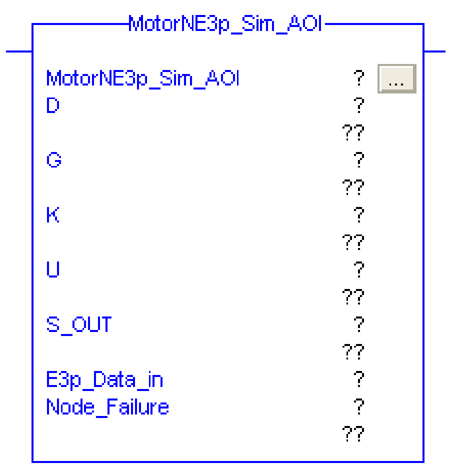
The following are features of MotorNE3p simulation:

- Display template for FactoryTalk View SE
- Node failure simulation
- Local buttons simulation
- Thermal utilization simulation
- MotorNE3p failures generation

Motor Normal Simulation with E3p InterfaceMotor Normal Drive



Simulation with E3p Simulation Ladder Block



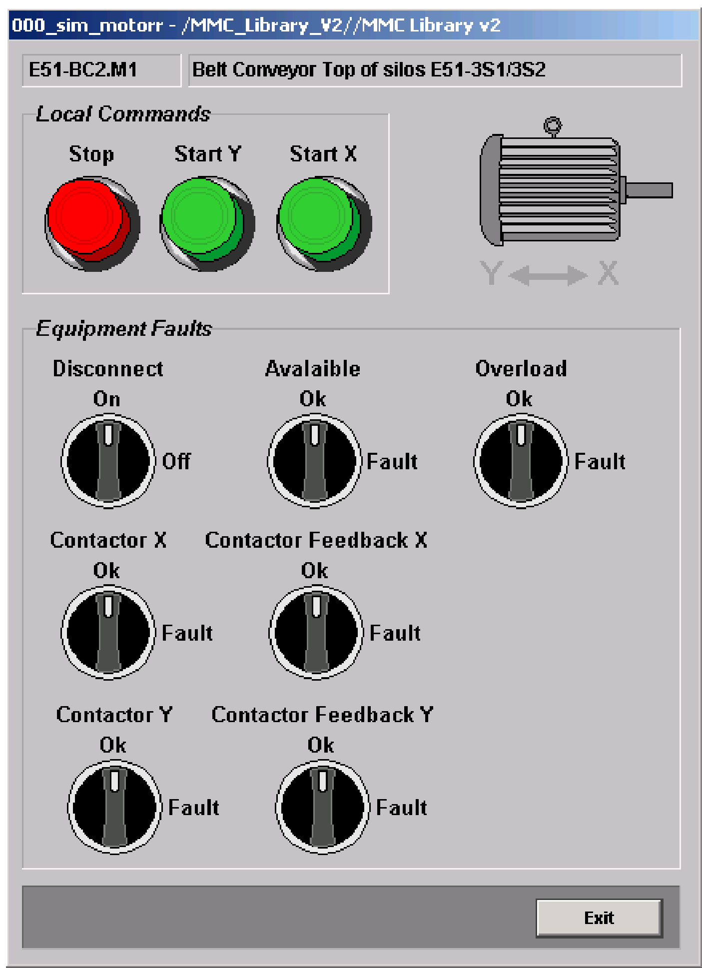
MotorR SIM – Motor Forward/Reverse Drive simulation

MotorR simulation block generates inputs for Motor Forward/Reverse Drive AOI block, it can replace real motor for code testing. Motor run feedback in X/Y direction is generated when Contactor input DX/DY is set to 1. Motor can be also controlled from HMI faceplate by StartX/StartY/Stop buttons which simulates Local buttons. All available failures for MotorR block can be simulated from HMI faceplate by using switches.

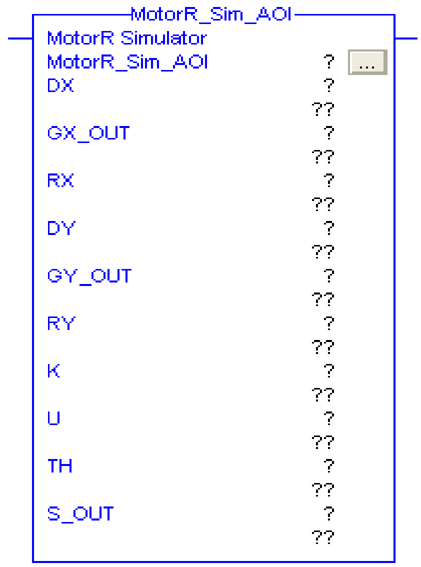
The following are features of MotorR simulation:

- Display template for FactoryTalk View SE
- Contactor feedback simulation
- Local buttons simulation
- Failure generation for all available MotorR failures

Motor Forward / Reverse Drive Simulation Interface



Motor Forward/Reverse Drive Simulation Ladder Block



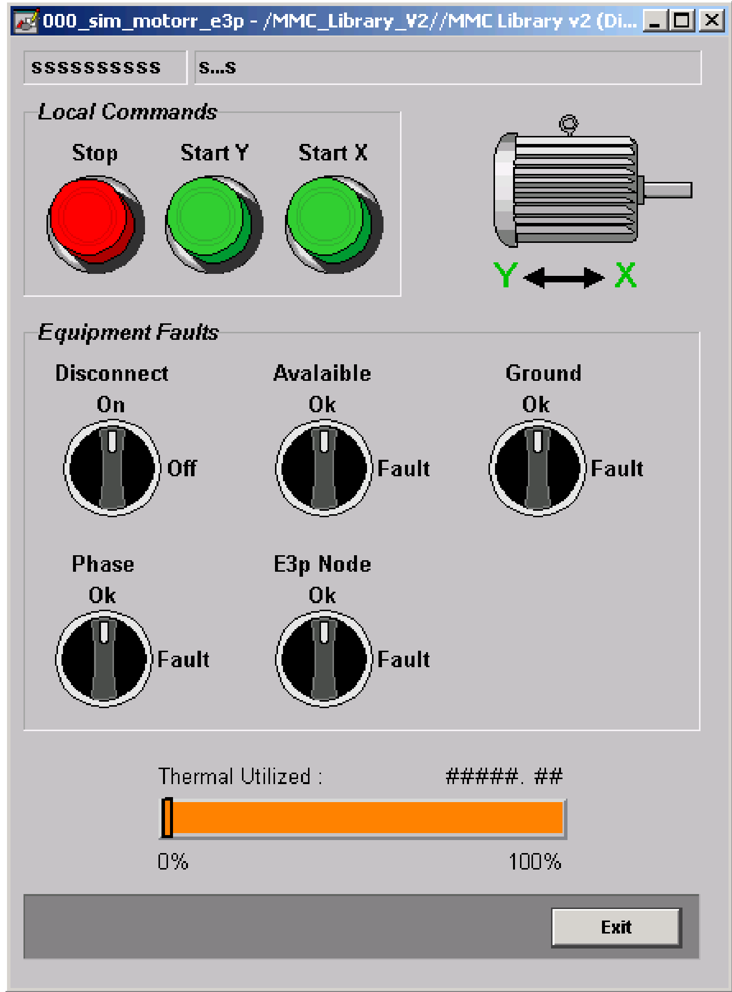
MotorRE3p SIM – Motor Forward/Reverse Drive with E3p simulation

MotorRE3p simulation block generates inputs for Motor Forward/Reverse Drive with E3p AOI block, it can replace real motor with E3p module for code testing. Motor run feedback in X/Y direction is generated when Contactor input DX/DY is set to 1. Motor can be also controlled from HMI faceplate by StartX/StartY/Stop buttons which simulates Local buttons. All available failures for MotorRE3p block can be simulated from HMI faceplate by using switches. Thermal utilization can be set by slider. Thermal overload warning and failure is generated automatically if utilization is set to 85-99% for warning and 100% for failure.

The following are features of MotorRE3p simulation:

- Display template for FactoryTalk View SE
- Node failure simulation
- Local buttons simulation
- Thermal utilization simulationMotorNE3p failures generation
- MotorRE3p failures generation

Motor Forward / Reverse Drive with E3p Simulation



Motor Forward/Reverse Drive with E3p Simulation Ladder Block

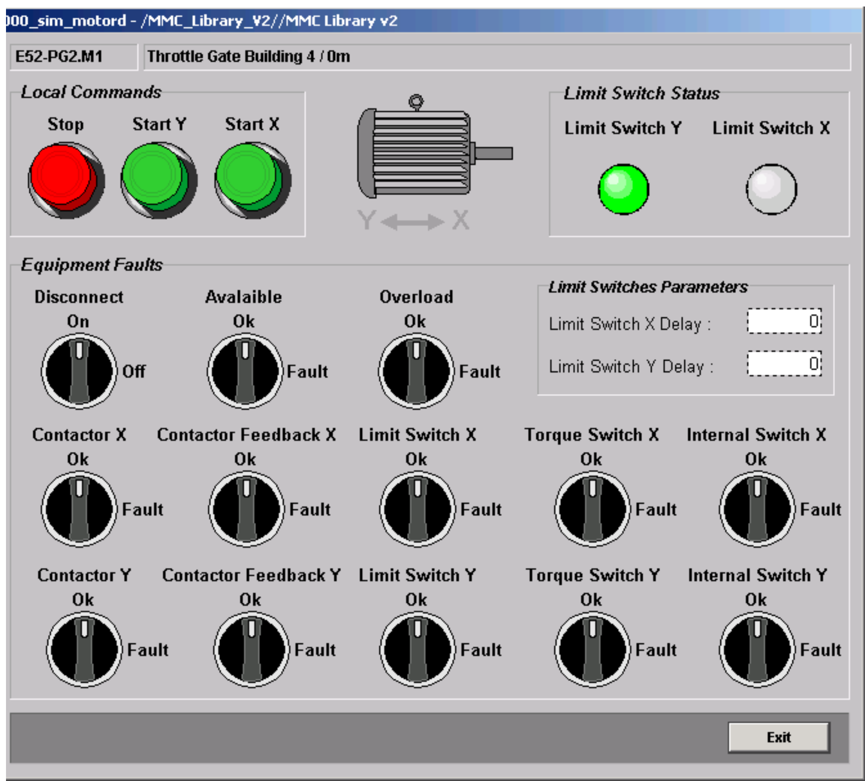
MotorD SIM – Motor Damper/Flap Drive simulation

MotorD simulation block generates inputs for Motor Damper/Flap Drive AOI block, it can replace real motor for code testing. Motor run feedback in X/Y direction is generated when Contactor input DX/DY is set to 1. Limit switch delay times can be set from HMI simulation faceplate. Motor can be also controlled from HMI faceplate by StartX/StartY/Stop buttons which simulates Local buttons. All available failures for MotorD block can be simulated from HMI faceplate by using switches.

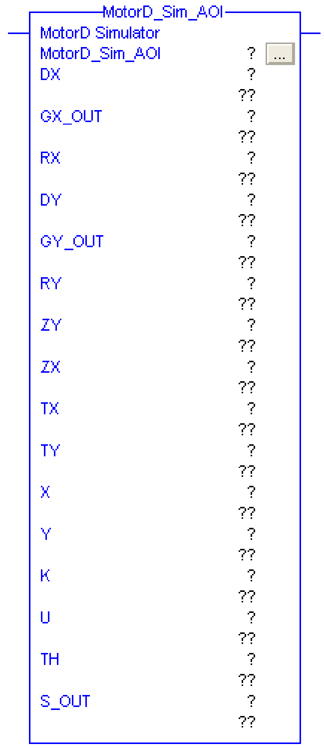
The following are features of MotorD simulation:

- Display template for FactoryTalk View SE
- Contactor feedback simulation
- Local buttons simulation
- MotorD failures generation

Motor Damper / Flap Drive Interface



Motor Damper / Flap Drive Simulation Ladder Block



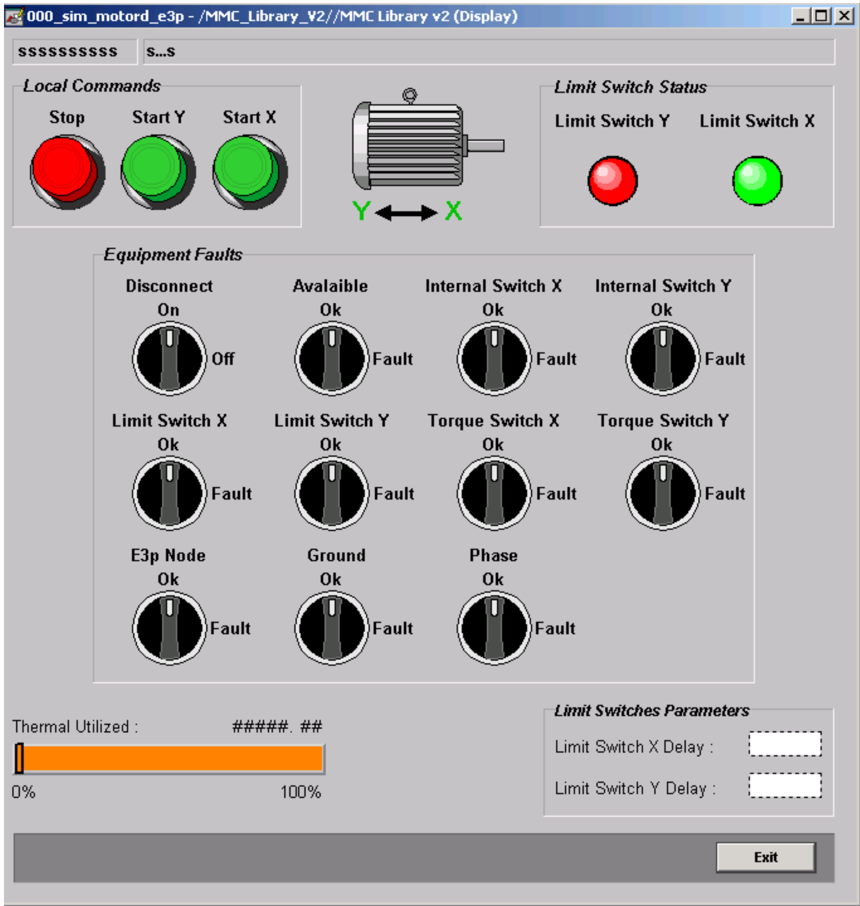
MotorDE3p SIM – Motor Damper/Flap Drive with E3p simulation

MotorDE3p simulation block generates inputs for Motor Damper/Flap Drive with E3p AOI block, it can replace real motor with E3p module for code testing. Motor run feedback in X/Y direction is generated when Contactor input DX/DY is set to 1. Limit switch delay times can be set from HMI simulation faceplate. Motor can be also controlled from HMI faceplate by Start/Stop buttons which simulates Local buttons. E3p failures can be simulated from HMI faceplate by using switches. Thermal utilization can be set by slider. Thermal overload warning and failure is generated automatically if utilization is set to 85-99% for warning and 100% for failure.

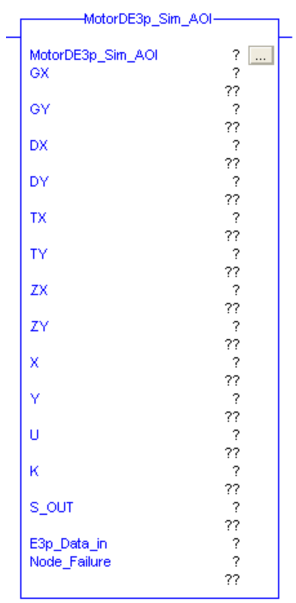
The following are features of MotorDE3p simulation:

- Display template for FactoryTalk View SE
- Node failure simulation
- Local buttons simulation
- MotorDE3p failures generation
- Thermal utilization simulation

Motor Damper/Flap Drive with E3p simulation interface



Motor Damper/Flap Drive with E3p Simulation Ladder Block



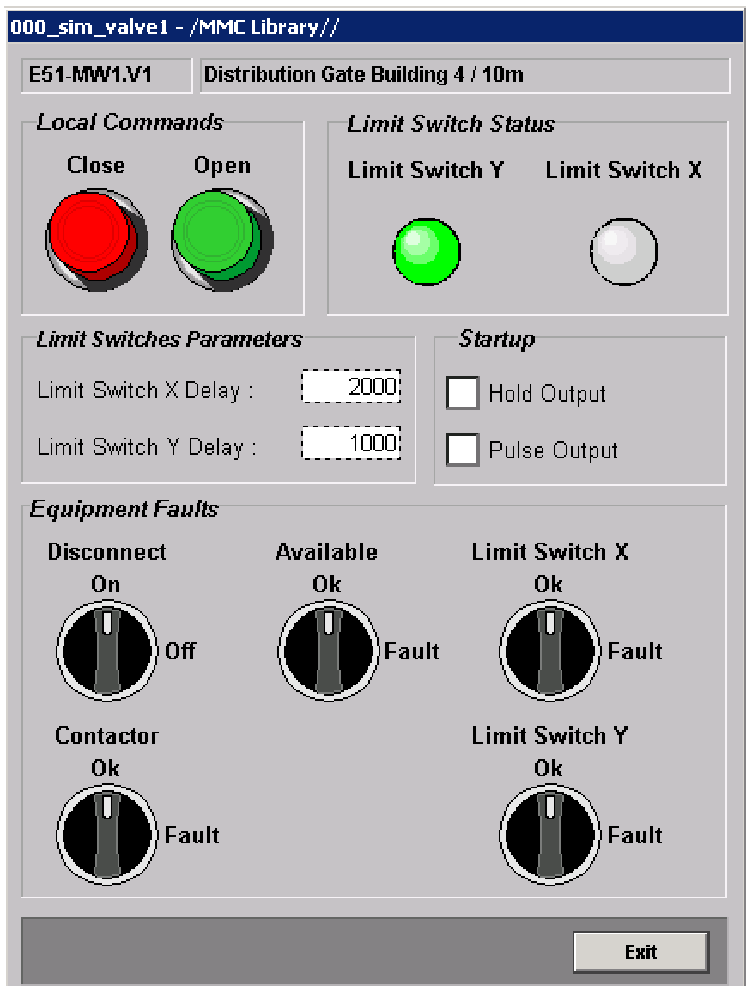
Valve1 SIM – Valve with one coil simulation

Valve one coil simulation block is used to generate input for Valve1 AOI, so no real valve is needed for testing code functionality. Valve Limit Switch X/Y feedback is generated when Contactor input D is set to 0/1 and switch time is reached. Valve can be also controlled from HMI faceplate by Open/Close buttons which simulates Local buttons. All available failures for Valve1 block can be simulated from HMI faceplate by using switches.

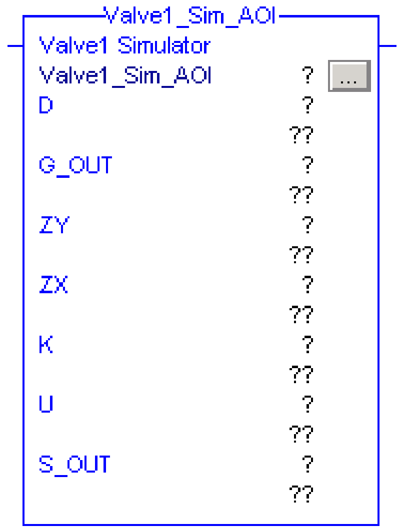
The following are features of Valve1 simulation:

- Display template for FactoryTalk View SE
- Local control buttons simulation
- Valve 1 module failure generation

Valve 1 Module simulation interface



Valve 1 Module simulation Ladder block



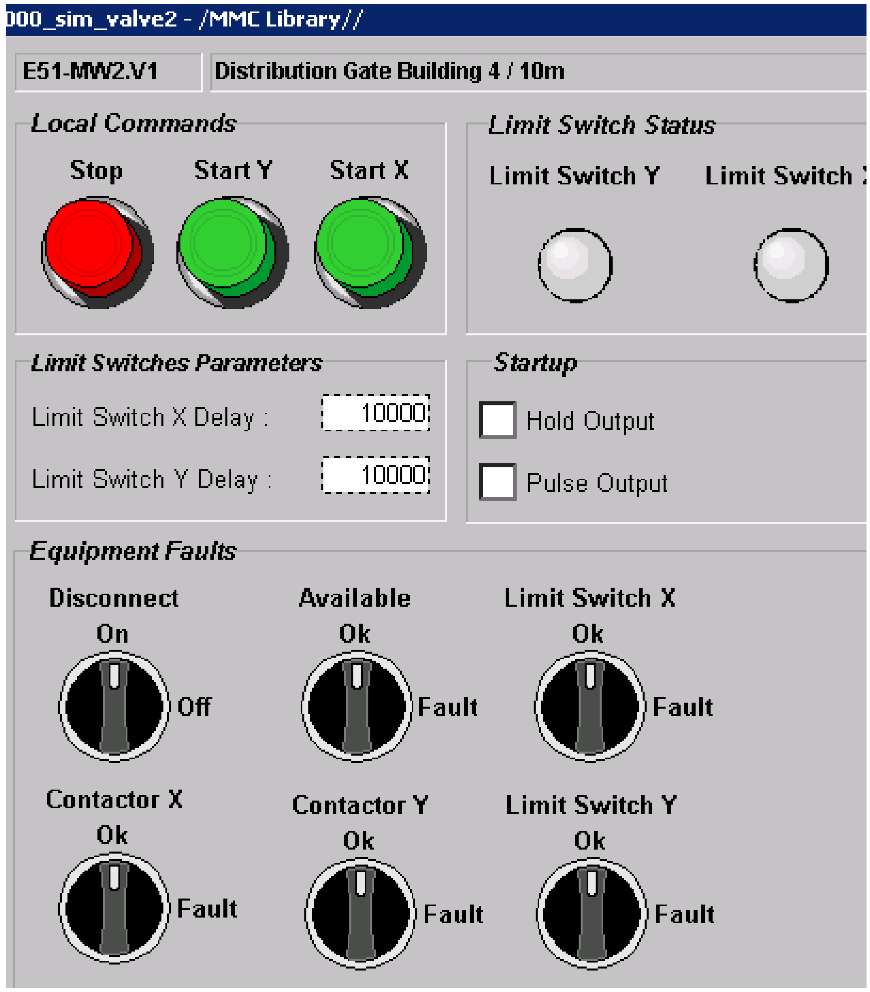
Valve2 SIM – Valve with two coils simulation

Valve with two coil simulation block is used to generate input for Valve2 AOI, so no real valve is needed for testing code functionality. Valve Limit Switch X/Y feedback is generated when Contactor input DX/DY is set to 0/1 and switch time is reached. Valve can be also controlled from HMI faceplate by StartX/StartY/Close buttons which simulates Local buttons. All available failures for Valve2 block can be simulated from HMI faceplate by using switches.

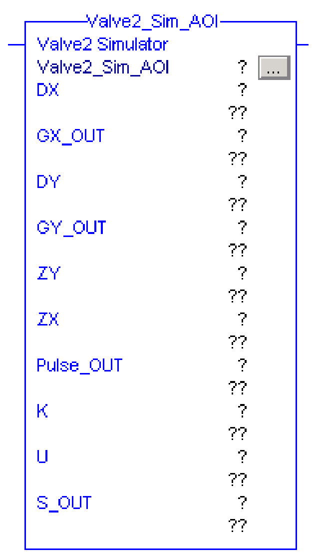
The following are features of Valve2 simulation:

- Display template for FactoryTalk View SE
- Local control buttons simulation
- Valve 2 module failure generation

Valve 2 Module simulation interface



Valve 2 Module simulation Ladder block



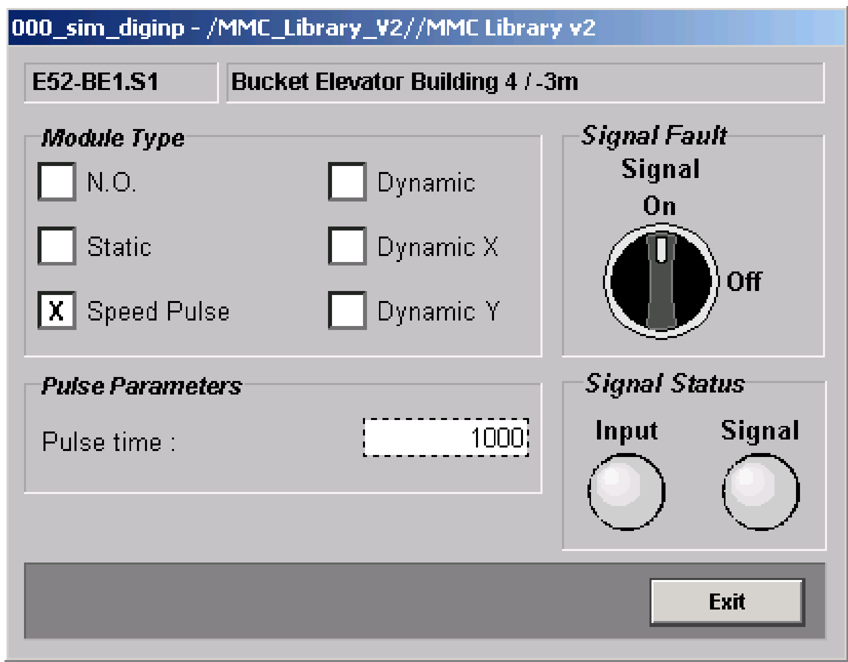
DigInp SIM – Digital Input simulation

Digital Input simulation block is used to generate input signal for Digital Input AOI, so no real digital input is needed for testing code functionality. Output of simulation block depend on type of device, it can be pulse, NO, NC device. Start condition of device depend on input bit called Activate, when this bit is set to logical 1 correct output is generated.

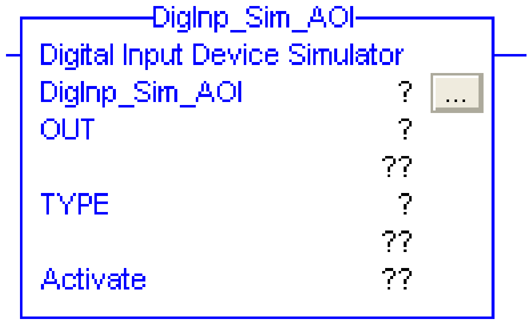
The following are features of DigInp simulation:

- Display template for FactoryTalk View SE
- Speed pulse, NC, NO simulation and other types supported by DigInp AOI
- Speed pulse frequency setting
- Digital Input Failure generation

Digital Input simulation interface



Digital Input simulation Ladder block

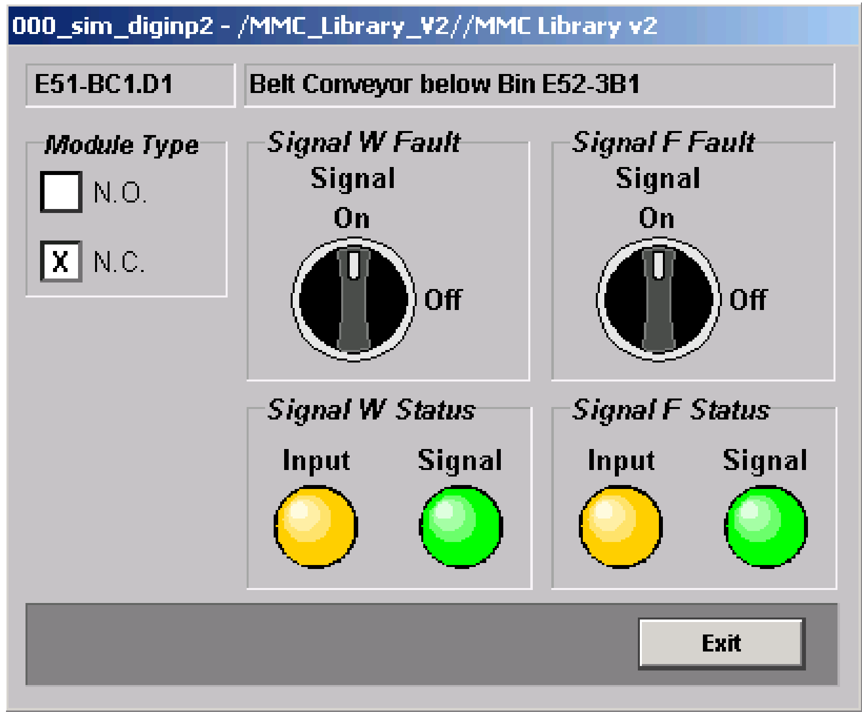


DigInp2 SIM – Digital Input 2 simulation

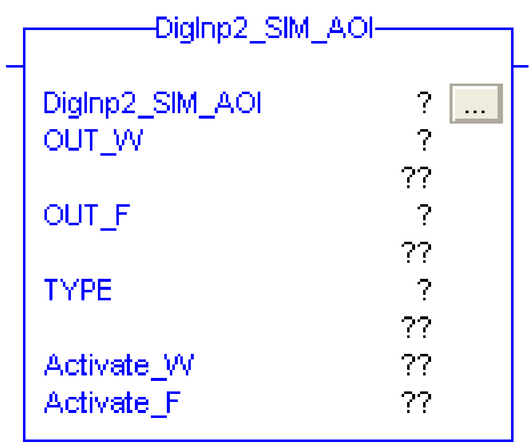
Digital Input 2 simulation block is used to generate input signal for Digital Input 2 AOI, so no real digital input is needed for testing code functionality. Digital Input 2 simulation block can generate Warning or Failure depending on Activate_W and Activate_F conditions.

- Display template for FactoryTalk View SE
- Warning or Failure output from simulation block
- Digital Input 2 Failure generation

Digital Input 2 simulation interface



Digital Input 2 simulation Ladder block



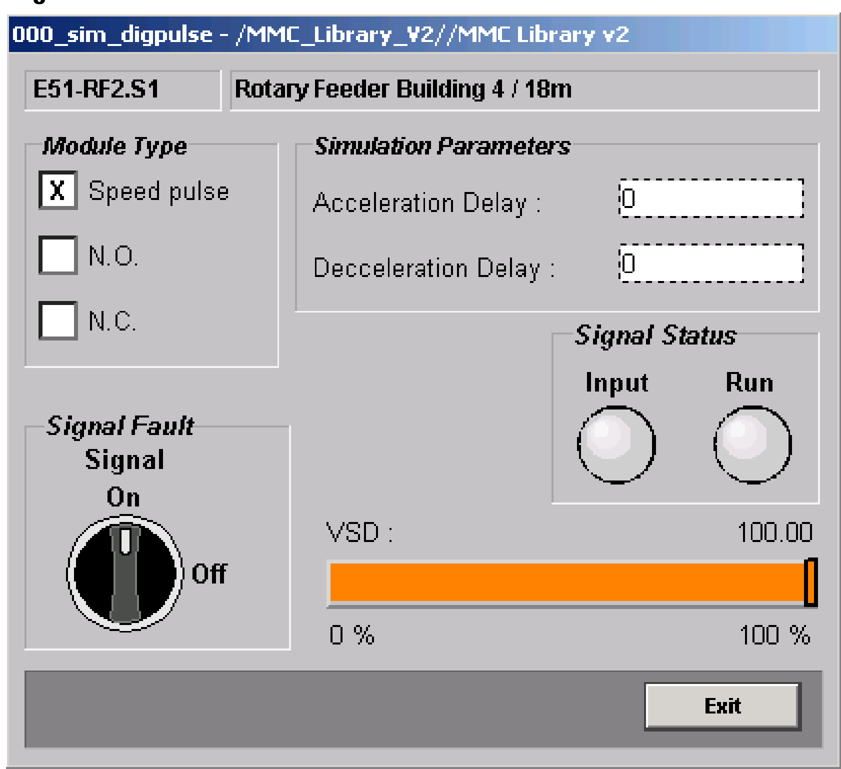
DigPulse SIM – Digital Pulse simulation

Digital Pulse simulation block is used to generate input signal for Digital Pulse AOI, so no real digital input is needed for testing of code functionality. Output is generated depending on Digital Pulse AOI type. Simulation block generate slow start up pulses with acceleration to normal speed and slow deceleration. That simulates startup/shutdown of motors with big inertia.
Startup/shutdown times can be set from HMI faceplate. Acceleration is started when input bit Start is set to logical 1. Deceleration starts when Start bit goes from 1 to 0. It's possible to set VSD speed from HMI faceplate. Input failure can be generated from HMI faceplate as well.

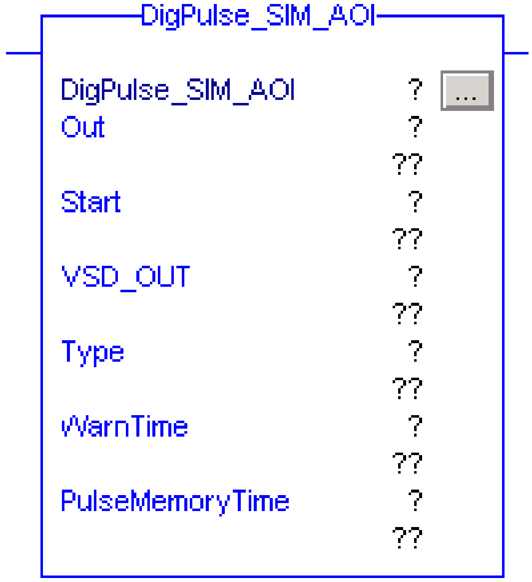
The following are features of DigPulse simulation:

- Display template for FactoryTalk View SE
- Speed pulse, NO and NC type simulation
- Slow acceleration and deceleration simulation
- VSD speed setting
- Digital Pulse failure generation

Digital Pulse simulation interface



Digital Pulse simulation Ladder block



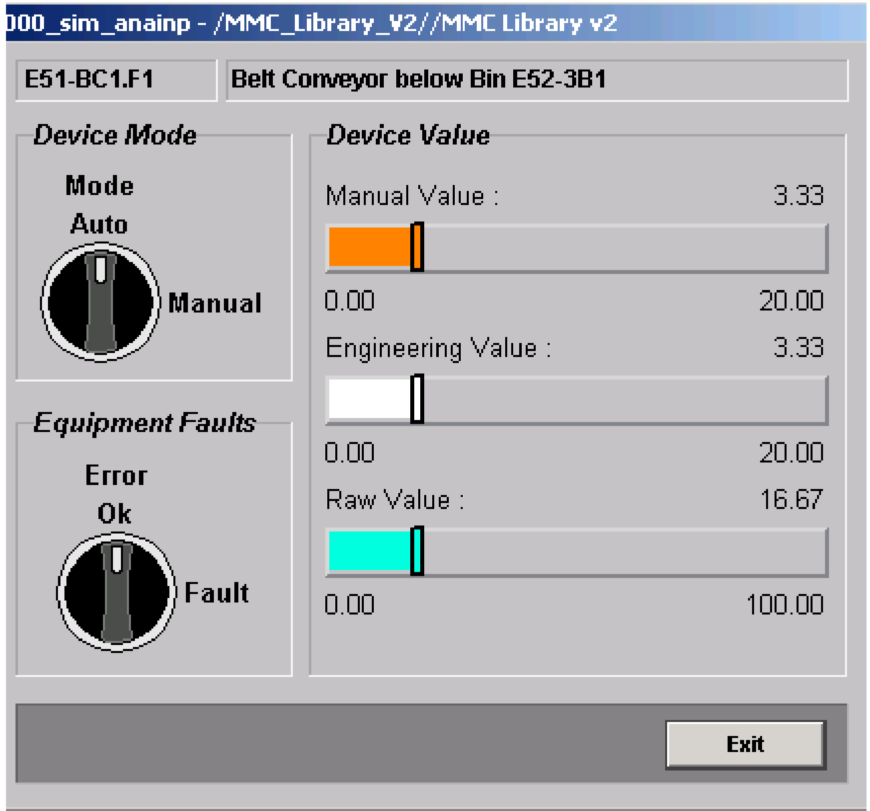
AnaInp SIM – Analog Input simulation

Analog input simulation block is used to generate input for AnaInp and AnaInpC add on instructions. Simulation output can come from two sources. First one is in Manual mode set from HMI faceplate using slider. Second source is from AnaInp simulation block input called Auto_Value. In both cases simulation output is scaled to raw input so it can be directly used as input for AnaInp or AnaInpC add on instructions.

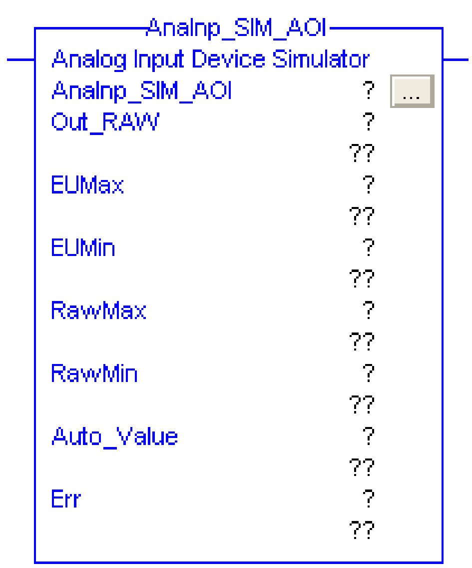
The following are features of AnaInp simulation:

- Display template for FactoryTalk View SE
- Automatic/Manual mode
- Output value comes from HMI faceplate or Auto_Value input
- Analog Input failure generation

Analog Input simulation interface



Analog Input simulation Ladder block



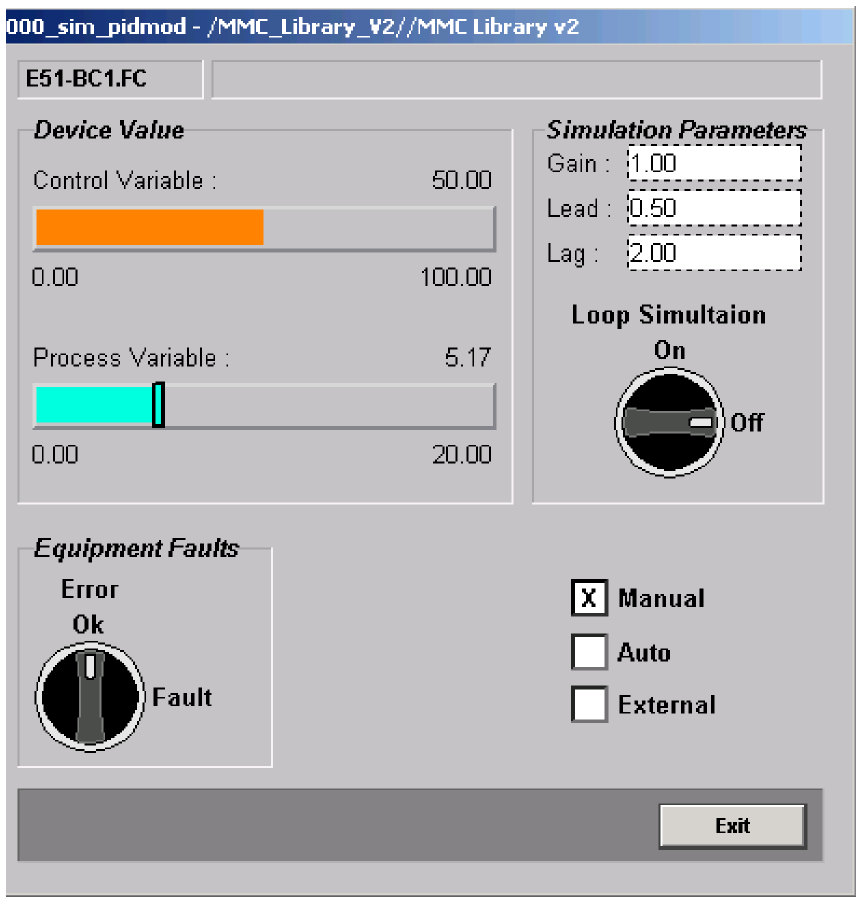
PIDMod SIM – PID Module simulation

PID Module simulation is used to simulate feedback from process with currently set PID parameters. Parameters of simulation process can be set from HMI faceplate. Process Variable can be calculated automatically by simulation block when Loop Simulation switch is set to “On” or operator can set it manually by using slider when Loop Simulation switch is set to “Off”.

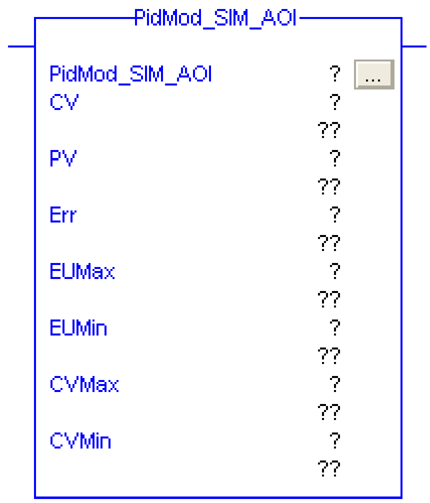
The following are features of PIDMod simulation:

- Display template for FactoryTalk View SE
- Process Variable simulation
- Possibility to set Process Variable manually
- PIDMod failure generation

PID Module simulation interface



PID Module simulation Ladder block



Process variable should be connected to Analog input simulation block Auto_value input so if AnaInp simulation block is set to Automatic PID block can control this value.

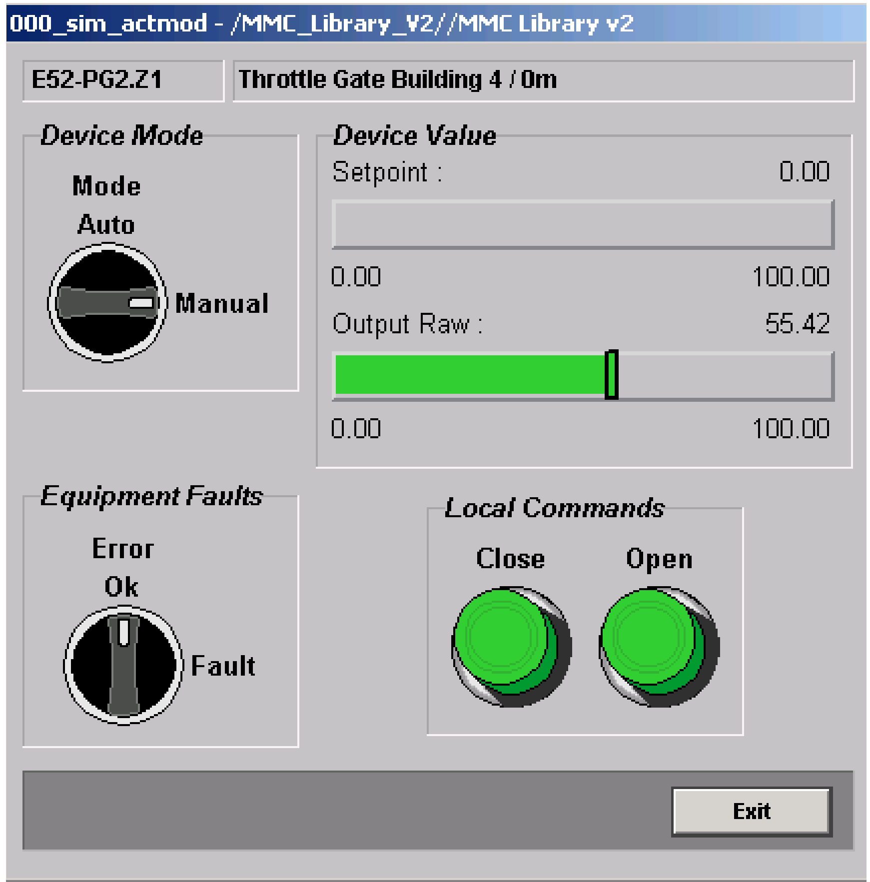
ActMod SIM – Actuator Module simulation

The Actuator Module simulation is used to generate inputs for Actuator Module without using real Actuator. Input for ActMod simulation is Set point from Actuator Module. In automatic mode Set point is scaled to Raw units and can be used as input for Actuator Module. Simulation output in manual mode is set from HMI faceplate. In Local mode Set point can be set by buttons on simulation faceplate.

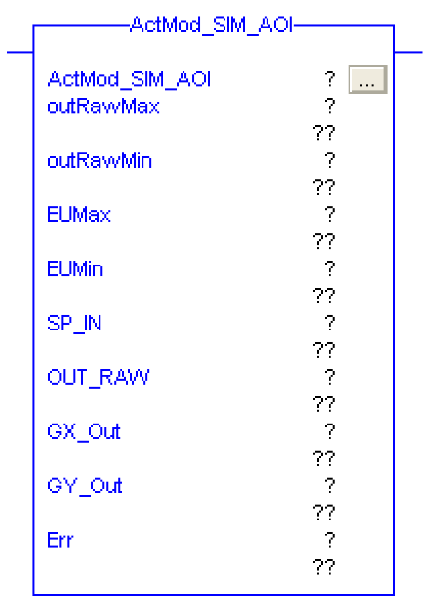
The following are features of ActMod simulation:

- Display template for FactoryTalk View SE
- Automatic/Manual mode
- Output value is computed from Actuator Module Set point or taken from HMI faceplate
- Local buttons to change Set point
- Actuator Module failure generation

Actuator Module simulation interface



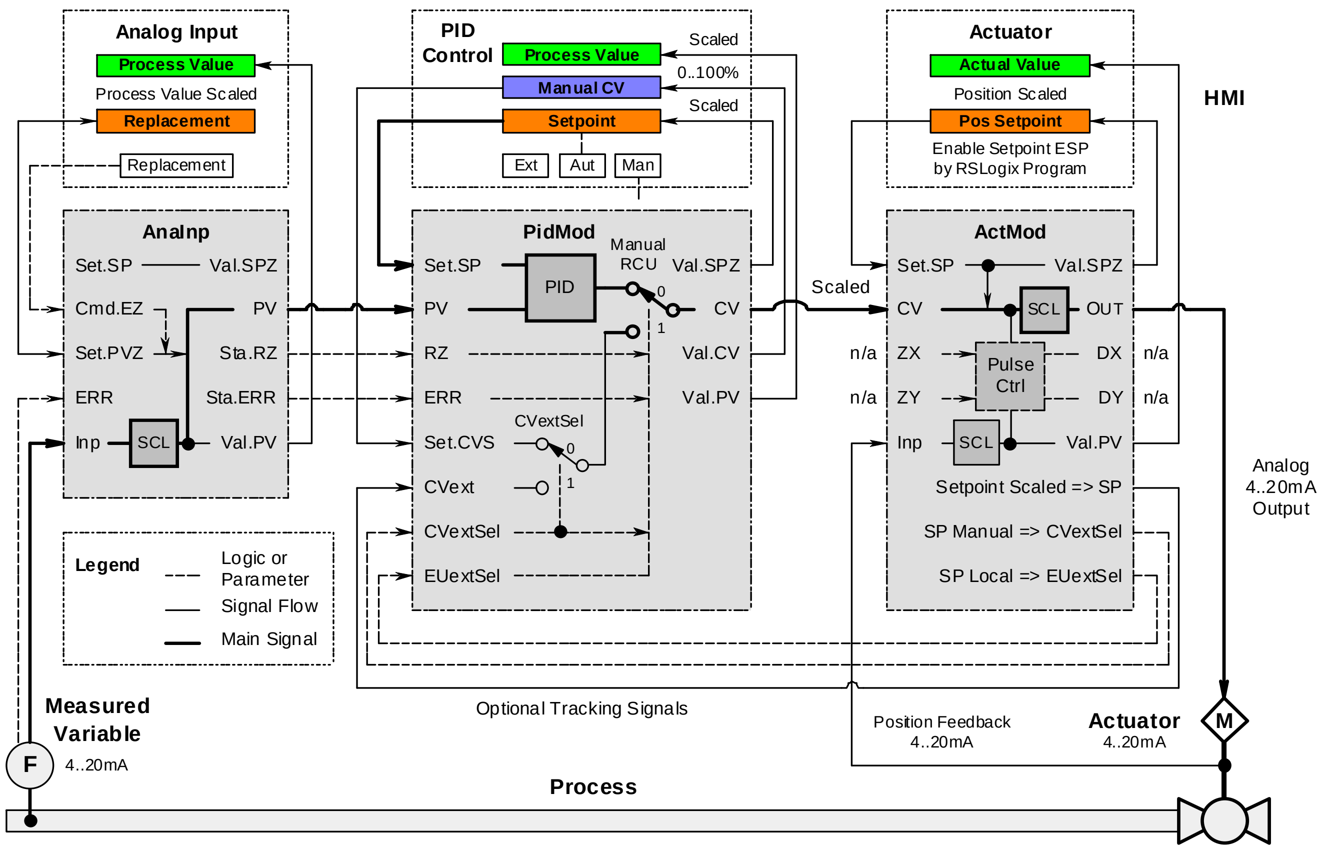
Actuator Module simulation Ladder block



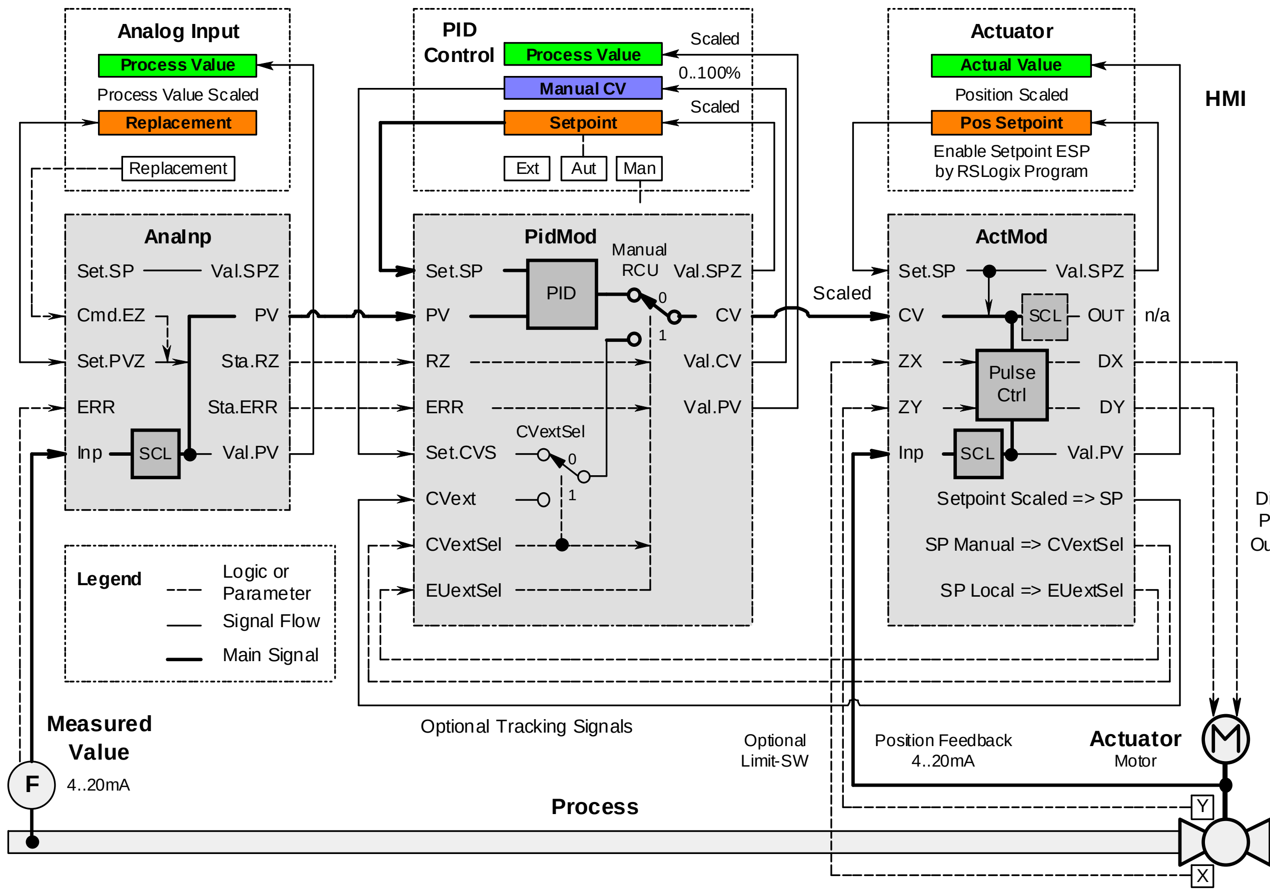
Notes:

Examples

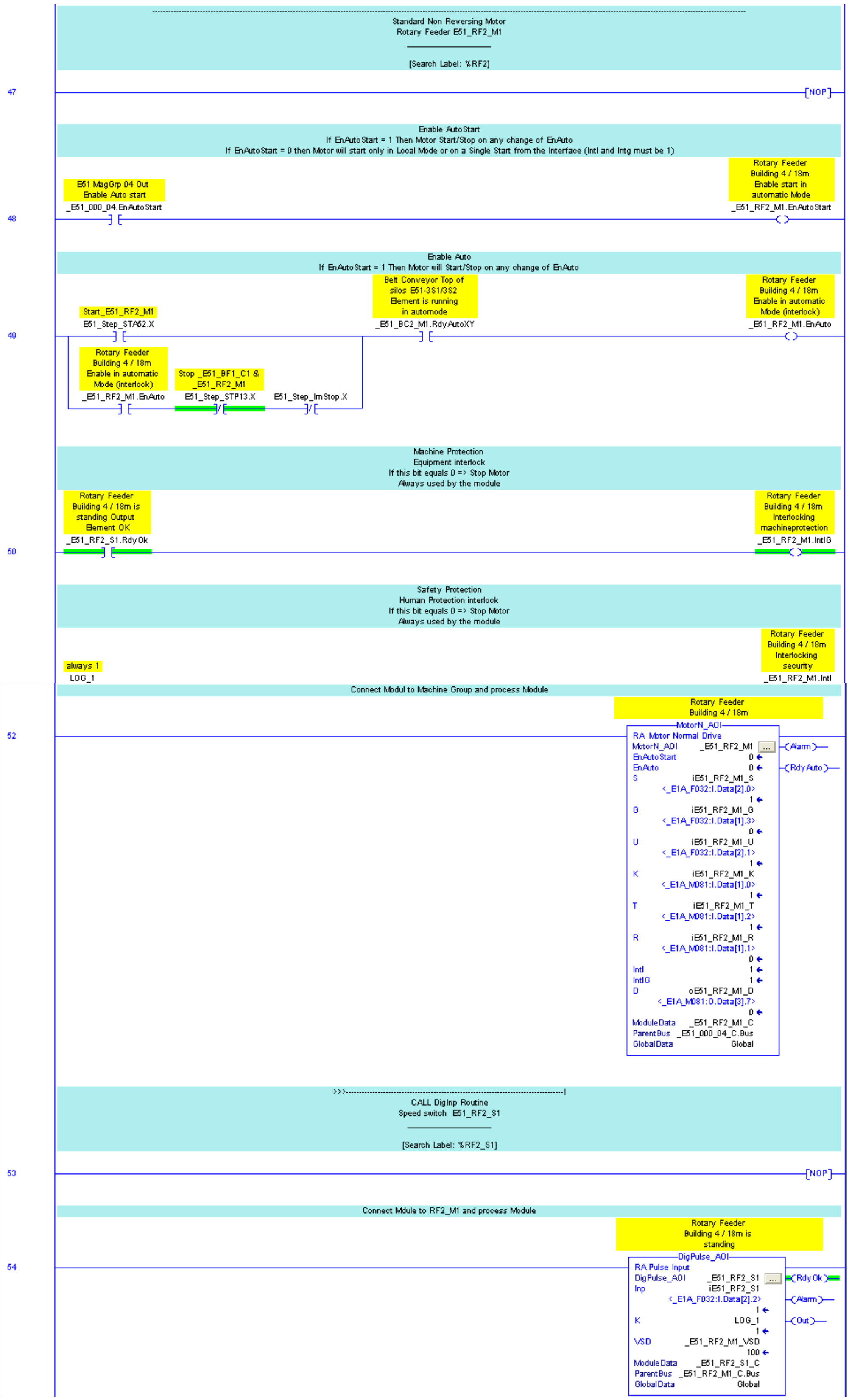
Wiring Diagram
Analog Input - PID Module - Actuator Analog Controlled



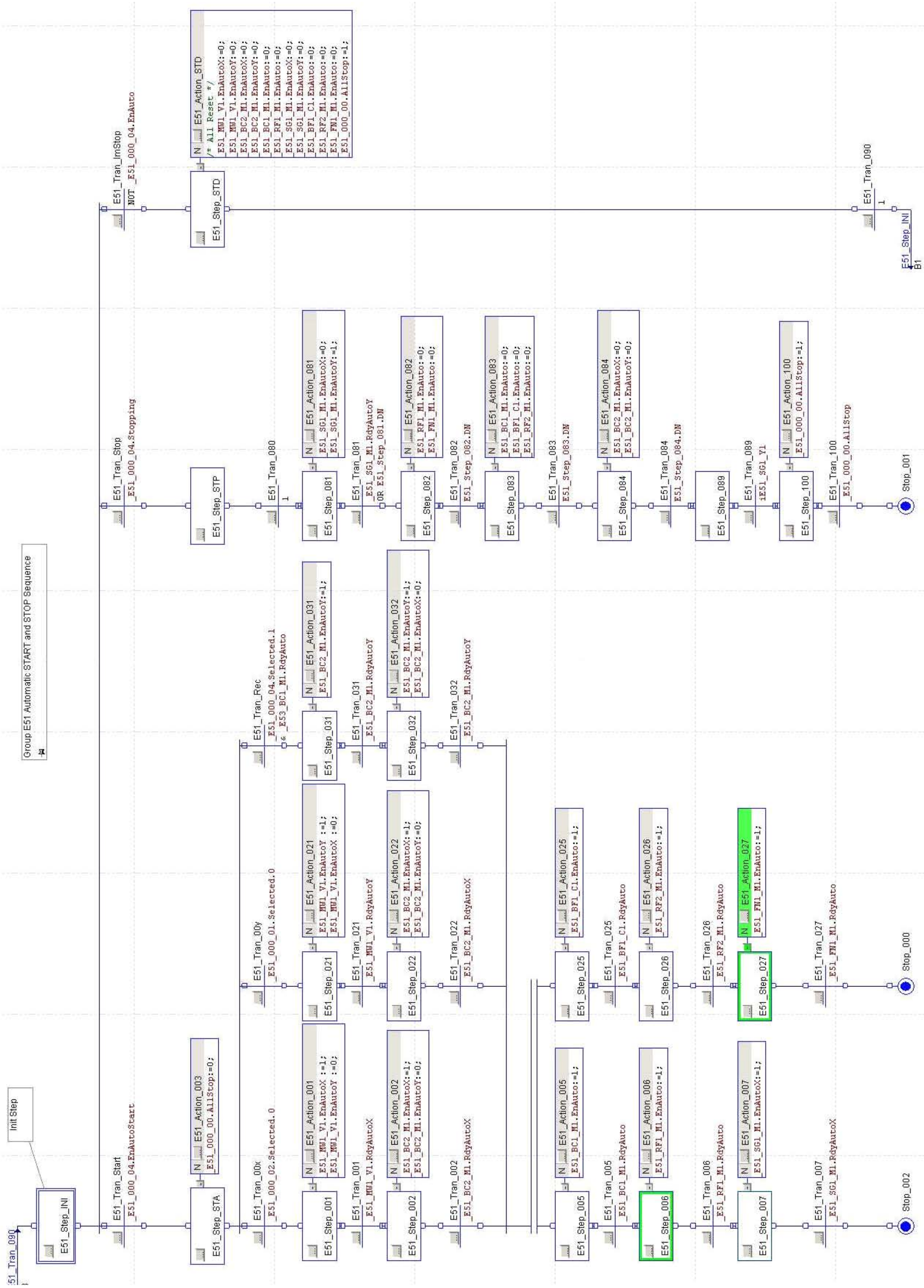
Wiring Diagram
Analog Input - PID Module - Actuator Pulse Controlled



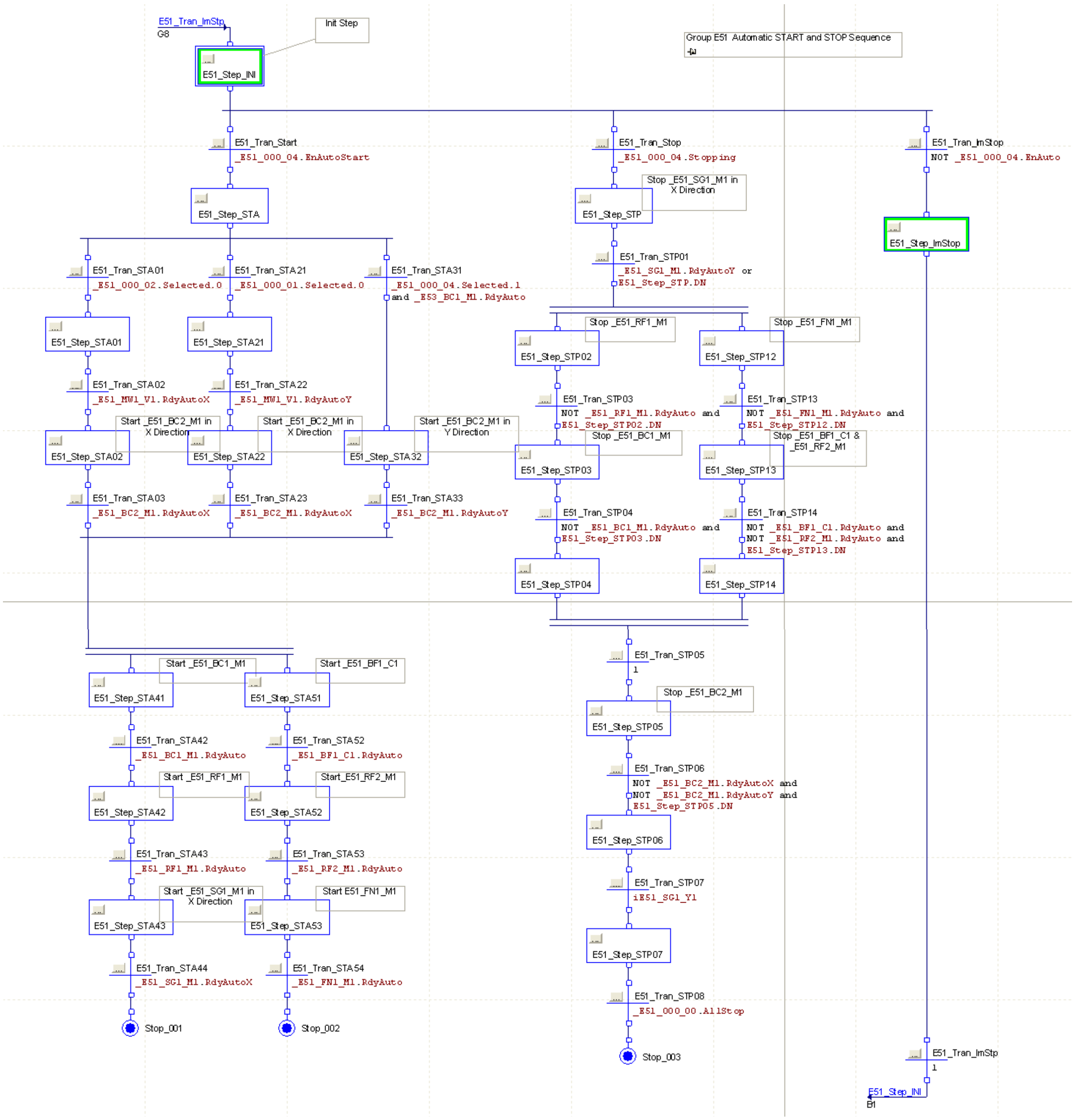
Ladder Diagram Using Add-On Instructions MotorN & DigInp



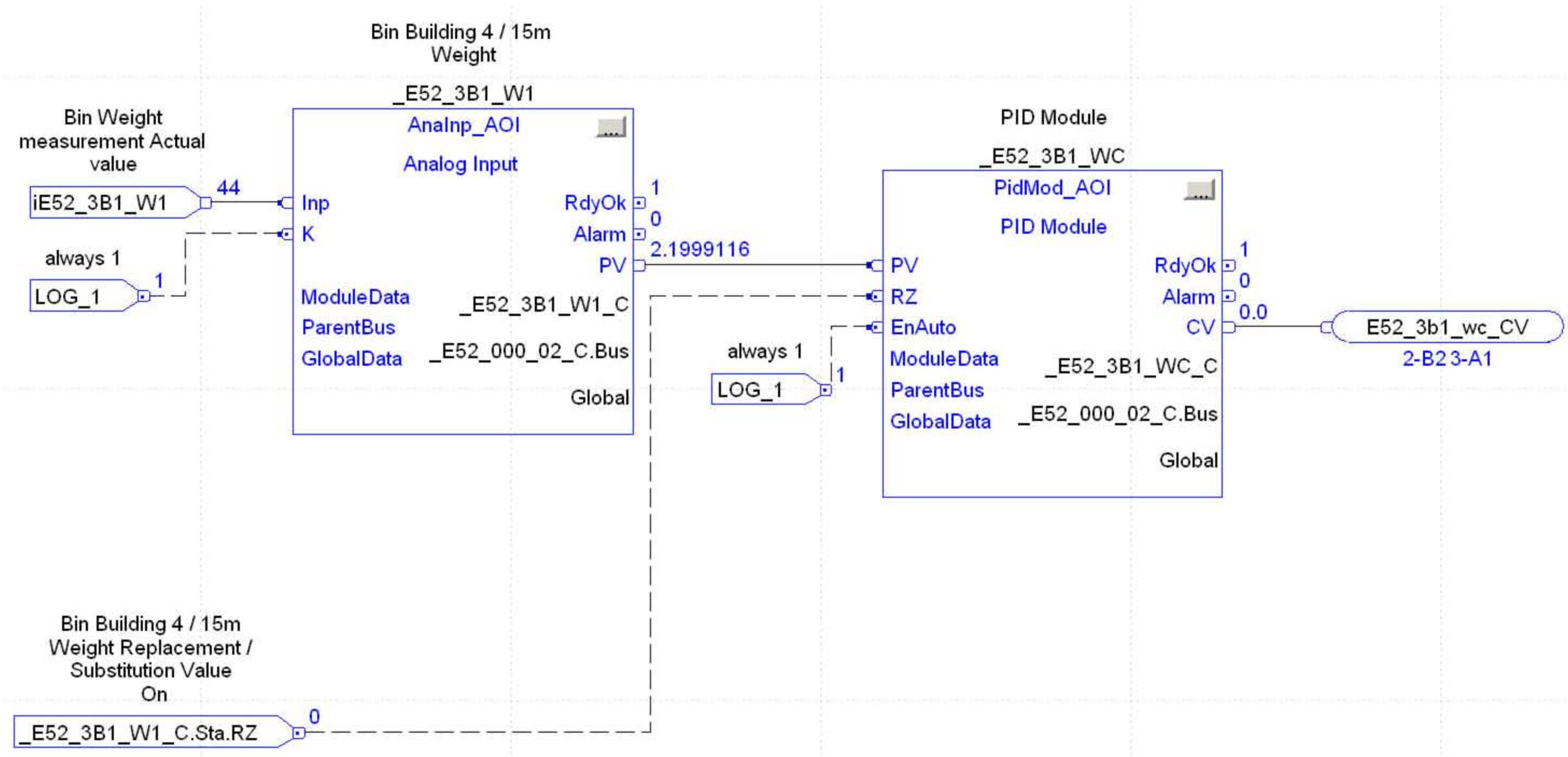
Function Block Diagram Using An Actuator Module (ActMod)



Sequential Function Chart for Machine Start and Stop Sequences



Function Block Diagram Using an Actuator Module (ActMod)



MMCL Module Definitions and Signal Descriptions

SysGrp

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Local	BOOL		Inp		Local Selector / Disable Single Start (same as **Cmd.EU**) Enables all Child Devices for Local Operation or Disables both Local and Single Start if previously ON->Sta.REU.	Set **Local** = 1 (Pulse) by the PLC application program to toggle the SysGrp Local Mode. This performs the same function as the Local button on the SysGrp HMI Faceplate (**Cmd.4**) Case 1: If Local Mode is ON for any device on the SysGrp bus (**<SysGrp>.Bus.21** = 1), then this **Local** pulse turn Local Mode Off for all CtrlGrp devices. Case 2: If Local Mode is OFF for all devices on the SysGrp bus (**<SysGrp>.Bus.21** = 0), then the **Local** pulse will turn Local Mode ON for all SysGrp devices.
ImmStop	BOOL		Inp		Immediate Stop (same as Cmd.SIR) ->Sta.IMS	Set **ImmStop** = 1 (pulse or maintained) by the PLC application program to immediately stop all SysGrp bus devices. This performs the same function as "Imm. Stop" on the SysGrp HMI Faceplate (**Cmd.6**).
ApplyPar	BOOL	x	Inp	**If you change analog scalings!** Refer to E50 ACD file, Main Routine, Rung 4	Apply Parameter Signal to all Modules -> **Global.ApplyPar**	Toggle **ApplyPar** = 1 and the SysGrp AOI will read the **ApplyPar** input and reset the value to 0. The SysGrp AOI will apply scaling & parameter changes to ALL **ActMod**, **PIDMod**, & **AnaInp/C** modules in the CLX. (i.e. new scaling, sample rate, min/max ranges, etc.) by using the **Global.ApplyPar** output tag. Parameters are also applied on a a system First Scan S:SF (e.g. at power-up of the processor). The **Global.ApplyPar** output is set to 1 when **Global.Scan1.Prog** = 0 and remains ON for the # of scans programmed for Global.Scan1.Preset (default=4). *For example: If you change analog scaling, then you must toggle this bit to apply that scaling change. This is global for the CLX processor.*
PowerDip	BOOL		Inp		PowerDip Message Input ->Sta.PowerDipFault ->Sta.W	Set **PowerDip** = 1 by the PLC application program. A field input (fast relay) signal is used to detect the power dip and drive this input signal. This will immediately cause the **<SysGrp>.Active** state to be set to 0, and **Sta.PowerDipFault** = 1. immediately. The SysGrp will alarm that a PowerDip was seen even if there is no reaction by the CtrlGrps.
WA.0..7	SINT of BOOLs		Inp	Ladder Coding above SysGrp AOI. HMI Tag Description that matches each bit	Warning Alarm Bit 0..7 (unspecified) ->Sta.WA.0..7->Sta.W 1=Warning / Minor Fault	Tags are warning conditions from the SubSystem. **WA.*** = 1 is a warning condition. The **Sta.W** will always report the **WA** Warnings and a corresponding warning message appears on the **SysGrp** HMI Faceplate. On the Faceplate, the lower bit has priority for display (only 1 message will appear at a time). Note: You have to configure the text message in the HMI Tag Database under **WAM.*** tags.

FA.0..7	SINT of BOOLs		Inp	Ladder Coding above SysGrp AOI. HMI Tag Description that matches each bit	Failure Alarm Bit 0..7 (unspecified) ->Sta.FA.0..7->Sta.F 1=Failure / Severe Fault	Tags are Failure Alarm conditions from the SubSystem. **FA.*** = 1 is a failure condition. The **Sta.F** will always report the **FA.0..7** Failures. For each FA.0..7 bit, a corresponding failure message appears on the **SysGrp** HMI Faceplate. On the Faceplate, the lower bit has priority for display (only 1 message will appear at a time). Note: You have to configure the text message in the HMI Tag Database under **CAM.*** tags.
AlarmReset	BOOL		Inp		System Group Alarm Reset (same as **Cmd.ACK**)	Set **AlarmReset** = 1 (pulse or maintained) by the PLC application program to reset SysGrp and bus device alarms through the SysGrp bus (**<SysGrp>.Bus.29**). This performs the same function as the Acknowledge button on the SysGrp HMI Faceplate (**Cmd.3**). If the AlarmReset signal is maintained (1), then the **AlarmReset** continously transmitted over the CtrlGrp bus.
GongReset	BOOL		Inp		Gong Reset (same as **Cmd.RES**) ->**Warning/FailureGong.0..31**=OFF	Set **GongReset** = 1 by the PLC application program to clear (set = 0) the **FailureGong** and **WarningGong** tags. Recommended to Pulse the GongReset, but it can be maintained to suppress the Warning & Failure Gongs.
Alarm	BOOL		Out		Any Device on Alarm (Warning or Failure) Common signal Sta.W or Sta.F	**Alarm** = 1 Identifies that the **<SysGrp>.Bus** reports either a Warning or Failure condition from a child device. See also **Sta.W** and **Sta.F**.
Warning	BOOL		Out	Used by PLC program (if needed).	Warningß**Sta.W** and/or Failureß**Sta.F**	**Warning** = 1 Identifies that the <SysGrp>.Bus(bit19) reports an active Warning condition. Same value as **Sta.W**.
Failure	BOOL		Out	Used by PLC program (if needed).	collected from Child Modules and SysGrp	**Failure** = 1 Identifies that the <SysGrp>.Bus(bit18) reports an active Failure condition. Same value as **Sta.F**.
WarningGong.0..31	DINT of BOOLs		Out	Mapping bits to HW outputs	Warning and Failure Gong/Sound, enabled on alarm of any module	Gongs are sound devices in the control room. The Global.WarningGongCode tag is used to transfer the warning gong command from CLX MMCL AOI modules to the SysGrp module. The SysGrp AOI module will map the Global.WarningGongCode to the WarningGong DINT tag. The WarningGong.0..31 will reset after time specified in Global.Par.AlarmGongTime has expired and no new warnings or failures have occurred. The PLC application program must map each of the WarningGong.0..31 bits to physical outputs that provide the warning sounds. See also GongReset.

FailureGong.0..31	DINT of BOOLs		Out	Mapping bits to HW outputs	Set by bit pattern to any {Module}.**Par.AlarmGongCode**. The outputs are set OFF after **Global.Par.AlarmGongTime** if no new gong is released within this time. Reset by **ResetGong** is always possible.	Gongs are sound devices in the control room. The Global.FailureGongCode tag is used to transfer the failure gong command from CLX MMCL AOI modules to the SysGrp module. The SysGrp AOI module will map the Global.FailureGongCode to the FailureGong DINT tag. The FailureGong.0..31 will reset after time specified in Global.Par.AlarmGongTime has expired and no new failures or warnings have occurred. The PLC application program must map each of the FailureGong.0..31 bits to physical outputs that provide the warning sounds..
Horn.0..31	DINT of BOOLs		Out	Mapping bits to HW outputs	Startup Warning Horn 0..31, enabled at startup, set by bit pattern to any {Module}.**Par.StartupHornCode**	Warning Horns are sound devices in the field used to warn of equipment startups. The Global.ComHornCode is used to transfer to warning horn command from the CLX MMCL modules to the SysGrp module. The SysGrp AOI maps the Global.ComHornCode to the <SysGrp>.Horn tag every fourth (value of Global.Scan1.Preset) scan of the CLX processor. The PLC application program must map each of the Horn.0..31 bits to physical outputs that provide the Horn in the field.
FlashLight.0..31	DINT of BOOLs		Out	Mapping bits to HW outputs	Startup Warning Flashlight 0..31, enabled by bit pattern set to any {Module}.**Par.StartupLightCode** at startup	Warning Lights are Flashlight devices in the field used to warn of equipment startups. The Global.ComLightCode is used to transfer to warning horn command from the CLX MMCL modules to the SysGrp module. The SysGrp AOI maps the Global.ComLightCode to the <SysGrp>.Flashligth tag every fourth (value of Global.Scan1.Preset) scan of the CLX processor. The PLC application program must map each of the Flashlight.0..31 bits to physical outputs that provide the Flashlight in the field.

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT		I/O		Interface from/to Child Modules (System Devices)	<SysGrp>.Bus is the bus signal. This signal is mapped as the input parent bus throughout the MMCL library. DO NOT WRITE TO ANY BUS DIRECTLY THROUGH PLC APPLICATION PROGRAM.
Cmd.2 \RES	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Global Gong Reset	**Cmd.2** = 1 is used to reset the warning and failure gongs. Same functionality as **GongReset**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.3 \ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the CtrlGrp child device alarms from the HMI. Same functionality as **AlarmReset**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4 \EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable/Disable Local Mode, Disable Single Start (Toggle)->**Sta.REU**	**Cmd.4** = 1 is used to toggle the SysGrp's "Local" mode status (**Sta.REU**). Same functionality as **Local**. Reserved for HMI Commands. Set in HMI, Reset in AOI.

Cmd.6 \SIR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Immediate Stop Remote	**Cmd.6** = 1 is used to Immediate Stop the SysGrp. Same Functionality as **ImmStop**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.31 \HeartbeatRet	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Heartbeat Return Signal from HMI	**Cmd.31** = 1 is a pulsed Heartbeat Return Signal from the HMI to the PLC. The signal must continually pulse from the HMI. The HMI Feeds back the **Sta.CLX_Heartbeat** into this **Cmd.31** tag. Each rising edge of the signal resets the Heartbeat Timeout Timer. If the Heartbeat Timeout Timer times out, an alarm (**Sta.HMI_Timeout**) will set indicating loss of communication between the PLC and the HMI. *Note: In the HMI application, there must be a Derived Tag called "HeartbeatFct", Event called "ComErr", HMI Tags "Heartbeat" & "HeartbeatRet", and VB code installed on a display that is always open (copied from VB on 00_Main display).*
Sta. CLX_Heartbeat	BOOL		Out	read only!	Heartbeat Signal to HMI	**Sta.CLX_Heartbeat** is a heartbeat signal generated in the CLX and sent to the HMI. The signal will cycle ON for 2 Seconds, then OFF for 2 Seconds and repeat.
Sta.WGong	BOOL		Out		Warning Gong/Sound Indication	**Sta.WGong** = 1 Identifies that one or more bits of **WarningGong.0..31** are set. There should be a warning gong sounding in this condition. (Read Only)
Sta.FGong	BOOL		Out		Failure Gong/Sound Indication	**Sta.FGong** = 1 Identifies that one or more bits of **FailureGong.0..31** are set. There should be a failure gong sounding in this condition. (Read Only)
Sta.ACT	BOOL		Out		System Group Active	**Sta.ACT** = 1 - Always (Read Only)
Sta.REU	BOOL		Out		Local Operation or Single Start Mode Enabled. Note, Alarm Gongs from the particular devices are not set if Local or Single Start is enabled.	**Sta.REU** = 1 Identifies that one or more SysGrp child devices is in Local Mode. Same value as <SysGrp>.Bus.21 (Read Only) Note, Alarm Gongs from the particular devices are not set if Local or Single Start is enabled.
Sta.CK	BOOL		Out		Check Feedback Ok	**Sta.CK** = 1 Identifies that no Warnings (**Sta.W**) or Failures (**Sta.F**) are present in the SysGrp. (Read Only)
Sta.W	BOOL		Out		Warning Alarm (Common Signal) set by Child Modules	**Sta.W** = 1 Identifies that **<SysGrp>.Bus.19** = 1. There is an active Warning on one or more of the SysGrp's child devices.
Sta.F	BOOL		Out		Failure Alarm (Common Signal) set by Child Modules	**Sta.F** = 1 Identifies that **<SysGrp>.Bus.18** = 1. There is an active Failure on one or more of the SysGrp's child devices.
Sta. BatteryFault	BOOL		Out		Message CLX Processor Battery Low ->Sta.W	**Sta.BatteryFault** = 1 Identifies that the CLX Processor Battery is Low or Missing. This condition is reported as a **Sta.W** and can be reset with **AlarmReset** when the monitored signal has returned to normal.

Sta. HMI_Timeout	BOOL		Out		Message HMI Connection Lost-> **Sta.W**	**Sta.HMI_Timeout** = 1 Identifies that HMI to PLC communications has been lost. This condition is reported as a **Sta.W** and can be reset with **AlarmReset**. See also **Cmd.31** when the communciations have been restored.
Sta. PowerDipFault	BOOL		Out		Message **PowerDip** detected ->**Sta.W**	**Sta.PowerDipFault** = 1 Identifies that the input **PowerDip** has been detected as 1 signal. This condition is reported as a **Sta.W** and can be reset with **AlarmReset** when the **PowerDip** signal returns to normal.
Sta.WA.0..7	BOOL		Out		Warning Message / Minor Fault 0..7 (unspecified) ->**Sta.W**	**Sta.WA.0..7** = 1 Identifies that there is an active **WA.0..7** warning alarm and reports a message to the HMI Alarm Log
Sta.FA.0..7	BOOL		Out		Failure Message / Severe Fault 0..7 (unspecified) ->**Sta.F**	**Sta.FA.0..7** = 1 Identifies that there is an active **FA.0..7** failure alarm and reports a message to the HMI Alarm Log
Sta.IMS	BOOL		Out		Immediate Stop releasedß**Cmd.SIR** or **ImmStop**. The status remains on for **Global.Par.MsgDelay** in order alert the operator and allow for message registration.	**Sta.IMS** = 1 Identifies that an Immediate Stop situation has occurred (**Cmd.6** or **ImmStop**). After the immediate stop command is detected then the **Sta.IMS** = 1 and the signal will transition 1->0 after the time specified in **Global.Par.MsgDelay** has expired. This status is used for HMI status.
Par. AlarmGong Code	DINT of BOOLs		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs from the particular devices are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this SysGrp. Set the **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** (*rising edge only*) occurs this gong code is mapped to the **Global.WarningGongCode** tag. (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** (*rising edge only*) causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag. *Note: If one device holds Sta.W to be active, there will be no rising edge of Sta.W if a second device goes into a warning condition.*
Par. AlarmGong Time	DINT		Inp	10000	Gong Autoreset Delay ->**AlarmGong.0..31**= OFF	Set the **Par.AlarmGongTime** to a time delay in ms. This specifies the period of time to sound the Warning and Failure Gongs before automatically silencing the sound device.
Val.RT	REAL		Out		Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the SysGrp. It is a totalize updated every 2 seconds.

Global Data

Input /Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Global. EmergencyPower_1…4	BOOL		Inp	User coding / setting	Emergency Power Priority to all Motor Modules (set by Application). Motors are stoped if their **Par.EmergencyPower** Priority is set 1..4: **EmergencyPower_1**=ON ->stop Prio 1..4 **EmergencyPower_2**=ON ->stop Prio 2..4 **EmergencyPower_3**=ON- >stop Prio 3..4 **EmergencyPower_4**=ON ->stop Prio 4	Global.EmergencyPower_1..4 is managed by the PLC application program entirely and it is recommended that only one bit is set simultaneously. Signal Maintained. MOTORS and SubSys MAY STOP IMMEDIATELY!! EmergencyPower_1..4 is used to stop and prevent starting motors to avoid overloading emergency generators. It is required to prioritize which motors can operate on emergency power. Set <Motor>.Par.EmergencyPower to the desired Priority (1, 2, 3, or 4). - If EmergencyPower_1 = 1 then Stop Priority 1, 2, 3, & 4 - If EmergencyPower_2 = 1 then Stop Priority 2, 3, & 4 - If EmergencyPower_3 = 1 then Stop Priority 3 & 4 - If EmergencyPower_4 = 1 then Stop Priority 4 *For example, if Par.EmergencyPower is set to 3 for a MotorN, the motor will shutdown if EmergencyPower_1 (or _2 or _3) is set to ON.In this example the MotorN will not shutdown if only EmergencyPower_4 is set to ON.* **Application software needs to manage this input entirely.** **(SysGrp doesn't manage!!)**
Global. OneSecClock	BOOL		Out	Value constant: 512	Clock 512 [ms] ON/OFF	Global.OneSecClock is a clock signal managed by SysGrp AOI. The signal value repeats and is ON for 512 ms, OFF for 512 ms.
Global. RunHourClock	BOOL		Out	Value constant: 1024	Clock 1024 [ms] ON/OFF ->{Module}**.Val.RT**	Global.RunHourClock is a clock signal managed by the SysGrp AOI. The signal value repeats and is ON for 1024 ms, OFF for 1024 ms. The signal is used by the MMCL modules for running hour counters.
Global. ApplyPar	BOOL		Out	None -- Just remember to toggle the **SysGrp.ApplyPar** Tag when analog scaling is changed or your scaling change won't go into effect.	Apply Parameter Signal to all Modules, set by **ApplyPar** or at System First Scan **S:SF** e.g. at power-up of the processor. The output is sysnchronized with **Global.Scan1.Prog**	Global.ApplyPar is written by the SubSys AOI. This tag is read by MMCL modules throughout the CLX and is used to apply new parameters in the analog modules globally in the CLX. Refer to <SysGrp>.ApplyPar

Global. HeavyStartup	BOOL		I/O	None! MMCL internal memory Tag	Shared Memory for all Modules set if {Module}.**Par.HeavyStartup**=ON	Global.HeavyStartup = 1 is written and monitored by Motor MMCL modules to identify that a heavystartup motor is currently starting. Refer to <Motor*>.Par.HeavyStartup. This tag allows a motor to delay starting if another heavystartup motor is currently starting.
Global. WarningGongCode	DINT of BOOLs		Inp	None! MMCL internal memory Tag	Common Warning Gong Bus of all Modules	Global.WarningGongCode is written to by MMCL modules to command the <SysGrp>.WarningGong to activate. Refer to <SysGrp>.WarningGong
Global. FailureGongCode	DINT of BOOLs		Inp	None! MMCL internal memory Tag	Common Failure Gong Bus of all Modules	Global.FailureGongCode is written to by MMCL modules to command the <SysGrp>.FailureGong to activate. Refer to <SysGrp>.FailureGong
Global. ComHornCode	DINT of BOOLs		Inp	None! MMCL internal memory Tag	Common Warning Horn Bus of all Modules	Global.ComHornCode is written to by MMCL modules to command the <SysGrp>.Horn to activate. Refer to <SysGrp>.Horn
Global. ComLightCode	DINT of BOOLs		Inp	None! MMCL internal memory Tag	Common FlashLight Bus of all Modules	Global.ComLightCode is written to by MMCL modules to command the <SysGrp>.Light to activate. Refer to <SysGrp>.Light
Global. Scan1.Prog	DINT		I/O	None! MMCL internal memory Tag	Slow Pragram Scan Counter ACC value	**Global.Scan1.Prog** is written by the SysGrp AOI and contains the Slow Program Scan Counter ACC value. INTERNAL USE ONLY
Global. Scan1.Mod	DINT		I/O	None! MMCL internal memory Tag	Slow Module Scan Counter ACC value	Global.Scan1.Mod is written by the MMCL Module AOIs and contains the Slow Module Scan Counter ACC value. INTERNAL USE ONLY
Global. Scan1.Preset	DINT		Out	None! MMCL internal memory Tag	Slow Scan Rate Preset (fixed in program=4)	Global.Scan1.Preset is written by the SysGrp_AOI. It is the Slow Scan Rate Preset (fixed in program=4). Every fourth Scan of the CLX, the MMCL Modules scanned at a slower rate are Scanned (i.e. AnaInp, PosMod, & PidMod). This setting is global to all MMCL modules in the CLX. CLEAR UNDERSTANDING OF SCAN IMPACT NEEDED BEFORE ANY ADJUSTMENT.
Global. Scan2.Prog	DINT		I/O	None! MMCL internal memory Tag	Fast Pragram Scan Counter ACC value	Global.Scan2.Prog is written by the SysGrp AOI and contains the Fast Program Scan Counter ACC value. INTERNAL USE ONLY
Global. Scan2.Mod	DINT		I/O	None! MMCL internal memory Tag	Fast Module Scan Counter ACC value	Global.Scan2.Mod is written by the MMCL Module AOIs and contains the Fast Module Scan Counter ACC value. INTERNAL USE ONLY

Global. Scan2.Preset	DINT		Out	None! MMCL internal memory Tag	Fast Scan Rate Preset (fixed in program=2)	Global.Scan2.Preset is written by the SysGrp AOI. It is Fast Scan Rate Preset (fixed in program=2). Every 2nd Scan of the CLX, the MMCL Modules scanned at a fast rate are Scanned (i.e. Motors, Valves, SubSys, DigIn, ActMod). This setting is global to all MMCL modules in the CLX. CLEAR UNDERSTANDING OF SCAN IMPACT NEEDED BEFORE ANY ADJUSTMENT. Note: SysGrp, CtrlGrp, & MaGrp are scanned every scan of the controller.
Global. Par.DisableTwoLocalStart	BOOL		Inp		Disable the requirement for two pushes of local start button from field to start device in local mode.	When set to one, allow device be started from field in local mode by only pressing start button once. By default Two Local Start requirement is enabled.
Global. Par.DisableJogging	BOOL		Inp		Disable jogging function	When set to one, jogging function is disabled. By default device is allowed to be started and run from field in local mode by keep holding start button down when device is faulted due to machine interlock failure.
Global. Par.EnLamp Test	BOOL		Inp		Enable lamp test function	When set to one, press the local start button while device is running will flash the light. By default lamp test function is disabled.
Global. Par.EnGrpIdentify	BOOL		Inp		Enable control group device identify	When set to one, all devices connected to the control bus could be identified by its control group issuing an identify command. By default this function is disabled.
Global. Par.StartupHornTime	DINT	x	Inp	10000 ms	Preset Startup Horn Time	Global.Par.StartupHornTime is a time setting in ms. This is the period of time that a warning horn will sound. once it is commanded to sound by a MMCL Module (i.e. CtrlGrp, Motor*, Valve*, etc). As a rule, the Global.Par.StartupHornTime < Global.Par.StartupWarningTime.
Global. Par.Startup WarningTime	DINT	x	Inp	15000 ms	Preset Startup Warning Time	Global.Par.StartupWarningTime is a time setting in ms. This is the period of time that a warning light will flash once commanded to flash by a MMCL module (i.e. CtrlGrp, Motor*, Valve*, etc). As a rule, the Global.Par.StartupHornTime < Global.Par.StartupWarningTime.

Global.Par. StartupCancelTime	DINT	x	Inp	20000 ms	Preset Startup Cancel Time	**Global.Par.StartupCancelTime** is a time setting in ms. When an action device MMCL Module (MotorN/D/R, SubSys, Valve1/2, or ActMod) has the parameter **Par.StartupWarningLocal** = 1, then there is a start warning provided before the device will start in Local Mode. The Local Start (G/GX/GY) must be set to the ON state two times. The first time the the Local Start is set, there will be a startup warning. After the start warning horn has completed (**Global.Par.StartupHornTime**), the Local Start needs to be set second time for the device to immediately start. The time specified in **Global.Par.StartupCancelTime** is the time window between the two rising edges of the Local Start signal. If the 2nd Local Start is not set a second time within this time, then startup is cancelled and the start warning would be repeated. **Global.Par.StartupCancelTime** must be > **Global.Par.StartupHornTime** (Recommend 5-10 second difference between the two times)
Global.Par. AlarmDelayTime	DINT	x	Inp	2000 ms	Preset Contactor Feedback Alarm Delay	**Global.Par.AlarmDelayTime** is a time setting in ms to delay the the contactor feedback alarm for Motors and SubSys. This allows for only a single alarm reporting in Motors & SubSys modules when there is a power loss at the MCC, both the R and K signals will be lost at the same time. Only 1 alarm message will appear for this motor: Availability (K)
Global.Par. PowerDipTime	DINT	x	Inp	300 ms	Preset PowerDip Shutdown 300 [ms]	**Global.Par.PowerDipTime** is a time setting in ms. This is the time setting to delay shutdown of a CtrlGrp upon detection of the **<CtrlGrp>.PowerDip** input. This is a debounce timer for **PowerDip** detection. The SysGrp PowerDip is not delayed.
Global.Par. MsgDelay	DINT	x	Inp	5000 ms	Preset CtrlGrp Message Delay	Global.Par.MsgDelay is a time setting in ms. This is the time that temporary messages (i.e. <CtrlGrp>.Sta.IMS and others) are displayed before automatically clearing by the respective MMCL AOI module.
Global.Par. PreCheckTime	DINT	x	Inp	1000 ms	Preset CtrlGrp Pre-Check Time	**Global.Par.PreCheckTime** is a time delay setting in ms. When a HMI Faceplate CtrlGrp start button is pressed, the CtrlGrp will check if any MaGrp is occupied and start the timer. If no MaGrp is occupied, then the CtrlGrp will start when this time delay has expired. (Recommended setting 1-2 sec. Do not set too long as this may be a confusing delay for operators.) *Application Note: Need to consider if this CLX has to check remote CLXs (use of IPComm module) for the MaGrp to report status back to CtrlGrp. When IPComm module is applied, then a longer time may be necessary.*

Global.Par. EUactiveTime	DINT	x	Inp	5000 ms	Preset ActMod Local Control Delay	**Global.Par.EUactiveTime** is a time delay in ms. This is the amount of time to delay before clearing (resetting) the **EUextSel** after a **GX** or **GY** signal is activated. Used only in the ActMod module
Global.Par. GrpIdentifyTime	DINT	x	Inp	10000 ms	Preset the duration of time for Group Devices being identified	**Global.Par.GrpIdentifyTime** is a time delay in ms. This is the amount of time for how long the control group identify status is keeping active after the Identify command was issued from CtrlGrp HMI faceplate.
Global.Par. AnaInpScale [x]	ARRAY	x	Inp		Data base (array of UDT) for different hardware analog modules configuration. Each index [x] corresponds to a different module type. In each analog modules. *E.g. [1] =Point I/O1734-IE2C 0-20mA*	A data base (array) for different hardware analog modules configuration. Each index [x] corresponds to a different analog input HWmodule type. Each analog modules (**AnaInp/C, ActMod**) us this Type No (**Par.InpScale**) to reference to the corresponding used Hardware typ configuration such as Input Scaling. Note: The value of **<modul>.ParInpScale** will refer the array index.
Global.Par.AnaInpScale[x] .InRawMax	REAL	x	Inp	Member of array	AnaInp/AnaInpC/ActMod Raw Value Max for {Module}.Par.InpScale=x (x=1..9)	**Global.Par.AnaInpScale[x].InRawMax** specifies the Raw Maximum value of the **Inp** signal of AnaInp and ActMod AOI modules. This is an array tag where x refers to the array element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaInpScale[x].*** tags. Refer to **<AnaInp>.Par.InpScale**
Global.Par.AnaInpScale[x] .InRawMin	REAL	x	Inp	Member of array	AnaInp/AnaInpC/ActMod Raw Value Min for {Module}.Par.InpScale=x (x=1..9)	**Global.Par.AnaInpScale[x].InRawMin** specifies the Raw Minimum value of the **Inp** signal of AnaInp and ActMod AOI modules. This is an array tag where x refers to the array index element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaInpScale[x].*** tags. Refer to **<AnaInp>.Par.InpScale**
Global.Par.AnaInpScale[x] .OverrangeLimit	REAL	x	Inp	Member of array	AnaInp/AnaInpC/ActMod Raw Overrange Limit for {Module}.Par.InpScale=x (x=1..9)	**Global.Par.AnaInpScale[x].OverrangeLimit** specifies the **Inp** signal value that will cause Over Range alarms of AnaInp and ActMod AOI modules. This is an array tag where x refers to the array element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaInpScale[x].*** tags. Refer to **<AnaInp>.Par.InpScale**

Global.Par.AnaInpScale[x].UnderrangeLimit	REAL	x	Inp	Member of array	AnaInp/AnaInpC/ActMod Raw Underrange Limit for {Module}.Par.InpScale=x (x=1..9)	**Global.Par.AnaInpScale[x].UnderrangeLimit** specifies the **Inp** signal value that will cause Under Range alarms of AnaInp and ActMod AOI modules. This is an array tag where x refers to the array element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaInpScale[x].*** tags. Refer to **<AnaInp>.Par.InpScale**
Global.Par.AnaOutScale[x]	ARRAY	x	Inp		Data base (array of UDT) for different hardware analog modules configuration. Each index [x] corresponds to a different module type. In each analog modules. *E.g. [1] =Point I/O 1734-OE2C 0-20mA*	A data base (array) for different hardware analog modules configuration. Each index [x] corresponds to a different analog output HWmodule type. Analog module **ActMod** us this Type No (**Par.InpScale**) to reference to the corresponding used Hardware typ configuration. Note: The value of **<modul>.ParOutScale** will refer the array index.
Global.Par.AnaOutScale[x].OutRawMax	REAL	x	Inp	Member of array	ActMod Raw Value Max Output for {Module}.Par.OutScale=x (x=1..9)	**Global.Par.AnaOutScale[x].OutRawMax** specifies the Maximum value of the **Out** signal of ActMod AOI modules. This is an array tag where x refers to the array element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaOutScale[x].*** tags. Refer to **<ActMod>.Par.OutScale**
Global.Par.AnaOutScale[x].OutRawMin	REAL	x	Inp	Member of array	ActMod Raw Value Min Output for {Module}.Par.OutScale=x (x=1..9)	**Global.Par.AnaOutScale[x].OutRawMin** specifies the Minimum value of the **Out** signal of ActMod AOI modules. This is an array tag where x refers to the array element #1-9. Array element 0 is not used. The scaling calculations in the respective AOI modules considers other **Global.Par.AnaOutScale[x].*** tags. Refer to **<ActMod>.Par.OutScale**

CtrlGrp

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Start	BOOL		Inp	(not typically needed)	Start or Restart Group (same as **Cmd.GR**)	Set **Start** = 1 (Pulse) by the PLC application program to start the CtrlGrp. This performs the same function as the Start button on the CtrlGrp HMI Faceplate (**Cmd.1**).
Stop	BOOL		Inp	(not typically needed)	Stop Group (same as **Cmd.SR**)	Set **Stop** = 1 (Pulse) by the PLC application program to stop the CtrlGrp. This performs the same function as the Stop button on the CtrlGrp HMI Faceplate (**Cmd.0**)
SBY	BOOL		Inp	(not typically needed)	Standby Selector (same as **Cmd.SBY**)	Set **SBY** = 1 (Pulse) by the PLC application program to toggle the **Standby** output. This performs the same function as the Standby button on the CtrlGrp HMI Faceplate (**Cmd.2**) It is used to show HMI information only (i.e. **SBY** has no influence on the CtrlGrp sequence/control.) The state bit can be used in the application as a memory bit
Local	BOOL		Inp		Local Selector / Disable Single Start (same as **Cmd.EU**) Enables all Child Devices for Local Operation or Disables both Local and Single Start if previously ON->Sta.REU	Set **Local** = 1 (Pulse) by the PLC application program to toggle the CtrlGrp Local Mode. This performs the same function as the Local button on the CtrlGrp HMI Faceplate (**Cmd.4**) Case 1: If Local Mode is ON for any device on the CtrlGrp bus , then this **Local** pulse turn Local Mode Off for all CtrlGrp devices. Case 2: If Local Mode is OFF for all devices on the CtrlGrp bus , then the **Local** pulse will turn Local Mode ON for all CtrlGrp devices.
ImmStop	BOOL		Inp		Immediate Stop (same as **Cmd.SIR**)->**Sta.IMS**	Set **ImmStop** = 1 (pulse or maintained) by the PLC application program to immediately stop all CtrlGrp bus devices. This performs the same function as "Imm. Stop" on the CtrlGrp HMI Faceplate (**Cmd.6**). If the **ImmStop** is maintained (1), then the CtrlGrp start is inhibited.
Clr	BOOL		Inp		Clear Memory "Active" (same as **Cmd.CLR**) Reset Status Active	Set **Clr** = 1 (pulse) by the PLC application program to clear CtrlGrp **Active** status. This performs the same function as the Clear button on the CtrlGrp HMI Faceplate (**Cmd.8**). HMI Application suggestion: May want to secure Faceplate Button
AlarmReset	BOOL		Inp		Alarm Reset (same as **Cmd.ACK**)	Set **AlarmReset** = 1 (pulse or maintained) by the PLC application program to reset CtrlGrp and bus device alarms through the CtrlGrp bus (**<CtrlGrp>.Bus.29**). This performs the same function as the Acknowledge button on the CtrlGrp HMI Faceplate (**Cmd.3**). If the AlarmReset signal is maintained (1), then the **AlarmReset** continously transmitted over the CtrlGrp bus.

PartRun	BOOL		Inp		Part of Group Runnning Feedback	Set **PartRun** = 1 (pulse or maintained) by the PLC application program to pause the startup of the CtrlGrp. The "Starting" CtrlGrp will transition to the "Ready" state and reports a "Restart" condition. A Restart is necessary (through HMI or **Start**) for the CtrlGrp to continue starting. The **PartRun** signal must be 0 before the CtrlGrp Restart is allowed. Refer to AppUserManual 7.1
AllRun	BOOL		Inp		All Devices of Group Running	Set **AllRun** = 1 (maintained) by the PLC application program to identify that ALL required equipment in the CtrlGrp are running. If **AllRun** signal transitions 1->0 during CtrlGrp "Running" state, the CtrlGrp state will change to a "Ready" state. i.e. Operator have the possibility to restart the Group again. Configuration may require application software to mask MaGrp equipment in the **AllRun** string based on MaGrp setting. Refer to AppUserManual 7.1
AllStop	BOOL		Inp		All Devices of Group Stopped	Set **AllStop** = 1 (maintained) by the PLC application program to identify that ALL CtrlGrp equipment is stopped. If the CtrlGrp is in the Stopping state when the **AllStop** transistions 0->1, then the CtrlGrp state will transition to "Stopped" This signifies a sucessful shutdown of the CtrlGrp and Resets the CtrlGrp **Active** status. AllStop must be configured by application software all the time.
EnStopTime	BOOL		Inp	Alternative to AllStop. Either always On or Always Off depending on desired functionality if the system cleanout.	Enable Stopping Time-> **Par.StopTime ->Sta.STP**	Set **EnStopTime** = 1 by PLC application program to enable the stop time (**Par.StopTime**). This will stop all CtrlGrp devices together during a CtrlGrp stop sequence. Set **EnStopTime** = 0 to disable the stop time (**Par.StopTime**) and allow the CtrlGrp devices to stop sequentially based on PLC application logic.
PowerDip	BOOL		Inp		Power Dip Input from Master Relay in HV Station	Set **PowerDip** = 1 by the PLC application program. A field input (fast relay) signal is used to detect the power dip and drive this input signal. The CtrlGrp will respond to the **PowerDip** input signal after the time specified in **Global.Par.PowerDipTime** is expired. The CtrlGrp response is to Immediate Stop all CtrlGrp devices and command all devices' Local Mode OFF . This action is identified with **Sta.PP**.

IntlRelease	BOOL		Inp		Interlock Release for Test (same as **Cmd.EIR**)+G38	Set **IntlRelease** = 1 (pulsed or maintained) by the PLC application program to toggle the IntlRelease mode. This performs the same function as the "Intl Release" button on the CtrlGrp HMI Faceplate (**Cmd.7**). When **IntlRelease** is maintained (1), the CtrlGrp **IntlStop**, **IntlStart**, & **IntlImmStop** are all bridged (bypassed). Extreme caution must be used if using this input as this is a **Source of DANGER!**
IntlStart.0..15	INT of BOOLs		Inp	Ladder Coding above CtrlGrp AOI. Configure HMI Tag Description text that matches each bit corresponding message text.	Interlock Start 0..15->**Val.INR.0..15**	Set any **IntlStart.*** = 1 by the PLC application program to prevent the CtrlGrp from starting. These 16 bits are CtrlGrp start permissives and for each bit there is a corresponding message on the HMI CtrlGrp Faceplate reported through the **Val.INR.0..15** tags. These interlocks can be bridged (bypassed) with the **IntlRelease** signal. **IntlStart.0..15** = 0 is the OK state allowing the CtrlGrp to start.
IntlStop.0..7	SINT of BOOLs		Inp	Ladder Coding above CtrlGrp AOI. Configure HMI Tag Description text that matches each bit corresponding message text.	Interlock Stop 0..7->**Val.INR.16..23**	Set any **IntlStop.0..7** = 1 by the PLC application program to trigger a controlled CtrlGrp shutdown. If IntlStop.0..7 = 1, then the CtrlGrp states of **Running**, **Starting**, And **Ready** will transition to **Stopping**. After the CtrlGrp state **Stopped** is reached, then **IntlStop.0..7** must be 0 to allow the CtrlGrp to start. Each of these 8 bits has a corresponding message on the HMI CtrlGrp Faceplate reported through the **Val.INR.16..23** tags. These interlocks can be bridged (bypassed) with the **IntlRelease** signal.
IntlImmStop.0..7	SINT of BOOLs		Inp	Ladder Coding above CtrlGrp AOI. Configure HMI Tag Description text that matches each bit corresponding message text.	Immediate Stop 0..7->**Val.INR.24..31**	Set any **IntlImmStop.0..7** = 1 by the PLC application program to trigger an immediate CtrlGrp shutdown. Each of these 8 bits has a corresponding message on the HMI CtrlGrp Faceplate reported through the **Val.INR.24..31** tags. These interlocks can be bridged (bypassed) with the **IntlRelease** signal. **IntlImmStop.0..7** = 0 is the OK state allowing the CtrlGrp to Run.
MsgDisp.0…15	INT of BOOLs		Inp	Ladder Coding above CtrlGrp AOI. Configure HMI Tag Description text that matches each bit corresponding message text.	User Message / Infos to diplay of HMI Grp Template (No Impact to Grp Control)	Set any **MsgDisp.0..15** = 1 by the PLC application program to provide a Text Message to the HMI CtrlGrp Faceplate. Each of the 16 bits has a corresponding text message. The faceplate has an display string and only one **MsgDisp.0..15** message will appear if it is active. The lower bits have higher display priority. *Note: HMI Tags (Com\Text\MsgDisp00..15) "On Label" Field text need to be configured to provide the text message.*

EnAutoStart	BOOL		Out	Used by PLC program to start other modules in the group	Enable Automatic Start (1=required for starting modules)	**EnAutoStart** is set to 1 when the CtrlGrp is in the **Starting**, **Running** state. Signal transitions 0->1 when the CtrlGrp transitions from Ready to Starting status (Re-Start situation).
EnAuto	BOOL		Out	Used by PLC program to start other modules in the group	Enable Auto (1=required for starting and running modules)	**EnAuto** is set to 1 when the CtrlGrp enters the **Starting** state and is reset to 0 when the CtrlGrp enters the **Stopped** State.
ResetSFC	BOOL		Out	Used by SFC routines	Trigger (One-Shot) to reset Sequential Function Chart (SFC) at any change to state **Sta.STD/STU/STA/STP**	Each group status change as Startup, Starting or Stopping , Standing, Set (one scan) this output **ResetSFC**, that can be used to initialize (reset) the Sequential Function Chart (SFC). The SFC then selects the actual sequence (e.g. stop sequence).
Alarm	BOOL		Out	Used by PLC program (if needed); set if Sta.W OR Sta.F	Any Device on Alarm (WarningorFailure) Common signal **Sta.W** or **Sta.F**	Group in Alarm condition **Alarm** = 1 Identifies that the **<CtrlGrp>.Bus** reports either a Warning or Failure condition from a child device. <CtrlGrp>.Bus.18 reports a Failure and <CtrlGrp>.Bus.19 reports a Warning. See also **Sta.W** and **Sta.F**.
Failure	BOOL		Out	Used by PLC program (if needed)	Failure<-**Sta.F** Collected from Child Modules	**Failure** = 1 Identifies that the CtrlGrp Bus reports an active Failure condition. Same value as **Sta.F**.
Warning	BOOL		Out	Used by PLC program (if needed)	Warning<-**Sta.W** Collected from Child Modules	**Warning** = 1 Identifies that the CtrlGrp Bus reports an active Warning condition. Same value as **Sta.W**.
Check	BOOL		Out	Use by user logic to do Machin Group(MaGrp) pre selection (.**PreSelection.x**) control.	Status Check Group	Check = 1 Identifies that the CtrlGrp is Not Stopped, OR is Active, OR Prechecked. When the Check signal is 1, then all the CtrlGrp child devices are supervised by the CtrlGrp unless the child device Par.DisableGrpCheck = 1). Note: First Layer child devices (i.e. motors) create & manage their own bus.
Active	BOOL		Out	Use this Active –bit for any user logic, to control **Start Interlocks** of a Group.	Status Active Group /**Sta.ACT**	**Active** is set to 1 when the CtrlGrp is Not **Stopped**. When the CtrlGrp completes a controlled shutsdown OR is Cleared (**Cmd.8** or **Clr**) the Active tag will reset to 0.
Stopped	BOOL		Out	Used by PLC program (if needed)	Status Stopped / Standing /-**Sta.STD**	**Stopped** = 1 Identifies that the CtrlGrp state sequencer is in the "Stopped State". **Stopped** = 1 is the normal initial state for the CtrlGrp. The CtrlGrp will transition from **Stopped** to **Startup** with a CtrlGrp **Start** command provided that interlocks and **Sta.Occ** (Occupied) is OK for the CtrlGrp to start.

Startup	BOOL		Out	Used by PLC program (if needed), used for indication for PLC programmer.	Status Startup Warning Horn and FlashLight /**Sta.STU**	**Startup** = 1 Identifies that the CtrlGrp state sequencer is in the "Startup State". When **Startup** = 1 the CtrlGrp provides a pre-start warning based on **Par.StartupHornCode** and **Par.StartupLightCode**. The CtrlGrp State will transition from **Startup** to **Waiting** after warning for the time specified in **GlobalData.Par.StartupHornTime**. The timer starts when the **Startup** State is entered.
Waiting	BOOL		Out	Used by PLC program (if needed)	Status Waiting with FlashLight /**Sta.WAI**	**Waiting** = 1 Identifies that the CtrlGrp state sequencer is in the "Waiting State". The CtrlGrp state will transition from **Waiting** to **Starting** after the time specified in **GlobalData.Par.StartupWarningTime** is expired. The timer starts when the **Startup** State is entered. The Warning horns will be OFF, and Flashlights will be ON during the Waiting State.
Starting	BOOL		Out	Used by PLC program (if needed)	Status Starting with FlashLight /**Sta.STA**	**Starting** = 1 Identifies that the CtrlGrp state sequencer is in the "Starting State". **EnAuto** is set to 1 when the **Starting** state is entered. Restart requested on the **<CtrlGrp>.Bus** (bit 17) will cause the CtrlGrp state to transition to **Ready**. The CtrlGrp state will transition from **Starting** to **Running** when **AllRun** = 1. Refer to **EnAuto** and **EnAutoStart**.
Running	BOOL		Out	Used by PLC program (if needed)	Status Running /**Sta.RUN**	**Running** = 1 Identifies that the CtrlGrp state sequencer is in the "Running State". **Failures** reported on the CtrlGrp Bus (bit18) OR loosing the **AllRun** will cause the CtrlGrp state to transistion from **Running** to **Ready**. The CtrlGrp state will transition from **Running** to **Stopping** when a CtrlGrp Stop command is provided, OR **IntlStop.0..7** = 1, OR (**<CtrlGrp>.Bus** bit 18 = 1 and **Par.FailureStopDisable = 0**)
Stopping	BOOL		Out	Used by PLC program (if needed), used for indication for PLC programmer.	Status Stopping /**Sta.STP**	**Stopping** = 1 Identifies that the CtrlGrp state sequencer is in the "Stopping State". This state is entered with by a CtrlGrp **Stop** command, OR **IntlStop.0..7**, OR Failures **<CtrlGrp>.Bus.18** with **Par.FailureStopDisable** = 0
Ready	BOOL		Out	Used by PLC program (if needed), used for indication for PLC programmer.	Status Ready to Restart /**Sta.RDY**	**Ready** = 1 Identifies that that CtrlGrp state sequencer is in the "Ready State" and that the CtrlGrp is requesting a Restart request. This state is entered when any of the following are true: CtrlGrp is **Starting** and **PartRun** = 1, OR CtrlGrp is Starting and paused by operator when **Par.AllowStartPause** = 1, OR CtrlGrp has been Starting too long and time out, OR CtrlGrp is Starting and child device on the control bus is requesting a restart, OR CtrlGrp is **Running** and **AllRun** = 0, OR CtrGrp is **Running** and **Failure** and **Par.FailureStopDisable** = 0 and **Par.RdyOnlyUesAllRun** = 0

Standby	BOOL		Out	Used by PLC program (if needed),	Status Standby Active /**Sta.RSB** selected by **SBY**	**Standby** = 1 Identifies that the CtrlGrp is Stopped and in Standby (**Sby** or **Cmd.2**). Status Identification Only -- No CtrlGrp Response other than CtrlGrp state status. The state -bit, can be used in the application as a memory flag, to trigger an automatic start of the sequence. E.g: to trigger an automatic start of a sequence. --[CtrlGrp.Standby]----[user_ready]--------(CtrlGrp.Start)---
u = required						

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT	read only!!	I/O		Interface from/to Child Modules (Machine, Device)	<CtrlGrp>.Bus is the bus signal that this CtrlGrp manages. This signal is mapped as the input parent bus throughout the MMCL library. DO NOT WRITE TO ANY BUS DIRECTLY THROUGH PLC APPLICATION PROGRAM.
Cmd.0 \SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Group)	**Cmd.0** = 1 is used to Stop the CtrlGrp from the HMI. Same functionality as **Stop**. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1 \GR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote (Start or Restart Group)	**Cmd.1** = 1 is used to Start the CtrlGrp from the HMI. Same functionality as **Start**. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.2 \SBY	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Standby Selector (Toggle)	**Cmd.2** = 1 is used to toggle CtrlGrp **Standby** state. Same functionality as **SBY**. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.3 \ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the CtrlGrp child device alarms from the HMI. Same functionality as **AlarmReset**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4 \EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable/Disable Local, Disable Single Start (Toggle)->**Sta.REU**	**Cmd.4** = 1 is used to toggle the CtrlGrp's "Local" mode status (**Sta.REU**). Same functionality as **Local**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.6 \SIR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Immediate Stop Remote	**Cmd.6** = 1 is used to Immediate Stop the CtrlGrp. Same Functionality as **ImmStop**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.7 \EIR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Interlock Release for Test	**Cmd.7** = 1 is used to Enable Interlock Release. Same functionality as **IntlRelease**. Reserved for HMI Commands
Cmd.8 \CLR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Clear Memory "Active"	**Cmd.8** = 1 is used to Clear (Reset) the CtrlGrp's memory of the **Active** condition. Same functionality as **CLR**. Reserved for HMI Commands
Cmd.9 \StartPause	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Pause control group starting sequence.	**Cmd.9** = 1 is used to pause the starting sequence when control group is starting up. Reserved for HMI Commands

Cmd.10 \GrpIdentify	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Identify devices belong to control group	**Cmd.10** = 1 is used to identify group children connected to the control bus. Reserved for HMI Commands
Sta.STD	BOOL		Out		Status Stopped / Standing ->**Bus.0**	**Sta.STD** = 1 Identifies that the CtrlGrp is in the Stopped state. Same value as **Stopped** and **<CtrlGrp>**.**Bus.0**. (Read Only Tag)
Sta.STU	BOOL		Out		Status Startup Warning Horn and FlashLight->**Bus.1**	**Sta.STU** = 1 Identifies that the CtrlGrp is in the Startup state. Same value as **Startup** and **<CtrlGrp>**.**Bus.1**. (Read Only Tag). Warning Horns and Flashlights will be active in this state.
Sta.WAI	BOOL		Out		Status Waiting with FlashLight ->**Bus.2**	**Sta.WAI** = 1 Identifies that the CtrlGrp is in the Startup state. Same value as **Waiting** and **<CtrlGrp>.Bus.2**. (Read Only Tag) Flashlights will be active in this state.
Sta.STA	BOOL		Out		Status Starting with FlashLight ->**Bus.3**	**Sta.STA** = 1 Identifies that the CtrlGrp is in the Starting state. Same value as **Starting** and **<CtrlGrp>.Bus.3**. (Read Only Tag) Flashlights will be active in this state.
Sta.RDY	BOOL		Out		Status Ready to Restart with FlashLight->**Bus.4**	**Sta.RDY** = 1 Identifies that the CtrlGrp is in the Ready state. Same value as **Ready** and **<CtrlGrp>.Bus.4**. (Read Only Tag) Flashlights will be active in this state.
Sta.RUN	BOOL		Out		Status Running->**Bus.5**	**Sta.RUN** = 1 Identifies that the CtrlGrp is in the Running state. Same value as **Running** and **<CtrlGrp>.Bus.5**. (Read Only Tag)
Sta.STP	BOOL		Out		Status Stopping ->**Bus.6**	**Sta.STP** = 1 Identifies that the CtrlGrp is in the Stopping state. Same value as **Stopping** and **<CtrlGrp>.Bus.6**. (Read Only Tag)
Sta.RSB	BOOL		Out		Run Standby, selected by **SBY**	**Sta.RSB** = 1 Identifies that the CtrlGrp is in the Standby mode. (Read Only Tag)
Sta.ACT	BOOL		Out		Group Selection Memory Active	**Sta.ACT** = 1 Identifies that the CtrlGrp is Active. Same value as **Active**. (Read Only)
Sta.REU	BOOL		Out		Local Operation or Single Start Mode Enabled Machine Protection Interlock **IntlG** may be overridden if Local Start activated by Input **G**or**GX/GY/GZ** Note, Alarm Gongs from the particular devices are not set if Local or Single Start is enabled.	**Sta.REU** = 1 Identifies that one or more CtrlGrp child devices is in Local Mode. (Read Only)
Sta.CK	BOOL		Out		Check Feedback Ok	**Sta.CK** = 1 Identifies that the CtrlGrp is being Checked and that no Warnings (**Sta.W**) or Failures (**Sta.F**) are present. (Read Only)
Sta.RRQ	BOOL		Out		Restart Requestedßfrom Child Modules	**Sta.RRQ** = 1 Identifies that there is a Restart Request from one of the CtrlGrp's child devices. (Read Only)
Sta.F	BOOL		Out		Failure Alarm (Common Signal) set by Child Modules	**Sta.F** = 1 Identifies that **<CtrlGrp>.Bus.18** = 1. There is an active Failure on one or more of the CtrlGrp's child devices.

Sta.W	BOOL		Out		Warning Alarm (Common Signal) set by Child Modules	**Sta.W** = 1 Identifies that **<CtrlGrp>.Bus.19** = 1. There is an active Warning on one or more of the CtrlGrp's child devices.
Sta.RIR	BOOL		Out		Group Interlocks released **Cmd.EIR** or **IntlRelease**	**Sta.RIR** = 1 Identifies that the CtrlGrp Interlocks are released from either **Cmd.7** Or **IntlRelease**. The CtrlGrp **IntlStop**, **IntlStart**, & **IntlImmStop** are all bridged (bypassed).
Sta.IMS	BOOL		Out	HMI Status & Used in PLC Program if needed	Immediate Stop released **Cmd.SIR** or **ImmStop**.	**Sta.IMS** = 1 Identifies that a CtrlGrp Immediate Stop has occurred (**Cmd.6** or **ImmStop or IntlImmStop**). This status will automatically clear after the time specified in **Global.Par.MsgDelay** has expired. (Read Only)
Sta.OCC	BOOL		Out	HMI Status & Used in PLC Program if needed	Machine Group Occupied (Active for other Group)	**Sta.OCC** = 1 Identifies that a MaGrp that is child to the CtrlGrp is not available to this CtrlGrp (i.e. Another CtrlGrp has control of this MaGrp active). This data is written by the MaGrp when the <CtrlGrp>.**Check** = 1 and is reported back to the CtrlGrp through the Bus (bit23). The status will automatically clear after the time specified in **Global.Par.MsgDelay** has expired. (Read Only)
Sta.PP	BOOL		Out	HMI Status & Used in PLC Program if needed	Shutdown by Power Dip->**PowerDip** Sta.IMS, Sta.OCC and Sta.PP remain indicated for Global.Par.MsgDelay in order alert the operator and allow for message registration.	**Sta.PP** = 1 Identifies that the CtrlGrp has been shut down due to a **PowerDip**. The status will automatically clear after the time specified in Global.Par.MsgDelay has expired. (Read Only)
Sta.MAT	BOOL		Out	HMI Status & Used in PLC Program if needed	Material (not Empty) Memory, cleared if group orderly stopped	**Sta.MAT** = 1 Identifies that material should be on the CtrlGrp's process equipment. The Sta.MAT is set to 1 with the **Running** state and is reset to 0 when a controlled shutdown completes sucessfully.
Sta.GrpIdentify	BOOL		Out	HMI Status & Used in PLC Program if needed	Group identify is active	**Sta.GrpIdentify** = 1 Identifies that children are being identified by control group. The status will automatically clear after preset secs (**GlobalData.Par.GrpIndentifyTime**) or when **GlobalData.Par.EnGrpIdentify** = 0 once the identify command is issued from HMI.
Par. FailureStop Disable	BOOL	x	Inp		Disable to select Stop Sequence **Sta.STP** in case of Failure	Set **Par.FailureStopDisable** = 1 to specify that the CtrlGrp should NOT stop when a Failure is detected.
Par. AllowStartPause	BOOL		Inp		Allow pause group start up sequence	Set **Par.AllowStartPause** = 1 to specify that the CtrlGrp start up sequence could be paused.
Par. RdyOnlyUsesAllRun	BOOL		Inp		Enable monitor AllRun input only to turn group from Running to Ready state	Set **Par.RdyOnlyUsesAllRun** = 1 to specify that the CtrlGrp will not go to Ready state from Running state even when the CtrlGrp has failure devices. The CtrlGrp will only monitor AllRun input signal.

Par. StartupHorn Code	DINT of BOOLs		Inp	Select by desired Bit Pattern.	Enable Startup WarningHorn0..31 Select Horn(s) by desiredBitPattern. Sets the Global.ComHornCode at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the CtrlGrp's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when the CtrlGrp state is in Startup. Before the CtrlGrp starts, the **Par.StartupHornCode** is mapped to the Global.ComHornCode tag (Indirectly the SysGrp_AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn tag**).
Par. StartupLight Code	DINT of BOOLs		Inp	Select by desired Bit Pattern.	Enable Startup Warning FlashLight 0..31 Select Flash(es) by desired Bit Pattern. Sets the Global.ComLightCode at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the CtrlGrp's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when this CtrlGrp state is **Startup**, **Waiting**, **Starting**, or **Ready**. When the lights need to provide warning, the **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the <SysGrp>.Light tag).
Par. StopTime	DINT	x	Inp	Enter time in [sec] Value necessary when Par.EnStopTime =1	Group Stopping Time (sec) ->**Sta.STP**	Set **Par.StopTime** to a Time delay setting in sec. This specifies the time delay when the CtrlGrp enters the Stopping state until the state transition to **Stopped**. When the CtrlGrp enters the **Stopped** state, the EnAuto signal is reset to 0 (stops all CtrlGrp child devices). Do not use this parameter when **EnStopTime** = 0 where it is desired to have a sequential shutdown. The sequential shutdown must be configured with PLC application software for each device.
Par. StartingTime OutPreset	DINT		Inp	Enter time in [sec]	Group Starting time out preset	Set **Par.StartingTimeOutPreset** to a Time delay setting in sec. This specifies the time delay when the CtrlGrp enters the Starting state until the state transition to **Ready**. When the CtrlGrp enters the **Starting** state, if could not entering **Running** state within the time range specified in Par.StartingTimeOutPreset the CtrlGrp will transition to **Ready** state. This behavior is only enabled when setting is a non zero value.
Val.RT	REAL		Out	Direct linked by RSViewSE –Group Template	Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the CtrlGrp. It is a totalize updated every 2 seconds.

Val.INR	DINT of BOOLs			Direct linked by RSViewSE –Group Template	Interlock Message Bit Number (Bit Pattern) **Val.INR.0..15<-MessageforIntlStart.0..15 Val.INR.16..23ßMessageforIntlStop.0..7 Val.INR.24..31ßMessage for IntlImmStop.0..7**	CtrlGrp Interlock Message **Val.INR.0..15** = 1 Identifies that the corresponding tag in **IntlStart.0..15** is not OK. For each **Val.INR.0..31** bit, there is a message for display on the HMI CtrlGrp Faceplate. Only 1 message is visable and the lower bits have higher display priorty. **Val.INR.16..23** = 1 Identifies that the correspondingtagin**IntlStop.0..7**isnotOK. **Val.INR.24..31** = 1 Identifies that the corresponding tag in **IntlImmStop.0..7** is not OK. *Note: HMI Tags (MSGText\IntlStart00..17, MSGText\IntlStop00..07, and MSGText\ImmStop00.07) "On Label" Field text needtobeconfiguredtoprovidethetextmessage.*
Val.STM	DINT		Out	Direct linked by RSViewSE –Group Template	Status Message Stepß0=Stopped, 1=Startup, 2=Waiting, 3=Starting, 4=Ready, 5=Running, 6=Stopping, 7=Standby	CtrlGrp Status Message **Val.STM** reports the CtrlGrp state to the HMI. No HMI configuration required (all part of the library) 1 = Startup, 2 = Waiting, 3 = Starting, 4 = Ready, 5 = Running, 6 = Stopping, 7 = Standby
Val.STA_RT	DINT		Out	Direct linked by RSViewSE –Group Template	Remaining time in sec during starting before time out.	**Val.STA_RT** is a value identifying the remaining seconds before starting time out for the CtrlGrp. It is calculated only when **Par.StartingTimeOutPreset** is set to a non zero value.

MaGrp

Input /Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
PreSelect.0..x	SINT of BOOLs	x	Inp	Ladder Coding above MaGrp AOI.	PreSelect MaGrp for Parent Bus Nr. 0..x. The corresponding bit **Selected.0..x** is set if the parent group is (re)starting and input **Share.0..x** is set or, if the MaGrp is not occupied by another active selection. In this case the parent group would be signalled to indicate Occupied **Sta.OCC**. Bit legend: <InputName>**.0** = Masterbus0 <InputName>**.1** = Slavebus1 <InputName>**.2** = Slavebus2	Set **PreSelect.0..2** = 1 (maintain signal) by the PLC application program to identify the incoming bus (**MasterBus0, SlaveBus1,** or **SlaveBus2**) that is desired to transfer to the <MaGrp>.Bus. The MaGrp module response varies with the setting of **EnShare.0..2**, **EnRestart.0..2**, and **EnSwitch.0..2**. Refer to **Selected.0..2** for the bus switch reactions. The selected <MaGrp>.Bus status corresponding action of this Input is the Output Selected=1. The PLC application program must manage all PreSelect inputs together. Only 1 PreSelect.0..2 bit should be simultaneously true unless one or more Share.0..2 bits are set to 1. If Share.0..2 = 0, then only one PreSelect.0..2 bit should 1 (i.e. no two bits of Preselect should be value 1). Recommended to use the CtrlGrp output **<CtrlGrp>.Check**, to control the Machine group pre selection by user logic. **Note:** new selection takes action in the moment if Group state change to Startup(e.g. after Restart). **Exception**:If EnSwitch is set, then will be the new selection immediately active.
Share.0..x	SINT of BOOLs		Inp	Ladder Coding above MaGrp AOI.	Share MaGrp for Parent Bus Nr. 0..x. If set, the corresponding channel can always be selected in parallel to other parent groups.	Set **Share.0..2** (maintain signal) by the PLC application program. This setting is similiar to a parameter (i.e. typically the value will not change, set and leave). **Share.0..2** = 1 specifies that the corresponding input bus (MasterBus0, SlaveBus1, or SlaveBus2) can share the **<MaGrp>.Bus** with another input bus. *For Example, if Share.0 = 0 and Share.1 = 1, then the SlaveBus1 can Share the MasterBus0 control of the MaGrp child devices, BUT the MasterBus0 can NOT Share the SlaveBus1 control of the MaGrp child devices.*

EnRestart.0..x	SINT of BOOLs		Inp	Ladder Coding above MaGrp AOI.	Enable Restart at 0 to 1 transition of **PreSelect.0..x**. The parent group will indicate this by **Sta.RRQ** (restart request). Restarting the parent group will set/reset the corresponding **Selected.0..x** and enable or disable the channel.	Set **EnRestart.0..2** (maintain signal) by the PLC application program. This setting is similiar to a parameter (i.e. typically the value will not change, set and leave). **EnRestart.0..2** = 1 specifies that the corresponding input bus (**MasterBus0**, **SlaveBus1**, or **SlaveBus**2) will receive a Restart Request (Bus bit 17) from the MaGrp when the corresponding **PreSelect.0..2** signal transitions 0->1. This restart request will be identified on the CtrlGrp faceplate and will allow the operator to Restart the CtrlGrp. Refer to **Selected.0..2** for actual Bus switching response. *For example, If EnRestart.0 = 1, then when PreSelect.0 transitions 0->1, the MaGrp will request that the MasterBus0-CtrlGrp be restarted.* After version 1.4:) - Restart will be set also, if we do a de-selection (PreSelect 1->0) after the CtrlGrp -Bus was Selectet.
EnSwitch.0..x	SINT of BOOLs		Inp	Ladder Coding above MaGrp AOI.	Enable immediate switching at change of **PreSelect.0..x**. The corresponding output **Selected.0..x** is set/reset directly in order to enable or disable the channel without calling for a restart. The **EnRestart.0..x** is superceded.	Set **EnSwitch.0..2** (maintain signal) by the PLC application program. This setting is similiar to a parameter (i.e. typically the value will not change, set and leave). **EnSwitch.0..2** = 1 specifies that the corresponding input bus (**MasterBus0**, **SlaveBus1**, or **SlaveBus2**) will immediately switch when the corresponding **Preselect.0..2** transitions either 0->1 or 1->0. The Occupied (Active) status of the currently selected bus is IGNORED (Bus bit 23). *For example, If EnSwitch.0 = 1, then when PreSelect.0 becomes 1, the MaGrp will immediately switch onto the selected input bus regardless if another input bus has control (Occupied) of the MaGrp.*
Selected.0..x	SINT of BOOLs		Out		MaGrp Selected for Parent Bus Nr. 0..x. The corresponding **PreSelect.0..x** has been accepted to use the MaGrp for the selected parent group. The channel is selected when (re)starting the parent group if input **Share.0..x** is set for sharing the MaGrp or, if the MaGrp is not occupied by another active selection. In this case the parent group would be signalled to indicate **Sta.OCC** (Occupied).	**Selected.0..2** = 1 Identifies which input bus (**MasterBus0**, **SlaveBus1**, and/or **SlaveBus2)** is currently selected by the MaGrp. When the input bus is selected (**Selected.0..2** = 1), then the respective input bus is transfered to the **<MaGrp>.Bus**. (i.e. **Selected.0** = 1 means Master Input Bus is selected and **MasterBus0** is then transfered to the **<MaGrp>.Bus**). If **Share.0..2** is set to 1, then more than 1 input bus can be selected simultaneously. In this case, the MaGrp module combines the two (or more) selected input Bus signals into a single Bus **<MaGrp>.Bus** The trigger to change the selected input bus varies with the **EnSwitch.0..2** setting. If the **EnSwitch.0..2** signal = 1 (respective to the **PreSelect.0..2** = 1), then the MaGrp will Ignore the Occupied (Active) Status of the currently selected bus (Bus bit 23) and switch the selected <MaGrp>.Bus to the **PreSelect.0..2** bus immediately. If the EnSwitch.0..2 signal = 0 then the CtrlGrp Bus Startup status (Bus bit 1) corresponding to the preselected bus (**PreSelect.0..2** = 1) is the MaGrp's trigger to switch to the selected bus.

AllRun.0..x	SINT of BOOLs		Out		All Devices for Parent Module Nr. x are running	**AllRun.0..2** = 1 Identifies that the the corresponding input bus (**MasterBus0**, **SlaveBus1**, and/or **SlaveBus2**) is selected by the MaGrp (**Selected.0..2** = 1) AND that all MaGrp equipment is running (**AllRunning** = 1). Recommendation: These status bits can be used in the corresponding input bus <**CtrlGrp>.AllRun** strings to simplify that logic string. (i.e. **AllRun.0** may be used in MasterBus0's <**CtrlGrp>.AllRun** string)
AllStop.0..x	SINT of BOOLs		Out		All Devices for Parent Module Nr. x are stopped	**AllStop.0..2** = 1 Identifies that the the corresponding input bus (**MasterBus0**, **SlaveBus1**, and/or **SlaveBus2**) is selected by the MaGrp (**Selected.0..2** = 1) AND that all MaGrp equipment is running (**AllStopped** = 1). Recommendation: These status bits can be used in the corresponding input bus **<CtrlGrp>.AllStop** strings to simplify that logic string. (i.e. AllStop.0 may be used in MasterBus0's <**CtrlGrp>.AllStop** string)
AllRunning	BOOL		Inp	Coded above the MaGrp AOI	All Devices of this MaGrp are running (Feedback)	**Set AllRunning** = 1 by the PLC application program when devices under this MaGrp are running. Similar to **CtrlGrp.AllRun**
AllStopped	BOOL		Inp	Coded above the MaGrp AOI	All Devices of this MaGrp are stopped (Feedback)	**Set AllStopped** = 1 by the PLC application program when devices under this MaGrp Bus are stopped. Similar to **CtrlGrp.AllStop**
EnAutoStart	BOOL	x	Out	Use by all MaGrp Child action modules (Motors/Valves) <module>.AutoStart tags	Enable Automatic Start from Parent CtrlGrp, 1=required to start modules	The **<MaGrp>.EnAutoStart** is similar to <**CtrlGrp>.EnAutoStart**. It is calculated based on the selected (**Selected.0..2 = 1**) input bus signals (**MasterBus0**, **SlaveBus1**, and **SlaveBus2**). **<MaGrp>.EnAutoStart** is used start the child modules on the <**MaGrp>.**Bus. Refer to the description for <**CtrlGrp>.EnAutoStart**
EnAuto	BOOL	x	Out	Use by 1st MaGrp Child <module>.EnAuto for interlocking	Enable Auto from Parent CtrlGrp, 1=required for starting and running modules	The **<MaGrp>.EnAuto** is similar to <**CtrlGrp>.EnAuto**. It is calculated based on the selected (**Selected.0..2 = 1**) input bus signals (**MasterBus0**, **SlaveBus1**, and **SlaveBus2**). **<MaGrp>.EnAuto** is used start and interlock the child modules on the <**MaGrp>.**Bus. Refer to the description for <**CtrlGrp>.EnAuto**
ResetSFC	BOOL		Out		Trigger to reset Sequential Function Chart (SFC) from the selected Parent CtrlGrp	Each group status change as Startup, Starting or Stopping , Standing, Set (one scan) this output ResetSFC, that can be used to initialize (reset) the Sequential Function Chart (SFC). The SFC then selects the actual sequence (e.g. stop sequence). Note: This Output is managed from the selected Control Group (CtrlGrp)
Stopping	BOOL		Out		Status Stopping from Parent CtrlGrp	**Stopping** = 1 Identifies that the MaGrp is Stopping. This condition occures when the **Selected.0..2** input bus is stopping (Bus Bit 6). If more than one input bus is **Selected.0..2**, then this indicates that the MaGrp is stopping (i.e. Case 1: All selected input buses are stopping OR Case 2: One or more selected input buses are stopping (while other buses are Stopped).

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT	X	I/O		Interface from/to Child Modules (Machine, Device)	<MaGrp>.Bus is the calculated Bus from the MaGrp module. It is calculated from the input bus signals (MasterBus0, SlaveBus1, and SlaveBus2) based on the Selected.0..2 status. **Warning:** The input bus connected to the MasterBus 0 ALWAYS passes critical signals to the <MaGrp>.Bus: Local Mode, AlarmReset, Immediate Stop, Power Dip, & Failure/Warning.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
MasterBus0	DINT	X	I/O		Bus from/to Parent Module Nr. 0 (Master Group or Machine)	Primary Input Bus
SlaveBus1	DINT		I/O		Bus from/to Parent Module Nr. 1 (Slave Group or Machine)	Secondary Input Bus
SlaveBus2	DINT		I/O		Bus from/to Parent Module Nr. x (Slave Group or Machine)	Tertiary Input Bus

IPCom

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ComError	BOOL		Out	Connect to CtrlGrp input *.MsgDisp.n*	Communication Error Lost communication link to other PLC (Timeout)	In case of Communication-Error, all Devices on Slave IPCom will stop immediately! The IPCom module does not have an HMI Template (popup), to indicate this Alarm to the Operator. To bring this information to the Operator Screen, we can use a special input at **CtrlGrp** module, to show this information on the HMI CtrlGrp Popup. Connect CtrlGrp input **.MsgDisp.n** to indicate our Communication Error situation. Furthermore, in case of failure, we have to switch off the CtrlGrp Input **AllRun**, in order to have the possibility to **restart** a CtrlGrp. In this case, the CtrlGrp changes into Ready-status.
Master	BOOL		Out	Set by user code or set Tag value.	Diagnostic output, this Module is as Master Configured	Indicate the status of the module configuration as Master or Slave. Represent parameter **Par.MasterMode** **1= Master Module, 0= Slave Module**
EnAutoStart	BOOL		Out	Mapping Code above the Device AOI	Enable Automatic Start (1=required for starting modules) (0->1=restart condition for modules)	The **<IPCom>.EnAutoStart** is similar to <**CtrlGrp>.EnAutoStart**. <IPCom>.EnAutoStart is used to start the child modules on the **<IPCom>.Bus.** Refer to the description for <**CtrlGrp>.EnAutoStart**
EnAuto	BOOL		Out	Mapping Code above the Device AOI	Enable Auto 1=required for starting and running modules (0=stop modules)	The **<IPCom>.EnAuto** is similar to <**CtrlGrp>.EnAuto**. <IPCom>.EnAuto is used to start and interlock the child modules on the **<IPCom>.Bus**. Refer to the description for <**CtrlGrp>.EnAuto**
MasterBus	DINT	X	In/Out	Link Group Bus	Bus Interface to connect an master site (CtrlGrp.Bus)	This Bus will distribute to the Slave (remote) IPCom module. **Note**: MasterBus is always to link to an GroupBus <CtrlGrp_C.Bus! (To connect a *<MaGrp>.Bus* is not allowed, the reason is that the "Occupied" -signal which is generated at the remote MaGrp does not function)

ProducedData	IPC_Data	X	In/Out	Manualy you have to create a Controller Tag from DataTyp "IPC_Data" and config it as "Produced"-Typ	Produced Data structure, the allocated Tag is linked to remote PLC	The communication basis of the IPCom module uses the CLX system **Produced/Consumed Tags**. After the programmer has created and configured a Produced/Consumed Tag-Structure, the IPCom modules will plug on to this Tag, as a communication channel.If we use more than one remote connection to the same Controller, we create an array of **IPC_Data** and extend the array on the required channels.
ConsumedData	IPC_Data	X	In/Out	Manualy you have to create a Controller Tag from DataTyp "IPC_Data" and config it as "Consumed"-Typ	Consumed Data structure, the allocated Tag is linked to remote PLC	

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT	X	In/Out	To link to a <MaGrp>SlaveBus1 or <MaGrp>SlaveBus2 or Device module	Bus Interface to connect an slave site. (Machine, Device)	This is the Bus Output by the Slave IPCom. Typically this bus is linked to a Maschine Group or Motor module as example.
Par. MasterModule	BOOL	X	Inp	controlled by user code if it's used or set Tag value.	Master - Slave configuration 1= Module is Master; 0= Slave	Configuration of IPCom module. The main function of IPCom is to distribute the Group-Bus -Data. Only one can be an Master IPCom per Control Group ("Master" Group is the bus keeper).
Par. HoldOutput	BOOL		Inp		1= Outputs are kept in last state in case of Alarm 0= Outputs are set to "0" in case of Alarm	If set to 0 all User Data will reset in case of communication Alarm (Sta.CTA). **UserRec.Data[n]** <- reset to"0".
Par. ErrorWarning	BOOL	X	Inp		Set module as warning device only. 1= Module is Warning Device(W); 0= If ComError then CtrlGrp go in Failure (F)	Set **Par.ErrorWarning** = 1 to specify that **Sta.CTA** alarms will be reported as Warning alarms (**Sta.W**) only. Set **Par.ErrorWarning** = 0 to specify that these alarms should be reported as Failure alarms (**Sta.F**). Examle: A device with set Par.ErrorWarning-bit is configured as a "Warnning device" only -> a failure will never stop the Group.
Par. Timeout	DINT	X	Inp		Communication Supervision Time in [ms]	Set **delay time** in [ms]. By a communication lost situation this time has to expire before we get a communication error **Sta.CTA**
Sta. CTA	BOOL		Out		Communication Timout Alarm,->Link lost to remote PLC	Don't forget to add the IPCom.Sta.*CTA* Tag in the HMI Tag Database. This HMI –Tag is to configure as an Alarm Tag! (Sta.CTA is equal to module Output ComError!)

Sta. KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready -> Communication error Sta.CTA=1.
Master. Select	BOOL		Inp	Mapping Code above the AOI only at the Master IPCom	Pre select remote MaGrp (output on Remote Slave site: SelectMaGrp)	Is the same as **PreSelect.x** of a MaGrp. If set **Master.Select**=1 this will set the **Slave.SelectMaGrp**=1 on the remote Slave IPCom. Appl example: With this bit you could select/deselect a MaGrp.PreSelect.x -Bit in a remote CPU.
Master. AllRun	BOOL		Out	Mapping Code above the AOI only at the Master IPCom	Feedback from remote Slave, all remote Devices are running	This output an **Master IPCom** is the feedback from the remote Slave IPCom and representing the running status of the remote devices. Controlled by<- **Slave.AllRunning**
Master. AllStop	BOOL		Out	Mapping Code above the AOI only at the Master IPCom	Feedback from remote Slave, all remote Devices are stopped	This output an **Master IPCom** is the feedback from the remote Slave IPCom and representing the device status all stopped at the remote site. Controlled by<- **Slave.AllStopped**
Slave. AllRunning	BOOL		Inp	Mapping Code above the AOI only at the Slave IPCom	All Devices are running	This input an **Slave IPCom** is the status feedback to the remote Master-IPCom, and is representing the all running device status of each motor at the slave site. **-> Master.AllRun** Appl. Example: On the Slave IPCom, all <Motor>.RdyOk=1 signal will control this input signal, where are controlled by IPCom.Bus.
Slave. AllStopped	BOOL		Inp	Mapping Code above the AOI only at the Slave IPCom	All Devices are stopped	This input an **Slave IPCom** is the status feedback to the remote Master-IPCom, and is representing the device status all standing at the slave site. **-> Master.AllStop** Appl. Example: On the Salve IPCom, all <Motor>.RdyOk=0 signal will control this input signal, where are controlled by IPCom.Bus.
Slave. SelectMaGrp	BOOL		Out	Mapping Code below the AOI only used at the Slave IPCom	Pre select MaGrp	Is the same as **PreSelect.x** of a MaGrp. Link this signal to the *MaGrp.PreSelect.x* to select/deselect a Machine Group. This bit is controlled by the remote Master IPCom. <- **Master.Select**
UserSend. Data[n]	DINT array	X	Inp		User Data send to corresponding IPCom Module (*.UserRec.Data[x]*)	**Optional** user data. This can be used for independent data exchange instead of MSG's.
UserRec. Data[n]	DINT array	X	Out		User Data receive from corresp. IPCom Module (*.UserSend.Data[x]*)	**Optional** user data. This can be used for independent data exchange instead of MSG's.. See also **Par.HoldOutput**

MotorN

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Input/Output	Data Type	Required	I/O	Default Value	Original Description (MotorN,R,D/SubSys/Valve1,2) From MMCLib RefGuidV101.doc Section 5.2.9 Pg 49	PD Recommended Description (MotorN,R,D/SubSys/Valve1,2
EnAutoStart	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Start (normally mastered by CtrlGrp) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAuto=1 only	The **EnAutoStart** is an "auto start permissive". The **EnAutoStart** is typically but not necessarily mastered by the **CtrlGrp** or **MaGrp.** To initially start the motor both the **EnAutoStart** AND **EnAuto** must be = 1. Only **EnAuto** = 1 is required to keep the motor running. The **EnAutoStart** 0 to 1 transition will restart the motor if **EnAuto** =1. A rung of logic must be setup outside of the AOI to set the **EnAutoStart.** Using the **CtrlGrp.EnAutoStart** output as the input condition, the motor **EnAutoStart** is transitioned from 0 to 1 on a "RESTART" condition. Application Conditions (Permissives) required to Start Motor but not required to keep motor running can be used to set this bit. If the group of equipment is running in auto, setting the **EnAutoStart** to 0 does not stop the equipment while the **EnAuto** is still maintained. **EnAutoStart** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual_V111, chapter 7*
EnAuto	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Operation (normally mastered by CtrlGrp and interlocked by application) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0 ->1 transition - Run requires EnAuto=1 only	The **EnAuto** may be better referred to as the "IntlAuto" as it is the process interlock. The **EnAuto** must be maintained in the ON state to keep Motor running. The **EnAuto** reset to 0 will stop the Motor. The **EnAuto** is used to sequence/start/stop equipment during the auto operation of equipment. **EnAuto** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual_V111, chapter 7*
S	BOOL	X	Inp	Direct allocated to the AOI	Local Stop Input (0=Stop)	**S** is the Fail-Safe Local (Field) Stop PB input signal. **S** = 1 for the Device to start in any mode. If **S** = 0, the Device will always Stop.

G	BOOL	X	Inp	Direct allocated to the AOI	Local Start Input (Go) Overrides Machine Protection Interlock IntlG if Local operation enabled	Local (Field) Start PB input signal (Direction specific). Device must be in "Local" mode to enable the use of the **GX/GY** signals to start the device. - If the device is in "Local" mode, the **GX/GY** 0->1 transition will initiate an alarm acknowledge for this device and all it's bus child devices. GX/GY PB could also be used to test flashlights while device is running if **GlobalData.Par.EnLampTest** = 1. - If the **GX/GY** signal is maintained during "Local" mode (i.e. holding the local start PB), then the Machine Interlocks (**IntlG**) are bridged (i.e. bypassed or jumpered) if GobalData.Par.DisableJogging is set to 0. - If the device is configured to have **Local warning**, (**Par.StartupWarningLocal =1**) then the required sequence to start the device in "Local" mode is as follows: The Local Start PB is momentarily pressed providing a **GX/GY** = 1 signal; the start warning horn will be provided; within X seconds of the start warning horn completion, the Local Start PB is pressed again and the device will immediately start. X represents the time value in **Global.Par.StartupCancelTime.** If **Global.Par.DisableTwoLocalStart** is set to one, then only one press of GX/GY is required to start the device. Function same as **G. GX** start for X direction(open) / **GY** start for Y direction(close). Device Input for Local (Field) Start PB for both directions Should not set GX and GY to 1 at the same time.
U	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no **U** IO signal.	Local Isolator / Safety Switch (0=OFF)	**U** is the Fail Safe Local Isolator Position input signal (i.e. field disconnect switch). **U** = 1 for Device to Start in any mode. If the **U** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.
K	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no K IO signal.	Ok Available 1=Ok, MCC Unit	**K** is the Fail Safe Device OK status input signal (typically from 480V or 120V breaker status). **K** = 1 for device to Start. If the **K** signal transitions to 0 when the device is running, device will Fail and a RESTART is required from Parent CtrlGrp.
T	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no T IO signal.	Thermal Overload Ok 0=Thermal	**T** is the Fail Safe Device Thermal Overload input signal. **T** = 1 for device to start. If the **T** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.

R	BOOL	X	Inp	Direct allocated to the AOI	Running Contactor Feedback. 0=OFF	**R** is the Device Running Status input signal (i.e. contactor position). After the device is commanded to run (**D** = 1), the **R** signal must transition to 1 before **Par.InpOutMonTime** expires or the device will Fail (message on HMI). If the **R** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.
Intl	BOOL	X	Inp	Coded above the AOI Instruction	Safety Interlock 0=Stop/1=Enable. - Required in any operation mode	**Intl** is a Fail Safe Safey Interlock input signal. The **Intl** signal is coded above the device AOI instance and is the result of a string of Safety device signals (i.e. rope switches) and is intended for safety related interlocks. **Intl** cannot be bridged (bypassed). **Intl** = 1 to run device.
IntlG	BOOL	X	Inp	Coded above the AOI Instruction	Machine Protection Interlock 0=Stop/1=Enable - Required in any operation mode - May be overridden if Local operation enabled and Local Start activated by Input G	**IntlG** is the Fail Safe Machine Protection Interlock input signal. The **IntlG** signal is coded above the device AOI instance and is the result of a string of Machine Protection signals (i.e. drift and speed switches). The signal is intended to protect the machinery. If the device is in "Local" mode, the **IntlG** signal can be bridged by holding the devices "Local Start Pushbutton" (**G**) to run device. To be able to use jog function, **GobalData.Par.DisableJogging** has be be set to 0.
D	BOOL	X	Out	Direct allocated to the AOI	Digital Output Contactor or Coil	**D** is the Device Command to Start output signal. **D** = 1 when device is commanded to start (run) and is mapped to the device's contactor (motor) or coil (valve).
Alarm	BOOL		Out	Typically not used in application programming.	Device Alarm (Warning or Failure) Common signal **Sta.W** or **Sta.F**	**Alarm** = 1 Identifies that the devices has an active alarm present or that the device failed during supervised operation. Any of the following conditions can cause **Alarm**: **Sta.UA**, **Sta.KA**, **Sta.TA**, **Sta.SA**, **Sta.KA**, **Sta.XA**, **Sta.YA**, **Sta.TXA**, **Sta.ZYA**, **Sta.ZXA**, **Sta.CA.***, or "Device Failures". When the device is active (Startup, Waiting, Starting, or Running states), "Device Failures" are caused from the Interlock input (**Intl** & **IntlG)** or contactor feedback (No **R** with **D)** failures. All alarm conditions must clear with an alarm acknowledge before the **Alarm** bit will reset. *Note: Not all Status alarms are present for all modules and status alarms require the Internal Check= 1*
Run	BOOL		Out		Running in Any Mode All interlocks are checked and ok	**Run** = 1 Identifies that the device is running. When the device is starting, the **Run** signal will transition 0->1 only after the **D** and **R** signals are both 1 and the **Par.InpOutMonTime** is expired, and when the device is starting after If On, the Device is Running. The Run signal will transition 1->0 immediately after the device is stopped.

RdyAuto	BOOL		Out	Signal is typically used as the auto interlock for the EnAuto string of the next upstream device.	Ready Running in Auto Mode Set if **Run** AND **Par.ReadyTime** Delay All interlocks are checked and ok, signal to enable/ start next devive in auto mode.	**RdyAuto** = 1 Identifies that the device is Running in "Auto" Mode and Ready for auto operation. When the device's **Run** signal transitions 0->1, delay **Par.ReadyTime** before transitioning the **RdyAuto** signal 0->1. The **RdyAuto** signal will transition 1->0 immediately after the device is stopped.

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT		I/O		Interface from/to Child Modules (Device)	**<Motor*>.Bus** is the calculated Bus from the Motor, Valve, or SubSys module. It is calculated from the input bus signal **<Parent>.Bus** and this modules AOI software. It is used to link child devices to this parent device. Example: Belt Conveyor Motor is this MotorN module and child devices include speed switches, drift switches, etc. The child devices would be linked to the Belt Conveyor Motor using this "Bus"
Cmd.0 \SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Single Start)	**Cmd.0** = 1 is used to stop the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1 \GR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote (Single Start)	**Cmd.1** = 1 is used to start the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.3 \ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the device and child device alarms from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4 \EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle) ->**Sta.REU**	**Cmd.4** = 1 is used to toggle the device's "Local" mode status (**Sta.REU**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.5 \EG	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Single Start (Toggle) ->Sta.REG	**Cmd.5** = 1 is used to toggle the device's "Single" mode status (**Sta.REG**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI
Sta.RP	BOOL		Out		Running in Auto Mode (Run Production)	**Sta.RP** = 1 Identifies that the device is starting or running in "Auto" Mode while being supervised by parent Group. (i.e. **Internal Check** = 1)
Sta.RU	BOOL		Out		Running in Local or Single Mode	**Sta.RU** = 1 Identifies that the device is starting or running in "Local" or "Single" Mode while being supervised by parent group (i.e. **Internal Check** = 1)
Sta.RM	BOOL		Out		RunninginAnyMode Set directly by Contactor Feedback Input R to indicate a possible blocked Contactor, for Valve1 Sta.RM=D	**Sta.RM** = 1 Identifies that the device is running. Set directly by the device's **R** input signal

Sta.KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **U**, **K**, **T**, **S,** Contactor feedback (**R** with **D**). This signal should not be confused with **K** or **KAM**. Indicated in HMI as purple
Sta.REU	BOOL		Out		Local Operation Mode Enabled Machine Protection Interlock IntlG may be overridden if Local Start activated by Input G	**Sta.REU** = 1 Identifies that the device is in "Local" mode. Machine protection interlocks (**IntlG)** may be bridged as described in **G** signal description
Sta.REG	BOOL		Out		Single Start Operation Mode Enabled Device can be started from HMI by Cmd.GR Note, Alarm Gongs are not set if Single Start is enabled.	**Sta.REG** = 1 Identifies that the device is in "Single" mode. Machine protection interlocks (**IntlG**) may be bridged as described in **G** signal description
Sta.STU	BOOL		Out		Startup Warning Horn and FlashLight	**Sta.STU** = 1 Identifies that the device is in startup. This signal will be 1 during startup warning until the **Run** signal is 1.
Sta.WAI	BOOL		Out		Waiting with FlashLight	**Sta.WAI** = 1 Identifies that the device is waiting. This signal will be 1 during a device startup after the startup warning horn has stopped (**GlobalDat.Par.StartupHornTime**). The signal will transition 1->0 when either the device is starting or the device is stopped.
Sta.W	BOOL		Out		Warning Alarm (CommonSignal) Set on Alarm if ErrorWarning=1	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.F	BOOL		Out		Failure Alarm (CommonSignal) Set on Alarm if ErrorWarning=0	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.SA	BOOL		Out		Local Stop Alarm (Message) Set if module check active AND Input S=0	**Sta.SA** = 1 Identifies a Local Stop Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **S** signal input or **Cmd.0** from the HMI when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.UA	BOOL		Out		Local Isolator / Safety Switch Alarm (Message) Set if module check active AND Input U=0	**Sta.UA** = 1 Identifies a Local Isolator / Safety Switch Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **U** signal input when device's parent group is acitive (i.e. **Internal Check** = 1)
Sta.KA	BOOL		Out		Availability Alarm (Message) Set if module check active AND Input K=0	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)

Sta.TA	BOOL		Out		Thermal Overload Alarm(Message) Set if module check active AND Input T=0	**Sta.TA** = 1 Identifies a Thermal Overload Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **T** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.RA	BOOL		Out		Contactor Feedback Alarm(Message) Set delayed if Output D/DX/DY AND its corresponding feedback Input R/RX/RY are not equal	**Sta.RA** = 1 Identifies a Contactor Feedback Alarm and reports a message to the HMI Alarm Log. The **R** signal input was not detected within **Par.InpOutMonTime** when **D** signal is active. Alternatively, the **R** signal may have been lost after the **Par.InpOutMonTime** has expired.
Sta.SAM	BOOL		Out		Local Stop Alarm Mimic Set if Input **S**=0	**Sta.SAM** = 1 Identifies that there is no **S** input signal or there is an active **Sta.SA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Local Stop problem.
Sta.UAM	BOOL		Out		Local Isolator / Safety SwitchAlarmMimic Set if Input **U=0**	**Sta.UAM** = 1 Identifies that there is no **U** input signal or there is an active **Sta.UA** alarm. This is used for HMI display (mimic) animiations to identify that there is a local isolator or safety switch problem.
Sta.KAM	BOOL		Out		Availability Alarm Mimic Set if Input **K**=0	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.
Sta.TAM	BOOL		Out		Thermal Overload AlarmMimic Set if Input **T**=0	**Sta.TAM** = 1 Identifies that there is no **T** input signal or there is an active **Sta.TA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Thermal Overload problem.
Sta.RAM	BOOL		Out		Contactor Feedback AlarmMimic Set delayed if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are not equal	**Sta.RAM** = 1 Identifies that there is a contactor failure (**R** when **D** after **Par.InpOutMonTime** expired) or there is an active **Sta.RA** alarm. This is used for HMI display (mimic) animiations to identify contactor feedback failures.
Sta.IDS	BOOL		Out		Safety Interlock Alarm IndicationMimic Set if Input Intl=0	**Sta.IDS** = 1 Identifies that there is no **Intl** input signal or there is an active **Intl** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Safety Interlock problem.
Sta.IDP	BOOL		Out		Maschine Protection Interlock Alarm Indication Mimic Set if Input IntlG=0	**Sta.IDP** = 1 Identifies that there is no **IntlG** input signal or there is an active **IntlG** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Machine Protection Interlock problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.RT	REAL		Out		Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the device. It is a totalize updated every 2 seconds.

Val.DC	DINT		Out		Number of Starts Counter	**Val.DC** is a value identifying the total number of Starts for the device. The value increments 1 for every start.
Val.RST_RT	DINT		Out		Remaining seconds before restart is enabled	**Val.RST_RT** is a value identifying the remaining time in second before the device could be started again.
Par. HeavyStartup	BOOL		Inp		Code for Heavy Start 0=No/1=Yes If ON, all other devices are signalled during startup of this device by common tag **Global.HeavyStartup**	Set **Par.HeavyStartup** = 1 to specify that this device is a heavy starting device (i.e. hard starting with high inrush current). When this device is starting, other heavy startup devices in the same PLC are prevented from starting until this device's **RdyAuto** = 1. This is accomplished using the **Global.HeavyStartup** tag.
Par. HeavyStartupIgn	BOOL		Inp		Heavy Start Ignore 0=No/1=Yes If ON, this device ignores the startup of other heavy starting devices	Set **Par.HeavyStartupIgn** = 1 to specify that this device should start when commanded to start regardless of any other device's **Par.HeavyStartup** configuration and starting status. (I.E. Disable the effect of the **Global.HeavyStartup** tag)
Par. StartupWarningLocal	BOOL		Inp		Device with Startup WarninginLocalMode If ON, this device will set the Horn and Flash outputs specified by **Par.StartupHornCode** and **Par.StartupLightCode** when started locally	Set **Par.StartupWarningLocal** = 1 to specify that this device will always provide a startup warning horn / light before starting the device in "Local" mode. When set, a second pulse is required on the **G** input signal to start the device. Refer to **Par.StartupHornCode** and **Par.StartupLightCode** for specific horn/light configurations. Note that MSHA requires a start warning horn when starting conveyors regardless of mode of operation. **Note**: If set this parameter you have to press twice the G-button! See G description.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableSingle	BOOL		Inp		Suppress visibility of Singlestart mode Button at HMI template.	Set Par.DisableSingle =1 to specify that the "Manual/Auto"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableGrpAlarm	BOOL		Inp		Disable Alarm Indication to Parent Groupmodules If ON, the Warning or Failure Alarm Bus of the Parent Group module is not set. Alarm messages are however still created and **Sta.W** or **Sta.F** is set.	Set **Par.DisableGrpAlarm** = 1 to specify that the device Alarms (**Alarm**) will NOT be reported on the **<Parent>.Bus.18** (Failure) or **<Parent>.Bus.19** (Warning). (This parameter has No affect on Device Alarm reporting i.e. Sta.* tags directly between the device AOI and the HMI.

Par. DisableGrp Check	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set Sta.W or Sta.F	Set **Par.DisableGrpCheck** = 1 to specify that the device will not report to the Control group(CtrlGrp): Restart requests, device Alarms (**Alarm)**, and inhibit the <Module>.Bus.26 (Check) from activating the Module's Internal**.** **Note: Module will not start in Automatic Mode (EnAuto) if this Parameter is set!**
Par. ErrorWarnin g	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by Sta.W IF ON, Alarms are indicated by Sta.F	Set **Par.ErrorWarning** = 1 to specify that device alarms (**Alarm**) will be reported as Device Warnings (**Sta.W**) and these device warnings will be reported as warnings to the **<Parent>.Bus.19**. Warnings will cause the device HMI animation object to be Yellow. Set **Par.DisableGrpCheck** = 0 to specify that device alarms (**Alarms**) will be reported as Device Failures (**Sta.F**) and these device failures will be reported as failures to the **<Parent>.Bus.18**. Failures will cause the device HMI animation object to be Red. Reporting to the Parent Bus is affected by **Par.DisableGrpAlarm** and **Par.DisableGrpCheck**.
Par. OverrideElm Chk	BOOL		Inp		Override element check	Set **Par.OverrideElmChk** = 1 to specify that the device will report contactor feedback failure to control group even when control group is not active.
Par. DisableGrpI dentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par. HasPermObj	BOOL		Inp		Has permissive object connected to Block	Set Par.HasPermObj = 1 to specify that a Permissive AOI is connected to this block and the permissive object on the HMI faceplate is visible to provide permissive diagnostic information.
Par. HasIntlkObj	BOOL		Inp		Has Interlock object connected to Block	Set Par.HasIntlkObj = 1 to specify that an Interlock AOI is connected to this block and the interlock object on the HMI faceplate is visible to provide interlock diagnostic information.

Par. AlarmGong Code	DINT of BOOLs	X	Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag.
Par. StartupHorn Code	DINT of BOOLs	X	Inp		Enable Startup WarningHorn0..31 Select Horn(s) by desiredBitPattern. Sets the **Global.ComHornCode** at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the device's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when starting this device. Before this device starts, the **Par.StartupHornCode** The **Global.ComHornCode** tag (Indirectly the SysGrp AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn** tag).
Par. StartupLight Code	DINT of BOOLs	X	Inp		Enable Startup Warning FlashLight 0..31 Select Flash(es) by desiredBitPattern. Sets the **Global.ComLightCode** at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the device's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when starting this device. The **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the **<SysGrp>.Light** tag).
Par. EmergencyPower	DINT		Inp	Configure only if there is Emergency power for this section of the plant	Priority for Emergency Power Operation (0..4). If set 0, the device is not stopped on Emergency,elseif… **Global.EmergencyPower_1**=ON->stop Prio 1,2,3 & 4 **Global.EmergencyPower_2**=ON->stop Prio 2,3 & 4 **Global.EmergencyPower_3**=ON->stop Prio 3 & 4 **Global.EmergencyPower_4**=ON->stop Prio 4	Set **Par.EmergencyPower** to a value 0 - 4. This is a priority setting that is used to conserve energy when emergency power generators are on-line. (i.e. incoming power from utility is OFF). Higher settings mean lower priority for operations. - Set to 0: the device is not stopped - Set to 1: Stop device if **Global.EmergencyPower_1** = 1; - Set to 2: Stop device if **Global.EmergencyPower_1** or **_2** = 1; - Set to 3: Stop device if **Global.EmergencyPower_1, _2,** or **_3** = 1; - Set to 4: Stop device if **Global.EmergencyPower_1, _2, _3,** or **_4** = 1.

Par. InpOutMon Time	DINT	X	Inp	Default setting Varies by the desired system response	Contactor Feedback Supervision Timer (default 2000 ms) Delay time to check if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are NOT equal->**Sta.RA/RAM**	Set the **Par.InpOutMonTime** to a time delay in ms. This specifies the maximum allowable time between the start command (**D**) and feedback signal (**R**). If this time is exceeded before the **D** & **R** signals are equal, a contactor failure alarm will occur. Recommended default 2000 ms
Par. RestartTime	DINT	X	Inp		Time Delay for Restarting(ms) Time to hold Status Stopping, during this time the device cannot be restarted	Set the **Par.RestartTime** to a time delay in ms. This specifies the minumum time delay between a device stop and the device's next restart. After the device stops, it cannot be restarted until this time expires. It is used for 2 purposes. 1 - Allow motors to coast to a stop before restarting. (Not all motors can "catch" the rotor to restart 2 - Allow motor windings to cool before restarting.
Par. ReadyTime	DINT	X	Inp		Time Delay to set Output Ready Auto (ms) Outputs **RdyAuto** or (RdyAutoX/Y) are set time delayed to allow the device to take up the load before the next device is started.	Set the **Par.ReadyTime** to a time delay in ms. This specifies the time delay between the device's **Run** output signal and **RdyAuto** output signal. This timer is responsible for the start delay of the next device in a startup sequence.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	Inp		Bus Input from/to Parent Module (Machine or Group)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This Motor* module reads information FROM the "Bus" and Writes status information TO the "Bus".
Reference Notes						
Note 1:						**Internal Check** is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: **<Parent>.Bus.26** with **DisableGrpCheck** = 0, **Sta.REU** (Local Mode), OR **Sta.REG** (Single Mode)

MotorN_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
D	BOOL	x	Inp	Direct allocated to the MotorN AOI	Digital Input Contactor or Coil	Input D bit come from MotorN AOI. If D is 1 then MotorN is started and simulation routine set R bit to 1 after a delay time (Contactor Feedback).
G_OUT	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Start Output (1=Start)	G bit can be set by button on HMI faceplate. Motor will be started if MotorN is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorN AOI	Ok Available 1=Ok	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorN faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Isolator / Disconnect Switch (0=OFF)	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorN faceplate and Motor will be stopped.
R	BOOL	x	Out	Direct allocated to the MotorN AOI	Contactor Feedback	R bit can be cleared by switch on HMI faceplate when D = 1. Contactor feedback alarm will be displayed on MotorN faceplate and Motor will be stopped.
TH	BOOL	x	Out	Direct allocated to the MotorN AOI	Thermal Overload	TH bit can be cleared by switch on HMI faceplate. Thermal overload alarm will be displayed on MotorN faceplate and Motor will be stopped.
90	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
D_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor fault from HMI	D_Fault bit is connected with HMI Contactor switch. (0=Ok, 1=Fault)
G_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start from HMI	G_IN bit is connected with HMI Local start button. 1=Start
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)
R_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor feedback fault from HMI	R_Fault bit is connected with HMI Contactor Feedback switch. (0=Ok, 1=Fault)

S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
TH_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Overload fault from HMI	TH_Fault bit is connected with HMI Thermal overload switch (0=Ok, 1=Fault)
tmrFeedback	TIMER				Contactor feedback timer	This timer is used to delay output **R** after receiving **D** from motor block
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

MotorNE3p_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
D	BOOL	x	Inp	Direct allocated to the MotorN AOI	Digital Input Contactor or Coil	Input D bit come from MotorN AOI. If D is 1 then MotorN is started and simulation routine set MotorCurrent bit in E3p module after a delay time.
G	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Start Output	G bit can be set by button on HMI faceplate. Motor will be started if MotorN is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorN AOI	Ok Available 1=Ok	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorN faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Isolator / Disconnect Switch (0=OFF)	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorN faceplate and Motor will be stopped.
S_OUT	BOOL	x	Out	Direct allocated to the MotorN AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.
E3p_Data_in	E3_Inp	x	In/Out	Direct allocated to the E3p module	DeviceNet E3-Node Input structure	DeviceNet E3 Node Input Structure
Node_Failure	BOOL	x	Out	Direct allocated to the User created Node Failure tag	E3p Node failure	This input has to be correctly mapped to allow generation of E3p module communication error.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
G_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start from HMI	G_IN bit is connected with HMI Local start button. 1=Start
Ground_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Ground Fault from HMI	Ground_Fault bit is connected with HMI Ground switch. (0=Ok, 1=Fault). Ground Fault alarm will be generated and E3p trip status will be set to 1.
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available Fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)

Node_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	E3p Node failure from HMI	Node_Fault bit is connected with HMI E3p Node switch. (0=Ok, 1=Fault)
Phase_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Phase Loss fault from HMI	Phase_Fault bit is connected with HMI Phase switch.(0=Ok, 1=Fault). Phase Loss alarm will be generated and E3p trip status will be set to 1.
S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
ThermalUtilized	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Utilized from HMI	Thermal utilized variable is connected to HMI Thermal utilized slider. Setting Thermal utilized value to 85-99% will generate Overload warning. Setting this value to 100% will generate Overload alarm and E3p trip status will be set to 1.
tmrFeedback	TIMER				Timer to simulate E3p running	This timer is used to delay E3p MotorCurrent output after receiving D from motor block
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

MotorR

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EnAutoStart	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Start (normally mastered by CtrlGrp) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAuto=1 only	The **EnAutoStart** is an "auto start permissive". The **EnAutoStart** is typically but not necessarily mastered by the **CtrlGrp** or **MaGrp.** To initially start the motor both the **EnAutoStart** AND **EnAuto** must be = 1. Only **EnAuto** = 1 is required to keep the motor running. The **EnAutoStart** 0 to 1 transition will restart the motor if **EnAuto** =1. A rung of logic must be setup outside of the AOI to set the **EnAutoStart.** Using the **CtrlGrp.EnAutoStart** output as the input condition, the motor **EnAutoStart** is transitioned from 0 to 1 on a "RESTART" condition. Application Conditions (Permissives) required to Start Motor but not required to keep motor running can be used to set this bit. If the group of equipment is running in auto, setting the **EnAutoStart** to 0 does not stop the equipment while the **EnAuto** is still maintained. **EnAutoStart** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual_V111, chapter 7*
EnAutoX/EnAutoY	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Operation X/Y (normally mastered by CtrlGrp and interlocked by application) - Start requires EnAutoStart=1 AND EnAutoX/Y=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAutoX/Y=1only	Similar to the EnAuto in operation but controls which direction the Motor/Valve will run/operate (X or Y). Motor/Valve will not operate if both the **EnAutoX** and **EnAutoY** are set = 1simultaneously.
S	BOOL	X	Inp	Direct allocated to the AOI	Local Stop Input (0=Stop)	**S** is the Fail-Safe Local (Field) Stop PB input signal. **S** = 1 for the Device to start in any mode. If **S** = 0, the Device will always Stop.

GX/GY	BOOL		Inp	Direct allocated to the AOI	Local Start Direction X/Y (Go X/Y) Overrides Machine Protection Interlock IntlG if Local operation enabled	Local (Field) Start PB input signal (Direction specific). Device must be in "Local" mode to enable the use of the **GX/GY** signals to start the device. - If the device is in "Local" mode, the **GX/GY** 0->1 transition will initiate an alarm acknowledge for this device and all it's bus child devices. GX/GY PB could also be used to test flashlights while device is running if **GlobalData.Par.EnLampTest** = 1. - If the **GX/GY** signal is maintained during "Local" mode (i.e. holding the local start PB), then the Machine Interlocks (**IntlG**) are bridged (i.e. bypassed or jumpered) if GobalData.Par.DisableJogging is set to 0. - If the device is configured to have **Local warning**, (**Par.StartupWarningLocal =1**) then the required sequence to start the device in "Local" mode is as follows: The Local Start PB is momentarily pressed providing a **GX/GY** = 1 signal; the start warning horn will be provided; within X seconds of the start warning horn completion, the Local Start PB is pressed again and the device will immediately start. X represents the time value in **Global.Par.StartupCancelTime.** If **Global.Par.DisableTwoLocalStart** is set to one, then only one press of GX/GY is required to start the device. Function same as **G. GX** start for X direction(open) / **GY** start for Y direction(close). Device Input for Local (Field) Start PB for both directions Should not set GX and GY to 1 at the same time.
U	BOOL	X	Inp	Use Log_1 if device has no U IO signal.	Local Isolator / Safety Switch (0=OFF)	**U** is the Fail Safe Local Isolator Position input signal (i.e. field disconnect switch). **U** = 1 for Device to Start in any mode. If the **U** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
K	BOOL	X	Inp	Use Log_1 if device has no U IO signal.	Ok Available 1=Ok, MCC Unit	**K** is the Fail Safe Device OK status input signal (typically from 480V or 120V breaker status). **K** = 1 for device to Start. If the **K** signal transitions to 0 when the device is running, device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
T	BOOL	X	Inp	Use Log_1 if device has no T IO signal.	Thermal Overload Ok 0=Thermal	**T** is the Fail Safe Device Thermal Overload input signal. **T** = 1 for device to start. If the **T** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)

RX/RY	BOOL	X	Inp	Direct allocated to the AOI	Running Contactor Feedback X/Y. 0=OFF	**RX/RY** is the Device Running Status input signal direction specific (i.e. contactor position). After the device is commanded to run (**DX/DY** = 1), the **RX/RY** signal must transition to 1 before **Par.InpOutMonTime** expires or the device will Fail (message on HMI). If the **RX/RY** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (**CtrlGrp**, **MaGrp,** etc.)
Intl	BOOL	X	Inp	Coded above the AOI Instruction	Safety Interlock 0=Stop/1=Enable. - Required in any operation mode	**Intl** is a Fail Safe Safey Interlock input signal. The **Intl** signal is coded above the device AOI instance and is the result of a string of Safety device signals (i.e. rope switches) and is intended for safety related interlocks. **Intl** = 1 to run device. If the **Intl** signal transitions to 0 when the device is running, the device will Fail and a "Restart" is required from the Parent AOI (CtrlGrp, MaGrp). **Intl** cannot be bridged (bypassed).
IntlG	BOOL	X	Inp	Coded above the AOI Instruction	Machine Protection Interlock 0=Stop/1=Enable - Required in any operation mode - May be overridden if Local operation enabled and Local Start activated by Input G or GX/GY	**IntlG** is the Fail Safe Machine Protection Interlock input signal. The **IntlG** signal is coded above the device AOI instance and is the result of a string of Machine Protection signals (i.e. drift and speed switches). The signal is intended to protect the machinery. If the device is in "Auto" mode, **IntlG** = 1 to allow device to run. If the **IntlG** signal transitions to 0 when the device is running in "Auto" mode, the device will fail and require a "Restart" from Parent AOI (CtrlGrp, MaGrp). If the device is in "Local" mode, the **IntlG** signal can be bridged by holding the devices "Local Start Pushbutton" (**G**, **GX**, or **GY**) as described above. To be able to use jog function, **GobalData.Par.DisableJogging** has be be set to 0.
DX/DY	BOOL	X	Out	Direct allocated to the AOI	Digital Output Contactor or Coil Direction X/Y	**DX/DY** is the Device Command to Start output signal (direction specific). **DX/DY** = 1 when device is commanded to start (run) in the X/Y direction respectively and signal is mapped to the device's contactor (motor) or coil (valve).
Alarm	BOOL		Out	Typically not used in application programming.	Device Alarm (Warning or Failure) Common signal Sta.W or Sta.F	**Alarm** = 1 Identifies that the devices has an active alarm present or that the device failed during supervised operation. Any of the following conditions can cause **Alarm**: **Sta.UA**, **Sta.KA**, **Sta.TA**, **Sta.SA**, **Sta.KA**, **Sta.XA**, **Sta.YA**, **Sta.TXA**, **Sta.ZYA**, **Sta.ZXA**, **Sta.CA.***, or "Device Failures". When the device is active (Startup, Waiting, Starting, or Running states), "Device Failures" are caused from the Interlock input (**Intl** & **IntlG)** or contactor feedback (No **R** with **D)** failures. All alarm conditions must clear with an alarm acknowledge before the **Alarm** bit will reset. *Note: Not all Status alarms are present for all modules and status alarms require the Internal Check= 1*

Run	BOOL		Out		Running in Any Mode All interlocks are checked and ok	**Run** = 1 Identifies that the device is running. When the device is starting, the **Run** signal will transition 0->1 only after the **D** and **R** signals are both 1 and the **Par.InpOutMonTime** is expired, and when the device is starting after If On, the Device is Running. The **Run** signal will transition 1->0 immediately after the device is stopped.
RdyAutoX/ RdyAutoY	BOOL		Out	Signal is typically used as the auto interlock for the EnAuto string of the next upstream device.	Ready Running/Positioned X/Y in Auto Mode Set if Run AND Par.ReadyTime Delay	**RdyAutoX/RdyAutoY** = 1 Identifies that the device is Running in "Auto" Mode in the respective direction (**X** or **Y**) and is Ready for auto operation (direction specific). When the device's **Run** signal transitions 0->1, delay **Par.ReadyTime** before transitioning the RdyAuto signal 0->1. The **RdyAuto** signal will transition 1->0 immediately after the device is stopped.
RdyAutoXY	BOOL		Out	Signal is typically used as the auto interlock for the next upstream device.	Ready Running/Positioned X or Y in Auto Mode Turns on when either RdyAutoX/RdyAutoY is on and delays to turn off for a precalculated time period depends on parameter settings.	**RdyAutoXY** = 1 Identifies that the device's **RdyAutoX** = 1 OR **RdyAutoY** = 1.

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT		I/O		Interface from/to Child Modules (Device)	**<Motor*>.Bus** is the calculated Bus from the Motor, Valve, or SubSys module. It is calculated from the input bus signal **<Parent>.Bus** and this modules AOI software. It is used to link child devices to this parent device. Example: Belt Conveyor Motor is this MotorN module and child devices include speed switches, drift switches, etc. The child devices would be linked to the Belt Conveyor Motor using this "Bus"
Cmd.0\SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Single Start)	**Cmd.0** = 1 is used to stop the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1\GR X,	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction X (Single Start X)	**Cmd.1** = 1 is used to start the device (**DX** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI. -Configure HMI Tag (TAGNAME) to identify logical direction meaning (i.e. X = to BC2)
Cmd.2\GRY	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction Y (Single Start Y)	**Cmd.2** = 1 is used to start the device (**DY** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI. Configure HMI Tag (TAGNAME) to identify logical direction meaning (i.e. Y = to BC3)

Cmd.3\ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the device and child device alarms from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4\EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle) ->Sta.REU	**Cmd.4** = 1 is used to toggle the device's "Local" mode status (**Sta.REU**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.5\EG	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Single Start (Toggle) ->Sta.REG	**Cmd.5** = 1 is used to toggle the device's "Single" mode status (**Sta.REG**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI
Sta.RPX/RPY	BOOL		Out		Running X/Y in Auto Mode (Production X/Y)	**Sta.RPX/RPY** = 1 Identifies that the device is starting or running in "Auto" Mode while being supervised by parent group. (i.e. **Internal Check** = 1). Signal is direction specific.
Sta.RUX/RUY	BOOL		Out		Running X/Y Local or Single Mode	**Sta.RUX/RUY** = 1 Identifies that the device is starting or running in "Local" or "Single" Mode while being supervised by parent group (i.e. **Internal Check** = 1). Signal is direction specific
Sta.RXM/RYM	BOOL		Out		Running Direction X/Y in Any Mode Set directly by Contactor Feedback Input RX/RY to indicate a possible blocked Contactor.	**Sta.RXM/RYM** = 1 Identifies that the device is running in the **X/Y** direction respectively. Set directly by the device's **RX/RY** input signal.
Sta.KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **U**, **K**, **T**, **S,** Contactor feedback (**R** with **D**). This signal should not be confused with **K** or **KAM**. Indicated in HMI as purple
Sta.REU	BOOL		Out		Local Operation Mode Enabled Machine Protection Interlock IntlG may be overridden if Local Start activated by Input G or GX/GY Note, Alarm Gongs are not set if Local is enabled.	**Sta.REU** = 1 Identifies that the device is in "Local" mode. Machine protection interlocks (**IntlG)** may be bridged as described in **G** signal description
Sta.REG	BOOL		Out		Single Start Operation Mode Enabled Device can be started from HMI by Cmd.GR/GRX/GRY Note, Alarm Gongs are not set if Single Start is enabled.	**Sta.REG** = 1 Identifies that the device is in "Single" mode. Machine protection interlocks (**IntlG**) may be bridged as described in **G** signal description
Sta.STU	BOOL		Out		Startup Warning Horn and FlashLight	**Sta.STU** = 1 Identifies that the device is in startup. This signal will be 1 during startup warning until the **Run** signal is 1.

Sta.WAI	BOOL		Out		Waiting with FlashLight	**Sta.WAI** = 1 Identifies that the device is waiting. This signal will be 1 during a device startup after the startup warning horn has stopped (**GlobalDat.Par.StartupHornTime**). The signal will transition 1->0 when either the device is starting or the device is stopped.
Sta.W	BOOL		Out		Warning Alarm (Common Signal) Set on Alarm if ErrorWarning=1	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.F	BOOL		Out		Failure Alarm (Common Signal) Set on Alarm if ErrorWarning=0	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.SA	BOOL		Out		Local Stop Alarm (Message) Set if module check active AND Input S=0	**Sta.SA** = 1 Identifies a Local Stop Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **S** signal input or **Cmd.0** from the HMI when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.UA	BOOL		Out		Local Isolator / Safety Switch Alarm (Message) Set if module check active AND Input U=0	**Sta.UA** = 1 Identifies a Local Isolator / Safety Switch Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **U** signal input when device's parent group is acitive (i.e. **Internal Check** = 1)
Sta.KA	BOOL		Out		Availability Alarm (Message) Set if module check active AND Input K=0	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.TA	BOOL		Out		Thermal Overload Alarm (Message) Set if module check active AND Input T=0	**Sta.TA** = 1 Identifies a Thermal Overload Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **T** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.RA	BOOL		Out		Contactor Feedback Alarm (Message) Set delayed if Output D/DX/DY AND its corresponding feedback Input R/RX/RY are not equal	**Sta.RA** = 1 Identifies a Contactor Feedback Alarm and reports a message to the HMI Alarm Log. The **R** signal input was not detected within **Par.InpOutMonTime** when **D** signal is active. Alternatively, the **R** signal may have been lost after the **Par.InpOutMonTime** has expired.
Sta.SAM	BOOL		Out		Local Stop Alarm Mimic Set if Input **S**=0	**Sta.SAM** = 1 Identifies that there is no **S** input signal or there is an active **Sta.SA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Local Stop problem.
Sta.UAM	BOOL		Out		Local Isolator / Safety Switch Alarm Mimic Set if Input **U=0**	**Sta.UAM** = 1 Identifies that there is no **U** input signal or there is an active **Sta.UA** alarm. This is used for HMI display (mimic) animiations to identify that there is a local isolator or safety switch problem.
Sta.KAM	BOOL		Out		Availability Alarm Mimic Set if Input **K**=0	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.

Sta.TAM	BOOL		Out		Thermal Overload Alarm Mimic Set if Input **T**=0	**Sta.TAM** = 1 Identifies that there is no **T** input signal or there is an active **Sta.TA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Thermal Overload problem.
Sta.RAM	BOOL		Out		Contactor Feedback Alarm Mimic Set delayed if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are not equal	**Sta.RAM** = 1 Identifies that there is a contactor failure (**R** when **D** after **Par.InpOutMonTime** expired) or there is an active **Sta.RA** alarm. This is used for HMI display (mimic) animiations to identify contactor feedback failures.
Sta.IDS	BOOL		Out		Safety Interlock Alarm Indication Mimic Set if Input Intl=0	**Sta.IDS** = 1 Identifies that there is no **Intl** input signal or there is an active **Intl** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Safety Interlock problem.
Sta.IDP	BOOL		Out		Maschine Protection Interlock Alarm Indication Mimic Set if Input IntlG=0	**Sta.IDP** = 1 Identifies that there is no **IntlG** input signal or there is an active **IntlG** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Machine Protection Interlock problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.RT	REAL		Out		Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the device. It is a totalize updated every 2 seconds.
Val.DCX/DCY	DINT		Out		Number of Starts Counter Direction X/Y	**Val.DCX/DCY** is a value identifying the total number of Starts for the device in that respective direction. The value increments 1 for every start.
Val.RST_RT	DINT		Out		Remaining seconds before restart is enabled	**Val.RST_RT** is a value identifying the remaining time in second before the device could be started again.
Par. HeavyStartup	BOOL		Inp		Code for Heavy Start 0=No/1=Yes If ON, all other devices are signalled during startup of this device by common tag **Global.HeavyStartup**	Set **Par.HeavyStartup** = 1 to specify that this device is a heavy starting device (i.e. hard starting with high inrush current). When this device is starting, other heavy startup devices in the same PLC are prevented from starting until this device's **RdyAuto** = 1. This is accomplished using the **Global.HeavyStartup** tag.
Par. HeavyStartupIgn	BOOL		Inp		Heavy Start Ignore 0=No/1=Yes If ON, this device ignores the startup of other heavy starting devices	Set **Par.HeavyStartupIgn** = 1 to specify that this device should start when commanded to start regardless of any other device's **Par.HeavyStartup** configuration and starting status. (I.E. Disable the effect of the **Global.HeavyStartup** tag)

Par. StartupWarningLocal	BOOL		Inp		Device with Startup Warning in Local Mode If ON, this device will set the Horn and Flash outputs specified by **Par.StartupHornCode** and **Par.StartupLightCode** when started locally	Set **Par.StartupWarningLocal** = 1 to specify that this device will always provide a startup warning horn / light before starting the device in "Local" mode. When set, a second pulse is required on the **G** input signal to start the device. Refer to **Par.StartupHornCode** and **Par.StartupLightCode** for specific horn/light configurations. Note that MSHA requires a start warning horn when starting conveyors regardless of mode of operation.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableSingle	BOOL		Inp		Suppress visibility of Singlestart mode Button at HMI template.	Set Par.DisableSingle =1 to specify that the "Manual/Auto"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableGrp Alarm	BOOL		Inp		Disable Alarm Indication to Parent Group modules If ON, the Warning or Failure Alarm Bus of the Parent Group module is not set. Alarm messages are however still created and **Sta.W** or **Sta.F** is set.	Set **Par.DisableGrpAlarm** = 1 to specify that the device Alarms (**Alarm**) will NOT be reported on the **<Parent>.Bus.18** (Failure) or **<Parent>.Bus.19** (Warning). (This parameter has No affect on Device Alarm reporting i.e. Sta tags directly between the device AOI and the HMI)
Par. DisableGrp Check	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** = 1 to specify that the device will not report restart requests to **<Parent>.Bus.17** (Restart Request), device Alarms (**Alarm)** to **<Parent>.Bus.18** (Failure), and inhibit the **<Parent>.Bus.26** (Check) from activating the Module's **Internal Check**. Use **Par.DisableGrpCheck** to prevent device failures from faulting the entire group. If **Par.DisableGrpCheck** = 1, the **Internal Check** will enable **Sta.W, Sta.F,** or other **Sta.** alarms. **Note: Module will not start in Automatic Mode (EnAuto) if this Parameter is set!**

Par. ErrorWarning	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by **Sta.W** IF ON, Alarms are indicated by **Sta.F**	Set **Par.ErrorWarning** = 1 to specify that device alarms (**Alarm**) will be reported as Device Warnings (**Sta.W**) and these device warnings will be reported as warnings to the **<Parent>.Bus.19**. Warnings will cause the device HMI animation object to be Yellow. Set **Par.DisableGrpCheck** = 0 to specify that device alarms (**Alarms**) will be reported as Device Failures (**Sta.F**) and these device failures will be reported as failures to the **<Parent>.Bus.18**. Failures will cause the device HMI animation object to be Red. Reporting to the Parent Bus is affected by **Par.DisableGrpAlarm** and **Par.DisableGrpCheck**.
Par. OverrideElmChk	BOOL		Inp		Override element check	Set **Par.OverrideElmChk** = 1 to specify that the device will report contactor feedback failure to control group even when control group is not active.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par. HasPermObj	BOOL		Inp		Has permissive object connected to Block	Set Par.HasPermObj = 1 to specify that a Permissive AOI is connected to this block and the permissive object on the HMI faceplate is visible to provide permissive diagnostic information.
Par. HasIntlkObj	BOOL		Inp		Has Interlock object connected to Block	Set Par.HasIntlkObj = 1 to specify that an Interlock AOI is connected to this block and the interlock object on the HMI faceplate is visible to provide interlock diagnostic information.
Par. AlarmGongCode	DINT of BOOLs		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or **.WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag

Par. StartupHorn Code	DINT of BOOLs		Inp		Enable Startup Warning Horn 0..31 Select Horn(s) by desired Bit Pattern. Sets the **Global.ComHornCode** at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the device's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when starting this device. Before this device starts, the **Par.StartupHornCode** is mapped to the **Global.ComHornCode** tag (Indirectly the SysGrp AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn** tag).
Par. StartupLight Code	DINT of BOOLs		Inp		Enable Startup Warning FlashLight 0..31 Select Flash(es) by desired Bit Pattern. Sets the **Global.ComLightCode** at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the device's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when starting this device. Before this device starts, the **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the **<SysGrp>.Light** tag).
Par. EmergencyPower	DINT		Inp	Configure only if there is Emergency power for this section of the plant	Priority for Emergency Power Operation (0..4). If set 0, the device is not stopped on Emergency, else if… **Global.EmergencyPower_1**=ON->stop Prio 1,2,3 & 4 **Global.EmergencyPower_2**=ON->stop Prio 2,3 & 4 **Global.EmergencyPower_3**=ON->stop Prio 3 & 4 **Global.EmergencyPower_4**=ON->stop Prio 4	Set **Par.EmergencyPower** to a value 0 - 4. This is a priority setting that is used to conserve energy when emergency power generators are on-line. (i.e. incoming power from utility is OFF). Higher settings mean lower priority for operations. - Set to 0: the device is not stopped - Set to 1: Stop device if **Global.EmergencyPower_1** = 1; - Set to 2: Stop device if **Global.EmergencyPower_1** or **_2** = 1; - Set to 3: Stop device if **Global.EmergencyPower_1, _2,** or **_3** = 1; - Set to 4: Stop device if **Global.EmergencyPower_1, _2, _3,** or **_4** = 1.
Par. InpOutMon Time	DINT	X	Inp	Default setting Varies by the desired system response	Contactor Feedback Supervision Timer (default 2000 ms) Delay time to check if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are NOT equal->**Sta.RA/RAM**	Set the **Par.InpOutMonTime** to a time delay in ms. This specifies the maximum allowable time between the start command (**D**) and feedback signal (**R**). If this time is exceeded before the **D** & **R** signals are equal, a contactor failure alarm will occur. Recommended default 2000 ms
Par. RestartTime	DINT		Inp		Time Delay for Restarting (ms) Time to hold Status Stopping, during this time the device cannot be restarted	Set the **Par.RestartTime** to a time delay in ms. This specifies the minumum time delay between a device stop and the device's next restart. After the device stops, it cannot be restarted until this time expires. It is used for 2 purposes. 1 - Allow motors to coast to a stop before restarting. (Not all motors can "catch" the rotor to restart 2 - Allow motor windings to cool before restarting

Par. ReadyTime	DINT	X	Inp		Time Delay to set Output Ready Auto (ms) Outputs **RdyAuto** or **RdyAutoX/Y** are set time delayed to allow the device to take up the load before the next device is started.	Set the **Par.ReadyTime** to a time delay in ms. This specifies the time delay between the device's **Run** output signal and **RdyAuto** output signal.
Par. ChangeTime	DINT		Inp		Change Time X to Y or vice versa (default 5000 ms) Time to wait in Status Stopping, during this time the device will not start in the reverse direction	Set the **Par.ChangeTime** to a time delay in ms. This specifies the minimum time delay between when the device can change directions (i.e. For a **DX** 1->0 transition followed by **DY** 0->1 transition. This is the minumum time between the falling **DX** signal and rising **DY** signal)

Parent Bus	**Data Type**	**User Config?**	**I/O**	**Configuration Required**	**Description**	**Notes**
ParentBus	DINT	X	Inp		Bus Input from/to Parent Module (Machine or Group)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This Motor* module reads information FROM the "Bus" and Writes status information TO the "Bus".
Reference Notes						
Note 1:						**Internal Check** is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: **<Parent>.Bus.26** with **DisableGrpCheck** = 0, **Sta.REU** (Local Mode), OR **Sta.REG** (Single Mode)

MotorR_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	x	Inp	Direct allocated to the MotorR AOI	Digital Input Contactor or Coil X	Input DX bit come from MotorR AOI. If DX is 1 then MotorR is started in X direction and simulation routine set RX bit to 1 after a delay time (contactor feedback).
DY	BOOL	x	Inp	Direct allocated to the MotorR AOI	Digital Input Contactor or Coil Y	Input DY bit come from MotorR AOI. If DY is 1 then MotorR is started in Y direction and simulation routine set RY bit to 1 after a delay time (contactor feedback).
GX_OUT	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Start X direction	GX bit can be set by button on HMI faceplate. Motor will be started in X direction if MotorR is in local mode.
GY_OUT	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Start Y direction	GY bit can be set by button on HMI faceplate. Motor will be started in Y direction if MotorR is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorR AOI	Available	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorR faceplate and Motor will be stopped.
RX	BOOL	x	Out	Direct allocated to the MotorR AOI	Contactor Feedback X	RX bit can be cleared by switch on HMI faceplate when DX =1. Contactor feedback alarm will be displayed on MotorR faceplate and Motor will be stopped.
RY	BOOL	x	Out	Direct allocated to the MotorR AOI	Contactor Feedback Y	RY bit can be cleared by switch on HMI faceplate when DY = 1. Contactor feedback alarm will be displayed on MotorR faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Disconnect Switch	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorR faceplate and Motor will be stopped.
TH	BOOL	x	Out	Direct allocated to the MotorR AOI	Thermal Overload	TH bit can be cleared by switch on HMI faceplate. Thermal overload alarm will be displayed on MotorR faceplate and Motor will be stopped.
S_OUT	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor X fault from HMI	DX_Fault bit is connected with HMI Contactor X switch. (0=Ok, 1=Fault)
DY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor Y fault from HMI	DY_Fault bit is connected with HMI Contactor Y switch. (0=Ok, 1=Fault)
GX_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start X from HMI	GX_IN bit is connected with HMI Local Start X button. 1=Start
GY_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start Y from HMI	GY_IN bit is connected with HMI Local Start Y button. 1=Start
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)
RX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor feedback RX fault from HMI	RX_Fault bit is connected with HMI Contactor Feedback X switch. (0=Ok, 1=Fault)
RY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor feedback RY fault from HMI	RY_Fault bit is connected with HMI Contactor Feedback Y switch. (0=Ok, 1=Fault)
S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
TH_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Overload fault from HMI	TH_Fault bit is connected with HMI Thermal overload switch (0=Ok, 1=Fault)
tmrFeedback_RX	TIMER				Contactor Feedback RX Timer	This timer is used to delay output **RX** after receiving **DX** from motor block
tmrFeedback_RY	TIMER				Contactor Feedback RY Timer	This timer is used to delay output **RY** after receiving **DY** from motor block
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

MotorRE3p_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	x	Inp	Direct allocated to the MotorR AOI	Digital Input Contactor or Coil X	Input DX bit come from MotorD AOI. If DX is 1 then MotorR is started in X direction and simulation routine set MotorCurent bit to 1 after a delay time.
DY	BOOL	x	Inp	Direct allocated to the MotorR AOI	Digital Input Contactor or Coil Y	Input DY bit come from MotorD AOI. If DY is 1 then MotorR is started in Y direction and simulation routine set MotorCurent bit to 1 after a delay time.
GX	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Start X direction	GX bit can be set by button on HMI faceplate. Motor will be started in X direction if MotorR is in local mode.
GY	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Start Y direction	GY bit can be set by button on HMI faceplate. Motor will be started in Y direction if MotorR is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorR AOI	Available	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorR faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Disconnect Switch	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorR faceplate and Motor will be stopped.
S_OUT	BOOL	x	Out	Direct allocated to the MotorR AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.
E3p_Data_in	E3_Inp	x	In/Out	Direct allocated to the E3p module	DeviceNet E3-Node Input structure	DeviceNet E3 Node Input Structure
Node_Failure	BOOL	x	Out	Direct allocated to the User created Node Failure tag	E3p Node failure	This input has to be correctly mapped to allow generation of E3p module communication error.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
GX_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start X direction from HMI	GX_IN bit is connected with HMI Local Start X button. 1=Start
GY_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start Y direction from HMI	GY_IN bit is connected with HMI Local Start Y button. 1=Start
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)
Node_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	E3pNode failure from HMI	Node_Fault bit is connected with HMI E3p Node switch. (0=Ok, 1=Fault)
Ground_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Ground Fault from HMI	Ground_Fault bit is connected with HMI Ground switch.(0=Ok, 1=Fault). Ground Fault alarm will be generated and E3p trip status will be set to 1.
Phase_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Phase Loss fault from HMI	Phase_Fault bit is connected with HMI Phase switch.(0=Ok, 1=Fault). Phase Loss alarm will be generated and E3p trip status will be set to 1.
RX	BOOL				Running X	Motor is Running in X direction.
RY	BOOL				Running Y	Motor is Running in Y direction.
S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
ThermalUtilized	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Utilized	Thermal utilized variable is connected to HMI Thermal utilized slider. Setting Thermal utilized value to 85-99% will generate Overload warning. Setting this value to 100% will generate Overload alarm and E3p trip status will be set to 1.
tmrFeedback_RX	TIMER				Timer to simulate E3p running in X direction	This timer is used to delay E3p MotorCurrent output after receiving DX from motor block
tmrFeedback_RY	TIMER				Timer to simulate E3p running in Y direction	This timer is used to delay E3p MotorCurrent output after receiving DY from motor block
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

MotorD

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EnAutoStart	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Start (normally mastered by CtrlGrp) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAuto=1 only	The **EnAutoStart** is an "auto start permissive". The **EnAutoStart** is typically but not necessarily mastered by the **CtrlGrp** or **MaGrp.** To initially start the motor both the **EnAutoStart** AND **EnAuto** must be = 1. Only **EnAuto** = 1 is required to keep the motor running. The **EnAutoStart** 0 to 1 transition will restart the motor if **EnAuto** =1. A rung of logic must be setup outside of the AOI to set the **EnAutoStart.** Using the **CtrlGrp.EnAutoStart** output as the input condition, the motor **EnAutoStart** is transitioned from 0 to 1 on a "RESTART" condition. Application Conditions (Permissives) required to Start Motor but not required to keep motor running can be used to set this bit. If the group of equipment is running in auto, setting the **EnAutoStart** to 0 does not stop the equipment while the **EnAuto** is still maintained. **EnAutoStart** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual_V111, chapter 7*
EnAutoX /EnAutoY	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Operation X/Y (normally mastered by CtrlGrp and interlocked by application) - Start requires EnAutoStart=1 AND EnAutoX/Y=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAutoX/Y=1only	Similar to the EnAuto in operation but controls which direction the Motor/Valve will run/operate (X or Y). Motor/Valve will not operate if both the **EnAutoX** and **EnAutoY** are set = 1simultaneously.
S	BOOL	X	Inp	Direct allocated to the AOI	Local Stop Input (0=Stop)	**S** is the Fail-Safe Local (Field) Stop PB input signal. **S** = 1 for the Device to start in any mode. If **S** = 0, the Device will always Stop.

GX/GY	BOOL		Inp	Direct allocated to the AOI	Local Start Direction X/Y (Go X/Y) Overrides Machine Protection Interlock IntlG if Local operation enabled	Local (Field) Start PB input signal (Direction specific). Device must be in "Local" mode to enable the use of the **GX/GY** signals to start the device. - If the device is in "Local" mode, the **GX/GY** 0->1 transition will initiate an alarm acknowledge for this device and all it's bus child devices. GX/GY PB could also be used to test flashlights while device is running if **GlobalData.Par.EnLampTest** = 1. - If the **GX/GY** signal is maintained during "Local" mode (i.e. holding the local start PB), then the Machine Interlocks (**IntlG**) are bridged (i.e. bypassed or jumpered) if **GobalData.Par.DisableJogging** is set to 0. - If the device is configured to have **Local warning**, (**Par.StartupWarningLocal =1**) then the required sequence to start the device in "Local" mode is as follows: The Local Start PB is momentarily pressed providing a **GX/GY** = 1 signal; the start warning horn will be provided; within X seconds of the start warning horn completion, the Local Start PB is pressed again and the device will immediately start. X represents the time value in **Global.Par.StartupCancelTime.** If **Global.Par.DisableTwoLocalStart** is set to one, then only one press of GX/GY is required to start the device. Function same as **G. GX** start for X direction(open) / **GY** start for Y direction(close). Device Input for Local (Field) Start PB for both directions Should not set GX and GY to 1 at the same time.
U	BOOL	X	Inp	Direct allocated to the AOI	Local Isolator / Safety Switch (0=OFF)	**U** is the Fail Safe Local Isolator Position input signal (i.e. field disconnect switch). **U** = 1 for Device to Start in any mode. If the **U** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
K	BOOL	X	Inp	Direct allocated to the AOI	Ok Available 1=Ok, MCC Unit	**K** is the Fail Safe Device OK status input signal (typically from 480V or 120V breaker status). **K** = 1 for device to Start. If the **K** signal transitions to 0 when the device is running, device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
T	BOOL	X	Inp	Direct allocated to the AOI	Thermal Overload Ok 0=Thermal	**T** is the Fail Safe Device Thermal Overload input signal. **T** = 1 for device to start. If the **T** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)

RX/RY	BOOL	X	Inp	Direct allocated to the AOI	Running Contactor Feedback X/Y. 0=OFF	**RX/RY** is the Device Running Status input signal direction specific (i.e. contactor position). After the device is commanded to run (**DX/DY** = 1), the **RX/RY** signal must transition to 1 before **Par.InpOutMonTime** expires or the device will Fail (message on HMI). If the **RX/RY** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (**CtrlGrp**, **MaGrp,** etc.)
TX/TY	BOOL		Inp	Use Log_1 if device has no **TX/TY** IO signal.	Torque Switch X=Open / Y=Close 0=Torque Alarm that stop the device, direction dependent	**TX/TY** are Fail-Safe Device Torque Switch input signals used to indicate Maximum Torque has been exceeded (direction specific). The signal is intended to prevent mechanical machinery damage. **TX/TY** = 1 for the device to start/run in that direction (**DX/DY** respectively). If the **TX** signal transitions 1->0 during a **RX** running condition, or if the **TY** signal transitions 1->0 during a **RY** running condition the device will fail.
X/Y	BOOL		Inp	Use Log_1 if device has no **X/Y** IO signal.	Internal Limit X=Open/Y=Closed. 0=Positioned that stop the device, direction dependent	**X/Y** are Fail-Safe Device Internal Limit Switch input signals used to indicate physical position (direction specific). The signal is intended to prevent mechanical machinery damage. **X/Y** = 1 for the device to start/run in that direction (**DX/DY** respectively). If the X signal transitions 1->0 during a **RX** running condition, or if the **Y** signal transitions 1->0 during a **RY** running condition the device will fail.
ZX/ZY	BOOL	X	Inp	Direct allocated to the AOI	Position X/Y Limit Switch. 1=Positioned that stop the device, direction dependent	**ZX/ZY** are External Limit Switch input signals used to indicate physical position for maximum travel (direction specific). Signal is used to Stop device after desired position is achieved. **ZX/ZY** = 0 to start/run in that direction (**DX/DY** respectively) **ZX/ZY** = 1 to stop traveling in the respective direction (**DX/DY** respectively)
Intl	BOOL	X	Inp	Coded above the AOI Instruction	Safety Interlock 0=Stop/1=Enable. - Required in any operation mode	**Intl** is a Fail Safe Safey Interlock input signal. The **Intl** signal is coded above the device AOI instance and is the result of a string of Safety device signals (i.e. rope switches) and is intended for safety related interlocks. **Intl** = 1 to run device. If the **Intl** signal transitions to 0 when the device is running, the device will Fail and a "Restart" is required from the Parent CtrlGrp. **Intl** cannot be bridged (bypassed).
IntlG	BOOL	X	Inp	Coded above the AOI Instruction	Machine Protection Interlock 0=Stop/1=Enable - Required in any operation mode - May be overridden if Local operation enabled and Local Start activated by Input GX/GY	**IntlG** is the Fail Safe Machine Protection Interlock input signal. The **IntlG** signal is coded above the device AOI instance and is the result of a string of Machine Protection signals (i.e. drift and speed switches). The signal is intended to protect the machinery. If the device is in "Local" mode, the **IntlG** signal can be bridged by holding the devices "Local Start Pushbutton" (**G, GX, or GY**) to run device. To be able to use jog function, **GobalData.Par.DisableJogging** has be be set to 0.

DX/DY	BOOL	X	Out	Direct allocated to the AOI	Digital Output Contactor or Coil Direction X/Y	**DX/DY** is the Device Command to Start output signal (direction specific). **DX/DY** = 1 when device is commanded to start (run) in the X/Y direction respectively and signal is mapped to the device's contactor (motor) or coil (valve).
Alarm	BOOL		Out	Typically not used in application programming.	Device Alarm (Warning or Failure) Common signal Sta.W or Sta.F	**Alarm** = 1 Identifies that the devices has an active alarm present or that the device failed during supervised operation. Any of the following conditions can cause **Alarm**: **Sta.UA**, **Sta.KA**, **Sta.TA**, **Sta.SA**, **Sta.KA**, **Sta.XA**, **Sta.YA**, **Sta.TXA**, **Sta.ZYA**, **Sta.ZXA**, **Sta.CA.***, or "Device Failures". When the device is active (Startup, Waiting, Starting, or Running states), "Device Failures" are caused from the Interlock input (**Intl** & **IntlG)** or contactor feedback (No **R** with **D)** failures. All alarm conditions must clear with an alarm acknowledge before the **Alarm** bit will reset. *Note: Not all Status alarms are present for all modules and status alarms require the Internal Check= 1*
Run	BOOL		Out		Running in Any Mode All interlocks are checked and ok	**Run** = 1 Identifies that the device is running. When the device is starting, the **Run** signal will transition 0->1 only after the **D** and **R** signals are both 1 and the **Par.InpOutMonTime** is expired, and when the device is starting after If On, the Device is Running. The **Run** signal will transition 1->0 immediately after the device is stopped.
RdyAutoX/ RdyAutoY	BOOL		Out	Signal is typically used as the auto interlock for the EnAuto string of the next upstream device.	Ready Running X/Y in Auto Mode Set if Run to ZX/ZY position AND Par.ReadyTime Delay	**RdyAutoX/RdyAutoY** = 1 Identifies that the device is Running in "Auto" Mode in the respective direction (**X** or **Y**) and is Ready for auto operation (direction specific). When the device's **Run** signal transitions 0->1, and ZX/ZY is reached, delay **Par.ReadyTime** before transitioning the RdyAuto signal 0->1. The **RdyAuto** signal will transition 1->0 immediately after the device is stopped or when there is a device failure. After a device failure occurrence, the RdyAuto signal will not be released after fault is cleared until a control group restart is issued.
RdyAutoXY	BOOL		Out	Signal is typically used as the auto interlock for the next upstream device.	Ready Running/Positioned X or Y in Auto Mode Turns on when either RdyAutoX/RdyAutoY is on and delays to turn off for a precalculated time period depends on parameter settings.	**RdyAutoXY** = 1 Identifies that the device's **RdyAutoX** = 1 OR **RdyAutoY** = 1. Signal is typically used as the auto interlock for the EnAuto string of the next upstream device if doesn't matter which direction the module running.

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT		I/O		Interface from/to Child Modules (Device)	**<Motor*>.Bus** is the calculated Bus from the Motor, Valve, or SubSys module. It is calculated from the input bus signal **<Parent>.Bus** and this modules AOI software. It is used to link child devices to this parent device. Example: Belt Conveyor Motor is this MotorN module and child devices include speed switches, drift switches, etc. The child devices would be linked to the Belt Conveyor Motor using this "Bus"
Cmd.0 \SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Single Start)	**Cmd.0** = 1 is used to stop the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1 \GRX,	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction X (Single Start X)	**Cmd.1** = 1 is used to start the device (**DX** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI. -Configure HMI Tag (TAGNAME) to identify logical direction meaning (i.e. X = to BC2)
Cmd.2 \GRY	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction Y (Single Start Y)	**Cmd.2** = 1 is used to start the device (**DY** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI. Configure HMI Tag (TAGNAME) to identify logical direction meaning (i.e. Y = to BC3)
Cmd.3 \ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the device and child device alarms from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4 \EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle) ->Sta.REU	**Cmd.4** = 1 is used to toggle the device's "Local" mode status (**Sta.REU**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.5 \EG	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Single Start (Toggle) ->Sta.REG	**Cmd.5** = 1 is used to toggle the device's "Single" mode status (**Sta.REG**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI
Sta.RPX/RPY	BOOL		Out		Running/Positioned X/Y in Auto Mode (Production X/Y)	**Sta.RPX/RPY** = 1 Identifies that the device is starting or running in "Auto" Mode while being supervised by parent group. (i.e. **Internal Check** = 1). Signal is direction specific.
Sta.RUX/RUY	BOOL		Out		Running X/Y Local or Single Mode	**Sta.RUX/RUY** = 1 Identifies that the device is starting or running in "Local" or "Single" Mode while being supervised by parent group (i.e. **Internal Check** = 1). Signal is direction specific
Sta.RXM/RYM	BOOL		Out		Running Direction X/Y in Any Mode Set directly by Contactor Feedback Input RX/RY to indicate a possible blocked Contactor.	**Sta.RXM/RYM** = 1 Identifies that the device is running in the **X/Y** direction respectively. Set directly by the device's **RX/RY** input signal.

Sta.KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **U**, **K**, **T**, **S,** Contactor feedback (**R** with **D**). This signal should not be confused with **K** or **KAM**. Indicated in HMI as purple
Sta.REU	BOOL		Out		Local Operation Mode Enabled Machine Protection Interlock IntlG may be overridden if Local Start activated by Input G or GX/GY Note, Alarm Gongs are not set if Local is enabled.	**Sta.REU** = 1 Identifies that the device is in "Local" mode. Machine protection interlocks (**IntlG)** may be bridged as described in **G** signal description
Sta.REG	BOOL		Out		Single Start Operation Mode Enabled Device can be started from HMI by Cmd.GR/GRX/GRY Note, Alarm Gongs are not set if Single Start is enabled.	**Sta.REG** = 1 Identifies that the device is in "Single" mode. Machine protection interlocks (**IntlG**) may be bridged as described in **G** signal description
Sta.STU	BOOL		Out		Startup Warning Horn and FlashLight	**Sta.STU** = 1 Identifies that the device is in startup. This signal will be 1 during startup warning until the **Run** signal is 1.
Sta.WAI	BOOL		Out		Waiting with FlashLight	**Sta.WAI** = 1 Identifies that the device is waiting. This signal will be 1 during a device startup after the startup warning horn has stopped (**GlobalDat.Par.StartupHornTime**). The signal will transition 1->0 when either the device is starting or the device is stopped.
Sta.W	BOOL		Out		Warning Alarm (Common Signal) Set on Alarm if ErrorWarning=1	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.F	BOOL		Out		Failure Alarm (Common Signal) Set on Alarm if ErrorWarning=0	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.SA	BOOL		Out		Local Stop Alarm (Message) Set if module check active AND Input S=0	**Sta.SA** = 1 Identifies a Local Stop Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **S** signal input or **Cmd.0** from the HMI when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.UA	BOOL		Out		Local Isolator / Safety Switch Alarm (Message) Set if module check active AND Input U=0	**Sta.UA** = 1 Identifies a Local Isolator / Safety Switch Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **U** signal input when device's parent group is acitive (i.e. **Internal Check** = 1)
Sta.KA	BOOL		Out		Availability Alarm (Message) Set if module check active AND Input K=0	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)

Sta.TA	BOOL		Out		Thermal Overload Alarm (Message) Set if module check active AND Input T=0	**Sta.TA** = 1 Identifies a Thermal Overload Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **T** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.RA	BOOL		Out		Contactor Feedback Alarm (Message) Set delayed if Output D/DX/DY AND its corresponding feedback Input R/RX/RY are not equal	**Sta.RA** = 1 Identifies a Contactor Feedback Alarm and reports a message to the HMI Alarm Log. The **R** signal input was not detected within **Par.InpOutMonTime** when **D** signal is active. Alternatively, the **R** signal may have been lost after the **Par.InpOutMonTime** has expired.
Sta.TXA/TYA	BOOL		Out		Torque Switch X=Open/Y=Close Alarm (Message)	**Sta.TXA/TYA** = 1 Identifies that the **TX/TY** signal respectivley was lost while running in the **X/Y** direction and being supervised by parent group (i.e. **Internal Check** = 1)
Sta.XA/YA	BOOL		Out		Internal Position X=NotOpen/Y=Not Closed Alarm (Message)	**Sta.XA/YA** = 1 Identifies that the **X/Y** signal respectivley was lost while running in the **X/Y** direction and being supervised by parent group (i.e. **Internal Check** = 1)
Sta.ZXA/ZYA	BOOL		Out		External Position X/Y Alarm (Message)	**Sta.ZXA/ZYA** = 1 Identifies that there is a device position alarm. Three possible conditions for each direction: 1. Device commanded to **X/Y** position and **Par.PosTimeX/Par.PosTimeY** expired before reaching the **ZX/ZY** limit or 2. **ZX** = 1 and **ZY** = 1 at the same time or 3. **Par.DisablePosAutoCorrect** = 1 and ZX/ZY limit switch input is lost (=0) after RdyAutoX/RdyAutoY has already been released.
Sta.SAM	BOOL		Out		Local Stop Alarm Mimic Set if Input **S**=0	**Sta.SAM** = 1 Identifies that there is no **S** input signal or there is an active **Sta.SA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Local Stop problem.
Sta.UAM	BOOL		Out		Local Isolator / Safety Switch Alarm Mimic Set if Input **U=0**	**Sta.UAM** = 1 Identifies that there is no **U** input signal or there is an active **Sta.UA** alarm. This is used for HMI display (mimic) animiations to identify that there is a local isolator or safety switch problem.
Sta.KAM	BOOL		Out		Availability Alarm Mimic Set if Input **K**=0	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.
Sta.TAM	BOOL		Out		Thermal Overload Alarm Mimic Set if Input **T**=0	**Sta.TAM** = 1 Identifies that there is no **T** input signal or there is an active **Sta.TA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Thermal Overload problem.

Sta.RAM	BOOL		Out		Contactor Feedback Alarm Mimic Set delayed if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are not equal	**Sta.RAM** = 1 Identifies that there is a contactor failure (**R** when **D** after **Par.InpOutMonTime** expired) or there is an active **Sta.RA** alarm. This is used for HMI display (mimic) animiations to identify contactor feedback failures.
Sta.TXAM /TYAM	BOOL		Out		Torque Switch **X**=Open/**Y**=Close Alarm Mimic	**Sta.TXAM/TYAM** = 1 Identifies that there is a torque limit failure or there is an active **Sta.TXA/TYA** alarm. This is used for HMI display (mimic) animations to identify that there is a torque problem.
Sta.XAM /YAM	BOOL		Out		Internal Position X=NotOpen/Y=Not Closed Alarm Mimic	**Sta.XAM/YAM** = 1 Identifies that there is an internal position limit failure or there is an active **Sta.XA/YA** alarm. This is used for HMI display (mimic) animations to identify that there is an internal limit switch problem.
Sta.ZXAM /ZYAM	BOOL		Out		External Position X/Y Alarm Mimic	**Sta.ZXAM/ZYAM** = 1 Identifies that there is an external device position limit failure or there is an active **Sta.ZXA/ZYA** alarm. This is used for HMI display (mimic) animations to identify that there is an external limit switch problem.
Sta.ZX/ZY	BOOL		Out		External Position X/Y Indication Mimic	**Sta.ZX/ZY** = 1 Identifies that the device is positioned at that position. This is used for HMI display (mimic) animations to identify the position.
Sta.IDS	BOOL		Out		Safety Interlock Alarm Indication Mimic Set if Input Intl=0	**Sta.IDS** = 1 Identifies that there is no **Intl** input signal or there is an active **Intl** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Safety Interlock problem.
Sta.IDP	BOOL		Out		Maschine Protection Interlock Alarm Indication Mimic Set if Input IntlG=0	**Sta.IDP** = 1 Identifies that there is no **IntlG** input signal or there is an active **IntlG** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Machine Protection Interlock problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.RT	REAL		Out		Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the device. It is a totalize updated every 2 seconds.
Val.DCX /DCY	DINT		Out		Number of Starts Counter Direction X/Y	**Val.DCX/DCY** is a value identifying the total number of Starts for the device in that respective direction. The value increments 1 for every start.
Val.RST_RT	DINT		Out		Remaining seconds before restart is enabled	**Val.RST_RT** is a value identifying the remaining time in second before the device could be started again.

Par. HeavyStartup	BOOL		Inp		Code for Heavy Start 0=No/1=Yes If ON, all other devices are signalled during startup of this device by common tag **Global.HeavyStartup**	Set **Par.HeavyStartup** = 1 to specify that this device is a heavy starting device (i.e. hard starting with high inrush current). When this device is starting, other heavy startup devices in the same PLC are prevented from starting until this device's **RdyAuto** = 1. This is accomplished using the **Global.HeavyStartup** tag.
Par. HeavyStartupIgn	BOOL		Inp		Heavy Start Ignore 0=No/1=Yes If ON, this device ignores the startup of other heavy starting devices	Set **Par.HeavyStartupIgn** = 1 to specify that this device should start when commanded to start regardless of any other device's **Par.HeavyStartup** configuration and starting status. (I.E. Disable the effect of the **Global.HeavyStartup** tag)
Par.SafetyPosition	BOOL		Inp		Positioning device is moved to Safety Position **ZY** (closed) if **EnAutoY**=ON regardless of **EnAutoStart**)	Set **Par.SafetyPosition** = 1 to allow the flap/gate to change to position **Y** when an immediate stop (**<Parent>.Bus.25**) occurs. **Y** is ALWAYS considered the Safe position. The gate movement to the safe position must be controlled from the PLC application program after teh immediate stop (i.e. Set **EnAutoY** = 1 to change the flap/gate position without the **EnAutoStart** signal)
Par. StartupWarningLocal	BOOL		Inp		Device with Startup Warning in Local Mode If ON, this device will set the Horn and Flash outputs specified by **Par.StartupHornCode** and **Par.StartupLightCode** when started locally	Set **Par.StartupWarningLocal** = 1 to specify that this device will always provide a startup warning horn / light before starting the device in "Local" mode. When set, a second pulse is required on the **G** input signal to start the device. Refer to **Par.StartupHornCode** and **Par.StartupLightCode** for specific horn/light configurations. Note that MSHA requires a start warning horn when starting conveyors regardless of mode of operation.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableSingle	BOOL		Inp		Suppress visibility of Singlestart mode Button at HMI template.	Set Par.DisableSingle =1 to specify that the "Manual/Auto"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableGrpAlarm	BOOL		Inp		Disable Alarm Indication to Parent Group modules If ON, the Warning or Failure Alarm Bus of the Parent Group module is not set. Alarm messages are however still created and **Sta.W** or **Sta.F** is set.	Set **Par.DisableGrpAlarm** = 1 to specify that the device Alarms (**Alarm**) will NOT be reported on the **<Parent>.Bus.18** (Failure) or **<Parent>.Bus.19** (Warning). (This parameter has No affect on Device Alarm reporting i.e. Sta tags directly between the device AOI and the HMI)

Par. DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** = 1 to specify that the device will not report restart requests to **<Parent>.Bus.17** (Restart Request), device Alarms (**Alarm)** to **<Parent>.Bus.18** (Failure), and inhibit the **<Parent>.Bus.26** (Check) from activating the Module's **Internal Check**. Use **Par.DisableGrpCheck** to prevent device failures from faulting the entire group. If **Par.DisableGrpCheck** = 1, the **Internal Check** will enable **Sta.W, Sta.F,** or other **Sta.** alarms. **Note: Module will not start in Automatic Mode (EnAuto) if this Parameter is set!**
Par. ErrorWarning	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by **Sta.W** IF ON, Alarms are indicated by **Sta.F**	Set **Par.ErrorWarning** = 1 to specify that device alarms (**Alarm**) will be reported as Device Warnings (**Sta.W**) and these device warnings will be reported as warnings to the **<Parent>.Bus.19**. Warnings will cause the device HMI animation object to be Yellow. Set **Par.DisableGrpCheck** = 0 to specify that device alarms (**Alarms**) will be reported as Device Failures (**Sta.F**) and these device failures will be reported as failures to the **<Parent>.Bus.18**. Failures will cause the device HMI animation object to be Red. Reporting to the Parent Bus is affected by **Par.DisableGrpAlarm** and **Par.DisableGrpCheck**.
Par. OverrideElmChk	BOOL		Inp		Override element check	Set **Par.OverrideElmChk** = 1 to specify that the device will report contactor feedback failure to control group even when control group is not active.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par. DisablePosAutoCorrect	BOOL		Inp		Disable Position Auto Correct behaviour when ZX/ZY is lost after RdyAutoX/RdyAutoY has been released	Set **Par.DisablePosAutoCorrect** = 1 to specify that the device will not automatically correct its position and announce a Extern Position alarm right away. By default (Par.DisablePosAutoCorrect = 0) the device will try to correct its position failure by starting automatically to reach the lost position ZX/ZY within position supervision time (Par.PosTimeX/Par.PosTimerY).
Par. HasPermObj	BOOL		Inp		Has permissive object connected to Block	Set Par.HasPermObj = 1 to specify that a Permissive AOI is connected to this block and the permissive object on the HMI faceplate is visible to provide permissive diagnostic information.
Par. HasIntlkObj	BOOL		Inp		Has Interlock object connected to Block	Set Par.HasIntlkObj = 1 to specify that an Interlock AOI is connected to this block and the interlock object on the HMI faceplate is visible to provide interlock diagnostic information.

Par. AlarmGong Code	DINT of BOOLs		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Par. StartupHorn Code	DINT of BOOLs		Inp		Enable Startup Warning Horn 0..31 Select Horn(s) by desired Bit Pattern. Sets the **Global.ComHornCode** at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the device's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when starting this device. Before this device starts, the **Par.StartupHornCode** is mapped to the **Global.ComHornCode** tag (Indirectly the SysGrp AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn** tag).
Par. StartupLight Code	DINT of BOOLs		Inp		Enable Startup Warning FlashLight 0..31 Select Flash(es) by desired Bit Pattern. Sets the **Global.ComLightCode** at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the device's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when starting this device. Before this device starts, the **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the **<SysGrp>.Light** tag).
Par. EmergencyPower	DINT		Inp	Configure only if there is Emergency power for this section of the plant	Priority for Emergency Power Operation (0..4). If set 0, the device is not stopped on Emergency, else if… **Global.EmergencyPower_1**=ON->stop Prio 1,2,3 & 4 **Global.EmergencyPower_2**=ON->stop Prio 2,3 & 4 **Global.EmergencyPower_3**=ON->stop Prio 3 & 4 **Global.EmergencyPower_4**=ON->stop Prio 4	Set **Par.EmergencyPower** to a value 0 - 4. This is a priority setting that is used to conserve energy when emergency power generators are on-line. (i.e. incoming power from utility is OFF). Higher settings mean lower priority for operations. - Set to 0: the device is not stopped - Set to 1: Stop device if **Global.EmergencyPower_1** = 1; - Set to 2: Stop device if **Global.EmergencyPower_1** or **_2** = 1; - Set to 3: Stop device if **Global.EmergencyPower_1, _2,** or **_3** = 1; - Set to 4: Stop device if **Global.EmergencyPower_1, _2, _3,** or **_4** = 1.

Par. InpOutMon Time	DINT	X	Inp	Default setting Varies by the desired system response	Contactor Feedback Supervision Timer (default 2000 ms) Delay time to check if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are NOT equal->**Sta.RA/RAM**	Set the **Par.InpOutMonTime** to a time delay in ms. This specifies the maximum allowable time between the start command (**D**) and feedback signal (**R**). If this time is exceeded before the **D** & **R** signals are equal, a contactor failure alarm will occur. Recommended default 2000 ms
Par. RestartTime	DINT		Inp		Time Delay for Restarting (ms) Time to hold Status Stopping, during this time the device cannot be restarted	Set the **Par.RestartTime** to a time delay in ms. This specifies the minumum time delay between a device stop and the device's next restart. After the device stops, it cannot be restarted until this time expires. It is used for 2 purposes. 1 - Allow motors to coast to a stop before restarting. (Not all motors can "catch" the rotor to restart 2 - Allow motor windings to cool before restarting
Par. ReadyTime	DINT	X	Inp		Time Delay to set Output Ready Auto (ms) Outputs **RdyAuto** or **RdyAutoX/Y** are set time delayed to allow the device to take up the load before the next device is started.	Set the **Par.ReadyTime** to a time delay in ms. This specifies the time delay between the device's **Run** output signal and **RdyAuto** output signal.
Par. ChangeTime	DINT		Inp		Change Time X to Y or vice versa (default 5000 ms) Time to wait in Status Stopping, during this time the device will not start in the reverse direction	Set the **Par.ChangeTime** to a time delay in ms. This specifies the minimum time delay between when the device can change directions (i.e. For a **DX** 1->0 transition followed by **DY** 0->1 transition. This is the minumum time between the falling **DX** signal and rising **DY** signal)
Par. PosTimeX/ PosTimeY	DINT	X	Inp		Positioning Time Supervision X/Y (ms)	Set the **Par.PosTimeX/PosTimeY** to a time delay in ms. This specifies the maximum time delay between when a device is commanded to the **X/Y** position (with **DX/DY** commands) and achieves the **ZX/ZY** position. If the time expires before achieve the position, a **Sta.ZXA** or **Sta.ZYA** alarm will occur.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	Inp		Bus Input from/to Parent Module (Machine or Group)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This Motor* module reads information FROM the "Bus" and Writes status information TO the "Bus".

Reference Notes						
Note 1:						**Internal Check** is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: **<Parent>.Bus.26** with **DisableGrpCheck** = 0, **Sta.REU** (Local Mode), OR **Sta.REG** (Single Mode)

MotorD_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	x	Inp	Direct allocated to the MotorD AOI	Digital Input Contactor or Coil X	Input DX bit come from MotorD AOI. If DX is 1 then MotorD is started in X direction and simulation routine set RX bit to 1 after a delay time (contactor feedback).
DY	BOOL	x	Inp	Direct allocated to the MotorD AOI	Digital Input Contactor or Coil Y	Input DY bit come from MotorD AOI. If DY is 1 then MotorD is started in Y direction and simulation routine set RY bit to 1 after a delay time (contactor feedback).
GX_OUT	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Start X direction	GX bit can be set by button on HMI faceplate. Motor will be started in X direction if MotorD is in local mode.
GY_OUT	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Start Y direction	GY bit can be set by button on HMI faceplate. Motor will be started in Y direction if MotorD is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorD AOI	Available	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorD faceplate and Motor will be stopped.
RX	BOOL	x	Out	Direct allocated to the MotorD AOI	Contactor Feedback X	RX bit can be cleared by switch on HMI faceplate when DX = 1. Contactor feedback alarm will be displayed on MotorD faceplate and Motor will be stopped.
RY	BOOL	x	Out	Direct allocated to the MotorD AOI	Contactor Feedback Y	RY bit can be cleared by switch on HMI faceplate when DY = 1. Contactor feedback alarm will be displayed on MotorD faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Disconnect switch	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorD faceplate and Motor will be stopped.
TX	BOOL	x	Out	Direct allocated to the MotorD AOI	Torque Switch X	TX bit can be cleared by switch on HMI faceplate. Internal torque switch X alarm message will be displayed on MotorD faceplate and Motor will be stopped.
TY	BOOL	x	Out	Direct allocated to the MotorD AOI	Torque Switch Y	TY bit can be cleared by switch on HMI faceplate. Internal torque switch Y alarm message will be displayed on MotorD faceplate and Motor will be stopped.
TH	BOOL	x	Out	Direct allocated to the MotorD AOI	Thermal Overload	TH bit can be cleared by switch on HMI faceplate. Thermal overload alarm will be displayed on MotorD faceplate and Motor will be stopped.
X	BOOL	x	Out	Direct allocated to the MotorD AOI	Internal Limit Switch X	X bit can be cleared by switch on HMI faceplate. Internal Limit Switch X alarm will be displayed on MotorD faceplate and Motor will be stopped.

Y	BOOL	x	Out	Direct allocated to the MotorD AOI	Internal Limit Switch Y	Y bit can be cleared by switch on HMI faceplate. Internal Limit Switch Y alarm will be displayed on MotorD faceplate and Motor will be stopped.
ZX	BOOL	x	Out	Direct allocated to the MotorD AOI	Position X external Limit Switch	ZX bit can be cleared by switch on HMI faceplate. External Limit Switch X alarm will be displayed on MotorD faceplate and Motor will be stopped.
ZY	BOOL	x	Out	Direct allocated to the MotorD AOI	Position Y external Limit Switch	ZY bit can be cleared by switch on HMI faceplate. External Limit Switch Y alarm will be displayed on MotorD faceplate and Motor will be stopped.
S_OUT	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor X fault from HMI	DX_Fault bit is connected with HMI Contactor switch. (0=Ok, 1=Fault)
DY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor Y fault from HMI	DY_Fault bit is connected with HMI Contactor switch. (0=Ok, 1=Fault)
GX_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start X from HMI	GX_IN bit is connected with HMI Local Start X button. 1=Start
GY_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start Y from HMI	GY_IN bit is connected with HMI Local Start Y button. 1=Start
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)
RX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor feedback RX fault from HMI	RX_Fault bit is connected with HMI Contactor Feedback X switch. (0=Ok, 1=Fault)
RY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Contactor feedback RY fault from HMI	RY_Fault bit is connected with HMI Contactor Feedback Y switch. (0=Ok, 1=Fault)

S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
SwitchX_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch X Feedback time	Time delay to set ZX bit after device is running in X direction
SwitchY_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch Y Feedback time	Time delay to set ZY bit after device is running in Y direction
TH_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Overload fault from HMI	TH_Fault bit is connected with HMI Thermal overload switch (0=Ok, 1=Fault)
tmrFeedback_RX	TIMER				Contactor Feedback RX Timer	This timer is used to delay output **RX** after receiving **DX** from motor block
tmrFeedback_RY	TIMER				Contactor Feedback RY Timer	This timer is used to delay output **RY** after receiving **DY** from motor block
tmrLimitSwitch_ZX	TIMER				External Limit Switch ZX Position Feedback Timer	This timer is used to delay output **ZX** after **RX** =1
tmrLimitSwitch_ZY	TIMER				External Limit Switch ZY Position Feedback Timer	This timer is used to delay output **ZY** after **RY** =1
TX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Torque Switch X Fault from HMI	TX_Fault bit is connected with HMI Torque Switch X switch. (0=Ok, 1=Fault)
TY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Torque Switch Y Fault from HMI	TY_Fault bit is connected with HMI Torque Switch Y switch. (0=Ok, 1=Fault)
X_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Internal Limit Switch X Fault from HMI	X_Fault bit is connected with HMI Internal Switch X switch. (0=Ok, 1=Fault)
Y_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Internal Limit Switch Y Fault from HMI	Y_Fault bit is connected with HMI Internal Switch Y switch. (0=Ok, 1=Fault)
ZX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch X Feedback Fault	ZX_Fault bit is connected with HMI External Limit Switch X switch. (0=Ok, 1=Fault)

ZY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch Y Feedback Fault	ZY_Fault bit is connected with HMI External Limit Switch Y switch. (0=Ok, 1=Fault)
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

MotorDE3p_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	x	Inp	Direct allocated to the MotorD AOI	Digital Input Contactor or Coil X	Input DX bit come from MotorD AOI. If DX is 1 then MotorD is started in X direction and simulation routine set MotorCurrent bit to 1 after a delay time.
DY	BOOL	x	Inp	Direct allocated to the MotorD AOI	Digital Input Contactor or Coil Y	Input DY bit come from MotorD AOI. If DY is 1 then MotorD is started in Y direction and simulation routine set MotorCurrent bit to 1 after a delay time.
GX	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Start X direction	GX bit can be set by button on HMI faceplate. Motor will be started in X direction if MotorD is in local mode.
GY	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Start Y direction	GY bit can be set by button on HMI faceplate. Motor will be started in Y direction if MotorD is in local mode.
K	BOOL	x	Out	Direct allocated to the MotorD AOI	Available	K bit can be cleared by switch on HMI faceplate. Not available alarm will be displayed on MotorD faceplate and Motor will be stopped.
U	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Disconnect Switch	U bit can be cleared by switch on HMI faceplate. Disconnect switch alarm will be displayed on MotorD faceplate and Motor will be stopped.
TX	BOOL	x	Out	Direct allocated to the MotorD AOI	Torque Switch X	TX bit can be cleared by switch on HMI faceplate. Internal torque switch X alarm message will be displayed on MotorD faceplate and Motor will be stopped.
TY	BOOL	x	Out	Direct allocated to the MotorD AOI	Torque Switch Y	TY bit can be cleared by switch on HMI faceplate. Internal torque switch Y alarm message will be displayed on MotorD faceplate and Motor will be stopped.
X	BOOL	x	Out	Direct allocated to the MotorD AOI	Internal Limit Switch X	X bit can be cleared by switch on HMI faceplate. Internal Limit Switch X alarm will be displayed on MotorD faceplate and Motor will be stopped.
Y	BOOL	x	Out	Direct allocated to the MotorD AOI	Internal Limit Switch Y	Y bit can be cleared by switch on HMI faceplate. Internal Limit Switch Y alarm will be displayed on MotorD faceplate and Motor will be stopped.
ZX	BOOL	x	Out	Direct allocated to the MotorD AOI	External Limit Switch X	ZX bit can be cleared by switch on HMI faceplate. External Limit Switch X alarm will be displayed on MotorD faceplate and Motor will be stopped.
ZY	BOOL	x	Out	Direct allocated to the MotorD AOI	External Limit Switch Y	ZY bit can be cleared by switch on HMI faceplate. External Limit Switch Y alarm will be displayed on MotorD faceplate and Motor will be stopped.
S_OUT	BOOL	x	Out	Direct allocated to the MotorD AOI	Local Stop Output (0=Stop)	S bit can be cleared by button on HMI faceplate. Motor will be stopped when set to 0.
E3p_Data_in	E3_Inp	x	In/Out	Direct allocated to the E3p module	DeviceNet E3-Node Input structure	DeviceNet E3 Node Input Structure

Node_Failure	BOOL	x	Out	Direct allocated to the User created Node Failure tag	E3p Node failure	This input has to be correctly mapped to allow generation of E3p module communication error.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
GX_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start X direction from HMI	GX_IN bit is connected with HMI Local Start X button. 1=Start
GY_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Start Y direction from HMI	GY_IN bit is connected with HMI Local Start Y button. 1=Start
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available fault from HMI	K_Fault bit is connected with HMI Available switch. (0=Ok, 1=Fault)
Ground_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Ground Fault from HMI	Ground_Fault bit is connected with HMI Ground switch.(0=Ok, 1=Fault). Ground Fault alarm will be generated and E3p trip status will be set to 1.
Node_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	E3p Node failure from HMI	Node_Fault bit is connected with HMI E3p Node switch. (0=Ok, 1=Fault)
Phase_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Phase Loss fault from HMI	Phase_Fault bit is connected with HMI Phase switch.(0=Ok, 1=Fault). Phase Loss alarm will be generated and E3p trip status will be set to 1.
RX	BOOL				Running X	Motor is Running in X direction.
RY	BOOL				Running Y	Motor is Running in Y direction.
S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Local stop button. 1=Stop
SwitchX_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch ZX Feedback time	Time delay to set ZX bit after device is running in X direction
SwitchY_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch ZY Feedback time	Time delay to set ZY bit after device is running in Y direction
ThermalUtilized	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Thermal Utilized	Thermal utilized variable is connected to HMI Thermal utilized slider. Setting Thermal utilized value to 85-99% will generate Overload warning. Setting this value to 100% will generate Overload alarm and E3p trip status will be set to 1.
tmrFeedback_RX	TIMER				Contactor Feedback RX Timer	This timer is used to delay output **RX** after receiving **DX** from motor block

tmrFeedback_RY	TIMER				Contactor Feedback RY Timer	This timer is used to delay output **RY** after receiving **DY** from motor block
tmrLimitSwitch_ZX	TIMER				External Limit Switch ZX Position Feedback Timer	This timer is used to delay output **ZX** after **RX** =1
tmrLimitSwitch_ZY	TIMER				External Limit Switch ZY Position Feedback Timer	This timer is used to delay output **ZY** after **RY** =1
TX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Torque Switch X Fault from HMI	TX_Fault bit is connected with HMI Torque Switch X switch. (0=Ok, 1=Fault)
TY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Torque Switch Y Fault from HMI	TY_Fault bit is connected with HMI Torque Switch Y switch. (0=Ok, 1=Fault)
X_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Internal Limit Switch X Fault from HMI	X_Fault bit is connected with HMI Internal Switch X switch. (0=Ok, 1=Fault)
Y_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Internal Limit Switch Y Fault from HMI	Y_Fault bit is connected with HMI Internal Switch Y switch. (0=Ok, 1=Fault)
ZX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch X Fault from HMI	ZX_Fault bit is connected with HMI External Limit Switch X switch. (0=Ok, 1=Fault)
ZY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	External Limit Switch Y Fault from HMI	ZY_Fault bit is connected with HMI External Limit Switch Y switch. (0=Ok, 1=Fault)
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local disconnect switch fault from HMI	U_Fault bit is connected with HMI Disconnect switch. (0=Ok, 1=Fault)

E3p

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	X	Inp		Contactor Coil X direction	Is the Command to Start the Motor Device. -> Set E3p device output **OUTA=1** Typically linked to MotorN block output **D** or by MotorR / MotorD to **DX.** **Note:** Linked mean this signal is direct allocated to the Motor AOI output **D/DX.**
DY	BOOL	X	Inp		Contactor Coil Y direction	Is the Command to Start the Motor Device. -> Set E3p device output **OUTB=1** Typically linked to MotorR / MotorD to **DY** **Note:** Linked mean this signal is direct allocated to the Motor AOI output **DY.**
NodeFailure	BOOL	X	Inp		Node failure bit mapped from DNET scanner module device failure register bit	This input has to be correctly mapped to allow E3p module monitor communication status for E3 Plus hardware device. When NodeFailure is on, the E3p input image table is cleared and E3p comm and trip fault is generated.
RX	BOOL	X	Out		Motor Status:Contactor feedback DX	RX is the Device Running feedback Status of the motor. Typically linked to MotorN input *R* or by MotorD / MotorR to *RX* Status "Motor is running" if **Motor Current >30% of the minimum FLA** Setting and the Motor was commanded to X direction (DX). Refer to the E3p Device "Current Rating", see also Catalog Number explanation: i.g. 193-EC2**C ;** C = **5**..25Amp -> 5Amp x 30% = 1,5Amp; if motor current >1.5Amp then **RX** set to 1.
RY	BOOL	X	Out		Motor Status:Contactor feedback DY	RY is the Device Running feedback Status of the motor in Y direction. Typically linked to MotorD / MotorR input *RY* Status "Motor is running" if **Motor Current >30% of the minimum FLA** Setting and the Motor was commanded to Y direction (DY). Refer to the E3p Device "Current Rating", see also Catalog Number explanation: i.g. 193-EC2**C** ; C = **5**..25Amp -> 5Amp x 30% = 1,5Amp; if motor current >1.5Amp then **RY** set to 1.
T	BOOL	X	Out		Motor Status: Trip Fault	T is the Tripped output signal including Fail Safe Device Thermal Overload signal. 1=OK; 0=Trip. The E3p module will trip with a overload indication if **ThermUtilized** reaches **100%**. Typically direct allocated to the Motor AOI-input *T*.
Trip	BOOL		Out		Module Trip status	Common Status -> E3p Module tripped. ->Motor OFF (OUTA=0,OUTB=0)

IN1	BOOL		Out		E3p Module Input 1	Status of E3p Device input **IN1** Usable for individual MCC feedbacks
IN2	BOOL		Out		E3p Module Input 2	Status of E3p Device input **IN2** Usable for individual MCC feedbacks
IN3	BOOL		Out		E3p Module Input 3	Status of E3p Device input **IN3** Usable for individual MCC feedbacks
IN4	BOOL		Out		E3p Module Input 4	Status of E3p Device input **IN4** Usable for individual MCC feedbacks
WA	BOOL		Out	Note: Tag direct linked by RSViewSE and is config.as a Alarm Tag	Overload Warnning Alarm Thermal Utilization > 85%	This output is set if the Motors Thermal Utilization reach the Overload Warn Level (Refer to the E3p Device -Advanced Setup **Parameter 32** "OL Warn Level". Default Value is set to **85%** of the Thermal Utilisation. Note: The Overload Warning status is part of the Device Status information (**Par 21**)
CA	BOOL		Out	Note: Tag direct linked by RSViewSE and is config.as a Alarm Tag	Comm. Fault	This output is set if there is a communication fault between E3p hardware device and DNET scanner module.
DeviceStatus	INT		Out	Noting, Tag direct linked by RSViewSE /FactoryTalk	Device Status Word	Device Status Word ->HMI Motor Template direct linked to this Word Refer E3p diagnostic parameter: DEVICE STATUS =**PAR21** **Bit0 = Trip** **Bit4 = Input#1** **Bit1 = Warning** **Bit5 = Input#2** **Bit2 = Output A** **Bit6 = Input#3** **Bit3 = Output B** **Bit7 = Input#4** **Bit8 = Motor Current (running)** **Bit9 =Ground Fault**
TripStatus	INT		Out	Noting, Tag direct linked by RSViewSE /FactoryTalk	Trip Status Word	Device Status Word ->HMI Motor Template direct linked to this Word Refer E3p diagnostic parameter: TRIP STATUS =**PAR14 -**by default are enabled: **-Bit1= Overload** **-Bit2= Phase Loss** **-Bit9= Comm Fault** (See PAR24 -Trip Enable)
AverageCurrent	INT		Out	Noting, Tag direct linked by RSViewSE /FactoryTalk	Av Current Value	Motor Monitor Average Current in Amps ->HMI Motor Template direct linked to this Word Refer E3p Monitor parameter: AVERAGE CURRENT =**PAR4**
ThermUtilized	SINT		Out	Noting, Tag direct linked by RSViewSE /FactoryTalk	Therm Utilized Value	Motor Monitor Thermal capacity utilization in % ->HMI Motor Template direct linked to this Word Refer E3p Monitor parameter: %THERM UTILIZED =**PAR9**

DataInp	E3_Inp	X	InOut		DeviceNet E3-Note Input struchture	DeviceNet E3 Note Input Structure Typically direct allocated by the DeviceNet Scanner Input structure Tag, created by “DN Tag Generator”
DataOut	E3_Out	X	InOut		DeviceNet E3-Note Output struchture	DeviceNet E3 Note Output Structure Typically direct allocated by the DeviceNet Scanner output structure Tag, created by “DN Tag Generator”

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	InOut	Typically <*Motor*_C>.Bus	Bus from/to Parent Module (MotorN/D/R)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This E3p_AOI module reads information FROM the "Bus" and Writes status information TO the "Bus". (Comm fault and WA)

SubSys

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EnAutoStart	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Start (normally mastered by CtrlGrp) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAuto=1 only	The **EnAutoStart** is an "auto start permissive". The **EnAutoStart** is typically but not necessarily mastered by the **CtrlGrp** or **MaGrp.** To initially start the motor both the **EnAutoStart** AND **EnAuto** must be = 1. Only **EnAuto** = 1 is required to keep the motor running. The **EnAutoStart** 0 to 1 transition will restart the motor if **EnAuto** =1. A rung of logic must be setup outside of the AOI to set the **EnAutoStart.** Using the **CtrlGrp.EnAutoStart** output as the input condition, the motor **EnAutoStart** is transitioned from 0 to 1 on a "RESTART" condition. Application Conditions (Permissives) required to Start Motor but not required to keep motor running can be used to set this bit. If the group of equipment is running in auto, setting the **EnAutoStart** to 0 does not stop the equipment while the **EnAuto** is still maintained. **EnAutoStart** is not active in "Local" or "Single Start" mode.
EnAuto	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Operation (normally mastered by CtrlGrp and interlocked by application) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0 ->1 transition - Run requires EnAuto=1 only	The **EnAuto** may be better referred to as the "IntlAuto" as it is the process interlock. The **EnAuto** must be maintained in the ON state to keep Motor running. The **EnAuto** reset to 0 will stop the Motor. The **EnAuto** is used to sequence/start/stop equipment during the auto operation of equipment. **EnAuto** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual*
S	BOOL	X	Inp	Direct allocated to the AOI	Local Stop Input (0=Stop)	**S** is the Fail-Safe Local (Field) Stop PB input signal. **S** = 1 for the Device to start in any mode. If **S** = 0, the Device will always Stop.

G	BOOL	X	Inp	Direct allocated to the AOI	Local Start Input (Go) Overrides Machine Protection Interlock IntlG if Local operation enabled	Local (Field) Start PB input signal (Direction specific). Device must be in "Local" mode to enable the use of the **GX/GY** signals to start the device. - If the device is in "Local" mode, the **GX/GY** 0->1 transition will initiate an alarm acknowledge for this device and all it's bus child devices. GX/GY PB could also be used to test flashlights while device is running if **GlobalData.Par.EnLampTest** = 1. - If the **GX/GY** signal is maintained during "Local" mode (i.e. holding the local start PB), then the Machine Interlocks (**IntlG**) are bridged (i.e. bypassed or jumpered) if GobalData.Par.DisableJogging is set to 0. - If the device is configured to have **Local warning**, (**Par.StartupWarningLocal =1**) then the required sequence to start the device in "Local" mode is as follows: The Local Start PB is momentarily pressed providing a **GX/GY** = 1 signal; the start warning horn will be provided; within X seconds of the start warning horn completion, the Local Start PB is pressed again and the device will immediately start. X represents the time value in **Global.Par.StartupCancelTime.** If **Global.Par.DisableTwoLocalStart** is set to one, then only one press of GX/GY is required to start the device. Function same as **G. GX** start for X direction(open) / **GY** start for Y direction(close). Device Input for Local (Field) Start PB for both directions Should not set GX and GY to 1 at the same time.
U	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no **U** IO signal.	Local Isolator / Safety Switch (0=OFF)	**U** is the Fail Safe Local Isolator Position input signal (i.e. field disconnect switch). **U** = 1 for Device to Start in any mode. If the **U** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.
K	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no K IO signal.	Ok Available 1=Ok, MCC Unit	**K** is the Fail Safe Device OK status input signal (typically from 480V or 120V breaker status). **K** = 1 for device to Start. If the **K** signal transitions to 0 when the device is running, device will Fail and a RESTART is required from Parent CtrlGrp.
T	BOOL	X	Inp	Direct allocated to the AOI.Use Log_1 if device has no T IO signal.	Thermal Overload Ok 0=Thermal	**T** is the Fail Safe Device Thermal Overload input signal. **T** = 1 for device to start. If the **T** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.

R	BOOL	X	Inp	Direct allocated to the AOI	Running Contactor Feedback. 0=OFF	**R** is the Device Running Status input signal (i.e. contactor position). After the device is commanded to run (**D** = 1), the **R** signal must transition to 1 before **Par.InpOutMonTime** expires or the device will Fail (message on HMI). If the **R** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent CtrlGrp.
WA.0..7	SINT of BOOLs		Inp	Code above the respective SubSys AOI and Configure HMI Tag (.WAM tag) with logical meaning of alarm. Alarms are displayed on HMI faceplate.	Warning Alarm Bit 0..7 (unspecified)->**Sta.W** 1=Warning that do not stop the device	**WA.0..7** are warning conditions from the SubSystem. **WA.*** = 1 is a warning condition. The **Sta.W** will always report the **WA** Warnings and a corresponding warning message appears on the **SubSys** faceplate. On the faceplate, the lower bit has priority for display (only 1 message will appear at a time). Note: You have to configure the text message in the HMI Tag Database under **WAM.*** tags.
CA.0..7	SINT of BOOLs		Inp	Code above the respective SubSys AOI and Configure HMI Tag (.CAM tag) with logical meaning of alarm	Control Alarm Bit 0..7 (unspecified)->**Sta.W** or **Sta.F** 1=Alarm that stop the device	**CA.0..7** are Failure Alarm conditions from the SubSystem. **CA.*** = 1 is a failure condition. When the SubSys is supervised by a parent group (**Check** bus signal) the **CA.*** failures are reported either as Failures (**Sta.F**) or Warnings (**Sta.W**) based on the selection of **Par.ErrorWarning.** A corresponding failure message always appears on the **SubSys** faceplate. On the faceplate, the lower bit has priority for display (only 1 message will appear at a time). Note: You have to configure the text message in the HMI Tag Database under **CAM.*** tags.
Intl	BOOL	X	Inp	Coded above the AOI Instruction	Safety Interlock 0=Stop/1=Enable. - Required in any operation mode	**Intl** is a Fail Safe Safey Interlock input signal. The **Intl** signal is coded above the device AOI instance and is the result of a string of Safety device signals (i.e. rope switches) and is intended for safety related interlocks. **Intl** cannot be bridged (bypassed). **Intl** = 1 to run device.
IntlG	BOOL	X	Inp	Coded above the AOI Instruction	Machine Protection Interlock 0=Stop/1=Enable - Required in any operation mode - May be overridden if Local operation enabled and Local Start activated by Input G	**IntlG** is the Fail Safe Machine Protection Interlock input signal. The **IntlG** signal is coded above the device AOI instance and is the result of a string of Machine Protection signals (i.e. drift and speed switches). The signal is intended to protect the machinery. If the device is in "Local" mode, the **IntlG** signal can be bridged by holding the devices "Local Start Pushbutton" (**G**) to run device. To be able to use jog function, **GobalData.Par.DisableJogging** has be be set to 0.
D	BOOL	X	Out	Direct allocated to the AOI	Digital Output Contactor or Coil	**D** is the Device Command to Start output signal. **D** = 1 when device is commanded to start (run) and is mapped to the device's contactor (motor) or coil (valve).

Alarm	BOOL		Out	Typically not used in application programming.	Device Alarm (Warning or Failure) Common signal **Sta.W** or **Sta.F**	**Alarm** = 1 Identifies that the devices has an active alarm present or that the device failed during supervised operation. Any of the following conditions can cause **Alarm**: **Sta.UA**, **Sta.KA**, **Sta.TA**, **Sta.SA**, **Sta.KA**, **Sta.XA**, **Sta.YA**, **Sta.TXA**, **Sta.ZYA**, **Sta.ZXA**, **Sta.CA.***, or "Device Failures". When the device is active (Startup, Waiting, Starting, or Running states), "Device Failures" are caused from the Interlock input (**Intl** & **IntlG)** or contactor feedback (No **R** with **D)** failures. All alarm conditions must clear with an alarm acknowledge before the **Alarm** bit will reset. *Note: Not all Status alarms are present for all modules and status alarms require the Internal Check= 1*
Run	BOOL		Out		Running in Any Mode All interlocks are checked and ok	**Run** = 1 Identifies that the device is running. When the device is starting, the **Run** signal will transition 0->1 only after the **D** and **R** signals are both 1 and the **Par.InpOutMonTime** is expired, and when the device is starting after If On, the Device is Running. The Run signal will transition 1->0 immediately after the device is stopped.
RdyAuto	BOOL		Out	Signal is typically used as the auto interlock for the EnAuto string of the next upstream device.	Ready Running in Auto Mode Set if **Run** AND **Par.ReadyTime** Delay All interlocks are checked and ok, signal to enable/ start next devive in auto mode.	**RdyAuto** = 1 Identifies that the device is Running in "Auto" Mode and Ready for auto operation. When the device's **Run** signal transitions 0->1, delay **Par.ReadyTime** before transitioning the **RdyAuto** signal 0->1. The **RdyAuto** signal will transition 1->0 immediately after the device is stopped.

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Bus	DINT		I/O		Interface from/to Child Modules (Device)	**<Motor*>.Bus** is the calculated Bus from the Motor, Valve, or SubSys module. It is calculated from the input bus signal **<Parent>.Bus** and this modules AOI software. It is used to link child devices to this parent device. Example: Belt Conveyor Motor is this MotorN module and child devices include speed switches, drift switches, etc. The child devices would be linked to the Belt Conveyor Motor using this "Bus"
Cmd.0 \SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Single Start)	**Cmd.0** = 1 is used to stop the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1 \GR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote (Single Start)	**Cmd.1** = 1 is used to start the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.3 \ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the device and child device alarms from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.

Cmd.4 \EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle) ->**Sta.REU**	**Cmd.4** = 1 is used to toggle the device's "Local" mode status (**Sta.REU**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.5 \EG	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Single Start (Toggle)->**Sta.REG**	**Cmd.5** = 1 is used to toggle the device's "Single" mode status (**Sta.REG**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI
Sta.RP	BOOL		Out		Running in Auto Mode (Run Production)	**Sta.RP** = 1 Identifies that the device is starting or running in "Auto" Mode while being supervised by parent Group. (i.e. **Internal Check** = 1)
Sta.RU	BOOL		Out		Running in Local or Single Mode	**Sta.RU** = 1 Identifies that the device is starting or running in "Local" or "Single" Mode while being supervised by parent group (i.e. **Internal Check** = 1)
Sta.RM	BOOL		Out		RunninginAnyMode Set directly by Contactor Feedback Input R to indicate a possible blocked Contactor, for Valve1 Sta.RM=D	**Sta.RM** = 1 Identifies that the device is running. Set directly by the device's **R** input signal
Sta.KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **U**, **K**, **T**, **S,** Contactor feedback (**R** with **D**). This signal should not be confused with **K** or **KAM**. Indicated in HMI as purple
Sta.REU	BOOL		Out		Local Operation Mode Enabled Machine Protection Interlock IntlG may be overridden if Local Start activated by Input G	**Sta.REU** = 1 Identifies that the device is in "Local" mode. Machine protection interlocks (**IntlG)** may be bridged as described in **G** signal description
Sta.REG	BOOL		Out		Single Start Operation Mode Enabled Device can be started from HMI by Cmd.GR Note, Alarm Gongs are not set if Single Start is enabled.	**Sta.REG** = 1 Identifies that the device is in "Single" mode. Machine protection interlocks (**IntlG**) may be bridged as described in **G** signal description
Sta.STU	BOOL		Out		Startup Warning Horn and FlashLight	**Sta.STU** = 1 Identifies that the device is in startup. This signal will be 1 during startup warning until the **Run** signal is 1.
Sta.WAI	BOOL		Out		Waiting with FlashLight	**Sta.WAI** = 1 Identifies that the device is waiting. This signal will be 1 during a device startup after the startup warning horn has stopped (**GlobalDat.Par.StartupHornTime**). The signal will transition 1->0 when either the device is starting or the device is stopped.

Sta.W	BOOL		Out		Warning Alarm (Common Signal) Set on Alarm if ErrorWarning=1	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.F	BOOL		Out		Failure Alarm (Common Signal) Set on Alarm if ErrorWarning=0	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.SA	BOOL		Out		Local Stop Alarm (Message) Set if module check active AND Input S=0	**Sta.SA** = 1 Identifies a Local Stop Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **S** signal input or **Cmd.0** from the HMI when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.UA	BOOL		Out		Local Isolator / Safety Switch Alarm (Message) Set if module check active AND Input U=0	**Sta.UA** = 1 Identifies a Local Isolator / Safety Switch Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **U** signal input when device's parent group is acitive (i.e. **Internal Check** = 1)
Sta.KA	BOOL		Out		Availability Alarm (Message) Set if module check active AND Input K=0	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.TA	BOOL		Out		Thermal Overload Alarm (Message) Set if module check active AND Input T=0	**Sta.TA** = 1 Identifies a Thermal Overload Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **T** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.RA	BOOL		Out		Contactor Feedback Alarm (Message) Set delayed if Output D/DX/DY AND its corresponding feedback Input R/RX/RY are not equal	**Sta.RA** = 1 Identifies a Contactor Feedback Alarm and reports a message to the HMI Alarm Log. The **R** signal input was not detected within **Par.InpOutMonTime** when **D** signal is active. Alternatively, the **R** signal may have been lost after the **Par.InpOutMonTime** has expired.
Sta.WA.0..7	BOOL		Out		Warning Alarm Message 0..7 (unspecified) ->**Sta.W** Warnings that do not stop the device	**Sta.WA.*** = 1 Identifies that there is an active **WA.*** warning alarm. *Note: If an HMI Alarm is desired, then configure Sta.WA.* tags as alarms in the HMI.*
Sta.CA.0..7	BOOL		Out		Control Alarm Message 0..7 (unspecified)->**Sta.W** or **Sta.F** Alarms that stop the device	**Sta.CA.*** = 1 Identifies that there is an active **CA.*** failure alarm. Note: If an HMI Alarm is desired, then configure **Sta.CA.*** tags as alarms in the HMI.
Sta.SAM	BOOL		Out		Local Stop Alarm Mimic Set if Input **S**=0	**Sta.SAM** = 1 Identifies that there is no **S** input signal or there is an active **Sta.SA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Local Stop problem.
Sta.UAM	BOOL		Out		Local Isolator / Safety Switch Alarm Mimic Set if Input **U=0**	**Sta.UAM** = 1 Identifies that there is no **U** input signal or there is an active **Sta.UA** alarm. This is used for HMI display (mimic) animiations to identify that there is a local isolator or safety switch problem.

Sta.KAM	BOOL		Out		Availability Alarm Mimic Set if Input **K**=0	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.
Sta.TAM	BOOL		Out		Thermal Overload Alarm Mimic Set if Input **T**=0	**Sta.TAM** = 1 Identifies that there is no **T** input signal or there is an active **Sta.TA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Thermal Overload problem.
Sta.RAM	BOOL		Out		Contactor Feedback Alarm Mimic Set delayed if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are not equal	**Sta.RAM** = 1 Identifies that there is a contactor failure (**R** when **D** after **Par.InpOutMonTime** expired) or there is an active **Sta.RA** alarm. This is used for HMI display (mimic) animiations to identify contactor feedback failures.
Sta.WAM.0..7	BOOL		Out	Sta.WAM HMI Tag Configuraiton Required. Sta\WAM_* tags "ON Label" provides the Faceplate Message for logical meaning of Warning	Warning 0..7 (Mimic unspecified)->**Sta.W** Warnings that do not stop the device	**Sta.WAM.*** = 1 Identifies that there is a **WA.*** warning alarm. These tags are used in the HMI for mimic animations and SubSys faceplate display messages. For each **WAM.*** tag that is used, the corresponding **Sta\WAM_*** HMI tag "ON Label" field must contain the desired SubSys faceplate message. *Note: If it is desired to have alarm indication on the WA.* tags, then the corresponding HMI tag (Sta\WA_* tag) needs to be configured as an HMI alarm.*
Sta.CAM.0..7	BOOL		Out	Sta.CAM HMI Tag Configuraiton Required. Sta\CAM_* tags "ON Label" provides the Faceplate Message for logical meaning of Failure	Alarm 0..7 (Mimic unspecified)-> **Sta.W** or **Sta.F** Alarms that stop the device	**Sta.CAM.*** = 1 Identifies that there is a C**A.*** failure alarm. These tags are used in the HMI for mimic animations and SubSys faceplate display messages. For each **CAM.*** tag that is used, the corresponding **Sta\CAM_*** HMI tag "ON Label" field must contain the desired SubSys faceplate message. *Note: If it is desired to have alarm indication on the CA.* tags, then the corresponding HMI tag (Sta\CA_* tag) needs to be configured as an HMI alarm.*
Sta.IDS	BOOL		Out		Safety Interlock Alarm Indication Mimic Set if Input Intl=0	**Sta.IDS** = 1 Identifies that there is no **Intl** input signal or there is an active **Intl** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Safety Interlock problem.
Sta.IDP	BOOL		Out		Maschine Protection Interlock Alarm Indication Mimic Set if Input IntlG=0	**Sta.IDP** = 1 Identifies that there is no **IntlG** input signal or there is an active **IntlG** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Machine Protection Interlock problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.

Val.RT	REAL		Out		Run Time / Running Hours [h] updated every 2 sec	**Val.RT** is a value identifying the total number of running hours for the device. It is a totalize updated every 2 seconds.
Val.DC	DINT		Out		Number of Starts Counter	**Val.DC** is a value identifying the total number of Starts for the device. The value increments 1 for every start.
Val.RST_RT	DINT		Out		Remaining seconds before restart is enabled	**Val.RST_RT** is a value identifying the remaining time in second before the device could be started again.
Par. HeavyStartup	BOOL		Inp		Code for Heavy Start 0=No/1=Yes If ON, all other devices are signalled during startup of this device by common tag **Global.HeavyStartup**	Set **Par.HeavyStartup** = 1 to specify that this device is a heavy starting device (i.e. hard starting with high inrush current). When this device is starting, other heavy startup devices in the same PLC are prevented from starting until this device's **RdyAuto** = 1. This is accomplished using the **Global.HeavyStartup** tag.
Par. HeavyStartupIgn	BOOL		Inp		Heavy Start Ignore 0=No/1=Yes If ON, this device ignores the startup of other heavy starting devices	Set **Par.HeavyStartupIgn** = 1 to specify that this device should start when commanded to start regardless of any other device's **Par.HeavyStartup** configuration and starting status. (I.E. Disable the effect of the **Global.HeavyStartup** tag)
Par. StartupWarningLocal	BOOL		Inp		Device with Startup Warning in Local Mode If ON, this device will set the Horn and Flash outputs specified by **Par.StartupHornCode** and **Par.StartupLightCode** when started locally	Set **Par.StartupWarningLocal** = 1 to specify that this device will always provide a startup warning horn / light before starting the device in "Local" mode. When set, a second pulse is required on the **G** input signal to start the device. Refer to **Par.StartupHornCode** and **Par.StartupLightCode** for specific horn/light configurations. Note that MSHA requires a start warning horn when starting conveyors regardless of mode of operation. **Note**: If set this parameter you have to press twice the G-button! See G description.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableSingle	BOOL		Inp		Suppress visibility of Singlestart mode Button at HMI template.	Set Par.DisableSingle =1 to specify that the "Manual/Auto"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableGrpAlarm	BOOL		Inp		Disable Alarm Indication to Parent Group modules If ON, the Warning or Failure Alarm Bus of the Parent Group module is not set. Alarm messages are however still created and **Sta.W** or **Sta.F** is set.	Set **Par.DisableGrpAlarm** = 1 to specify that the device Alarms (**Alarm**) will NOT be reported on the **<Parent>.Bus.18** (Failure) or **<Parent>.Bus.19** (Warning). (This parameter has No affect on Device Alarm reporting i.e. Sta.* tags directly between the device AOI and the HMI.

Par. DisableGrp Check	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set Sta.W or Sta.F	Set **Par.DisableGrpCheck** = 1 to specify that the device will not report to the Control group(CtrlGrp): Restart requests, device Alarms (**Alarm)**, and inhibit the <Module>.Bus.26 (Check) from activating the Module's Internal**. Note: Module will not start in Automatic Mode (EnAuto) if this Parameter is set!**
Par. ErrorWarnin g	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by Sta.W IF ON, Alarms are indicated by Sta.F	Set **Par.ErrorWarning** = 1 to specify that device alarms (**Alarm**) will be reported as Device Warnings (**Sta.W**) and these device warnings will be reported as warnings to the **<Parent>.Bus.19**. Warnings will cause the device HMI animation object to be Yellow. Set **Par.DisableGrpCheck** = 0 to specify that device alarms (**Alarms**) will be reported as Device Failures (**Sta.F**) and these device failures will be reported as failures to the **<Parent>.Bus.18**. Failures will cause the device HMI animation object to be Red. Reporting to the Parent Bus is affected by **Par.DisableGrpAlarm** and **Par.DisableGrpCheck**.
Par. OverrideElm Chk	BOOL		Inp		Override element check	Set **Par.OverrideElmChk** = 1 to specify that the device will report contactor feedback failure to control group even when control group is not active.
Par. DisableGrpI dentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par. HasPermObj	BOOL		Inp		Has permissive object connected to Block	Set Par.HasPermObj = 1 to specify that a Permissive AOI is connected to this block and the permissive object on the HMI faceplate is visible to provide permissive diagnostic information.
Par. HasIntlkObj	BOOL		Inp		Has Interlock object connected to Block	Set Par.HasIntlkObj = 1 to specify that an Interlock AOI is connected to this block and the interlock object on the HMI faceplate is visible to provide interlock diagnostic information.

Par. AlarmGong Code	DINT of BOOLs	X	Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag.
Par. StartupHorn Code	DINT of BOOLs	X	Inp		Enable Startup Warning Horn 0..31 Select Horn(s) by desired Bit Pattern. Sets the **Global.ComHornCode** at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the device's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when starting this device. Before this device starts, the **Par.StartupHornCode** The **Global.ComHornCode** tag (Indirectly the SysGrp AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn** tag).
Par. StartupLight Code	DINT of BOOLs	X	Inp		Enable Startup Warning FlashLight 0..31 Select Flash(es) by desired Bit Pattern. Sets the **Global.ComLightCode** at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the device's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when starting this device. The **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the **<SysGrp>.Light** tag).
Par. EmergencyPower	DINT		Inp	Configure only if there is Emergency power for this section of the plant	Priority for Emergency Power Operation (0..4). If set 0, the device is not stopped on Emergency, else if… **Global.EmergencyPower_1**=ON->stop Prio 1,2,3 & 4 **Global.EmergencyPower_2**=ON->stop Prio 2,3 & 4 **Global.EmergencyPower_3**=ON->stop Prio 3 & 4 **Global.EmergencyPower_4**=ON->stop Prio 4	Set **Par.EmergencyPower** to a value 0 - 4. This is a priority setting that is used to conserve energy when emergency power generators are on-line. (i.e. incoming power from utility is OFF). Higher settings mean lower priority for operations. - Set to 0: the device is not stopped - Set to 1: Stop device if **Global.EmergencyPower_1** = 1; - Set to 2: Stop device if **Global.EmergencyPower_1** or **_2** = 1; - Set to 3: Stop device if **Global.EmergencyPower_1, _2,** or **_3** = 1; - Set to 4: Stop device if **Global.EmergencyPower_1, _2, _3,** or **_4** = 1.

Par. InpOutMon Time	DINT	X	Inp	Default setting Varies by the desired system response	Contactor Feedback Supervision Timer (default 2000 ms) Delay time to check if Output **D/DX/DY** AND its corresponding feedback Input **R/RX/RY** are NOT equal->**Sta.RA/RAM**	Set the **Par.InpOutMonTime** to a time delay in ms. This specifies the maximum allowable time between the start command (**D**) and feedback signal (**R**). If this time is exceeded before the **D** & **R** signals are equal, a contactor failure alarm will occur. Recommended default 2000 ms
Par. RestartTime	DINT	X	Inp		Time Delay for Restarting (ms) Time to hold Status Stopping, during this time the device cannot be restarted	Set the **Par.RestartTime** to a time delay in ms. This specifies the minumum time delay between a device stop and the device's next restart. After the device stops, it cannot be restarted until this time expires. It is used for 2 purposes. 1 - Allow motors to coast to a stop before restarting. (Not all motors can "catch" the rotor to restart 2 - Allow motor windings to cool before restarting.
Par. ReadyTime	DINT	X	Inp		Time Delay to set Output Ready Auto (ms) Outputs **RdyAuto** or (RdyAutoX/Y) are set time delayed to allow the device to take up the load before the next device is started.	Set the **Par.ReadyTime** to a time delay in ms. This specifies the time delay between the device's **Run** output signal and **RdyAuto** output signal. This timer is responsible for the start delay of the next device in a startup sequence.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	Inp		Bus Input from/to Parent Module (Machine or Group)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This Motor* module reads information FROM the "Bus" and Writes status information TO the "Bus".
Reference Notes						
Note 1:						**Internal Check** is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: <Parent>.Bus.26 with DisableGrpCheck = 0, Sta.REU (Local Mode), OR Sta.REG (Single Mode)

Valve1 and Valve2

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EnAutoStart	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Start (normally mastered by CtrlGrp) - Start requires EnAutoStart=1 AND EnAuto=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAuto=1 only	The **EnAutoStart** is an "auto start permissive". The **EnAutoStart** is typically but not necessarily mastered by the **CtrlGrp** or **MaGrp.** To initially start the motor both the **EnAutoStart** AND **EnAuto** must be = 1. Only **EnAuto** = 1 is required to keep the motor running. The **EnAutoStart** 0 to 1 transition will restart the motor if **EnAuto** =1. A rung of logic must be setup outside of the AOI to set the **EnAutoStart.** Using the **CtrlGrp.EnAutoStart** output as the input condition, the motor **EnAutoStart** is transitioned from 0 to 1 on a "RESTART" condition. Application Conditions (Permissives) required to Start Motor but not required to keep motor running can be used to set this bit. If the group of equipment is running in auto, setting the **EnAutoStart** to 0 does not stop the equipment while the **EnAuto** is still maintained. **EnAutoStart** is not active in "Local" or "Single Start" mode. *Note: For a timing diagram see AppUserManual_V111, chapter 7*
EnAutoX/EnAutoY	BOOL		Inp	Mapping Code above the Device AOI	Enable Automatic Operation X/Y (normally mastered by CtrlGrp and interlocked by application) - Start requires EnAutoStart=1 AND EnAutoX/Y=1 - Restart requires EnAutoStart 0->1 transition - Run requires EnAutoX/Y=1only	Similar to the EnAuto in operation but controls which direction the Motor/Valve will run/operate (X or Y). Motor/Valve will not operate if both the **EnAutoX** and **EnAutoY** are set = 1simultaneously.
S	BOOL	X	Inp	Direct allocated to the AOI	Local Stop Input (0=Stop)	**S** is the Fail-Safe Local (Field) Stop PB input signal. **S** = 1 for the Device to start in any mode. If **S** = 0, the Device will always Stop.

G GX/GY	BOOL		Inp	Direct allocated to the AOI	Local Start Direction (Go) (Go X/Y) Overrides Machine Protection Interlock IntlG if Local operation enabled	Local (Field) Start PB input signal (Direction specific). Device must be in "Local" mode to enable the use of the **G** or **GX/GY** signals to start the device. - If the device is in "Local" mode, the **G/ GX/GY** 0->1 transition will initiate an alarm acknowledge for this device and all it's bus child devices. GX/GY PB could also be used to test flashlights while device is running if **GlobalData.Par.EnLampTest** = 1. - If the **G / GX/GY** signal is maintained during "Local" mode (i.e. holding the local start PB), then the Machine Interlocks (**IntlG**) are bridged (i.e. bypassed or jumpered) if GobalData.Par.DisableJogging is set to 0. - If the device is configured to have **Local warning**, (**Par.StartupWarningLocal =1**) then the required sequence to start the device in "Local" mode is as follows: The Local Start PB is momentarily pressed providing a **G / GX/GY** = 1 signal; the start warning horn will be provided; within X seconds of the start warning horn completion, the Local Start PB is pressed again and the device will immediately start. X represents the time value in **Global.Par.StartupCancelTime.** If **Global.Par.DisableTwoLocalStart** is set to one, then only one press of GX/GY is required to start the device. Function same as **G. GX** start for X direction(open) / **GY** start for Y direction(close). Device Input for Local (Field) Start PB for both directions Should not set GX and GY to 1 at the same time.
U	BOOL	X	Inp	Use Log_1 if device has no U IO signal.	Local Isolator / Safety Switch (0=OFF)	**U** is the Fail Safe Local Isolator Position input signal (i.e. field disconnect switch). **U** = 1 for Device to Start in any mode. If the **U** signal transitions to 0 when the device is running, the device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
K	BOOL	X	Inp	Use Log_1 if device has no U IO signal.	Ok Available 1=Ok, MCC Unit	**K** is the Fail Safe Device OK status input signal (typically from 480V or 120V breaker status). **K** = 1 for device to Start. If the **K** signal transitions to 0 when the device is running, device will Fail and a RESTART is required from Parent AOI (CtrlGrp, MaGrp, etc.)
ZX/ZY	BOOL	X	Inp	Direct allocated to the AOI	Position X/Y Limit Switch. 1=Positioned that stop the device, direction dependent	**ZX/ZY** are External Limit Switch input signals used to indicate physical position for maximum travel (direction specific). Signal is used to Stop device after desired position is achieved. **ZX/ZY** = 0 to start/run in that direction (**DX/DY** respectively) **ZX/ZY** = 1 to stop traveling in the respective direction (**DX/DY** respectively)

Intl	BOOL	X	Inp	Coded above the AOI Instruction	Safety Interlock 0=Stop/1=Enable. - Required in any operation mode	**Intl** is a Fail Safe Safey Interlock input signal. The **Intl** signal is coded above the device AOI instance and is the result of a string of Safety device signals (i.e. rope switches) and is intended for safety related interlocks. **Intl** = 1 to run device. If the **Intl** signal transitions to 0 when the device is running, the device will Fail and a "Restart" is required from the Parent AOI (CtrlGrp, MaGrp). **Intl** cannot be bridged (bypassed).
IntlG	BOOL	X	Inp	Coded above the AOI Instruction	Machine Protection Interlock 0=Stop/1=Enable - Required in any operation mode - May be overridden if Local operation enabled and Local Start activated by Input G or GX/GY	**IntlG** is the Fail Safe Machine Protection Interlock input signal. The **IntlG** signal is coded above the device AOI instance and is the result of a string of Machine Protection signals (i.e. drift and speed switches). The signal is intended to protect the machinery. If the device is in "Auto" mode, **IntlG** = 1 to allow device to run. If the **IntlG** signal transitions to 0 when the device is running in "Auto" mode, the device will fail and require a "Restart" from Parent AOI (CtrlGrp, MaGrp). If the device is in "Local" mode, the **IntlG** signal can be bridged by holding the devices "Local Start Pushbutton" (**G**, **GX**, or **GY**) as described above. To be able to use jog function, **GobalData.Par.DisableJogging** has be be set to 0.
D DX/DY	BOOL	X	Out	Direct allocated to the AOI	Digital Output Contactor or Coil Direction X/Y (Valve2)	**D** (Vlave1) **or DX/DY** (Valve2) is the Device Command to Start output signal (direction specific). **DX/DY** = 1 (Valve2) when device is commanded to start (run) in the X/Y direction respectively and signal is mapped to the device's contactor (motor) or coil (valve).
Alarm	BOOL		Out	Typically not used in application programming.	Device Alarm (Warning or Failure) Common signal Sta.W or Sta.F	**Alarm** = 1 Identifies that the devices has an active alarm present or that the device failed during supervised operation. Any of the following conditions can cause **Alarm**: **Sta.UA**, **Sta.KA**, **Sta.TA**, **Sta.SA**, **Sta.KA**, **Sta.XA**, **Sta.YA**, **Sta.TXA**, **Sta.ZYA**, **Sta.ZXA**, **Sta.CA.***, or "Device Failures". When the device is active (Startup, Waiting, Starting, or Running states), "Device Failures" are caused from the Interlock input (**Intl** & **IntlG)** or contactor feedback (No **R** with **D)** failures. All alarm conditions must clear with an alarm acknowledge before the **Alarm** bit will reset. *Note: Not all Status alarms are present for all modules and status alarms require the Internal Check= 1*
Run	BOOL		Out		Running in Any Mode All interlocks are checked and ok	**Run** = 1 Identifies that the device is running. When the device is starting, the **Run** signal will transition 0->1 only after the **D** and **R** signals are both 1 and the **Par.InpOutMonTime** is expired, and when the device is starting after If On, the Device is Running. The **Run** signal will transition 1->0 immediately after the device is stopped.

RdyAutoX/ RdyAutoY	BOOL		Out	Signal is typically used as the auto interlock for the EnAuto string of the next upstream device.	Ready Positioned X/Y in Auto Mode Set if Run AND Par.ReadyTime Delay	**RdyAutoX/RdyAutoY** = 1 Identifies that the device is Running in "Auto" Mode in the respective direction (**X** or **Y**) and is Ready for auto operation (direction specific). When the device's **Run** signal transitions 0->1, delay **Par.ReadyTime** before transitioning the RdyAuto signal 0->1. The **RdyAuto** signal will transition 1->0 immediately after the device is stopped.
RdyAutoXY	BOOL		Out	Signal is typically used as the auto interlock for the next upstream device.	Ready Positioned X or Y in Auto Mode Turns on when either RdyAutoX/RdyAutoY is on and delays to turn off for a precalculated time period depends on parameter settings.	**RdyAutoXY** = 1 Identifies that the device's **RdyAutoX** = 1 OR **RdyAutoY** = 1. As soon as the module is in Auto (EnAutoStart) and one of the positions are reached, then this signal (RdyAutoXY) is set to 1. Will be reset by a Alarm or Auto off (EnAutoX /Y =0).

Module Data	**Data Type**	**User Config?**	**I/O**	**Configuration Required**	**Description**	**Notes**
Bus	DINT		I/O		Interface from/to Child Modules (Device)	**<Motor*>.Bus** is the calculated Bus from the Motor, Valve, or SubSys module. It is calculated from the input bus signal **<Parent>.Bus** and this modules AOI software. It is used to link child devices to this parent device. Example: Belt Conveyor Motor is this MotorN module and child devices include speed switches, drift switches, etc. The child devices would be linked to the Belt Conveyor Motor using this "Bus"
Cmd.0\SR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Stop Remote (Stop Single Start)	**Cmd.0** = 1 is used to stop the device in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.1\GR X \GR	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction X (Single Start X)	**Cmd.1** = 1 is used to start the device (**DX** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.2\GRY	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Go Remote Direction Y (Single Start Y)	**Cmd.2** = 1 is used to start the device (**DY** direction) in "Single" mode from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI. (Only by Valve2)
Cmd.3\ACK	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge (Remote Reset)	**Cmd.3** = 1 is used to acknowledge the device and child device alarms from the HMI. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4\EU	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle)->**Sta.REU**	**Cmd.4** = 1 is used to toggle the device's "Local" mode status (**Sta.REU**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI.

Cmd.5\EG	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Single Start (Toggle)->**Sta.REG**	**Cmd.5** = 1 is used to toggle the device's "Single" mode status (**Sta.REG**) to the opposite of it's current state. Reserved for HMI Commands. Set in HMI, Reset in AOI
Cmd.6\SEL	BOOL		I/O	Noting, Tag direct linked by RSViewSE /FactoryTalk	Bypss Limit switch input (Toggle)->**Sta.BA**	**Cmd.6** = 1 is used to Enable / Disable the signal Bypass from the HMI. The Signal Bypass is indicated by the **Sta.BA** status**.** Reserved for HMI Commands. Set in HMI, Reset in AOI. See Also **Par.EnBypass**
Sta.RPX/RPY	BOOL		Out		Running X/Y in Auto Mode (Production X/Y)	**Sta.RPX/RPY** = 1 Identifies that the device is starting or running in "Auto" Mode while being supervised by parent group. (i.e. **Internal Check** = 1). Signal is direction specific.
Sta.RUX/RUY	BOOL		Out		Running X/Y Local or Single Mode	**Sta.RUX/RUY** = 1 Identifies that the device is starting or running in "Local" or "Single" Mode while being supervised by parent group (i.e. **Internal Check** = 1). Signal is direction specific
Sta.RXM/RYM	BOOL		Out		Running Direction X/Y in Any Mode Set directly by Contactor Feedback Input RX/RY to indicate a possible blocked Contactor.	**Sta.RXM/RYM** = 1 Identifies that the device is running in the **X/Y** direction respectively. Set directly by the device's **RX/RY** input signal.
Sta.BA	BOOL		Out		Bypass Enabled Limit switches	**Sta.BA** = 1 Identifies that the **ZX** and **ZY** input signals are bypassed and internally simulated. *Note:* It isn't a alarm bit at the HMI. The template push button creates a remark text in the FT Event history file.
Sta.KM	BOOL		Out		Device Ok/Available Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **U**, **K**, **T**, **S,** Contactor feedback (**R** with **D**). This signal should not be confused with **K** or **KAM**. Indicated in HMI as purple
Sta.REU	BOOL		Out		Local Operation Mode Enabled Machine Protection Interlock IntlG may be overridden if Local Start activated by Input G or GX/GY Note, Alarm Gongs are not set if Local is enabled.	**Sta.REU** = 1 Identifies that the device is in "Local" mode. Machine protection interlocks (**IntlG)** may be bridged as described in **G** signal description
Sta.REG	BOOL		Out		Single Start Operation Mode Enabled Device can be started from HMI by Cmd.GR/GRX/GRY Note, Alarm Gongs are not set if Single Start is enabled.	**Sta.REG** = 1 Identifies that the device is in "Single" mode. Machine protection interlocks (**IntlG**) may be bridged as described in **G** signal description
Sta.STU	BOOL		Out		Startup Warning Horn and FlashLight	**Sta.STU** = 1 Identifies that the device is in startup. This signal will be 1 during startup warning until the **Run** signal is 1.

Sta.WAI	BOOL		Out		Waiting with FlashLight	**Sta.WAI** = 1 Identifies that the device is waiting. This signal will be 1 during a device startup after the startup warning horn has stopped (**GlobalDat.Par.StartupHornTime**). The signal will transition 1->0 when either the device is starting or the device is stopped.
Sta.W	BOOL		Out		Warning Alarm (Common Signal) Set on Alarm if ErrorWarning=1	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.F	BOOL		Out		Failure Alarm (Common Signal) Set on Alarm if ErrorWarning=0	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.SA	BOOL		Out		Local Stop Alarm (Message) Set if module check active AND Input S=0	**Sta.SA** = 1 Identifies a Local Stop Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **S** signal input or **Cmd.0** from the HMI when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.UA	BOOL		Out		Local Isolator / Safety Switch Alarm (Message) Set if module check active AND Input U=0	**Sta.UA** = 1 Identifies a Local Isolator / Safety Switch Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **U** signal input when device's parent group is acitive (i.e. **Internal Check** = 1)
Sta.KA	BOOL		Out		Availability Alarm (Message) Set if module check active AND Input K=0	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.ZXA/ZYA	BOOL		Out		External Position X/Y Alarm (Message)	**Sta.ZXA/ZYA** = 1 Identifies that there is a device position alarm. Two possible conditions for each direction: 1. Device commanded to **X/Y** position and **Par.PosTimeX/Par.PosTimeY** expired before reaching the **ZX/ZY** limit or 2. **ZX** = 1 and **ZY** = 1 at the same time.
Sta.SAM	BOOL		Out		Local Stop Alarm Mimic Set if Input **S**=0	**Sta.SAM** = 1 Identifies that there is no **S** input signal or there is an active **Sta.SA** alarm. This is used for HMI display (mimic) animiations to identify that there is a Local Stop problem.
Sta.UAM	BOOL		Out		Local Isolator / Safety Switch Alarm Mimic Set if Input **U=0**	**Sta.UAM** = 1 Identifies that there is no **U** input signal or there is an active **Sta.UA** alarm. This is used for HMI display (mimic) animiations to identify that there is a local isolator or safety switch problem.
Sta.KAM	BOOL		Out		Availability Alarm Mimic Set if Input **K**=0	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.
Sta.ZXAM/ZYAM	BOOL		Out		External Position X/Y Alarm Mimic Set delayed if Input **ZX/ZY**=0 AND Output **DX/DY**=1	**Sta.ZXAM/ZYAM** = 1 Identifies that there is an external device position limit failure or there is an active **Sta.ZXA/ZYA** alarm. This is used for HMI display (mimic) animations to identify that there is an external limit switch problem.

Sta.ZX/ZY	BOOL		Out		External Position X/Y Indication Mimic	**Sta.ZX/ZY** = 1 Identifies that the device is positioned at that position. This is used for HMI display (mimic) animations to identify the position.
Sta.IDS	BOOL		Out		Safety Interlock Alarm Indication Mimic Set if Input Intl=0	**Sta.IDS** = 1 Identifies that there is no **Intl** input signal or there is an active **Intl** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Safety Interlock problem.
Sta.IDP	BOOL		Out		Maschine Protection Interlock Alarm Indication Mimic Set if Input IntlG=0	**Sta.IDP** = 1 Identifies that there is no **IntlG** input signal or there is an active **IntlG** alarm. This is used for HMI display (mimic) animiations to identify that there is or was a Machine Protection Interlock problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.DCX/DCY	DINT		Out		Number of Starts Counter Direction X/Y	**Val.DCX/DCY** is a value identifying the total number of Starts for the device in that respective direction. The value increments 1 for every start.
Par.HoldOutput	BOOL		Inp		Outputs **D**, **DX**, **DY** are kept in its current (last) state in case of an Position Alarm.	Set **Par.HoldOutput** = 1 to hold the device output (**D**) in last state if an Position Alarm (Sta.ZXA/ZYA) condition comes true for the device.
Par.PulseOutput	BOOL		Inp		Outputs **DX**, **DY** are setOFFifappropriete Position **ZX** or **ZY** is reached ->-**Only Valve2!**	Set **Par.PulseOutput** = 1 to transition **DX/DY** outputs 1->0 after the **ZX/ZY** positions are reached (direction specific). This works with a 2 coil solenoid where the solenoid pin will not change position when **DX** transitions 1->0 until **DY** transitions 0->1. This is similar to operation of MotorD.
Par. StartupWarningLocal	BOOL		Inp		Device with Startup Warning in Local Mode If ON, this device will set the Horn and Flash outputs specified by **Par.StartupHornCode** and **Par.StartupLightCode** when started locally	Set **Par.StartupWarningLocal** = 1 to specify that this device will always provide a startup warning horn / light before starting the device in "Local" mode. When set, a second pulse is required on the **G** input signal to start the device. Refer to **Par.StartupHornCode** and **Par.StartupLightCode** for specific horn/light configurations. Note that MSHA requires a start warning horn when starting conveyors regardless of mode of operation.
Par. EnBypass	BOOL		Inp		Enable Bypass Function (HMI visibility)	Set **Par.EnBypass** =1 to specify that the "Bypass" -push button at the HMI is visible Added a “bypass limit switch” option to the valve modules. When bypass, then the limit switches are internally simulated controlled by **:DX :DY** resp **:D** -signal by Vlave1 . Usable when an Limitswich is broken OR we don't have Limitswitches installed.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)

Par. DisableSingle	BOOL		Inp		Suppress visibility of Singlestart mode Button at HMI template.	Set Par.DisableSingle =1 to specify that the "Manual/Auto"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. DisableGrpAlarm	BOOL		Inp		Disable Alarm Indication to Parent Group modules If ON, the Warning or Failure Alarm Bus of the Parent Group module is not set. Alarm messages are however still created and **Sta.W** or **Sta.F** is set.	Set **Par.DisableGrpAlarm** = 1 to specify that the device Alarms (**Alarm**) will NOT be reported on the **<Parent>.Bus.18** (Failure) or **<Parent>.Bus.19** (Warning). (This parameter has No affect on Device Alarm reporting i.e. Sta tags directly between the device AOI and the HMI)
Par. DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** = 1 to specify that the device will not report restart requests to **<Parent>.Bus.17** (Restart Request), device Alarms (**Alarm)** to **<Parent>.Bus.18** (Failure), and inhibit the **<Parent>.Bus.26** (Check) from activating the Module's **Internal Check**. Use **Par.DisableGrpCheck** to prevent device failures from faulting the entire group. If **Par.DisableGrpCheck** = 1, the **Internal Check** will enable **Sta.W, Sta.F,** or other **Sta.** alarms. Note: Module will not start in Automatic Mode (EnAuto) if this Parameter is set!
Par. ErrorWarning	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by **Sta.W** IF ON, Alarms are indicated by **Sta.F**	Set **Par.ErrorWarning** = 1 to specify that device alarms (**Alarm**) will be reported as Device Warnings (**Sta.W**) and these device warnings will be reported as warnings to the **<Parent>.Bus.19**. Warnings will cause the device HMI animation object to be Yellow. Set **Par.DisableGrpCheck** = 0 to specify that device alarms (**Alarms**) will be reported as Device Failures (**Sta.F**) and these device failures will be reported as failures to the **<Parent>.Bus.18**. Failures will cause the device HMI animation object to be Red. Reporting to the Parent Bus is affected by **Par.DisableGrpAlarm** and **Par.DisableGrpCheck**.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par. HasPermObj	BOOL		Inp		Has permissive object connected to Block	Set Par.HasPermObj = 1 to specify that a Permissive AOI is connected to this block and the permissive object on the HMI faceplate is visible to provide permissive diagnostic information.

Par. HasIntlkObj	BOOL		Inp		Has Interlock object connected to Block	Set Par.HasIntlkObj = 1 to specify that an Interlock AOI is connected to this block and the interlock object on the HMI faceplate is visible to provide interlock diagnostic information.
Par. AlarmGong Code	DINT of BOOLs		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local or Single Start is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Par. StartupHorn Code	DINT of BOOLs		Inp		Enable Startup Warning Horn 0..31 Select Horn(s) by desired Bit Pattern. Sets the **Global.ComHornCode** at startup.	Warning Horns are sound devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Horns. Set the device's **Par.StartupHornCode** bits that are mapped to the desired warning horns that should sound when starting this device. Before this device starts, the **Par.StartupHornCode** is mapped to the **Global.ComHornCode** tag (Indirectly the SysGrp AOI maps the **Global.ComHornCode** to the **<SysGrp>.Horn** tag).
Par. StartupLight Code	DINT of BOOLs		Inp		Enable Startup Warning FlashLight 0..31 Select Flash(es) by desired Bit Pattern. Sets the **Global.ComLightCode** at startup.	Warning Lights are flashing light devices in the field used to warn of equipment startups. For each CLX PLC, there are 32 configurable Warning Lights. Set the device's **Par.StartupLightCode** bits that are mapped to the desired warning lights that should flash when starting this device. Before this device starts, the **Par.StartupLightCode** is mapped to the **Global.ComLightCode** tag (Indirectly the SysGrp AOI maps the **Global.ComLightCode** to the **<SysGrp>.Light** tag).
Par. ReadyTime	DINT	X	Inp		Time Delay to set Output Ready Auto (ms) Outputs **RdyAuto** or **RdyAutoX/Y** are set time delayed to allow the device to take up the load before the next device is started.	Set the **Par.ReadyTime** to a time delay in ms. This specifies the time delay between the device's **Run** output signal and **RdyAuto** output signal.

Par. PosTimeX/ PosTimeY	DINT	X	Inp		Positioning Time Supervision X/Y (ms)	Set the **Par.PosTimeX/PosTimeY** to a time delay in ms. This specifies the maximum time delay between when a device is commanded to the **X/Y** position (with **DX/DY** commands) and achieves the **ZX/ZY** position. If the time expires before achieve the position, a **Sta.ZXA** or **Sta.ZYA** alarm will occur.
Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	Inp		Bus Input from/to Parent Module (Machine or Group)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This Motor* module reads information FROM the "Bus" and Writes status information TO the "Bus".
Reference Notes						
Note 1:						
						Internal Check is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: <Parent>.Bus.26 with DisableGrpCheck = 0, Sta.REU (Local Mode), OR Sta.REG (Single Mode)

Valve1_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
D	BOOL	x	Inp	Direct allocated to the Valve1 AOI	Solenoid	Solenoid input from Valve1_AOI.
G_OUT	BOOL	x	Out	Direct allocated to the Valve1 AOI	Local Open	Local command to open valve. It is controlled by HMI Open simulation button.
ZY	BOOL	x	Out	Direct allocated to the Valve1 AOI	Limit Switch Y Feedback	Limit switch Y feedback is set to 1 when SwitchY_Time times out after D = 0 .
ZX	BOOL	x	Out	Direct allocated to the Valve1 AOI	Limit Switch X Feedback	Limit switch X feedback is set to 1 when SwitchX_Time times out after D = 1.
K	BOOL	x	Out	Direct allocated to the Valve1 AOI	Available	Available status can be set from HMI simulation faceplate. 0=Ok, 1=Fault.
U	BOOL	x	Out	Direct allocated to the Valve1 AOI	Local Disconnect Switch	Local Disconnect Switch status can be set from HMI simulation faceplate. 0=Ok, 1=Fault.
S_OUT	BOOL	x	Out	Direct allocated to the Valve1 AOI	Local Stop	Local command to Stop/Close valve. It is controlled by HMI Close simulation button.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
D_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Solenoid Fault	D_Fault bit is connected with HMI Contactor switch. 0=Ok, 1=Fault
G_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Open From HMI	G_IN bit is connected with HMI Open button. 1=Open
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available Fault	K_Fault bit is connected with HMI Available switch. 0=Ok, 1=Fault

S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Close button. 1=Close
SwitchX_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch X Feedback time	Time between input D is set to 1 and Limit Switch X Feedback is set to 1.
SwitchY_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch Y Feedback time	Time between input D is set to 0 and Limit Switch Y Feedback is set to 1.
tmrFeedback_ZX	TIMER				Limit Switch X Position feedback Timer	This timer is used to delay output **ZX** after **D** =1
tmrFeedback_ZY	TIMER				Limit Switch Y Position feedback Timer	This timer is used to delay output **ZY** after **D** =0
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Disconnect Switch Fault	U_Fault bit is connected with HMI Disconnect switch. 0=Ok, 1=Fault
ZX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch X Feedback Fault	ZX_Fault bit is connected with HMI Limit Switch X switch. 0=Ok, 1=Fault
ZY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch Y Feedback Fault	ZY_Fault bit is connected with HMI Limit Switch Y switch. 0=Ok, 1=Fault

Valve2_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX	BOOL	x	Inp	Direct allocated to the Valve2 AOI	Solenoid X	Solenoid X input from Valve2_AOI
DY	BOOL	x	Inp	Direct allocated to the Valve2 AOI	Solenoid Y	Solenoid Y input from Valve2_AOI
Pulse_OUT	BOOL	x	Inp	Direct allocated to the Valve2 AOI	Pulse type output	If Valve2 is configured as pulse type, then ZX/ZY feedback stays on even when DX/DY input drop off.
GX_OUT	BOOL	x	Out	Direct allocated to the Valve2 AOI	Local Open X	Local command to open valve in X direction. It is controlled by HMI Start X simulation button.
GY_OUT	BOOL	x	Out	Direct allocated to the Valve2 AOI	Local Open Y	Local command to open valve in Y direction. It is controlled by HMI Start Y simulation button.
ZY	BOOL	x	Out	Direct allocated to the Valve2 AOI	Limit Switch Y Feedback	Limit switch Y feedback is set to 1 when SwitchY_Time times out after DY = 1.
ZX	BOOL	x	Out	Direct allocated to the Valve2 AOI	Limit Switch X Feedback	Limit switch X feedback is set to 1 when SwitchX_Time times out after DX = 1.
K	BOOL	x	Out	Direct allocated to the Valve2 AOI	Available	Available status can be set from HMI simulation faceplate. 0=Ok, 1=Fault.
U	BOOL	x	Out	Direct allocated to the Valve2 AOI	Local Disconnect Switch	Local Disconnect Switch status can be set from HMI simulation faceplate. 0=Ok, 1=Fault.
S_OUT	BOOL	x	Out	Direct allocated to the Valve2 AOI	Local Stop	Local command to Stop/Close valve. It is controlled by HMI Stop simulation button.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
DX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Solenoid X Fault	DX_Fault bit is connected with HMI Contactor X switch. 0=Ok, 1=Fault
DY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Solenoid Y Fault	DY_Fault bit is connected with HMI Contactor Y switch. 0=Ok, 1=Fault
GX_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Open X From HMI	GX_IN bit is connected with HMI Start X button. 1=Open
GY_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Open Y From HMI	GY_IN bit is connected with HMI Start Y button. 1=Open
K_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Available Fault	K_Fault bit is connected with HMI Available switch. 0=Ok, 1=Fault
S_IN	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Stop from HMI	S_IN bit is connected with HMI Stop button. 1=Close
SwitchX_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch X Feedback time	Time between input DX is set to 1 and Limit Switch X Feedback is set to 1.
SwitchY_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch Y Feedback time	Time between input DY is set to 1 and Limit Switch Y Feedback is set to 1.
tmrFeedback_ZX	TIMER				Limit Switch X Position feedback Timer	This timer is used to delay output ZX after DX =1
tmrFeedback_ZY	TIMER				Limit Switch Y Position feedback Timer	This timer is used to delay output ZY after DY =1
U_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Disconnect Switch Fault	U_Fault bit is connected with HMI Disconnect switch. 0=Ok, 1=Fault

ZX_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch X Feedback Fault	ZX_Fault bit is connected with HMI Limit Switch X switch. 0=Ok, 1=Fault
ZY_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Limit Switch Y Feedback Fault	ZY_Fault bit is connected with HMI Limit Switch Y switch. 0=Ok, 1=Fault

DigInp and DigInp2

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Check	BOOL		Inp		Used to Enable Module Alarms without the **ParentBus.Check** Signal	The **"Internal AOI Check"** signal enables module alarms and is driven by the Input **Check**, **<Parent>.Bus.26 (Check)**, or **<Parent>.Bus.8 (LocalCheck)**. Set **Check** = 1 (maintained) with the PLC application software to enable module alarms when the **Inp** signal should be supervised without check signals from a parent CtrlGrp or Device (i.e. Motor). See Also **Par.DisableGrpCheck**
K	BOOL	X	Inp	Typically this would be the voltage source signal for an instrument. I.e. the signal from the instrument's breaker is called K before going to the instrument and back to the PLC as a Inp signal. Set to Log_1 if No the K signal.	Device Ok/Available (Power supply, etc)	Set **K** to 1 when the Field Device providing the Inp signal is powered and considered OK.
Inp	BOOL	X	Inp	Typically the Tag how is connected to this input would have is an Alias tag for the DI signal.	Signal Input (maintain or pulse for type=4)	**Inp** is this DigIn's supervised signal . The signal state (NO/NC) and signal type is specified in **Par.Type**. If the **Inp** signal is not OK, the **DigIn** module will eventually go into an alarm condition (indicated by the **Alarm** output).
InpW *	BOOL	X	Inp	Typically the Tag how is connected to this input would have is an Alias tag for the DI signal.	Signal Input Warning HAC signal **:W** Only available by DigInp2_AOI.	**InpW** is this DigIn2's supervised signal . The signal state (NO/NC) and signal type is specified in **Par.Type**. If the **InpW** signal is not OK, the **DigIn2** module will eventually go into an alarm condition (indicated by the **Alarm** output). This input generate only warning alarms. (**Par.WarnTime**) Appl example: Used by Drift switch with 2 contacts wired (Warning and Failure).

InpF *	BOOL	X	Inp	Typically the Tag how is connected to this input would have is an Alias tag for the DI signal.	Signal Input Failure HAC signal **:F** Only available by DigInp2_AOI.	**InpF** is this DigIn2's supervised signal . The signal state (NO/NC) and signal type is specified in **Par.Type**. If the **InpF** signal is not OK, the **DigIn2** module will eventually go into an alarm condition (indicated by the **Alarm** output). This input generate only warning alarms. (**Par.FailTime**) Appl example: Used by Drift switch with 2 contacts wired (Warning and Failure).
Alarm	BOOL		Out		Alarm indication	**Alarm** = 1 Identifies the supervised signal (**Inp**) has an active Warning (**Sta.W**) or Failure (**Sta.F**). The warning and failure conditions are latched and require acknowledge to reset the **Alarm** signal.
RdyOk	BOOL		Out	Used in the application program as an interlocks as appropriate.		**RdyOK** = 1 Identifies that the **MaxTime** Alarm (**Sta.MA**) is not Active and the **K** input is 1. **RdyOK** = 0 is latched (will not return to 1 automatically) and requires both Alarm Acknowledge and **Inp** to be normal before returning **RdyOk** to 1.
Out	BOOL		Out	Used in the application program as an interlocks or control signal	Filtered Signal Output	Represents the module's signal logic status after debouncing (signal filter/ Par.TrueDelay-Par.FalseDelay). 1= Signal ok 0= Signal not ok
* Only by DigInp2						

Module Data	**Data Type**	**User Config?**	**I/O**	**Configuration Required**	**Description**	**Notes**
Cmd.0 \SEL	BOOL		Inp		Toggle Bypass (from HMI)	**Cmd.0** = 1 is used to Enable / Disable the signal Bypass from the HMI. The Signal Bypass is indicated by the **Sta.BA** status.. Reserved for HMI Commands. Set in HMI, Reset in AOI. See Also **Par.EnBypass**
Cmd.3 \ACK	BOOL		Inp		Acknowledge / Alarm reset (from HMI)	**Cmd.3** = 1 is used to Reset Alarms from the HMI. Set in HMI, Reset in AOI. Reserved for HMI Commands. **Note:** -**Alarm** output is reset if alarm condition is no longer present! -The local start input (**G**) of a parent module (e.g. MotorN), will also initiate the acknowledge command IF the **Par.DisableGrpCheck** = 0!

Par.Type	DINT	X	Inp	For most instances where a Control Group (or Machine Group) is the Parent for the DigInp use type 0 For Most instances where a Device (motor) is the Parent for the DigInp can use type 0 or type 3	Module Type: NC Inp: 0=Static NC, 1=DynX NC, 2=DynY NC, 3=Dyn NC, 4=Speed Switch Pulse 8= Static NC without alarming NO Inp: 10=Static NO, 11=DynX NO, 12=DynY NO, 13=Dyn NO, 18= Static NO without alarming NC normally close (ok=1) / NO normally open (ok=0) **Note: Types 4,8 and 18 are only avaiable by DigIn_AOI**	**Par.Type** specifies the type of input signal (**Inp**)Type 0 - Static: Inp signal is always Tested for alarm conditions. Normally Closed (NC Inp - "1" = OK): 0 = Static NC - Always Supervised, 1 = DynX NC - Supervised when **<Parent>.Bus.11** = 1, (running X direc.) 2 = DynY NC - Supervised when **<Parent>.Bus12** = 1, (running Y direc.) 3 = Dyn NC - Supervised when **<Parent>.Bus10** = 1, (running X or Y) 4 = Speed Switch Pulse Supervised when **<Parent>.Bus 10** = 1, (running X or Y) 8= Static NC without alarming. Normally Open (NO Inp - "0" = OK): 10=Static NO - Always Supervised, 11=DynX NO - Supervised when **<Parent>.Bus11** = 1, 12=DynY NO - Supervised when **<Parent>.Bus12** = 1, 13= Dyn NO - Supervised when **<Parent>.Bus10** = 1, 18= Static NO without alarming
Par.TrueDelay	DINT	X	Inp	Default value could change based on reaction time of other software	Input ON Delay for Debouncing	Debounce Time setting in [ms]. When the **Inp** signal goes from False to True, delay this amount of time before using the change of signal. If the signal returns to 0 before the time expires, then the module output would be unchanged and the timer will reset. This delay is not applied when **Par.Type** = 4
Par.FalseDelay	DINT	X	Inp	Default value could change based on reaction time of other software	Input OFF Delay for Debouncing	Debounce Time setting in [ms]. When the **Inp** signal goes from True to False, delay this amount of time before using the change of signal. If the signal returns to 1 before the time expires, then the module output would be unchanged and the timer will reset. This delay is not applied when **Par.Type** = 4
Par. WarnTime	DINT	X	Inp	Value=0 or Must be greater then Par.MaxPuls	Warning Timeout Alarm, **Sta.WA** if 0->No Warning	Alarm Time setting in [ms] to delay before setting the Warning Timeout Alarm (**Sta.WA**) to 1. The alarm will occur after this time delay when the debounced **Inp** signal (after **FalseDelay** or **TrueDelay** expires) is in the incorrect (or alarm) state. By **DigInp_AOI** setting the **Par.WarnTime** to 0 will disables the Warning Alarm. *Note: The Internal Check must be 1 to enable the alarm.*

Par. FailTime *	DINT	X	Inp	Should be greater then Par.WarnTime	Failure signal Timeout Alarm, **Sta.MA**	Only available by DigInp2_AOI. Alarm Time setting in [ms] to delay before setting the Failure signal Timeout Alarm (**Sta.WA**) to 1. The alarm will occur after this time delay when the debounced **InpF** signal (after **FalseDelay** or **TrueDelay** expires) is in the incorrect (or alarm) state. *Note: The Internal Check must be 1 to enable the alarm.*
Par. MaxTime	DINT	X	Inp	Must be greater then Par.MaxPuls	Maximum Timeout Alarm,**Sta.MA** sets RdyOk=OFF	Alarm Time setting in [ms] to delay before setting the Maximum Timeout Alarm (**Sta.MA)**. The alarm will occur after this time delay when the debounced **Inp** signal (after FalseDelay or TrueDelay expires) is in the incorrect (or alarm) state. The **RdyOK** will be cleared (OFF) when a **Sta.MA** alarm is active (1). *Note: The Internal Check must be 1 to enable the alarm.*
Par. MaxPulse	DINT	X	Inp	Configure MaxTime and WarnTime appropriately for the device application	Max Pulse Duration in [ms] for Speed Detector, **Par.Type** 4	Alarm Maximum Time setting in [ms] between two adjacent Rising Edges of **Inp** signal. Use this setting only for a pulsed speed **Inp** signal (**Par.Type** = 4). The calculation result is used with the **MaxTime** and **WarnTime** timers internal to the AOI. (i.e. A failure due to MaxPulse will report as a **WarnTime** or **MaxTime** alarm)
Par. EnBypass	BOOL		Inp		Enable Bypass, 0=No Bypass allowed	Set **Par.EnBypass** to 1 to allow the **Inp** / **InpF** signal to be bypassed from the HMI using **Cmd.0**. If **Par.EnBypass** is set to 0, then NO bypass is allowed.
Par. DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules. If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set Sta.W or Sta.F	Set **Par.DisableGrpCheck** to 1 to prevent **<Parent>.Bus.26** (Check) from enabling the Module's **Internal Check**. The **Check** Input and **<Parent>.Bus.8** still enable the Module's **Internal Check**. This setting inhibits the Parent Check signal from enabling module alarms that provide Warning (**Sta.W**) or Failure (**Sta.F**) conditions
Par. ErrorWarning	BOOL		Inp		Severity/Errors are only warnings. If "1", Alarms are only indicated as warnings (Sta.W)	Set **Par.ErrorWarning** to 0 to indicate MaxTime (**Sta.MA**) and Availability (**Sta.KA**) Alarms as Failures (**Sta.F**). Set **Par.ErrorWarning** to 1 to indicate MaxTime (**Sta.MA**) and Availability (**Sta.KA**) alarms as Warnings (**Sta.W**). Note that **Sta.WA** alarms are reported as Warnings (**Sta.W**) regardless of this setting.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.

Par. AlarmGong Code	DINT of BOOL		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the Global.FailureGongCode or Global.WarningGongCode on the respective alarm.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Sta.RB	BOOL		Out		Input Signal OK Remove Bypass without effect	**Sta.RB** = 1 Identifies that there would be no affect by removing the Signal Bypass (**Sta.BA**). (i.e. instrument is OK) Note: Not avaiable at DigInp2_AOI.
Sta.WM *	BOOL		Out		Input Signal InpW OK- Mimic	Only available by DigInp2_AOI. **Sta.WM** = 1 Identifies that the **InpW** signal is supervised and that the debounced **InpW** is 1. (module internal signal status/filtered; 1=ok)
Sta.FM *	BOOL		Out		Input Signal InpF OK- Mimic Remove Bypass without effect	Only available by DigInp2_AOI. **Sta.FM** = 1 Identifies that the **InpF** signal is supervised and that the debounced **InpF** is 1. (module internal signal status/filtered; 1=ok)
Sta.KM	BOOL		Out		Device Ok/Available Mimic (allways active)	**Sta.KM** = 1 Identifies that the **Inp** signal is supervised and that the debounced **Inp** is 1. Note: For a DigIn with **Par.Type** = 4, that the "debounced **Inp"** means that the **MaxPulse** Time has not expired
Sta.F	BOOL		Out		Device Failure Mimc (Bus to Parent Module)	**Sta.F = 1** will identify that there is an Active MaxTime (**Sta.MA**) or Availability (**Sta.KA**) Alarm. Requires that the module is supervised (i.e. **Internal Check** = 1) and **Par.ErrorWarning** = 0.
Sta.W	BOOL		Out		Device Warning Mimc (Bus to Parent Module)	The **Sta.W** = 1 will identify that there is an active WarnTime (**Sta.WA**) alarm when **Par.ErrorWarning** = 0. If the **Par.ErrorWarning** = 1, then **Sta.W** will identify that there is an active WarnTime (**Sta.WA**), MaxTime (**Sta.MA**), or Availability (**Sta.KA**) Alarm. All Alarms Require that the module is supervised (i.e. **Internal Check** = 1).
Sta.BA	BOOL		Out		Bypass Enabled Alarm (Diagnostic)	**Sta.BA** = 1 Identifies that the **Inp** signal is bypassed.
Sta.KA	BOOL		Out		Availablability Alarm (Failure or Warning)	**Sta.KA** = 1 Identifies that the **Internal Check** is 1 and the **K** input signal is 0. This alarm is latched and reset by alarm acknowledge. Alarm is masked if **Internal Check** = 0. When this signal is latched the **Sta.KM** is reset.

Sta.WA	BOOL		Out		Warning Timeout Alarm (Warning)	**Sta.WA** = 1 Identifies that the **Internal Check** is 1 and there is an active **WarnTime** Alarm. Alarm is latched and reset by Alarm Acknowledge. Alarm is masked if **Internal Check** = 0. Refer to **Par.WarnTime**
Sta.MA	BOOL		Out		Maximum Timeout Alarm (Failure or Warning)	**Sta.MA** = 1 Identifies that the **Internal Check** is 1 and there is an active **MaxTime** Alarm. Alarm is latched and reset by Alarm Acknowledge. Alarm is masked if **Internal Check** = 0. Refer to **Par.MaxTime**
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
* Only by DigInp2						
Parent Bus	**Data Type**	**User Config?**	**I/O**	**Configuration Required**	**Description**	**Notes**
ParentBus	DINT	X	Inp		Bus from/to Parent Module (Group, Machine or Motor/SubSys/Valve)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This DigIn module reads information FROM the "Bus" and Writes status information TO the "Bus".

DigInp_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
TYPE	SINT	x	Inp	Direct allocated to the DigInp AOI	Module Type (0=Static; 1=DynX; 2=DynY; 3=Dyn; 4=SpeedD)	Input from DigInp module .Par.Type. 4/14 - Pulse type, <10 NC type, >10 NO type
Activate	BOOL		Inp	Input into simulation block	Activate the digital output simulation	Activation bit to enable **Out** signal simulation. Activation bit is recommended to tie to parent device running signal when device type is dynamic.
Out	BOOL	x	Out	Direct allocated to the DigInp AOI	Signal Output	Output signal to DigInp module from simulation.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
IN_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Input Fault from HMI	IN_Fault bit is connected with HMI Signal Fault switch. 0=Ok, 1=Fault
PULSE	BOOL				Internal bit to identify if device type is pulse	PULSE internal bit indicates if device type is pulse.
PULSE_TIME	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Pulse Time preset (= On Time + Off Time)	Pulse_time is used to set the length of pulse from HMI simulation faceplate
PULSE_TIME_ON	DINT				Signal On Time	Pulse_Time_On internal variable is used to store the pulse on time
Pulse_Timer	TIMER				Pulse timer used to generate pulse output	Pulse timer is used to generate pulse output signal when device type is pulse

DigInp2_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
TYPE	DINT	x	Inp	Direct allocated to the DigInp2 AOI	Module Type (0=Static; 1=DynX; 2=DynY; 3=Dyn)	Input from DigInp2 module .Par.Type. <10 NC type, >10 NO type
Activate_W	BOOL		Inp	Input into simulation block	Activate the digital Out_W simulation	Activation bit to enable **Out_W** signal simulation. Activation bit is recommended to tie to parent device running signal when device type is dynamic.
Activate_F	BOOL		Inp	Input into simulation block	Activate the digital Out_F simulation	Activation bit to enable **Out_F** signal simulation. Activation bit is recommended to tie to parent device running signal when device type is dynamic.
OUT_W	BOOL	x	Out	Direct allocated to the DigInp2 AOI	Signal Warning Output	Output variable W to DigInp2 module from simulation.
OUT_F	BOOL	x	Out	Direct allocated to the DigInp2 AOI	Signal Failure output	Output variable F to DigInp2 module from simulation.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
IN_Fault_F	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Input F Fault from HMI	IN_Fault_F bit is connected with HMI Signal F Fault switch. 0=Ok, 1=Fault
IN_Fault_W	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Input W Fault from HMI	IN_Fault_W bit is connected with HMI Signal W Fault switch. 0=Ok, 1=Fault

DigPulse

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Inp	BOOL	X	Inp	Typically the Tag how is connected to this input would have is an Alias tag for the DI signal.	Signal Input pulse or maintain	**Inp** is this DigIn's supervised signal . The signal type is specified in **Par.Type**. If the **Inp** signal is not OK, the **DigPulse** module will eventually go into an alarm condition (indicated by the **Alarm** output).
K	BOOL	X	Inp	Typically this would be the voltage source signal for an instrument. I.e. the signal from the instrument's breaker is called **K** before going to the instrument and back to the PLC as a Inp signal. Set to Log_1 if No the K signal.	Device Ok/Available (Power supply, etc)	Set **K** to 1 when the Field Device providing the Inp signal is powered and considered OK.
VSD	DINT	X	Inp	**Set to 100** if no Tag variable from the VSD used.	Variable Speed Device Input [0-100%] (Internal range 25-100%)	The VSD input value will adjust the supervision timer during the **Acc** and **Run** state accordingly. ALL Timer parameter are related to 100% VSD. Note: The module use a fix factor(x4.0) if the value <=25% e.g. if VSD input=20% then time factor is still x4.0 VSD input=100% then the time factor is 1.0 VSD input=55% then the time factor is 1.82 etc
Running	BOOL		Inp	**Optional** This could be a alternative input to <ParentBus> runX or runY. Using if DigPulse not a child of a Motor.	Monitoring Device is running (optional for BusRun)	**Optional** Input for Bus signal "Run" **<ParentBus>.10** . If this module is used as a stand alone module, then we have the possibility to start the monitoring states with this input. See Sta.Acc/Run/Dec Any transition 0->1 edge set the internal monitor state to "**Acc**" accelerating. After the timer **Par.DelayRunTimer** expires he change to state "**Run**" running. If **Running** change from 1->0 then the monitor state change to "**Dec**" deceleration. After the timer **Par.DelayStopTimer** expires he change to state "**Stopped**" .

Check	BOOL		Inp		Used to Enable Module Alarms without the **ParentBus.Check** Signal	The **"Internal AOI Check"** signal enables module alarms and is driven by the Input **Check**, **<Parent>.Bus.26 (Check)**, or **<Parent>.Bus.8 (LocalCheck)**. Set **Check** = 1 (maintained) with the PLC application software to enable module alarms when the **Inp** signal should be supervised without check signals from a parent CtrlGrp or Device (i.e. Motor). See Also **Par.DisableGrpCheck**
RdyOk	BOOL		Out	Used in the application program as an interlocks as appropriate.		**RdyOK** = 1 Identifies that the **MaxTime** Alarm (**Sta.MA**) is not Active and the **K** input is 1. **RdyOK** = 0 is latched (will not return to 1 automatically) and requires both Alarm Acknowledge and **Inp** to be normal before returning **RdyOk** to 1.
Alarm	BOOL		Out		Alarm indication	**Alarm** = 1 Identifies the supervised signal (**Inp**) has an active Warning (**Sta.W**) or Failure (**Sta.F**). The warning and failure conditions are latched and require acknowledge to reset the **Alarm** signal.
Out	BOOL		Out	Used in the application program as an interlocks or control signal	Signal ok Output	Represents the module's Input signal logic status **Inp** 1= Signal ok, 0= Signal not ok

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Cmd.0 \SEL	BOOL		Inp		Toggle Bypass (from HMI)	**Cmd.0** = 1 is used to Enable / Disable the signal Bypass from the HMI. The Signal Bypass is indicated by the **Sta.BA** status.. Reserved for HMI Commands. Set in HMI, Reset in AOI. See Also **Par.EnBypass**
Cmd.3 \ACK	BOOL		Inp		Acknowledge / Alarm reset (from HMI)	**Cmd.3** = 1 is used to Reset Alarms from the HMI. Set in HMI, Reset in AOI. Reserved for HMI Commands. **Note:** -**Alarm** output is reset if alarm condition is no longer present! -The local start input (**G**) of a parent module (e.g. MotorN), will also initiate the acknowledge command IF the **Par.DisableGrpCheck** = 0!
Par. EnBypass	BOOL		Inp		Enable Bypass, 0=No Bypass allowed	Set **Par.EnBypass** to 1 to allow the **Inp** / **InpF** signal to be bypassed from the HMI using **Cmd.0**. If **Par.EnBypass** is set to 0, then NO bypass is allowed.

Par. DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules. If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set Sta.W or Sta.F	Set **Par.DisableGrpCheck** to 1 to prevent **<Parent>.Bus.26** (Check) from enabling the Module's **Internal Check**. The **Check** Input and **<Parent>.Bus.8** still enable the Module's **Internal Check**. This setting inhibits the Parent Check signal from enabling module alarms that provide Warning (**Sta.W**) or Failure (**Sta.F**) conditions
Par. ErrorWarning	BOOL		Inp		Severity/Errors are only warnings. If "1", Alarms are only indicated as warnings (Sta.W)	Set **Par.ErrorWarning** to 0 to indicate MaxTime (**Sta.MA**) and Availability (**Sta.KA**) Alarms as Failures (**Sta.F**). Set **Par.ErrorWarning** to 1 to indicate MaxTime (**Sta.MA**) and Availability (**Sta.KA**) alarms as Warnings (**Sta.W**). Note that **Sta.WA** alarms are reported as Warnings (**Sta.W**) regardless of this setting.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par.Type	SINT	X	Inp		Function Type: 0= Pulse Input 1= NC Maintain Input 2= NO Maintain Input	**Par.Type** specifies the type of input signal (**Inp**). Function Typ: 0=Pulse Inp; 1=NCmaintain; 2=Nomaintain Note: Maximal Input Frequency is given by the Program Scantime x2, so we can detect any signal edges.
Par. PulseMemoryTime	DINT	X	Inp		This is amount of time [ms] to remember a rising edge of Pulse signal during Motor stop.	The Parameter is used to identifying the amount of time to remember that a rising edge of a pulse occured. This is intended to validate the pulse signal only during the **"Dec"** or **"Stopped"**-monitoring states.
Par. DelayRunTime	DINT	X	Inp		Time delay after motor ON- until state running[ms]	Time to delay between the **localBus bit "Run"** or Input **Running** 0->1 transition and considering the equipment Running. This time is in charge of the monitor states control. Time delay between **Sta.Acc** till **Sta.Run** is active.
Par. DelayStopTime	DINT	X	Inp		Time delay after motor OFF- until state stop[ms]	Time to delay between the **localBus bit "Run"** or Input **Running** 1->0 transition and considering the equipment Stopped. This time is in charge of the monitor states control. Time delay between **Sta.Dec** till **Sta.Stopped** is active.

Par. MaxTime	DINT	X	Inp		Failure delay Timer during run [ms]	Alarm Time setting in [ms] to delay before setting the Maximum Timeout Alarm (**Sta.MA)**. The alarm will occur after this time delay when the **Inp** signal is in the incorrect (or alarm) state. The **RdyOK** will be cleared (OFF) when a **Sta.MA** alarm is active (1). *Note: This timer will ajusted with the VSD % input if VSD >25%.* *e.g. if VSD input=20% then time factor is still x4* *If Sta.Acc: FailureTimer = (100 / VSD input) * MaxTime +ExtraAccTime* *If Sta.Run: FailureTimer = (100 / VSD input) * MaxTime* *If Sta.Dec: FailureTimer = MaxTime + ExtraDecTime*
Par. WarnTime	DINT	X	Inp	Par.WarnTime =0 disable warning alarm	Warning delay Timer during run [ms]	Alarm Time setting in [ms] to delay before setting the Warning Timeout Alarm (**Sta.WA)**. The alarm will occur after this time delay when the **Inp** signal is in the incorrect (or alarm) state. **Par.WarnTime =0** disable warning alarm Sta.WA Note: This timer will ajusted with the VSD % input if VSD >25%. e.g. if VSD input=10% then time factor is still x4 *If Sta.Acc: WarnTimer = (100 / VSD input) * WarnTime +ExtraAccTime* *If Sta.Run: WarnTimer = (100 / VSD input) * WarnTime* *If Sta.Dec: FailureTimer = WarnTime + ExtraDecTime*
Par. ExtraAccTime	DINT	X	Inp		Add time to Alarm timer in Acc state[ms]	Time added to the Warn & MaxTime during **Sta.Acc** monitor state. 0ms = no additional time.
Par. ExtraDecTime	DINT	X	Inp		Add time to Alarm timer in Dec state[ms]	Time added to the Warn & MaxTime during **Sta.Dec** monitor state. 0ms = no additional time.
Par. AlarmGongCode	DINT of BOOL		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the Global.FailureGongCode or Global.WarningGongCode on the respective alarm.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag

Sta.BA	BOOL		Out		Bypass Enabled Alarm (Diagnostic)	**Sta.BA** = 1 Identifies that the **Inp** signal is bypassed.
Sta.RB	BOOL		Out		Input Signal OK Remove Bypass without effect	**Sta.RB** = 1 Identifies that there would be no affect by removing the Signal Bypass (**Sta.BA**). (i.e. instrument is OK)
Sta.KM	BOOL		Out		Device Ok/Available Mimic (allways active)	**Sta.KM** = 1 Identifies that the **Inp** signal is supervised and that the **Inp** signal is OK. If **K** input =0 or the FailureTimer expire then the **Sta.KM** =0. Show the moduls healthy.
Sta.F	BOOL		Out		Device Failure Mimc (Bus to Parent Module)	**Sta.F = 1** will identify that there is an Active MaxTime (**Sta.MA**) or Availability (**Sta.KA**) Alarm. Requires that the module is supervised (i.e. **Internal Check** = 1) and **Par.ErrorWarning** = 0.
Sta.W	BOOL		Out		Device Warning Mimc (Bus to Parent Module)	The **Sta.W** = 1 will identify that there is an active WarnTime (**Sta.WA**) alarm when **Par.ErrorWarning** = 0. If the **Par.ErrorWarning** = 1, then **Sta.W** will identify that there is an active WarnTime (**Sta.WA**), MaxTime (**Sta.MA**), or Availability (**Sta.KA**) Alarm. All Alarms Require that the module is supervised (i.e. **Internal Check** = 1).
Sta.KA	BOOL		Out		Availablability Alarm (Failure or Warning)	**Sta.KA** = 1 Identifies that the **Internal Check** is 1 and the **K** input signal is 0. This alarm is latched and reset by alarm acknowledge. Alarm is masked if **Internal Check** = 0. When this signal is latched the **Sta.KM** is reset.
Sta.WA	BOOL		Out		Warning Timeout Alarm (Warning)	**Sta.WA** = 1 Identifies that the **Internal Check** is 1 and there is an active **WarnTime** Alarm. Alarm is latched and reset by Alarm Acknowledge. Alarm is masked if **Internal Check** = 0. Refer to **Par.WarnTime**
Sta.MA	BOOL		Out		Maximum Timeout Alarm (Failure or Warning)	**Sta.MA** = 1 Identifies that the **Internal Check** is 1 and there is an active **MaxTime** Alarm. Alarm is latched and reset by Alarm Acknowledge. Alarm is masked if **Internal Check** = 0. Refer to **Par.MaxTime**
Sta.Pmode	BOOL		Out		Module is in pulse mode	Module is operating as pulse motion dedector. The moduel expect signal edges at the **Inp** which have to be faster then the Par.MaxTime.
Sta.Acc	BOOL		Out		Monitoring at accelerating state	Device (Motor) is accelerating. This state is active if the motor starts (by a transition **"Running"** 0->1)
Sta.Run	BOOL		Out		Monitoring at running state	Device (Motor) is running and on his operating speed. This state is active if the motor is started and the delay timer (**Par.DelayRunTime**) is expired.
Sta.Dec	BOOL		Out		Monitoring at deceleration state	Device (Motor) are stopping (slow down) and switched off. This state is active if the motor is Off (by a transition "Running" 1->0)and is still active till the delay timer **(Par.DelayStopTime)** is expired.

Sta.Stopped	BOOL		Out		Monitoring stopped	Device (Motor) is stopped. This state is active if the motor is Off and the delay timer **(Par.DelayStopTime)** is expired.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	Inp		Bus from/to Parent Module (Group, Machine or Motor/SubSys/Valve)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This DigIn module reads information FROM the "Bus" and Writes status information TO the "Bus".

DigPulse_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Start	BOOL	x	Inp	Parent device running Input into simulation block	To activate the digital pulse output simulation	Start bit indicate when parent device of DigPulse is started
Type	DINT	x	Inp	Direct allocated to the DigPulse AOI	Function type, 0=Pulse Inp; 1= NC Maintain; 2= NO Maintain	Input from DigPulse module .Par.Type
PulseMemoryTime	DINT	x	Inp	Direct allocated to the DigPulse AOI	Amount of time to remember a rising edge of Pulse signal	Input from DigPulse module .Par.PulseMemoryTime
WarnTime	DINT	x	Inp	Direct allocated to the DigPulse AOI	Warning delay time during run (ms)	Input from DigPulse module .Par.WarnTime
Out	BOOL	x	Out	Direct allocated to the DigPulse AOI	Output	Output signal to DigPulse module from simulation
VSD_OUT	DINT	x	Out	Direct allocated to the DigPulse AOI	Variable speed device output (0-100%)	VSD_OUT variable to DigPulse module from simulation

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Acc_time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Acceleration time from HMI	Acceleration time is set from HMI simulation faceplate. During this period frequency of pulses increase from 0 to run frequency. Acceleration pulse frequency is calculated from WarnTime, Acc_Time and VSD_Out variables.
Acc_Timer	TIMER				Acceleration timer	Timer block for Acceleration. Preset is set from Acc_time variable.
Act_pulse_len	DINT				Internal bit to store actual pulse-on length	Internal variable to store the pulse on time. Actual pulse-on length is depended on parameter settings and which phase is currently active (Acceleration, Run, Deceleration).
Dec_Time	DINT			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Deceleration time from HMI	Deceleration time is set from HMI simulation faceplate. During this period frequency of pulses decrease from run frequency to 0. Deceleration frequency is calculated from WarnTime, Dec_Time and VSD_out variables.
Dec_Timer	TIMER				Deceleration timer	Timer block for Deceleration. Preset is set from Dec_Time variable.

IN_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Input signal fault from HMI	IN_Fault bit is connected with HMI Signal Fault switch. 0=Ok, 1=Fault
PULSE	BOOL				Internal bit to Identify if device type is pulse	PULSE internal bit indicates if device type is pulse.
Pulse_Time	TIMER				Pulse timer used to generate pulse output	Pulse timer is used to generate pulse output signal when device type is pulse

AnaInp and AnaInpC

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Inp	REAL		Inp	Ideally this tag value will represent the true field measurement of the signal (i.e. 4-20 mA)	Raw Value analog Input(**$-3.4e^{+38}$..$3.4e^{+38}$) ->** moved to **PV** if **Scaling**=0	**Inp** is configured on the AOI instance and is the raw signal to be monitored by the AnaInp module. The input value can be either real or integer. Oftentimes, the **Inp** tag is aliased to a hardware module input signal. The AnaInp module Scales (**Par.InpScale**) and Filters (**Par.Filtering**) this Inp signal to calculate the **PV** signal output.
Check	BOOL		Inp		Enable Alarm Outputs (Override Parent Module Check)	The **"Internal AOI Check"** signal enables module alarms and is driven by the Input **Check**, **<Parent>.Bus.26 (Check)**, or **<Parent>.Bus.8 (LocalCheck)**. Set **Check** = 1 (maintained) with the PLC application software to enable module alarms when the Inp signal should be supervised without check signals from a parent CtrlGrp or Device (i.e. Motor). See Also **Par.DisableGrpCheck** **An unchecked module will not fault or alarm.**
K	BOOL	X	Inp	Enter Log_1 if not used.	Ok/Available	**K** is configured on the AOI instance and is set to 1 when the field device providing the **Inp** signal is powered and considered OK. Set **K** = 1 (Log_1) if the signal is not physically available. *Example: For a four wire analog signal, this could be the On/Off status of the instruments 120V power. See also Sta.KA/KAM*
ERR	BOOL		Inp	controlled by user code if it's used or set Tag value.	Transmitter Error ->**Sta.ERR**	**ERR** is set to 1 by the PLC application program when there is a sensor error is present and set to 0 if no error is present or the signal is physically not available. Not configured on the AOI instruction. Example: The Inp signal may come from a pressure transducer and through a signal transmitter. This could be an error status signal from the transmitter or transducer. See also **Sta.ERR/ERRM**
ESP	BOOL		Inp	controlled by user code if it's used or set Tag value.	Enable Setpoint ->sent to **Sta.ESP**	**ESP** is set by the PLC application software to Enable the setpoint. If **ESP** = 1 then **SP** = **Set.SP** and the Set.SP will be visible on the HMI Faceplate. There is no control ability internal to the AnaInpC module only data transfer. See also **SP** & **Set.SP**
EMA/EHA /ELA/ENA	BOOL		Inp	controlled by user code or set Tag value.	Enable Max..Min Alarm->**Sta.MA/HA /LA/NA** if **Par.Type**=2/3/6/7	**EMA/EHA/ELA/ENA** is set to 1 to individually enable the threshold alarms (**Sta.MA/HA/LA/NA**) if **Par.Type** = 2, 3, 6, or 7 (enable Alarm). The PLC application program can control these bits if desired or set directly at the Tag structure.
AlarmReset	BOOL		Inp		Alarm Reset	Set **AlarmReset** = 1 (Pulse) by the PLC application program to reset all alarms if the alarm condition has returned to normal. Same functionality as Acknowledge (**Cmd.3)** on the HMI Faceplate.

CM/CH/CL /CN	BOOL		Out	For example, with a bin level signal, the CH bit would be used to stop filling the bin.	Control Bits Max/High, ON if **PV**>**Par.CMV/CHV /CLV/CNV**	**CM/CH/CL/CN** are are control output bits that are used in the PLC application software to trigger other events. The control setpoints are specified in **Par.CMV/CHV/CLV/CNV**. Refer to the Reference Guide Deadband Diagram for a signal diagram. Example excluding deadband influence: If **PV** > **Par.CMV/CHV** then **CM/CH** = 1. If **PV** < **Par.CLV/CN** then **CL/CN** = 1.
Alarm	BOOL		Out		Alarm Indication <- **Sta.F** OR **Sta.W** (Failure or Warning)	**Alarm** = 1 Identifies that either a Failure (**Sta.F**) or Warning (**Sta.W**) alarm is active. The raw alarm signals include threshold alarms (**Sta.MA/HA/LA/NA**), Sensor Error alarms (**Sta.ERR**), and Availability alarms (**Sta.KA**). *Note: Internal Check must be 1 for this condition. See also Check*
RdyOk	BOOL		Out	Interlock to Parent Module	Ready Ok Signal -> Interlock to Parent Module	**RdyOk** = 1 Identifies that the **PV** signal is Valid. This means that there is no Max/Min Threshold alarm (**Sta.MA/NA**), Sensor Error alarm (**Sta.ERR**), or Availbility alarm (**Sta.KA**) active. The **RdyOK** signal is used in the PLC application software where needed to validate the **PV** signal prior to calculation/usage. See also **Check**
PV	REAL		Out		Process Value Scaled if **Par.InpScale**>0 else **PV**=**Inp**	**PV** contains the output value of the module. It is either the scaled filtered **Inp** signal or the bypassed signal value (**Set.PVZ**). **PV** should be used throughout the PLC application software for any calculations or control (if outside of the **CM/CH/ CL/CN** abilities)
SP	REAL		Out		Setpoint Output Scaled (in PV units) Setpoint (SP) indication only for reference reason.	**SP** contains a setpoint command from the HMI Faceplate. This provides the functionality to enter a setpoint command **SP** on the same HMI faceplate as monitoring the **PV** signal. By using this SP it gives the operator only a reference on which value the analog input have to be. The Setpoint entry can further also use for some user analog control. Application Example: An analog valve has both postion command (**<AnaInp>.SP**) and position feedback (**<AnaInp>.Inp**) .In this example, this SP signal may be used in the <ActMod> module as a input. The **Set.SP** is written into the **SP** when enabled by **ESP**. To provide bumpless transfer functionality, if **ESP** = 0, then **Set.SP** = **SP**.
Orange Highlight	= only available with AnaInp C					

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Cmd.0\EZ	BOOL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk -Unit depending [PV] units	Toggle **Sta.RZ** to select **Set.PVZ** if enabled by **Par.EnBypass**	**Cmd.0** = 1 will Toggle the Bypass (Replacement) status **Sta.RZ** if Bypass functionality is enabled by **Par.EnBypass**. Reserved for HMI Faceplate Interface. (Set in HMI, Reset in AOI)
Cmd.3\ACK	BOOL		Inp		Alarm Acknowledge	**Cmd.3** = 1 will Reset all alarms if the alarm condition has returned to normal. Reserved for HMI Faceplate Interface. Same functionality as **AlarmReset**. (Set in HMI, Reset in AOI)
Set.SP	REAL		Inp		Setpoint Input if enabled by **Sta.ESP** indication only for reference reason.	**Set.SP** contains the Setpoint value from the HMI Faceplate. *See also Sta.ESP and SP*
Set.PVZ	REAL		Inp		Replacement/Substitution Value if enabled by **Sta.RZ**	**Set.PVZ** contains the Bypass (Replacement) value to be used for **PV** if enabled by **Par.EnBypass** and **Sta.RZ** = 1. The value has the same units as the **PV** signal. *See also Par.EnBypass and Par.AutoBypass*
Set.CMV/CHV/CLV/CNV	REAL		Inp		Control Max..Min Threshold Values ->Ctrl Bits **CM/CH/CL/CN**	**Set.CMV/CHV/CLV/CNV** contains the values specifying the desired Threshold Control Bit setpoints for the **CM/CH/CL/CN** alarms. The value has the same units as the **PV** signal. *See also Par.Deadband*
Set.MV/HV/LV/NV	REAL		Inp		Max..Min Alarm Limit Value ->**Sta.MA/HA/LA/NA**	**Set.MV/HV/LV/NV** contains values specifying the desired Threshold Limit Alarm setpoints for the **Sta.MA/HA/LA/NA** alarms. The value has the same unit as the **PV** signal. *See also Par.AlarmDelay_MA/HA/LA/NA and Par.Deadband*
Par.Type **Par.Type.0** = Dyn **Par.Type.1** = Alarm **Par.Type.2** = Control	SINT	X	Inp		0=**Disable** Control and Alarm (Scaling only) 2=Static, 3=Dyn Enable Alarm->Bits Sta.MA/HA/LA/NA 4=Static, 5=Dyn Enable Control->Bits CM/CH/CL/CN 6=Static, 7=Dyn Enable both, Control and Alarm **Note**, Alarms are set only if **EMA/EHA/ELA/ENA** =1	Set **Par.Type** to specify the desired AnaInp module release of Alarm (**Sta.MA/HA/LA/NA**) and Control Bits (**CM/CH/CL/CN**). *Static* specifies that the bits are always and *Dynamic (Dyn)* specifies that the bits are only released when Parent device is running.(<Parent>.Bus.10 = 1 **Run**). The **Internal Check** signal is required for alarm bits to activate. Valid **Par.Type** settings are: 0 --> Disable Control & Alarm Bits (Scaling Only) 2 (Static), 3 (Dyn) --> Enable Alarm Bits 4 (Static), 5 (Dyn) --> Enable Control Bits 6 (Static), 7 (Dyn) --> Enable Alarm & Control Bits
Par.InpScale	SINT	X	Inp			Set **Par.InpScale** to an integer (0..7) to specify the raw scaling data (InRawMax/Min) to be used for scaling the **Inp** signal . The setting specifies the x element of the **Global.Par.AnaInpScale[x]** array. Set **Par.InpScale** = 0 there is no scaling performed (i.e. **PV** = **Inp**). See also **Global.Par.AnaInpScale**

Par.InEUMax	REAL	X	Inp	PV-units	Max/Min **PV**/**SP** Scaling Range in Engineering Units (PV units)	Set **Par.InEUMax** to specify the **PV** signal Engineering Maximum value to be used for scaling the **Inp** signal. *Example: Set Par.EUMax = 0 and Par.EUMin = 100 for a 4-20 mA Inp signal that measures 100 - 0% respectively. In this case the Sta.NegGrad = 1. See also Val.MZ/NZ*
Par.InEUMin	REAL	X	Inp	PV-units	If **InEUMax**<**InEUMin** then Negative Gradient **Sta.NegGrad**=ON in this case PV/SP Max Val.MZ=InEUMin and Min NZ=InEUMax	Set **Par.InEUMin** to specify the **PV** signal Engineering Minimum value to be used for scaling the **Inp** signal. *Example: Set Par.EUMax = 0 and Par.EUMin = 100 for a 4-20 mA Inp signal that measures 100 - 0% respectively. In this case the Sta.NegGrad = 1. See also Val.MZ/NZ*
Par.Filtering	SINT	X	Inp	Typically > 70. Unit [%]	0%=Disable Filter ->Use external Ladder or FBD Filter instruction ->PV =Aux.PVY 1..99%=Filter ->PV=PVold+1..99%(PVnew-PVold)/100*SampleTime/s 100%=Bypass Filter ->PV scaled or PV=Inp if Par.InpScale=0 Note, 1..99% is the max change (gradient) in % of PV per sec.	Set **Par.Filtering** to a value (0-100%) to specify the filtering effect internal to the AnaInp module. The internal filter is a low pass filter. **Par.Filtering** = 0 will disable the internal filter and enable external filtering PV = **Val.PVY** (See **Aux.PVX/PVY**). **Par.Filtering** = 100 will Bypass the internal filter (**PV** = scaled **Inp**) **Par.Filtering** = 1..99 will enable the internal filter and the filter algorithm is **PV** = **PV**$_{old}$ + **Par.Filtering***SampleTime*(**PV**$_{new}$-**PV**$_{old}$) / 100 where SampleTime is the time between **Inp** samples. *-Note: Par.Filtering is the maximum % change (gradient) of the PV per second. (i.e. 70% relates to a 1 second low pass filter) A higher Par.Filtering setting will allow the PV signal to change faster (i.e. less filtering). See also Par.SampleRate and GlobalScan. Par.SampleRate is related to SampleTime but they are different!* *Note: Parameter are tested by: Par.Filter[%] x Par.SampleRate[ms] >99'999 (i.e. filter factor >1) then we bypass the Filter -> PVY=PVX.* *Important: Par.SampleRate take influence to the filter behavior.*
Par.Deadband	REAL		Inp	Typically 1% PV-units	Deadband/Hysteresis **0..3.4e**$^{+38}$ used for all Alarms and Ctrl Bits	Set **Par.Deadband** to specify the value of deadband used for alarming bits (**Sta.MA/HA/LA/NA**) and control bits (**CM/CH/CL/CN**). The deadband value has the same units as the **PV**. Refer to the Reference Guide Deadband Diagram for a signal diagram. Max & High control and alarm signals are considered normal after the **PV** <= **Set.MA** - **Par.Deadband**. Min & Low control and alarm signals are considered alarm whe **PV** >= **Set.NA** + **Par.Deadband**. *Example: If the high alarm limit is 10.5 and the deadband is 0.5, then the signal must return to less than 10.0 before the alarm will clear.*

Par. MaxClamp	REAL		Inp	PV-units	Maximum Range Clamping Deadband	Set **Par.MaxClamp** to specify the value of clamping. Clamping is the tolerance from Max signal value to consider the signal to really be the Max value. Refer to the Reference Guide Clamping Diagram for the signal diagram. *Example: Par.InEUMin = 0, Par.In.EUMax = 100, and Par.MaxClamp = 1.0. Case 1: If the scaled filtered Inp signal > 99 then PV = 100 Case 2: If the scaled filtered Inp signal is between 1 and 99 then PV equals the scaled filtered Inp value.*
Par. MinClamp	REAL		Inp	PV-units	Minimum Range Clamping Deadband	Set **Par.MinClamp** to specify the value of clamping. Clamping is the tolerance from Min signal value to consider the signal to really be the Min value. Refer to the Reference Guide Clamping Diagram for the signal diagram. *Example: Par.InEUMin = 0, Par.In.EUMax = 100, and Par.MinClamp = 1. Case 1: If the scaled filtered Inp signal < 1 then PV = 0. Case 2: If the scaled filtered Inp signal is between 1 and 99 then PV equals the scaled filtered Inp value.*
Par.SampleRate	DINT	X	Inp	units [ms] Typically set to 1000ms	0=Always or >400 Preset time [ms] to update all Inputs. The Accum time since the last sample is used for the Filter SampleRate	Set Par.SampleRate to specify the desired frequency to execute the internal AnaInp logic. This is the frequency that the PV will be updated. Note: The SampleRate can't go faster then 400ms! This is given by the MMCL Scan Control Function, see also Global.Scan1.Preset. Important: The ApplyPar input in the SysGrp must be set for a change to the SampleRate parameter to take effect. Note: SampleRate take influence to the Filter configuration; see Par.Filter.
Par.AlarmDelay_MA	DINT		Inp	units [ms]	Delay Time for maximum Alarm ->Sta.MA	Set **Par.AlarmDelay_MA** to a value to specify the Time Delay in ms used for maximum Threshold alarm. (**Sta.MA**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.AlarmDelay_HA	DINT		Inp	units [ms]	Delay Time for high Alarm ->Sta.HA	Set **Par.AlarmDelay_HA** to a value to specify the Time Delay in ms used for high Threshold alarm. (**Sta.HA**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.AlarmDelay_LA	DINT		Inp	units [ms]	Delay Time for low Alarm ->Sta.LA	Set **Par.AlarmDelay_LA** to a value to specify the Time Delay in ms used for low Threshold alarm. (**Sta.LA**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.AlarmDelay_NA	DINT		Inp	units [ms]	Delay Time for minimum Alarm ->Sta.NA	Set **Par.AlarmDelay_NA** to a value to specify the Time Delay in ms used for minimum Threshold alarm. (**Sta.NA**). The alarm condition must be present for this amount of time before the alarm will become active.

Par.EnBypass	BOOL		Inp		Enable Bypass ->Manually **Set.PVZ**	Set **Par.EnBypass** = 1 to specify that the **PV** signal can be bypassed (replaced) and that bypass information should be visible on the HMI Faceplate. This setting only enables the ability to bypass the **PV**. Set Par.EnBypass = 0 to prevent the **PV** signal from being bypassed. See also **Cmd.0**, **Par.AutoBypass**, **Set.PVZ**, and **Sta.RZ**.
Par.AutoBypass	BOOL		Inp		Auto Bypass ->Automatically **Set.PVZ** if **Sta.ERR**, Reset Manually	Set **Par.AutoBypass** = 1 to specify that the **PV** bypassed should be automatically bypassed (**Sta.RZ**) if a sensor error alarm (**Sta.ERR**) or Availability alarm (**Sta.KA**) become active. Resetting the bypass condition is Always done manually with **Cmd.0**. See also **Set.PVZ**. **Note:** When this bit is set then **Sta.ERR** and **Sta.KA** -Alarms send only a warning to the Control Group, to prevent a group stop!
Par.DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** to 1 to prevent **Check -**condition from Parent module (<Parent>.Bus.26) to enabling the Module's **Internal Check**. The **Check** Input and **<Parent>.Bus.8** still enable the Module's **Internal Check**. If set to 1 then no warning(**Sta.W**) or error(**Sta.F**) will be passed on ParentBus. The **group does not** receive fault / warnings Info .Alarms however is Indicated on HMI. Note: An unchecked module will not fault or alarm.
Par.ErrorWarning	BOOL		Inp		Set module as warning device only. Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by Sta.W IF ON, Alarms are indicated by Sta.F	Set **Par.ErrorWarning** = 1 to specify that **Sta.ERR**, **Sta.KA**, **Sta.MA**, and **Sta.NA** alarms will be reported as Warning alarms (**Sta.W**) only. Set **Par.ErrorWarning** = 0 to specify that these alarms should be reported as Failure alarms (**Sta.F**). Examle: A device with set Par.ErrorWarning bit is configured as a "Warnning device" only -> a failure will never stop the Group.
Par. DisableSensorErr	BOOL		Inp		Disable sensor error reporting	Set **Par.DisableSensorErr** = 1 to specify that the device will not report sensor error to parent control group.
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.

Par.AlarmGongCode	DINT of BOOLs		Inp		Set Alarm Gong adresse# Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Sta.ESP	BOOL		Out		Enable Setpoint Input	**Sta.ESP** = 1 Identifies that the SP is enabled and **SP** = **Set.SP**. If **Sta.ESP** = 0, then **Set.SP** = **SP**.
Sta.RZ	BOOL		Out		Bypass Enabled, **PV**=**Set.PVZ** Replacement/Substitution Value	**Sta.RZ** = 1 Identifies that **PV** signal is bypassed (Replaced). In this condition the **PV = Set.PVZ**. This condition is toggled with **Cmd.0** from the HMI**.** *Refer also to Par.EnBypass.*
Sta.KM	BOOL		Out		Device Available Ok Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **K, ERR** or **limit alarms**. Indicated in HMI as purple color.
Sta.F	BOOL		Out		Device Failure Mimc (Bus to Module)	**Sta.F** = 1 Identifies that the module's **Internal Check** = 1 and there is an active failure alarm. Failure alarms are masked (inhibited) if **Par.ErrorWarning** = 1. Otherwise Failure alarms include **Sta.ERR**, **Sta.KA**, **Sta.MA**, and **Sta.NA** alarms.
Sta.W	BOOL		Out		Device Warning Mimc (Bus to Parent Module)	**Sta.W** = 1 Identifies that the module's **Internal Check** = 1 is and there is an active warning alarm. Warning alarms always include **Sta.HA/LA** alarms. If **Par.ErrorWarning** = 1, then Warning alarms also include **Sta.ERR**, **Sta.KA**, **Sta.MA**, and **Sta.NA** alarms.
Sta.ERR	BOOL		Out		Sensor Error Message	**Sta.ERR** = 1 Identifies that there is a Sensor Error (**ERR**) or Out of Range problem when the module is supervised. **Inp** Under / Over -Range (broken wire) See *Global.Par.AnaInpScale[x].OverrangeLimit -// .UnderrangeLimit*
Sta.ERRM	BOOL		Out		Sensor Error Mimic	**Sta.ERRM** = 1 Identifies that there is a Sensor Error (ERR) or Out of Range problem.
Sta.KA	BOOL		Out		Availability Alarm Message	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)

Sta.KAM	BOOL		Out		Availability Alarm Mimic	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.
Sta.MA/HA/LA/NA	BOOL		Out		Max..Min Limit Alarmß**Set.MV/HV/LV/NV** and **Par.Deadband** Alarms set only if **EMA/EHA/ELA/ENA=1** and **Par.Type=2/3/6/7**	**Sta.MA/HA/LA/NA** = 1 Identifies a Threshold Limit Alarm is active when the module is supervised (i.e. **Internal Check** = 1). Alarm settings are specified in **Set.MA/HV/LV/NV** and are influenced by **Par.Deadband**. Alarms are individually Enabled by **EMA/EHA/ELA/ENA** = 1 and require **Par.Type** setting of 2, 3, 6, or 7. Alarms can be reset if the alarm condition has cleared and the **AlarmReset** (**Cmd.3** or input) is triggered. *Refer to Deadband Diagram*
Sta.NegGrad	BOOL		Out	Tag direct linked by RSViewSE /FactoryTalk	Scaling Negative Gradient <-ON if **Par.InEUMax**<**Par.InEUMin,** in this case **PV/SP Max MZ=InEUMin** and **Min NZ=InEUMax**	**Sta.NegGrad** =1 That flag be used to flip the bargraph to display from top down in RSView. The method allows to indicate "under pressures" the same way as positive/negative pressures
Sta.ParErr	BOOL		Out		Parameter Error (check scaling min/max values and Set.MV/HV/LV/NV values)	**Sta.ParErr** = 1 identifies that there is a AnaInp module scaling or alarm threshold limit configuration error. Refer also to **Par.InpScale InRawMax** = **InRawMin** or **Par.InEUMin** = **Par.InEUMin or Not Set.NV < LV < HV < MV.**
Sta.PAM	BOOL		Out		Parameter alarm mimic if NOT (NV < LV < HV < MV)	**Sta.PAM** = 1 identifies that there is a analog module alarm threshold limit configuration error. Correct limit setting should be NV < LV < HV < MV. This is used for HMI display (mimic) animations to identify that there is an alarm limit configuration problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.PV	REAL		Out		Process/Actual or replaced Value=**PV** for Indication	**Val.PV** contains the **PV** signal value. It is typically the value of the scaled & filtered **Inp** signal or if the signal is bypassed (**Sta.RZ** = 1) then it is **Set.PVZ.**
Val.PVA	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Process Value Alarm=**PV** at Detection of Alarm	Value for HMI threshold value alarm. The RSViewSE alarm setup will create his owen alarm messages with threshold limit value.
Val.PVY	REAL		Out		Real Process Value (not corrected by Replacement Value)	**Val.PVY** contains the value of the scaled, clamped & filtered **Inp** signal which is called the Real Process Value. This signal is not influenced by bypass status (**Sta.RZ**).
Val.SPZ	REAL		Out		Actual Setpoint Feedback	**Val.SPZ** contains the **SP** value for feedback to the HMI display animations. **Val.SPZ** = **SP**

Val.MZ/NZ	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Max/Min PV/SP Range **MZ**=**InEUMax** and **NZ**=**InEUMin** if **InEUMax >** **InEUMin** else **MZ=InEUMin** and **NZ=InEUMax**	**Val.MZ/NZ** are maximum and minimum limits that are used by the HMI for **SP** and **PV** data entry animations. - **Val.MZ** is the higher of **Par.InEUMax** or **Par.InEUMin** - **Val.NZ** is the lower of **Par.InEUMax** or **Par.InEUMin**
Aux.PVX	REAL		Out	Module uses Low Pass Filter; if an external filter is needed, the PVX/PVY I/O would be used.	Input to external Filter by Ladder or Function Block Diagram (FBD)	**Aux.PVX** is an auxilary output of the AnaInp module that contains the clamped & scaled **Inp** signal. Example: If the application needs a high pass filter (HPF) in instead of the module internal low pass filter. Then you have to set Par.Filtering =0 and connect the external filter at Aux.PVX - Aux.PVY.
Aux.PVY	REAL		Inp		Output from external Filter. Valid only if **Par.Filtering**=0	**Aux.PVY** is an input to the AnaInp module that contains the filtered scaled **Inp** signal from an an external filtering algorithm.
Orange Highlight	= only available with AnaInp C					

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	I/O		Bus from/to Parent Module (Group, Machine or Motor/SubSys/Valve)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This AnaInp module reads information FROM the "Bus" and Writes status information TO the "Bus".
The following table shows a set of possible functions if **Par.Filtering**=0 (option):						
Function Name					Description (AnaInpC option only)	
Low Pass Filter (LPF)					filter input frequencies that are above the cutoff frequency	
High Pass Filter (HPF)					filter input frequencies that are below the cutoff frequency	
Notch Filter (NTCH)					filter input frequencies that are at the notch frequency	
Maximum Capture (MAXC)					find the maximum signal in time	
Minimum Capture (MINC)					find the minimum signal in time	
Function Generator (FGEN)					convert an input based on a piece-wise linear function	
High/Low Limit (HLL)					limit an analog input between two values	
Lead-Lag (LDLG)					provide a phase lead-lag compensation for an input signal	
Second-Order Lead Lag (LDL2)					filter with a pole pair and a zero pair	
Moving Average (MAVE)					calculate a time average value	
Moving Standard Deviation (MSTD)					calculate a moving standard deviation	
Ramp/Soak (RMPS)					provide for alternating ramp and soak periods	
Totalizer (TOT)					provide a time-scaled accumulation of an analog input value	
Derivative (DERV)					calculate the amount of change of a signal over time in per-second units	

Enhanced Select (ESEL)					select one of as many as six inputs	
Multiplexer (MUX)					select one of eight inputs	
Rate Limiter (RLIM)					limit the amount of change of a signal over time	
Select (SEL)					select one of two inputs	
Selected Negate (SNEG)					select between the input value and the negative of the input value	
Selected Summer (SSUM)					select real inputs to be summed	

AnalInp and AnalInpC_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EUMax	REAL	x	Inp	Direct allocated to the AnaInp-C AOI	Maximum Scaling Range in Engineering Units	Input from AnaInp-C module .Par.EUMax. Parameter is used to set correct limit for analog output variable.
EUMin	REAL	x	Inp	Direct allocated to the AnaInp-C AOI	Minimum Scaling Range in Engineering Units	Input from AnaInp-C module .Par.EUMin. Parameter is used to set correct limit for analog output variable.
RawMax	REAL	x	Inp	Direct allocated to the AnaInp-C AOI	Maximum RAW Value	Input from Global.Par.AnaInpScale[AnaInp-C module .Par.InpScale].InRawMax. Parameter is used to set correct limit for analog output variable.
RawMin	REAL	x	Inp	Direct allocated to the AnaInp-C AOI	Minimum RAW Value	Input from Global.Par.AnaInpScale[AnaInp-C module .Par.InpScale].InRawMin. Parameter is used to set correct limit for analog output variable.
Auto_Value	REAL	x	Inp	Direct allocated to PID AOI	PV input from PID module when working together with PID in automatic mode	Auto_value variable is used to pass PV output from PID simulation block to Out_Raw when in automatic mode.
Err	BOOL	x	Out	Direct allocated to the AnaInp-C AOI	Transmitter Error	Err bit can be set by switch on HMI faceplate. Sensor Error alarm will be displayed on AnaInp-C faceplate.
Out_RAW	REAL	x	Out	Direct allocated to the AnaInp-C AOI	Output Value in RAW	Out_RAW is the output from simulation block to AnaInp-C block. It may come from two sources. One is to Manually set value from HMI simulation faceplate and the other one is from Auto_Value input variable.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Auto	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Automatic/ Manual simulation mode 0=manual, 1=automatic	Auto bit is connected with HMI Simulation mode switch. (0=Manual, 1=Automatic)
Err_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Error fault from HMI	Err bit is connected with HMI Error switch. (0=Ok, 1=Fault)
EURange	REAL				Range in Engineering Units	EURange is used for output scaling calculation.
Man_Value	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Manual Value from HMI	Manual output value from HMI faceplate.

Out_EU	REAL				Output Value in Engineering Units	Output engineering value before scaled to RAW value .
RawRange	REAL				Range in RAW value	RawRange is used for output scaling calculation.

PidMod

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Check	BOOL		Inp		Enable Alarm Outputs (Override Parent Module Check)	The "**Internal AOI Check**" signal enables module alarms and is driven by the Input **Check**, **<Parent>.Bus.26 (Check)**, or **<Parent>.Bus.8 (LocalCheck)**. Set **Check** = 1 (maintained) with the PLC application software to enable module alarms when the Inp signal should be supervised without check signals from a parent CtrlGrp or Device (i.e. Motor). See Also **Par.DisableGrpCheck**
RZ	BOOL		Inp		Bad **PV ->Sta.RZ** if AnaInp Replacement Value Set	If **RZ** input is set, PID will revert to manual mode (Sta.BCU = 1) and identify this condition as **Sta.RZ**. **RZ** =1 indicate that the **PV** analog input isn't valid! To use this functionality, logic must be generated external to the PIDMod AOI. Example: The RZ output of the PV AnaInp AOI module is mapped to the PIDMod RZ input.
ERR	BOOL		Inp		Error bad **PV** ->**Sta.ERR** if AnaInp Erroneous	Set **ERR** with the PLC application software to the **<AnaInp>.Sta.ERR** tag to identify that the PID's process variable signal is in Error. If **ERR** = 1, then the PID will change to manual mode (**Sta.BCU** = 1) and identify this condtion as **Sta.ERR**.
EnAuto	BOOL		Inp	Set to always on (Log_1) if not used.	Enable Automatic Operation if Actuator ready	**EnAuto** = 1 is a master Enable switch for the internal PID instruction, indicated by **Sta.E** =1. If **EnAuto** = 0, the PID instruction will NOT execute. Note: The EnAuto does not place the PID into automatic mode.
EnFF	BOOL		Inp		Enable Feed Forward ->**Sta.CFF**, **Val.FF**	Set **EnFF** = 1 (maintained) to enable the Feed Forward (**FF**) in the PID instruction. The **FF** is used to bias controller CV. The Val.FF is added to the CV of the internal PID instruction. This can be done with PLC application software or at the Tag value. Feed Forward status is reported as **Sta.CFF**. *See Also FF & Val.FF*
ESP	BOOL		Inp	controlled by user code if it's used or set Tag value.	Enable Setpoint ->**Sta.ESP**	Set **ESP** = 1 (maintained) to transfer the HMI Faceplate setpoint (**Set.SP**) to into the PID instruction **SP**. Set ESP = 0 to prevent the operator from writing a SP to the PID. HMI Faceplate setpoint from This could be used to disable the ability to feed a HMI SP to the PID controller. The PidMod provides for a bumpless signal transfer for both **ESP** signal transitions.

EUextSel	BOOL		Inp	controlled by user code if it's used.	Local Extern Selected, Actuator local	Set **EUextSel** = 1 (maintained) to identify that the PID's CV actuator is in local mode or disabled. In this condition, the PID cannot affect the actuator position; therefore, the PID instruction is in manual mode. When **EUextSel** = 1 the internal PID instruction is not executed and the PID mode is set to manual (**Sta.BCU**). See Also **Set.CVS** & **Val.BCZ** *Application Note: EUextSel transition 1 -> 0, then PID requires an Auto mode change command from HMI or Application if Auto mode is desired.* *If the PidMod is controlling a ActMod then connect the PidMod.EUextSel to the Actuators output: <ActMod>.EUextSel.*
SPextSel	BOOL		Inp	Used by PID cascading.	Setpoint Extern Selected	Set **SPextSel** = 1 (maintained) by the PLC application software to use an external setpoint (**SPext**) from the PLC application as the PID setpoint. The PIDMod will report that the PID is running with external setpoint (**Sta.RCX**) when **SPextSel** = 1 and **Sta.BCU** = 0. *Application Note: Used for cascaded controllers or remote setpoint. SPextSel and CVextSel can NOT be 1 simultaneously.*
CVextSel	BOOL		Inp	This is typically used during start-up of the PIDMod. Used to load the initial CV prior to PID automatic operation.	Control Variable Extern Selected	Set **CVextSel** = 1 (maintained) by the PLC application software to use an external control variable (**CVext**) from the PLC application as the output of the **CV** (Internally PID.SO). The PID will be placed in Manual mode (**Sta.BCU**) when the **CVextSel** = 1 and the PIDMod will report that the PID is running in Manual Mode (**Sta.RCU** = 1). *Application Note: Use for PLC control of CV. SPextSel and CVextSel can NOT be 1 simultaneously.* *If the PidMod is controlling a ActMod then connect the PidMod.CVextSel to the Actuators output: <ActMod>.CVextSel.*
AlarmReset	BOOL		Inp		Alarm Reset	Set **AlarmReset** = 1 (Pulse) to reset the module alarms if the alarm conditons have returned to normal. Alarms include the Controller Deviation Alarm (**Sta.CDA**), Sensor Error (**Sta.ERR**), and Deviation Alarms (**PID.DVNA** & **PID.DVPA**).
PV	REAL	X	Inp		Process Value Scaled Input	**PV** is configured on the AOI instance to the scaled process variable **PV** (i.e. already in EU) for input into the internal **PID.PV**. This signal is typical the **<AnaInp>.PV** signal.

SPext	REAL		Inp	in PV units	Setpoint Extern (in PV units) ->**Pid.SP** if **SPextSel**	**SPext** contains a value written by the PLC application software logic that will be used as the PID **SP** when **Sta.RCX** = 1. *Application Note: This can be used to cascade PIDs or allow the PLC application to control the SP, but the setting will only be moved when Case 1: PID is not in Manual (Sta.BCU = 0) and CVextSel = 0; or Case 2: PID is in External mode (Sta.BCX = 1)*
CVext	REAL		Inp	unit [%]	Control Variable Extern 0..100% ->**Pid.SO** if **CVextSel**	**CVext** contains a value written by the PLC application software that will be used as the PID **CV** when **CVextSel** = 1. *Application Note: This can be used to control of the CV during system startup. Appl example:This input could set form an Actuator output <ActMod>.SP. Refer to Appendix "Wiring Diagram"*
FF	REAL		Inp	unit [%]	Feed Forward or Bias Value 0..100%	**FF** contains a Feed Forward value (0-100%) written by the PLC application software that is used to bias the internal PID controller using the internal **PID.Bias** tag. An external (i.e. PLC application logic). **FF** will be used by the PID instruction only if it is selected by the **EnFF** input. *See also Val.FF*
CU	BOOL		Inp		Controller MANUAL Selector	**CU** is pulsed ON by the PLC application software to trigger a PID mode change to Manual (**Sta.BCU**) . Same as **Cmd.0**
CC	BOOL		Inp		Controller AUTO Selector	**CC** is pulsed ON by the PLC application software to trigger a PID mode change to Auto (**Sta.BCC**) . Same as **Cmd.1**
CX	BOOL		Inp		Controller EXTERN Selector	**CX** is pulsed ON by the PLC application software to trigger a PID mode change to External (**Sta.BCX**) Same as **Cmd.2**
CV	REAL	X	Out	unit [%]	Controlled Variable Output, Scaled by **Pid.MAXCV/Pid.MINCV**	**CV** contains the control variable value (**Pid.CV**) of the internal PID instruction which is scaled by **Pid.MinCV** and **Pid.MaxCV**. As a standard practice, the CV output should be scaled as 0-100%.
Alarm	BOOL		Out		Alarm Indication <-**Sta.F** OR **Sta.W** (Failure or Warning)	**Alarm** = 1 Identifies that either a Failure (**Sta.F**) or Warning (**Sta.W**) alarm is active. The raw alarm signals include controller deviation alarm (**Sta.CDA**) and Sensor Error alarms (**Sta.ERR**). Requires **Internal Check** = 1. *See also Check*
RdyOk	BOOL		Out	Interlock to Parent Module	Ready Ok Signal -> for cascade	**RdyOK** = 1 Identifies that there is No active Failure (**Sta.F**) or Warning (**Sta.W**) Alarm. RdyOk is always set, until one of the Alarms coming up. The alarm signals include controller deviation alarm (**Sta.CDA**) and Sensor Error Alarms (**Sta.ERR**). If **RdyOK** = 0, then **<PidMod>.CV** holds last state and PID mode is changed to manual (**Sta.BCU**). The **RdyOK** signal is used in the PLC application software where needed to ensure that the PID module is functioning properly such as cascaded controllers. *See also Check*

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Cmd.0\CU	BOOL		Inp		Controller MANUAL Selector ->Feedback **Sta.BCU**, Run **Sta.RCU**	**Cmd.0** = 1 to change the PID mode to Manual (**Sta.BCU**) from the HMI. Cmd.0 is direct linked to HMI Faceplate MAN -Button. Set in HMI, Reset in AOI. *Note: If it is desired to control PID mode from the PLC application software see EUExtSel and CVExtSel.*
Cmd.1\CC	BOOL		Inp		Controller AUTO Selector ->Feedback **Sta.BCC**, Run **Sta.RCC**	Cmd.1 = 1 to change the PID mode to Auto (Sta.BCC) from the HMI. Cmd.1 is direct linked to HMI Faceplate AUT -Button. Set in HMI, Reset in AOI. There is no MMCL Lib provision to set to Auto from PLC application code.
Cmd.2\CX	BOOL		Inp		Controller EXTERN Selector ->Feedback **Sta.BCX**, Run **Sta.RCX**	**Cmd.2** = 1 to change the PID Mode to External (**Sta.BCX**) from the HMI. Cmd.2 is direct linked to HMI Faceplate EXT -Button. Set in HMI, Reset in AOI. *Note: If it is desired to control PID mode from the PLC application software see EUExtSel and CVExtSel.*
Cmd.3\ACK	BOOL		Inp		Acknowledge	**Cmd.3** = 1 is used to acknowledge the PIDMod alarms from the HMI. Same functionality as **AlarmReset**. Reserved for HMI Commands. Set in HMI, Reset in AOI.
Cmd.4\LAC	BOOL		Inp		Load Accu=Last Setpoint stored in Auto Mode < -Val.ACC if Sta.EAC	**Cmd.4** = 1 is used to restore the last Auto Setpoint from the HMI. The **Val.ACC** contains the **PID.SP** value when the PID is taken out of Auto Mode. This command loads the **Val.ACC** value into the **PID.SP** value if the PID is running Auto Mode (**Sta.RCC**). Reserved for HMI commands. Set in HMI, Reset in AOI.
Set.SP	REAL		Inp		Setpoint Input (in PV units)	**Set.SP** always contains the **SP** value on the HMI Faceplate. **SP** has the same units as the **PV**. When the PID is in Auto mode (**Sta.RCC**) and Setpoint Enabled (**Sta.ESP**), the **Set.SP** value is used as the **PID.SP**. If Not, the **PID.SP** is transferred back to the HMI Setpoint to provide for a bumpess **SP** transfer when going back into Auto Mode.
Set.CVS	REAL		Inp		Controlled Variable Manual Input 0..100% ->**Pid.SO** if MAN Selected	**Set.CVS** always contains the **CV** value [0..100%] as shown on the HMI Facepalte. When the PID is in Manual Mode (**Sta.RCU**), **CVextSel** = 0, and **EUextSel** = 0, the **Set.CVS** value is moved to the **PID.SO**. If Not, the **PID.SO** is transferred back to the **Set.CVS** to provide for a bumpless CV transfer.
Set.CP	REAL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Proportional Gain [unitless]	**Set.CP** contains the value of Proportional Gain [unitless] used in the PID calculation. This data is transferred to the PID instruction's **KP** signal. Refer to PID documentation for PID.KP.
Set.CI	REAL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Integral Gain [1/s]	**Set.CI** contains the value of Integral Gain [1/s] used in the PID calculation. This data is transferred to the PID instruction's **KI** signal. Refer to PID documentation for PID.KI.

Set.CD	REAL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Derivative Gain [s]	**Set.CD** contains the value of Derivative Gain [s] used in the PID calculation. This data is transferred to the PID instruction's **KD** signal. Refer to PID documentation for PID.KD.
Par.UpdateRate	DINT	X	Inp		0=Always, >400=Preset time [ms] to update all Inputs and PID. The Accum time since the last sample is used as PID update time **Pid.UPD**	Set Par.UpdateRate to a time setting in [ms] to execute the PIDMod module (i.e. sample rate). Execution of the module will update the module IO and execute the PID calculation. By setting the Par.UpdateRate = 0, the PID will calculate each time the PIDMod instruction is executed. The accumulated time since the last PID calculation update is used as the PID update time (Pid.UPD) Note: The UpdateRate can't go faster then 400ms! This is given by the MMCL Scan Control Function, see also Global.Scan1.Preset. Set as"Always" the module execution will be at the same time as the module is executing by Scan Control Func.(Scan balance) Important: The ApplyPar input in the SysGrp must be set for a change to the UpdateRate parameter to take effect.
Par.MZ	REAL	X	Inp		Maximum **PV/SP** Range Value in Engineering Units (PV units) Valid = any real value, Default =100	Set **Par.MZ** to a real value that specifies the maximum limit for the **PV** & **SP** signals. This is used by the HMI for **SP** data entry animations. Typically the **Par.MZ** is identical to the **<AnaInp>.Par.InEUMax** where **<AnaInp>** is the signal providing the PID's PV signal.
Par.NZ	REAL	X	Inp		Minimum PV/SP Range Value in Engineering Units (PV units) Valid = any real value, Default = 0	Set **Par.NZ** to a real value that specifies the minimum limit for the **PV** & **SP** signals. This is used by the HMI for **SP** data entry animations. Typically the **Par.NZ** is identical to the **<AnaInp>.Par.InEUMin** where **<AnaInp>** is the signal providing the PID's PV signal.
Par.AlarmDelay	DINT		Inp	Units [ms]	Delay Time for Deviation Alarm ->**Sta.CA**	Set the **Par.AlarmDelay** to a time setting in ms to delay deviation alarms ((**PID.DVNA** (low) or **PID.DVPA** (high)) from activating the (**Sta.CDA**).
Par.DisableGrpCheck	BOOL		Inp		Disable Alarm Check by Parent Group modules If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** to 1 to prevent **Check -**condition from Parent module (<Parent>.Bus.26) to enabling the Module's **Internal Check**. The **Check** Input and **<Parent>.Bus.8** still enable the Module's **Internal Check**. If set to 1 then no warning(**Sta.W**) or error(**Sta.F**) will be passed on ParentBus. The **group does not** receive fault / warnings Info .Alarms however is Indicated on HMI. Note: An unchecked module will not fault or alarm.
Par.EnableEXTSel	BOOL		Inp		Enable EXTERN -HMI selection push button	Set Par.EnableEXTSel to 1 to show the EXT -selection button at the HMI template. If set to 0, then the Operator isn't able to select the PID in externel mode.

Par.ErrorWarning	BOOL		Inp		Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by Sta.W IF ON, Alarms are indicated by Sta.F	Set **Par.ErrorWarning** = 1 to specify that **Sta.ERR** and **Sta.CDA** alarms will be reported as Warning alarms (**Sta.W**). Set **Par.ErrorWarning** = 0 to specify that these alarms should be reported as Failure alarms (**Sta.F**)
Par. DisableGrpIdentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.
Par.AlarmGongCode	DINT of BOOLs		Inp		Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Sta.F			Out		Device Failure Mimc (Bus to Parent Module)	**Sta.F** = 1 Identifies that the module is supervised (i.e. **Internal Check** = 1) and there is an active failure alarm. Failure alarms are masked (inhibited) if **Par.ErrorWarning** = 1. Otherwise Failure alarms include **Sta.ERR** and **Sta.CDA.**
Sta.W	BOOL		Out		Device Warning Mimc (Bus to Parent Module)	**Sta.W** = 1 Identifies that the module is supervised (i.e. **Internal Check** = 1) and there is an active Warning alarm. Warning alarms are masked (inhibited) if **Par.ErrorWarning** = 0 (*only for PidMod*). Otherwise Warning alarms include **Sta.ERR** and **Sta.CDA.**
Sta.CDA	BOOL		Out		Controller Deviation Alarm <- **Pid.DVNA** OR **Pid.DVPA**	**Sta.CDA** = 1 Identifies a Controller Deviation Alarm (**PID.DVNA** (low) or **PID.DVPA** (high)) is present when the module's **Internal Check** = 1. See also **PID.DVP** and **PID.DVN** for deviation alarm settings.
Sta.RZ	BOOL		Out		PV bad ->Replacement Value Auto Switch to MAN ->Sta.BCU/RCU	**Sta.RZ** = 1 identifies that the input **RZ** = 1. This means that the PID's **PV** signal is not valid. The PID will be placed in manual mode.
Sta.ERR	BOOL		Out		Error **PV** bad ->Auto Switch to MAN ->**Sta.BCU/RCU**	**Sta.ERR** = 1 Identifies that the input **ERR** = 1. This means that the PID's **PV** signal is not valid. The PID will be placed in manual mode and **Alarm** will be set.

Sta.E	BOOL		Out		Controller Enable	**Sta.E** = 1 Identifies that the PID instruction is Enabled (i.e. Instruction Rung is Enabled). The internal PID instruction is operational.
Sta.ESP	BOOL		Out		Enable Setpoint Input	**Sta.ESP** = 1 Identifies that the input **ESP** = 1. This means that the HMI setpoint **Set.SP** can be transferred to the **PID.SP** if the PID is in Auto (Sta.RCC).
Sta.EAC	BOOL		Out		Enable to load Accu=Last Setpoint stored in Auto Mode	**Sta.EAC** = 1 Identifies that the **Sta.RCX** = 0 meaning that the PID is not running in External mode. In this condition it is allowable to restore the last known auto setpoint to the current setpoint. ->**Cmd.4/ACC** is not available on the HMI when the **Sta.EAC** = 0.
Sta.CFF	BOOL		Out		Feed Forward **Val.FF** Enabled <-**EnFF**	**Sta.CFF** = 1 identifies that Feed Forward is enabled with **EnFF**. *See Also Val.FF*
Sta.ECU/ECC/ECX	BOOL		Out	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable ECU=MANUAL, ECC=AUTO, ECX=EXTERN	**Sta.ECU/ECC/ECX** identifies the PID Enabled Status. -**Sta.ECU** = 1 --> Manual Mode -**Sta.ECC** = 1 --> Auto Mode -**Sta.ECX** = 1 --> External Mode
Sta.RCU/RCC/RCX	BOOL		Out		Run RCU=MAN ->Set CV, RCC=AUT ->Set SP, RCX=EXT ->Set SP	**Sta.RCU/RCC/RCX** identifies the PID Running status. -**Sta.RCU** = 1 --> Running in Manual -**Sta.RCC** = 1 --> Running in Auto -**Sta.RCX** = 1 --> Running in External.
Sta.BCU/BCC/BCX	BOOL		Out		Select Feedback ->BCU=MANUAL, BCC=AUTO, BCX=EXTERN	**Sta.BCU/BCC/BCX** Identifies the PID Mode status. -**Sta.BCU** = 1 --> Enabled Manual Mode -**Sta.BCC** = 1 --> Enabled Auto Mode -**Sta.BCX** = 1 --> Enabled External Mode.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.BCZ	DINT		Out		Select Feedback Value 1=BCU/2=BCC/3=BCX	**Val.BCZ** contains an integer value identifying the PID Mode status. - **Val.BCZ** = 1 --> Manual Mode from (**Sta.BCU**) **- Val.BCZ** = 2 --> Auto Mode from (**Sta.BCC**) - **Val.BCZ** = 3 --> External Mode from (**Sta.BCX**)
Val.ACC	REAL		Out		Accu Value=Last Setpoint stored in Auto Mode (in PV units)	**Val.ACC** contains the last **SP** value (**Pid.SP**) from the internal PID instruction when the PID transitions out of Auto Mode (**Sta.RCC**). This is only a history of the last known Auto Setpoint.
Val.SPZ	REAL		Out		Actual Setpoint **SP** at Controller (in PV units)	**Val.SPZ** contains the **SP** value (**Pid.SP**) from the internal PID instruction.
Val.PV	REAL		Out		Process Value at Controller (in PV units)	**Val.PV** contains the **PV** value (**Pid.PV**) from the internal PID instruction.
Val.CE	REAL		Out		Control Error = Difference between **SP** and **PV** (in PV units)	**Val.CE** contains the error value (**Pid.ERR**) from the internal PID instruction. This is the difference between the **Pid.SP** and **Pid.PV** in the same units as **PV**.

Val.CV	REAL		Out		Controlled Variable Output, range **MINCV..MAXCV** (based on **Pid.OUT** 0..100%)	**Val.CV** contains the output value (**Pid.Out**) from the internal PID instruction. This value varies between **Pid.MinCV** and **Pid.MaxCV** in units of %.
Val.FF	REAL		Out		Feed Forward or Bias Value 0..100%	**Val.FF** contains the bias value (**Pid.Bias**) from the internal PID instruction Bias. This value varies between 0-100%.

Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	I/O		Bus from/to Parent Module (Group, Machine)	This is refered to as the <Parent>.Bus. This bus is managed by another MMCL Module and allows the modules to exchange information. This PidMod module reads information FROM the "Bus" and Writes status information TO the "Bus".
PidMod Module Data of integrated PID -RSLogix5000 Instruction						

Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Pid.CTL	DINT		I/O	See descriptions below for each individual bit.	The .CTL member provides access to the status members (bits) in one, 32-bit word. Bits 20-31 are Inputs. The PID instruction sets Output bits 07 -15.	The .CTL member provides access to the status members (bits) in one, 32-bit word. Bits 20-31 are Inputs. The PID instruction sets Output bits 07 -15.
Pid.EN=CTL.31	BOOL	NA	I/O	Status Only. Value written to **<PidMod>. Sta.E**	enabled < -Sta.E	PID Enabled; **Sta.E**
Pid.CT=CTL.30	BOOL	NA	I/O	Do Not Use! Functionality provided in PidMod by other means	cascade type (0=slave; 1=master)	Cascade type (0=slave; 1=master)
Pid.CL=CTL.29	BOOL	NA	I/O	Do Not Use! Functionality provided in PidMod by other means	cascade loop (0=no; 1=yes)	Cascade loop (0=no; 1=yes).
Pid.PVT=CTL.28	BOOL	NA	I/O	Do Not Configure! Unlatched by **PIDMod** AOI.	**PV** process variable tracking (0=no; 1=yes)	**PV** process variable tracking (0=no; 1=yes).

Pid.DOE=CTL.27	BOOL	X	I/O	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.DOE**	derivative of (0=**PV**; 1=error)	Derivative of (0=**PV**; 1=error)
Pid.SWM=CTL.26	BOOL		I/O	Do Not Configure! Same value as <**PIDMod>. Sta.RCU**.	software manual mode (0=no-auto; 1=yes- sw manual) < -Sta.BCU	PID Software Manual Mode (0=no-auto; 1=yes-sw manual); **Sta.BCU**.
Pid.CA=CTL.25	BOOL	X	I/O	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.CA**	control action (0 means ERR=**SP**-**PV**; 1 means ERR=**PV**-**SP**)	Control Action (0 means ERR=**SP**-**PV**; 1 means ERR=**PV**-**SP**)
Pid.MO=CTL.24	BOOL		I/O	DoNotUse! Unlatched by **PIDMod** AOI.	station mode (**0=automatic=default**; 1=manual)	PID Manual Mode (**0=automatic=default**; 1=manual).
Pid.PE=CTL.23	BOOL		I/O	DoNotUse! Unlatched by **PIDMod** AOI.	PID equation (**0=independent=default**; 1=dependent)	PID equation (**0=independent=default**; 1=dependent).
Pid.NDF=CTL.22	BOOL	X	I/O	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.NDF**	no derivative smoothing(0=derivative smoothing filter enabled; 1=derivative smoothing filter disabled)	No derivative smoothing (0=derivative smoothing filter enabled; 1=derivative smoothing filter disabled).
Pid.NOBC=CTL.21	BOOL		I/O	Do Not Configure! Samevalueas Not <**PIDMod>. EnFF.**	no bias back calculation (0=bias back calculation enabled; 1=bias back calculation disabled) < -NOT EnFF	No bias back calculation (0=bias back calculation enabled; 1=bias back calculation disabled); <u>NOT **EnFF**</u>.
Pid.NOZC=CTL.20	BOOL	X	I/O	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.NOZC**	no zero crossing deadband(0=deadband is zero crossing; 1=deadband is not zero crossing)	No zero crossing of deadband. (0=deadband is zero crossing; 1=deadband is not zero crossing)
Pid.INI=CTL.15	BOOL		I/O	Do Not Configure! Trigger only with Global.ApplyPar.	PID initialized (0=no; 1=yes) < -Global.Sys.ApplyPar ONS at startup	PID initialized (0=no; 1=yes); Tag is reset when **Global.ApplyPar** = 1.
Pid.SPOR=CTL.14	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	**Pid.SP** setpoint out of range (0=no; 1=yes)	**Pid.SP** setpoint out of range (0=no; 1=yes)

Pid.OLL=CTL.13	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	Pid.OUT ->Val.CV is below minimum output limit (0=no; 1=yes)	**Pid.OUT Val.CV** is below minimum output limit (0=no; 1=yes)
Pid.OLH=CTL.12	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	Pid.OUT ->Val.CV is above maximum output limit (0=no; 1=yes)	**Pid.OUT Val.CV** is above maximum output limit (0=no; 1=yes)
Pid.EWD=CTL.11	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	error is within deadband (0=no; 1=yes)	Error is within deadband (0=no; 1=yes)
Pid.DVNA=CTL.10	BOOL		Out	DoNotUse! Unlatched by **PIDMod** AOI.	deviation is alarmed low (0=no; 1=yes) ->used in Alarm Sta.CDA	Deviation is alarmed low (0=no; 1=yes); used in Alarm **Sta.CDA.**
Pid.DVPA=CTL.09	BOOL		Out	DoNotUse! Unlatched by **PIDMod** AOI.	deviation is alarmed high (0=no; 1=yes) ->used in Alarm Sta.CDA	Deviation is alarmed high (0=no; 1=yes); used in Alarm **Sta.CDA.**
Pid.PVLA=CTL.08	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	**PV** is alarmed low (0=no; 1=yes)	**PV** is alarmed low (0=no; 1=yes).
Pid.PVHA=CTL.07	BOOL		Out	Status Only. Not used by **PIDMod** AOI.	**PV** is alarmed high (0=no; 1=yes)	**PV** is alarmed high (0=no; 1=yes).
Pid.SP	REAL		Inp	Do Not Configure! Value is written from **<PidMod>. Set.SP**.	setpoint < - taken from HMI Set.SP or Input SPext	Setpoint taken from HMI **Set.SP** or Input **Spext**.
Pid.KP	REAL		Inp	Do Not Configure! Value is written from <**PIDMod**>. **Set.CP**.	independent proportional gain (unitless) < - taken fromHMISet.CP (dependent controller gain (unitless)<-if Pid.PE=1)	Independent proportional gain (unitless); taken from HMI **Set.CP** (dependent controller gain (unitless)<-if **Pid.PE**=1).
Pid.KI	REAL		Inp	Do Not Configure! Value is written from <**PIDMod**>. **Set.CI**.	independent integral gain [1/s] < - taken fromHMISet.CI (dependent reset time [min/repeat])<-if Pid.PE=1)	Independent integral gain [1/s]; taken from HMI **Set.CI** (dependent reset time [min/repeat])<-if **Pid.PE**=1).
Pid.KD	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. Set.CD**.	independent derivative gain [s] < - taken from HMISet.CD (dependent rate time [min]<-if Pid.PE=1)	Independent derivative gain [s]; taken from HMI **Set.CD** (dependent rate time [min]<-if **Pid.PE**=1).

Pid.BIAS	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. FF**.	feed forward or bias % < - taken from Input FF	Feed forward or bias % taken from Input **FF**.
Pid.MAXS	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. Par.MZ.**	maximum SP and PV eng. unit scaling value < -taken from Par.MZ	Maximum **SP** and **PV** eng. unit scaling value; taken from **Par.MZ**.
Pid.MINS	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. Par.NZ**.	minimum SP and PV eng. unit scaling value < -taken from Par.NZ	Minimum **SP** and **PV** eng. unit scaling value; taken from **Par.NZ**.
Pid.DB	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.DB**	deadband engineering units	Deadband Value (Engineering Units)
Pid.SO	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. CVext** or **<PIDMod>. Set.CVS**.	set output % < - taken from HMI Set.CVS or Input CVext	Set output % taken from HMI **Set.CVS** or Input **CVext**
Pid.MAXO	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MAXO**	maximum Pid.OUT limit (% of output, default 100%) ->limits Val.CV	CV High Limit, Typically 100% (not default)
Pid.MINO	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MINO**	minimum Pid.OUT limit (% of output, default 0%) ->limits Val.CV	CV Low Limit, Typically 0% (default)
Pid.UPD	REAL		Inp	Do Not Configure! Set by *UpdateRateT* Timer (**PIDMod** AOI).	loop update time [s] < -specified by Par.UpdateRate[ms] The Accumulated time since the last sample is used as PID update time	Loop update time [s] specified by **Par.UpdateRate** [ms]. The Accumulated time since the last sample is used as PID update time.

Pid.PV	REAL		Inp	Do Not Configure! Value Moved into **<PIDMod>. Val.PV**.	scaled PV value < -taken from Input PV	Scaled **PV** value taken from Input **PV**
Pid.ERR	REAL		Out	Do Not Configure! Value Moved into **<PIDMod>. Val.CE**.	scaled error value ->moved to Val.CE	Scaled error value moved to **Val.CE**
Pid.OUT	REAL		Out	Do Not Configure! Value Moved into **<PIDMod>. Val.CV**.	output 0..100% ->converted to Val.CV within range MINCV..MAXCV	Output 0..100% ->converted to Val.CV within range MINCV..MAXCV
Pid.PVH/PVL	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.PVH/PVL**	**PV** process variable high/low alarm limit	**PV** process variable high/low alarm limit
Pid.DVP	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.DVP**	positive deviation alarm limit (e.g. +5) ->Pid.DVPA sets Alarm Sta.CDA	Positive deviation alarm limit (e.g. +5). Note: **Pid.DVPA** sets Alarm **Sta.CDA**
Pid.DVN	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.DVN**	negative deviation alarm limit (e.g. -5) ->Pid.DVNA sets Alarm Sta.CDA	Negative deviation alarm limit (e.g. -5). Note: **Pid.DVNA** sets Alarm **Sta.CDA**
Pid.PVDB	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.PVDB**	**PV** process variable alarm deadband	**PV** process variable alarm deadband
Pid.DVDB	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.DVDB**	deviation alarm deadband	Deviation alarm deadband

Pid.MAXI	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. Par.MZ**.	maximum PV raw value (unscaled input) < -taken from Par.MZ	Maximum **PV** raw value (unscaled input. Taken from **Par.MZ**.
Pid.MINI	REAL		Inp	Do Not Configure! Value is written from **<PIDMod>. Par.NZ**.	minimum PV raw value (unscaled input) < -taken from Par.NZ	Minimum **PV** raw value (unscaled input). Taken from **Par.NZ**.
Pid.TIE	REAL		Inp	Do Not Use! Functionality provided in PidMod by other means	tieback value for manual control	Tieback value for manual control.
Pid.MAXCV	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MAXCV**	maximum CV scaling value (corresponding to 100% Pid.OUT < -default 100)	Maximum **CV** scaling value (corresponding to 100% **Pid.OUT,** default **100**)
Pid.MINCV	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MINCV**	minimum **CV** scaling value (corresponding to 0% **Pid.OUT** <-default **0**)	Minimum **CV** scaling value (corresponding to 0% **Pid.OUT,** default **0**)
Pid.MINTIE	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MINTIE**	minimum tieback value (corresponding to 100%)	Minimum tieback value (corresponding to 100%)
Pid.MAXTIE	REAL	X	Inp	Configure using "PID Configuration Dialog" or with the tag **<PidMod>. PID.MAXTIE**	maximum tieback value (corresponding to 0%)	Maximum tieback value (corresponding to 0%)
Pid.DATA[0]..[16]	REAL		Out	Do not use.	internal PID data (see manual)	Internal PID data (see manual)
Notes:						

* Portions of the UDT intended to interface to the HMI (CMD, SET) have been used internally in PLC logic
"PID Configuration Dialog" is the RSLogix5000 configuration faceplate for the PID instruction. It is recommendation to configure an unscanned Routine with DISABLED PID instructions. These PID instructions would be configured with alias tags to individual instances of the PID internal to the PidMOD so that the PID Configuration Dialog can be shown to organize the PID elements in a more user friendly manner. *Note: When initially configuring PID instruction data via the PLC faceplate, an "Update Time" value may need to be entered to allow the entry of other data; this will be overwritten by the PIDMod AOI.*

PidMod_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EUMax	REAL	x	Inp	Direct allocated to the PIDMod AOI	Scaled maximum Value in Engineering Units	Input from PidMod module .Par.MZ. EUMax value is used to set correct limit for PID simulation block PV output.
EUMin	REAL	x	Inp	Direct allocated to the PIDMod AOI	Scaled minimum Value in Engineering Units	Input from PidMod module .Par.NZ. EUMin value is used to set correct limit for PID simulation block PV output.
CV	REAL	x	Inp	Direct allocated to the PIDMod AOI	Controlled variable from PidMod	Control Variable input is used in Automatic mode. Lead-Lag instruction is used to simulate process behavior, parameters can be set on simulation faceplate. Output is scaled to correct limits.
CVMax	REAL	x	Inp	Direct allocated to the PIDMod AOI	Maximum CV value from PidMod	Input from PidMod module .Par.Pid.MaxCV. CVMax value is used to scale input CV to PV output.
CVMin	REAL	x	Inp	Direct allocated to the PIDMod AOI	Minimum CV value from PidMod	Input from PidMod module .Par.Pid.MinCV. CVMin value is used to scale input CV to PV output.
PV	REAL	x	Out	Direct allocated to the PIDMod AOI	Process variable to PidMod	Process variable output. In automatic mode PV is computed from CV after a Lead-Lag function. In manual mode PV is set from HMI simulation faceplate.
Err	BOOL	x	Out	Direct allocated to the PIDMod AOI	Transmiter Error	Err bit can be set by switch on HMI faceplate. Sensor Error alarm will be displayed on PIDMod faceplate.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Auto	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Automatic/ Manual mode 0=manual, 1=automatic	Auto bit is connected with HMI Loop Simulation switch. (0=Manual/Off, 1=Automatic/On)
Err_HMI	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Error fault from HMI	Err_HMI bit is connected with HMI Error switch. (0=Ok, 1=Fault)
Gain	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	The process gain multiplier. This value allows the simulation of a process gain.	Gain can be set from HMI simulation faceplate. This parameter is input for Lead lag instruction

Lag	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	The lag time in seconds. The minimum lag time is DeltaT/2.	Lag can be set from HMI simulation faceplate. This parameter is input for Lead lag instruction
Lead	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	The lead time in seconds. Set Lead = 0.0 to disable the lead control algorithm.	Lead can be set from HMI simulation faceplate. This parameter is input for Lead lag instruction
LDLG_Sim	LEAD_LAG				Lead lag instruction block	Lead lag instruction simulates behavior of process. Parameters can be set from HMI simulation faceplate.
SCL_01	SCALE				PV scaling block	PV scaling block takes output from lead-lag instruction and scale it to correct units for Process Variable.
SEL_01	SELECT				Output selector block	Select PV value output between HMI manual value or scaled Lead-lag instruction output. 0=Manual, 1=Automatic
PV_HMI	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Process variable in manual mode from HMI	Process variable value can be set from HMI faceplate by PV slider in manual mode.

ActMod

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Inp	REAL	X	Inp	Ideally this tag value will represent the true field measurement of the signal (i.e. 4-20 mA)	Raw Value analog Input(**$-3.4e^{+38}..3.4e^{+38}$) ->** moved to **PV** if **Scaling**=0	**Inp** is configured on the AOI instance and is the raw signal to be monitored by the ActMod module. The input value can be either real or integer. Oftentimes, the **Inp** tag is aliased to a hardware analog module input signal. The AnaInp module Scales (**Par.InpScale**) and Filters (**Par.Filtering**) this Inp signal to calculate the **PV** signal output.Similar to AnaInp.Inp. The **Inp** is moved to the **PV** output of the ActMod if the Scaling is 0 (Par.InpScale). This is an analog input value often from the 1756 AI module. Appl Example: Valve Position Feedback.
CV	REAL		Inp		Control Variable from PID->enabled if **ESP**=OFF	CV is used as an input to the ActMod when driving an Analog Output from application software. As an example, if a PID were driving the valve position, the PID.CV would be mapped to the **ActMod.CV** and it would be necessary to set the ActMod.ESP to 0
ESP	BOOL		Inp		Enable Setpoint ->sent to **Sta.ESP,** Disable Auto Control by PID	Either the ActMod.Out comes from the CV or the **Set.SP**. Set ESP = 1 to use the Set.SP and Set ESP = 0 to use the CV. The Set.SP comes from the HMI Faceplate
Check	BOOL		Inp		Enable Alarm Outputs (Override Parent Module Check)	The **"Internal Check"** signal enables module alarms and is driven by the Input **Check**, (<Parent>.Bus.26), **LocalCheck** (<Parent>.Bus.8) , or **Sta.REU**. -Set **Check** = 1 (maintained) with the PLC application software to enable module alarms when the **Inp** signal should be supervised without check signals from a parent CtrlGrp or Device (i.e. Motor). See Also **Par.DisableGrpCheck**
K	BOOL	X	Inp	Enter Log_1 if not used.	Ok/Available	**K** is configured on the AOI instance and is set to 1 when the field device providing the **Inp** signal is powered and considered OK. Set **K** = 1 (Log_1) if the signal is not physically available. *Example: For a four wire analog signal, this could be the On/Off status of the instruments 120V power. See also Sta.KA/KAM*
GX	BOOL		Inp		Local Start X Input Open/Raise	Device Input for Local (Field) Start PB for open/rise direction. Should not set GX and GY to 1 at the same time. Increase SP in local mode. Functionality is similar to G input per the MotorN. Input can be pulse or maintain, if maintain each second edge or pulse will increaseing value by **Par.RampRate** up to **MZ**. When press GX in local mode then output set **EUextSel =1** for a time of **GlobalData.Par.EUactiveTime.** i.e for this time length the module set the PID to CV externel Select (CVextSel) by pre called PID.

GY	BOOL		Inp		Local Start Y Input Close/Lower	Device Input for Local (Field) Start PB for close/decrease direction. Should not set GX and GY to 1 at the same time. Decrease SP in local mode. Functionality is similar to G input per the MotorN. Input can be pulse or maintain, if maintain each second edge or pulse will increaseing value by **Par.RampRate** up to **NZ**. When press GY in local mode then output set **EUextSel =1** for a time of **GlobalData.Par.EUactiveTime.** i.e for this time length the module set the PID to CV externel Select (CVextSel) by pre called PID.
ERR	BOOL		Inp	Enter Log_0 if not used.	Transmitter Error->**Sta.ERR**	**ERR** is set to 1 by the PLC application program when there is a sensor error is present and set to 0 if no error is present or the signal is physically not available. Not configured on the AOI instruction. Example: The Inp signal may come from a pressure transducer and through a signal transmitter. This could be an error status signal from the transmitter or transducer. See also **Sta.ERR/ERRM**
EMA/ENA	BOOL		Inp	To only work with Par.Type other than 0.	Enable Max/Min Limit ->**Sta.MA/NA** Stop Actuator Open/Close None module alarms!	Enable Max and Min alarms are only enabled if **Par.Type** is 2 or 3. These alarms would stop the travel of the signal in that direction (**ActPos**). These don't generate HMI-threshold alarms or module alarms! (W / F)
EHA/ELA	BOOL		Inp	To only work with Par.Type other than 0.	Enable High/Low Alarm ->**Sta.HA/LA** Only Warnings alarm!	Enable High and Low Alarms. Alarm status: Sta.HA/LA are only **Warnings** alarms! These generate HMI threshold alarms are only enabled if **Par.Type** is 2 or 3 (enable alarm).
AlarmReset	BOOL		Inp		Alarm Reset	Alarm reset for all ActMod alarms (HA, LA, ERR, KA).
CVextSel	BOOL		Out		Ctrl Variable Extern Selected (Signal to PID)	A command to tell the PID that the PID.CV is being controlled by the ActMod(external). If this functionality is to be used, the ActMod.CVextSel input must be written to the **PID.CVextSel** output. The ActMod.SP is then written to the PID.CVext. This logical mapping is done external to the AOIs. If the ActMod.CVextSel=1, the PID will be placed in manual mode.
EUextSel	BOOL		Out		Local Extern Selected (Signal to PID)	A command to the PID to tell the PID that the PID.CV is being not being used when the ActMod is in Local (Sta.REU). If this functionality is to be used, the ActMod.EUextSel output must be written to the **PID.EUextSel** input. This logical mapping is done external to the AOIs. If the ActMod.EUextSel =1, the PID will be placed in manual mode. Field Pushbuttons GX/GY cause EUextSel to latch. Time in **Global.Par.EUactiveTime** to reset.
Alarm	BOOL		Out		Alarm Indicationß**Sta.F** OR **Sta.W** (Failure or Warning)	Indicates that either a failure or warning alarm is present with this device. The alarm output is set (HA, LA, ERR, KA, DevErr). Check must be in place for the Alarm output to work.

RdyOk	BOOL		Out	Used by Interlocks	Ready Ok Signal -> Interlock to Parent Module	Provides Ready OK for following equipment in auto sequence. Is always 1 if we don't have a alarm (NOT **Alarm -> Sta.KA, Sta.ERR, Sta.DevErr**).
SP	REAL		Out		Setpoint Output Scaled (in PV units)	The SP is the ActMod output in PV-units. *(i.e. the SP may be 100 representing % open but may equate to 20 ma on the analog output.)* This setpoint may be used by a PIDMod connect to **PID.CVext**. *See Wiring Diagram at the appendix* .
PV	REAL		Out		Process Value Scaled if **Par.InpScale** >0 else **PV**=**Inp**	The PV (Process Value) is similar to the AnaInp.PV. This is the EU scaled **Inp** value. The PV=**Inp** if scaling = 0 (Par.InpScale).
Out	REAL		Out		Setpoint Output Raw Value <-Scale specified by **Par.OutScale**	**ActMod.Out** is the output value that is mapped to the physical AnalogOutput module channel (i.e. 1756 hardware module). This scale using the **Par.OutScale** parameter. Set Par.OutScale = 0 there is no scaling performed (i.e. SP = Out)
Module Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Cmd.0\EZ	BOOL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Toggle **Sta.RZ** to select **Set.PVZ** if enabled by **Par.EnBypass**	**Cmd.0** = 1 will Toggle the Bypass (Replacement) status **Sta.RZ** if Bypass functionality is enabled by **Par.EnBypass**. Reserved for HMI Faceplate Interface. (Set in HMI, Reset in AOI)
Cmd.3\ACK	BOOL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Alarm Acknowledge	**Cmd.3** = 1 will Reset all alarms if the alarm condition has returned to normal. Reserved for HMI Faceplate Interface. Same functionality as **AlarmReset**. (Set in HMI, Reset in AOI)
Cmd.4\EU	BOOL		Inp	Noting, Tag direct linked by RSViewSE /FactoryTalk	Enable Local Operation (Toggle)->**Sta.REU**	Local mode allows to operate the device with Field interface (push buttons)
Set.SP	REAL		Inp	Tag direct linked by RSViewSE /FactoryTalk	Setpoint Input if enabled by **Sta.ESP**	**Set.SP** contains the Setpoint value from the HMI Faceplate. *See also Sta.ESP and SP*
Set.PVZ	REAL		Inp	Tag direct linked by RSViewSE /FactoryTalk	Replacement/Substitution Value if enabled by **Sta.RZ**	**Set.PVZ** contains the Bypass (Replacement) value to be used for **PV** if enabled by **Par.EnBypass** and **Sta.RZ** = 1. The value has the same units as the **PV** signal. *See also Par.EnBypass and Par.AutoBypass*
Set.MV/NV	REAL		Inp	Tag direct linked by RSViewSE /FactoryTalk	Max/Min Limit Value->**Sta.MA/NA** Stop Actuator Open/Close	Set Limit to Stop Actuator movement (ActPos). If PV > Set.MV then stop open. If PV < Set.NV then stop close. Note: this limit's don't generate Failure (Sta.F)
Set.HV/LV	REAL		Inp	Tag direct linked by RSViewSE /FactoryTalk	High/Low Alarm Limit Value->**Sta.HA/LA**	Set Limit for High and Low alarms. If PV > Set.HV then Sta.HA warning alarm. If PV < Set.NV then Sta.LA warning alarm.

Par.Type	SINT	X	Inp		Adjust function of module: 0=**Disable** Pulse Control and Limit/Alarm (Inp/Outp Scaling only) 1=Enable Pulse Control (by ActPos_AOI) 2=Enable Limit/Alarm->Bits **Sta.MA/HA/LA/NA** 3=Enable both, Pulse Control and Limit/Alarm	Set **Par.Type** to specify the desired ActMod module release of Alarms (**Sta.MA/HA/LA/NA**) and Pulse Control.The **Internal Check** signal is required for alarm bits to activate. Valid **Par.Type** settings are: 0 --> Disable Pulse Control & Alarm Bits (Scaling Only) 1 --> Enable Output Pulse Control (ActPos) 2 --> Enable Alarm BIts (**Sta.MA/HA/LA/NA**) 3 --> Enable Both Pulse Control and Alarm Bits -If the Par.Type is Pulsed (1 or 3), then the ActMod must be used in conjunction with the **ActPos** as an Sub block.
Par. InpScale	SINT	X	Inp		Selecting of the input presetting scales: 0=No Scaling ->**PV**=**Inp** **$-3.4e^{+38}..3.4e^{+38}$** x=Raw Scale **Par.InRawMax/Min** from **Global.Par.AnaInpScale[x]**; For 1..x -> specify PV Range by **Par.InEUMax/Min**	Set **Par.InpScale** to an integer (0,..7) to specify the raw scaling data (InRawMax/Min) to be used for scaling the **Inp** signal . The setting specifies the x element(index) of the **Global.Par.AnaInpScale[x]** array. Set **Par.InpScale** = 0 there is no scaling performed (i.e. **PV** = **Inp**). See also **Global.Par.AnaInpScale**
Par. InEUMax	REAL	X	Inp		Max/Min **PV**/**SP** Scaling Range in Engineering Units (PV units). If **InEUMax** < **InEUMin** then Negative Gradient Sta.NegGrad =ON in this case PV/SP Max will be **Val.MZ=InEUMin** and Min **NZ=InEUMax.**	Set **Par.InEUMax** to specify the **PV** signal Engineering Maximum value to be used for scaling the **Inp** signal. *Example: Set Par.EUMax = 0 and Par.EUMin = 100 for a 4-20 mA Inp signal that measures 100 - 0% respectively. In this case the Sta.NegGrad = 1. See also Val.MZ/NZ*
Par. InEUMin	REAL	X	Inp			Set **Par.InEUMin** to specify the **PV** signal Engineering Minimum value to be used for scaling the **Inp** signal. *Example: Set Par.EUMax = 0 and Par.EUMin = 100 for a 4-20 mA Inp signal that measures 100 - 0% respectively. In this case the Sta.NegGrad = 1. See also Val.MZ/NZ*
Par. OutScale	SINT		Inp	Set up the global tags to fit with 4-20, 0-5, etc and select the appropriate set for each AO instance.	Selecting of the analog output presetting scales: 0=No Scaling ->**Out**=**SP** **$-3.4e^{+38}..3.4e^{+38}$** x=Output Raw Scale from **Global.Par.AnaOutScale[x];**	Same strategy as the Inp Scaling, but this is the scaling to go from the SP (EU Out) to the Raw Out signal. The setting specifies the x element(index) of the **Global.Par.AnaOutScale**[x] array. Set **Par.OutScale** = 0 there is no scaling performed (i.e. Out = SP). See also **Global.Par.AnaOutScale**

Par. Filtering	SINT	X	Inp		Analog Input Signal Filter 1..99%=Filter -> **PV**=PV_{old}+1..99%(PV_{new}-PV_{old})/100*SampleTime/s 100%=Bypass Filter ->**PV** scaled or **PV**=**Inp** if **Par.InpScale**=0 Note, 1..99% is the max change (gradient) in % of PV per sec	Set **Par.Filtering** to a value (0-100%) to specify the filtering effect internal to the AnaInp. The internal filter is a low pass filter. **Par.Filtering** = 0 will disable the filter, PV = Inp. **Par.Filtering** = 100 will bypass the internal filter (**PV** = scaled **Inp**) **Par.Filtering** = 1..99 will enable the internal filter and the filter algorithm is **PV** = $\mathbf{PV}_{old}$ + **Par.Filtering***SampleTime*($\mathbf{PV}_{new}$-$\mathbf{PV}_{old}$) / 100 where SampleTime is the time between **Inp** samples. *Note: Par.Filtering is the maximum % change (gradient) of the PV per second. (i.e. 70% relates to a 1 second low pass filter) A higher Par.Filtering setting will allow the PV signal to change faster (i.e. less filtering).* *Note: Parameter are tested by: Par.Filter[%] x Par.SampleRate[ms] >99'999 (i.e. filter factor >1) then we bypass the Filter -> PVY=PVX.* *Important: Par.SampleRate take influence to the filter behavior.*
Par. RampRate	REAL		Inp		Ramp Rate for Local Setpoint Control in [PV units]	Step value to Add/Subtract to SP for each rising edge of the **GX/GY** PushButton signals. Units are same as PV.
Par. Deadband	REAL		Inp		Deadband/Hysteresis **(0..3.4e^{+38})** used for all Limits and Alarms	Set **Par.Deadband** to specify the value of deadband used for alarming bits (**Sta.MA/HA/LA/NA**). The deadband value has the same units as the **PV**. Refer to the Reference Guide Deadband Diagram for a signal diagram. Max & High control and alarm signals are considered normal after the **PV** <= **Set.MA** - **Par.Deadband**. Min & Low control and alarm signals are considered alarm whe **PV** >= **Set.NA** + **Par.Deadband**. *Example: If the high alarm limit is 10.5 and the deadband is 0.5, then the signal must return to less than 10.0 before the alarm will clear.*
Par. MaxClamp	REAL		Inp		Maximum Range Clamping Deadband	Set **Par.MaxClamp** to specify the value of clamping. Clamping is the tolerance from Max signal value to consider the signal to really be the Max value. Refer to the Reference Guide Clamping Diagram for the signal diagram. *Example: Par.InEUMin = 0, Par.In.EUMax = 100, and Par.MaxClamp = 1.0. Case 1: If the scaled filtered Inp signal > 99 then PV = 100 Case 2: If the scaled filtered Inp signal is between 1 and 99 then PV equals the scaled filtered Inp value.*

Par. MinClamp	REAL		Inp		Minimum Range Clamping Deadband	Set **Par.MinClamp** to specify the value of clamping. Clamping is the tolerance from Min signal value to consider the signal to really be the Min value. Refer to the Reference Guide Clamping Diagram for the signal diagram. *Example: Par.InEUMin = 0, Par.In.EUMax = 100, and Par.MinClamp = 1. Case 1: If the scaled filtered Inp signal < 1 then PV = 0. Case 2: If the scaled filtered Inp signal is between 1 and 99 then PV equals the scaled filtered Inp value.*
Par. DevDeadBand	REAL		Inp		Deadband used for deviation error between PV and SP	Set **Par.Deadband** to specify the value of deadband used for deviation alarm **Sta.DevErr**. The deadband value has the same units as the **PV**. Deviation is considered normal after the ABS(**PV** - **SP**) <= **Par.Deadband**.
Par. SampleRate	DINT	X	Inp	units [ms] Typically set to 1000ms	0=Always or >400 Preset time [ms] to update all Inputs. The Accum time since the last sample is used for the Filter SampleRate	Set Par.SampleRate to specify the desired frequency to execute the internal logic. This is the frequency that the PV will be updated. Note: The SampleRate can't go faster then 400ms! This is given by the MMCL Scan Control Function, see also Global.Scan1.Preset. Important: The ApplyPar input in the SysGrp must be set for a change to the SampleRate parameter to take effect. Note: SampleRate take influence to the Filter configuration; see Par.Filter.
Par.AlarmDelay_MA	DINT		Inp	units [ms]	N/A	N/A
Par.AlarmDelay_HA	DINT		Inp	units [ms]	Delay Time for high Alarm ->Sta.HA	Set **Par.AlarmDelay_HA** to a value to specify the Time Delay in ms used for high Threshold alarm. (**Sta.HA**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.AlarmDelay_LA	DINT		Inp	units [ms]	Delay Time for low Alarm ->Sta.LA	Set **Par.AlarmDelay_LA** to a value to specify the Time Delay in ms used for low Threshold alarm. (**Sta.LA**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.AlarmDelay_NA	DINT		Inp	units [ms]	N/A	N/A
Par.AlarmDelay_Dev	DINT		Inp	units [ms]	Delay Time for deviation Alarm ->Sta.DevErr	Set **Par.AlarmDelay_DEV** to a value to specify the Time Delay in ms used for deviation alarm between **PV** and **SP**. (**Sta.DevErr**). The alarm condition must be present for this amount of time before the alarm will become active.
Par.FailedStatePos	INT		Inp		Setpoint of failed state position if deviation error occurred	Set **Par.FailedStatePos** to a value to specify the desired module output setpoint in case of a deviation error between **PV** and **SP**. 0 = Last position; 1 = 100% UOM; 2 = 0% UOM;

Par. EnBypass	BOOL		Inp		Enable Bypass ->Manually **Set.PVZ**	Set **Par.EnBypass** = 1 to specify that the **PV** signal can be bypassed (replaced) and that bypass information should be visible on the HMI Faceplate. This setting only enables the ability to bypass the **PV**. Set Par.EnBypass = 0 to prevent the **PV** signal from being bypassed. See also **Cmd.0**, **Par.AutoBypass**, **Set.PVZ**, and **Sta.RZ**.
Par. AutoBypass	BOOL		Inp		Enabling Auto Bypass ->Automatically **Set.PVZ** if **Sta.ERR**, Reset Manually	Set **Par.AutoBypass** = 1 to specify that the **PV** bypassed should be automatically bypassed (**Sta.RZ**) if a sensor error alarm (**Sta.ERR**) or Availability alarm (**Sta.KA**) become active. Resetting the bypass condition is Always done manually with **Cmd.0**. See also **Set.PVZ**.
Par. DisableGrp Check	BOOL		Inp		Disable Alarm Check by Parent Group modules. If ON, Warning or Failure Alarms are not released by the check signal from the Parent Group module. Local check is however still active and may set **Sta.W** or **Sta.F**	Set **Par.DisableGrpCheck** to 1 to prevent **Check -**condition from Parent module (<Parent>.Bus.26) to enabling the Module's **Internal Check**. The **Check** Input and **<Parent>.Bus.8** still enable the Module's **Internal Check**. If set to 1 then no warning(**Sta.W**) or error(**Sta.F**) will be passed on ParentBus. The **group does not** receive fault / warnings Info .Alarms however is Indicated on HMI.
Par. ErrorWarnin g	BOOL		Inp		Set module as warning device only. Severity/Error Code 0=Failure, 1=Warning If OFF, Alarms are indicated by **Sta.W** IF ON, Alarms are indicated by **Sta.F**	Set **Par.ErrorWarning** = 1 to specify that **Sta.ERR**, **Sta.KA**, **Sta.MA**, and **Sta.NA** alarms will be reported as Warning alarms (**Sta.W**). Set **Par.ErrorWarning** = 0 to specify that these alarms should be reported as Failure alarms (**Sta.F**) Examle: A device with set Par.ErrorWarning bit is configured as a "Warnning device" only -> a failure will never stop the Group.
Par. DisableLocal	BOOL		Inp		Suppress visibility of Local Button at HMI template.	Set Par.DisableLocal =1 to specify that the "Local"-push button at the HMI-template are not visible. This parameter is direct linked by RSView. (none module logic influence)
Par. EnDevErr	BOOL		Inp		Enable deviation error alarm	Set **Par.EnDevErr** = 1 to specify that the deviation error alarm **Sta.DevErr** is enabled.
Par. DisableSens orErr	BOOL		Inp		Disable sensor error alarm	Set **Par.DisableSensorErr** = 1 to specify that sensor error **Sta.ERR** will not be generated and reported.
Par. DisableGrpI dentify	BOOL		Inp		Disable device being identified	Set **Par.DisableGrpIdentify** = 1 to specify that the device will not be identified by its parent control group. This parameter allow individual device ignore group identify command respectively.

Par. AlarmGong Code	DINT		Inp		Set Alarm Gong adresse# . Enable Gong/Sound 0..31 for Failures and Warnings. Select a pair of Gongs by bit pattern, this will set the **Global.FailureGongCode** or .**WarningGongCode** on the respective alarm. Note, Alarm Gongs are not set if Local is enabled.	Gongs are sound devices in the control room. For each CLX PLC, there are 32 configurable Failure Alarm Gongs and 32 configurable Warning Alarm Gongs. When a Warning (**Sta.W**) occurs, the warning gong will sound and when a Failure (**Sta.F**) occurs, the failure gong will sound. A common parameter **Par.AlarmGongCode** is used to identify which gongs should sound for this device. Set the device's **Par.AlarmGongCode** bits that are mapped to the desired gongs. When a **Sta.W** occurs this gong code is mapped to the **Global.WarningGongCode** tag (Indirectly the SysGrp AOI maps the **Global.WarningGongCode** is mapped to the **<SysGrp>.WarningGong** tag). Similarly, the **Sta.F** causes the **Par.AlarmGongCode** to map to the **Global.FailureGongCode** and indirectly to the **<SysGrp>.FailureGong** tag
Sta.ESP	BOOL		Out		Status: Enable Setpoint Input	**Sta.ESP** = 1 Identifies that the device is taking the Set.SP as the actual setpoint.
Sta.RZ	BOOL		Out		Bypass Enabled, **PV=Set.PVZ** Replacement/Substitution Value	**Sta.RZ** = 1 Identifies that the device is "Replacement" his PV (analog Input)
Sta.KM	BOOL		Out		Device Available Ok Mimic	**Sta.KM** = 1 Identifies that the device is ready. **Sta.KM** = 0 Identifies that the device is not ready due to one of the device input signals being in the incorrect state: **K, ERR, DevErr** or **limit alarms**. Indicated in HMI as purple color.
Sta.REU	BOOL		Out		Run Local Mode (Feedback to Group Module) Note, Alarm Gongs are not set if Local is enabled.	**Sta.REU** = 1 Identifies that the device is in "Local" mode.
Sta.F	BOOL		Out		Device Failure Mimc (Bus to Parent Module)	**Sta.F** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 0. Module must be supervised (i.e. **Internal Check** = 1)
Sta.W	BOOL		Out		Device Warning Mimc (Bus to Parent Module)	**Sta.W** = 1 Identifies that the device is in **Alarm** and the **Par.ErrorWarning** = 1. Module must be supervised (i.e. **Internal Check** = 1)
Sta.ERR	BOOL		Out		Sensor Error Message	**Sta.ERR** = 1 Identifies an SensorError such as input **ERR** =1or **Inp** Under / Over -Range (broken wire)
Sta.ERRM	BOOL		Out		Sensor Error Mimic	**Sta.ERRM** = 1 shows an SensorError message at the HMI
Sta.KA	BOOL		Out		Availability Alarm Message	**Sta.KA** = 1 Identifies an Availability Alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the **K** signal input when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.KAM	BOOL		Out		Availability Alarm Mimic	**Sta.KAM** = 1 Identifies that there is no **K** input signal or there is an active **Sta.KA** alarm. This is used for HMI display (mimic) animiations to identify that there is an Availability problem.

Sta.MA/NA	BOOL		Out	(HMI Alarm Tag)	Max/Min Limit <-**Set.MV/NV** Stop Actuator Open/Close Only Overtravel indication (Max/Min Position) Stopped open/close at POSP (ActPos_AOI)	**Sta.MA/NA** = 1 Identifies a **max** or **min** limit alarm overtravel. A failure has occurred due to the enabled (EMA/ENA) alarm Limits when the device's parent group is active (i.e. **Internal Check** = 1) No Sta.F or Sta.W!
Sta.HA/LA	BOOL		Out		High/Low Limit Alarm <-**Set.HV/LV** and **Par.Deadband**	**Sta.HA/LA** = 1 Identifies a **high** or **low** limit alarm and reports a message to the HMI Alarm Log. A failure has occurred due to the enabled (EHA/ELA) alarm Limits when the device's parent group is active (i.e. **Internal Check** = 1)
Sta.NegGrad	BOOL		Out	Tag direct linked by RSViewSE /FactoryTalk	Scaling Negative Gradient =ON if **Par.InEUMax<Par.InEUMin,** in this case **PV/SP Max MZ=InEUMin** and **Min NZ=InEUMax**	**Sta.NegGrad** =1 That flag be used to flip the bargraph to display from top down in RSView. The method allows to indicate "under pressures" the same way as positive/negative pressures
Sta.ParErr	BOOL		Out		Parameter Error (check scaling min/max values and Set.MV/HV/LV/NV values)	Sta.ParErr =1 Identifies that there is a incorrect parameter set. **Par.InEUMin** equal **Par.InEUMax** OR GlobalData.Par.AnaInpScale.**InRawMin** >= **...InRawMax** OR **Not Set.NV < LV < HV < MV.**
Sta.PAM	BOOL		Out		Parameter alarm mimic if NOT (NV < LV < HV < MV)	**Sta.PAM** = 1 identifies that there is a analog module alarm threshold limit configuration error. Correct limit setting should be NV < LV < HV < MV. This is used for HMI display (mimic) animations to identify that there is an alarm limit configuration problem.
Sta.DevErr	BOOL		Out		Deviation error alarm between PV and SP	**Sta.DevErr** = 1 identifies that there is a deviation error alarm. Alarm is generated if the difference between **PV** and **SP** is greater than the specified deadband value **Par.DevdeadBand** over a period of time specified in **Par.AlarmDelay_DEV.**
Sta.DevErrM	BOOL		Out		Deviation Alarm Mimic	**Sta.DevErrM** = 1 Identifies that there is a deviation error alarm. This is used for HMI display (mimic) animations to identify that there is an deviation problem.
Sta.GrpIdentify	BOOL		Out		Group device identify indication	**Sta.GrpIdentify** = 1 Identifies that control group is requesting to identify its children over the control bus and this module device is also set to enable identify.
Val.PV	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Value status: Process/Actual Value=**PV** for Indication	Value for HMI indication.
Val.PVA	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Value status: Process Value Alarm=**PV** at Detection of Alarm	Value for HMI threshold value alarm. The RSViewSE alarm setup will create his owen alarm messages with threshold limit value.

Val.PVY	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Value status: Indicate Real Process Value **PV** healthiness regardless of Replacement Value set	Value for HMI indication.
Val.SPZ	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Value status: Actual Setpoint **SP** Feedback	Value for HMI indication.
Val.MZ/NZ	REAL		Out	Tag direct linked by RSViewSE /FactoryTalk	Max/Min PV/SP Range **MZ=InEUMax** and **NZ=InEUMin** if **InEUMax > InEUMin** else **MZ=InEUMin** and **NZ=InEUMax**	**Val.MZ/NZ** are maximum and minimum limits that are used by the HMI for **SP** and **PV** data entry animations. - **Val.MZ** is the higher of **Par.InEUMax** or **Par.InEUMin** - **Val.NZ** is the lower of **Par.InEUMax** or **Par.InEUMin**
Parent Bus	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ParentBus	DINT	X	I/O	Link to the Parent Module	Bus from/to Parent Module (Group, Machine)	
Reference Notes						
Note 1:						
Internal Check is refered to in the module descriptions above and the drivers that can create an **Internal Check** condition are: <Parent>.Bus.26 with DisableGrpCheck = 0, Sta.REU (Local Mode), OR Sta.REG (Single Mode)						

ActMod_SIM

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
EUMax	REAL	x	Inp	Direct allocated to the ActMod AOI	Maximum scaling range in Engineering Units	Input from ActMod module .Par.InEUMax. EUMax value is used to scale SP_IN to Out_Raw in automatic mode.
EUMin	REAL	x	Inp	Direct allocated to the ActMod AOI	Minimum scaling range in Engineering Units	Input from ActMod module .Par.InEUMin. EUMin value is used to scale SP_IN to Out_Raw in automatic mode.
outRawMax	REAL	x	Inp	Direct allocated to the ActMod AOI	Maximum raw value for analog scaling from ActMod	Input from Global.Par.AnaInpScale[ActMod .Par.InpScale].InRawMax. outRawMax value is used to scale SP_IN to Out_Raw in automatic mode.
outRawMin	REAL	x	Inp	Direct allocated to the ActMod AOI	Minimum raw value for analog scaling from ActMod	Input from Global.Par.AnaInpScale[ActMod .Par.InpScale].InRawMin. outRawMin value is used to scale SP_IN to Out_Raw in automatic mode.
SP_IN	REAL	x	Inp	Direct allocated to the ActMod AOI	Set Point output from ActMod	SP_IN input is used to be converted to ActMod simulation output when in Automatic mode.
OUT_RAW	REAL	x	Out	Direct allocated to the ActMod AOI	Raw analog value Input to ActMod	ActMod simulation output in RAW value
GX_OUT	BOOL	x	Out	Direct allocated to the ActMod AOI	Local Start X (Open/Rise) to ActMod	GX_OUT bit can be set by button on HMI simulation faceplate.
GY_OUT	BOOL	x	Out	Direct allocated to the ActMod AOI	Local Start Y (Close/Lower) to ActMod	GY_OUT bit can be set by button on HMI simulation faceplate.
Err	BOOL	x	Out	Direct allocated to the ActMod AOI	Transmitter Error to ActMod	Err bit can be set by switch on HMI faceplate. Sensor Error alarm will be displayed on ActMod faceplate.

Local Data	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Err_Fault	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Error fault from HMI	Err_Fault bit is connected with HMI Error switch. (0=Ok, 1=Fault)
GX	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Open control from HMI	GX bit is connected with HMI Local Open button. 1=Open
GY	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Local Close control from HMI	GY bit is connected with HMI Local Close button. 1=Close
Manual_Automatic	BOOL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Manual/Automatic mode (0=manual, 1=automatic) from HMI	Manual_Automatic bit is connected with HMI Mode switch. When simulation is set to Automatic, output value is taken from SP_IN and scaled to correct RAW value. In Manual Mode output can be set by Output Raw slider on HMI faceplate.
OUT_HMI	REAL			Nothing. Tag directly linked by RSViewSE/FactoryTalk	Raw value in manual mode from HMI	Output value from HMI in Raw value

ActPos

Input/Output	Data Type	User Config?	I/O	Configuration Required	Description	Notes
ZX	BOOL	X	Inp	Direct allocated to the AOI	Endposition Limit Swich Open 1=Open	**ZX/ZY** are External Limit Switch input signals used to indicate physical position for maximum travel (direction specific). Signal is used to Stop device after desired position is achieved. **ZX/ZY** = 0 to start/run in that direction (**DX/DY** respectively) **ZX/ZY** = 1 to stop traveling in the respective direction (**DX/DY** respectively)
ZY	BOOL	X	Inp	Direct allocated to the AOI	Endposition Limit Swich Closed 1=Closed	
DX	BOOL	X	Out	Typically coded in a rung after the AOI Instruction	Digital Output X Open/Raise	**DX** is the Device Command to Start output signal. **DX** = 1 when device is commanded to start (run) and is mapped to the device's contactor (motor) or coil (valve).
DY	BOOL	X	Out	Typically coded in a rung after the AOI Instruction	Digital Output Y Close/Lower	**DY** is the Device Command to Start output signal. **DY** = 1 when device is commanded to start (run) and is mapped to the device's contactor (motor) or coil (valve).

POSITION_PROP	Data Type	User Config?	I/O	Configuration Required	Description	Notes
Pos.EnableIn	BOOL		Inp		**Function Block:** If cleared, the instruction does not execute and outputs are not updated. If set, the instruction executes. Set by **Par.Type**	
Pos.SP	REAL		Inp		Setpoint. This is the desired value for the position. This value must use the same engineering units as Positionßautomatically taken from **SP** Valid = any float, Default = 0.0	
Pos.Position	REAL		Inp		Position feedback. This analog input comes from the position feedback from the deviceßautomatically taken from **PV** Valid = any float, Default = 0.0	
Pos.OpenedFB	BOOL		Inp	Direct linked by ActMOD_MOI	Opened feedback. This input signals when the device is fully opened. When set, the open output is not allowed to turn onß**ZX**. Default is cleared.	See Logix5000 Online Help **POSP**
Pos.ClosedFB	BOOL		Inp		Closed feedback. This input signals when the device is fully closed. When set, the close output is not allowed to turn onß**ZY**. Default is cleared.	

Pos.PositionEUMax	REAL		Inp		Maximum scaled value of Position and SPßtaken from **Val.MZ** Valid = any float, Default = 100.0	
Pos.PositionEUMin	REAL		Inp		Minimum scaled value of Position andSPßtakenfrom**Val.NZ** Valid = any float, Default = 0.0	
Pos.CycleTime	REAL	X	Inp	Enter a test value of e.g. 8.0 [s]	Period of the output pulse in [s]. A value of zero clears both OpenOut and CloseOut. If this value is invalid, the instruction assumes a value of zero and sets the appropriate bit in Pos.Status. Valid = any positive float. **Enter a test value of e.g. 8.0 [s]**	
Pos.OpenRate	REAL	X	Inp	Enter a test value of e.g. 1.0 [%/s]	Open rate of the device in [%/s]. A value of zero clears OpenOut. If this value is invalid, the instruction assumes a value of zero and sets the appropriate bit in Pos.Status. Valid = any positive float. **Enter a test value of e.g. 1.0 [%/s]**	
Pos.CloseRate	REAL	X	Inp	Enter a test value of e.g. 1.0 [%/s]	Close rate of the device in [%/s]. A value of zero clears CloseOut. If this value is invalid, the instruction assumes a value of zero and sets the appropriate bit in Pos.Status. Valid = any positive float. **Enter a test value of e.g. 1.0 [%/s]**	
Pos.MaxOnTime	REAL	X	Inp	Enter a test value of e.g. 5.0 [s]	Maximum time in [s] that an open or close pulse can be on. If OpenTime or CloseTime is calculated to be larger than this value, they are limited to this value. If this value is invalid, the instruction assumes a value of CycleTime and sets the appropriate bit in Pos.Status. Valid = 0.0 to CycleTime. **Enter a test value of e.g. 5.0 [s]**	
Pos.MinOnTime	REAL	X	Inp	Enter a test value of e.g. 2.0 [s]	Minimum time in [s] that an open or close pulse can be on. If OpenTime or CloseTime is calculated to be less than this value, they are set to zero. If this value is invalid, the instruction assumes a value of zero and sets the appropriate bit in Pos.Status. Valid = 0.0 to MaxOnTime. **Enter a test value of e.g. 2.0 [s]**	

Pos.Deadtime	REAL	X	Inp	Enter a test value of e.g. 0.1 [s]	Additional pulse time in [s] to overcome friction in the device. Deadtime is added to the OpenTime or CloseTime when the device changes direction or is stopped. If this value is invalid, the instruction sets the appropriate bit in Status and uses a value of Deadtime = 0.0. Valid = 0.0 to MaxOnTime. **Enter a test value of e.g. 0.1 [s]**	
Pos.EnableOut	BOOL		Out		Enable output.	
Pos.OpenOut	BOOL		Out		This output is pulsed to open the device ->Moved to DX	
Pos.CloseOut	BOOL		Out		This output is pulsed to close the device ->Moved to DY	
Pos.PositionPercent	REAL		Out		Position feedback is expressed as percent of the Position span. Arithmetic status flags are set for this output.	
Pos.SPPercent	REAL		Out		Setpoint is expressed as percent of the Position span.	
Pos.OpenTime	REAL		Out		Pulse time in [s] of OpenOutput for the current cycle.	
Pos.CloseTime	REAL		Out		Pulse time in [s] of CloseOutput for the current cycle.	
Pos.Status	DINT		Out		Status of the function block.	
Pos.InstructFault = Pos.Status.0	BOOL		Out		The instruction detected one of the following execution errors. This is not a minor or major controller error. Check the remaining status bits to determine what occurred.	
Pos.CycleTimeInv = Pos.Status.1	BOOL		Out		Invalid CycleTime value. The instruction uses zero.	
Pos.OpenRateInv = Pos.Status.2	BOOL		Out		Invalid OpenRate value. The instruction uses zero.	
Pos.CloseRateInv = Pos.Status.3	BOOL		Out		Invalid CloseRate value. The instruction uses zero.	
Pos.MaxOnTimeInv = Pos.Status.4	BOOL		Out		Invalid MaxOnTime value. The instruction uses the CycleTime value.	
Pos.MinOnTimeInv = Pos.Status.5	BOOL		Out		Invalid MinOnTime value. The instruction uses zero.	
Pos.DeadtimeInv = Pos.Status.6	BOOL		Out		Invalid Deadtime value. The instruction uses zero.	
Pos.PositionPctInv = Pos.Status.7	BOOL		Out		The calculated PositionPercent value is out of range.	

Pos.SPPercentInv = Pos.Status.8	BOOL		Out		The calculated SPPercent value is out of range.	
Pos.PositionSpanInv = Pos.Status.9	BOOL		Out		PositionEUMax = PositionEUMin.	

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication RA-RM002B-EN-P - November 2010
Supersedes Publication RA-RM002A-EN-P - August 2007