



# PlantPax Control Strategies

**PlantPax Process Library Release 5.20**



***Allen-Bradley***

by ROCKWELL AUTOMATION

**Reference Manual**

Original Instructions

# Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

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.

---

**IMPORTANT** Identifies information that is critical for successful application and understanding of the product.

---

These labels may also be on or inside the equipment to provide specific precautions.



**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.

---



**ARC FLASH HAZARD:** Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

---

The following icon may appear in the text of this document.



Identifies information that is useful and can help to make a process easier to do or easier to understand.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

	<b>Preface</b> .....	<b>19</b>
	About This Publication .....	19
	Download Firmware, AOP, EDS, and Other Files .....	19
	Summary of Changes .....	19
	Additional Resources .....	20
	<b>Chapter 1</b>	
<b>PlantPax Control Strategies</b>	Library Prerequisites .....	21
	How to use PlantPax Control Strategies .....	21
	Import with Studio 5000 Logix Designer .....	22
	Import with Application Code Manager Plug-ins .....	24
	Interlock Options .....	29
	Interlock Sheet .....	30
	PINTLK Input Reference .....	30
	PINTLK Output Reference .....	30
	PINTLK Configuration Considerations .....	30
	I/O Connections .....	30
	Map Device Tags to Input Data .....	31
	Map Tags to Output Data .....	32
	Map I/O Faults .....	33
	<b>Chapter 2</b>	
<b>ACM Considerations</b>	Process Controller .....	36
	ACM_Based Parameters for a Process Controller .....	37
	I/O Mapping .....	38
	I/O Map Strategy = 0 .....	39
	IO_Map_Strategy = 1 .....	40
	IO_Map_Strategy = 2 .....	40
	IO_Map_Strategy = 3 .....	41
	IO_Map_Strategy = 4 .....	43
	IO_Map_Strategy = 5 .....	44
	Skip_IO and Skip_UnusedIO .....	45
	Ownership/Arbitration .....	47
	Organization Bus .....	48
	Hardware Bus and Hardware Module Status .....	48
	Sub-Objects .....	49
	Event Logging .....	49
	Interlocks .....	49
	Permissives .....	50
	Device Object [Cfg_HasDvcObj] .....	51
	Extended Alarms .....	54
	Parameters .....	54
	Reports .....	55
	EM and EP Additional	
	Sub-Objects .....	55
	Local Tags .....	55

	Associated Devices.....	56
	Local Logic.....	56
	Transition Diagnostics.....	57
	Statistics Objects.....	57
	Runtime Object.....	57
	Restart Inhibit.....	58
	Valve Statistics.....	58
	ACM Considerations for a Prompt Instance.....	60
	ACM-Based Parameters for a Prompt Instance.....	60
	Additional Sub-Objects for an Area Instance.....	60
	<b>Chapter 3</b>	
<b>HART Integration</b>	HART Data.....	61
	Fully Integrated HART with FLEX 5000 I/O.....	62
	Integrate FLEX 5000 I/O with HART Device Using PAH and PAI Instructions.....	62
	Integrate FLEX 5000 with HART Device via PV, SV, TV, or QV Values.....	67
	Integrate Other HART Modules with the Process Controller.....	71
	Map HART Device to PAH from Non-FLEX 5000 I/O.....	73
	Map HART Device Digital Data to PAI from Non-FLEX 5000 I/O.....	79
	<b>Chapter 4</b>	
<b>EtherNet/IP Integration</b>	EtherNet/IP Data.....	85
	EtherNet/IP Instructions.....	86
	Flowmeter Example - raP_Dvc_EH_Flowmeter.....	86
	Heartbeat Example - raP_Dvc_EH_Heartbeat.....	89
	Sensor Example - raP_Dvc_EH_Sensor.....	90
	Integrate an EtherNet/IP E+H Flowmeter Device.....	91
	Integrate an EtherNet/IP E+H Sensor Device.....	95
	<b>Chapter 5</b>	
<b>FOUNDATION Fieldbus and Profibus PA Integration</b>	FOUNDATION Fieldbus and Profibus PA Instructions.....	99
	Foundation Fieldbus Link Example - raP_Dvc_AP_FFLink.....	99
	Profibus PA Link Example - raP_Dvc_AP_PALink.....	100
	Integrate a FOUNDATION Fieldbus Device.....	101
	Integrate a Profibus PA Device.....	105
	<b>Chapter 6</b>	
<b>Controller Fault Handler Control Strategy</b>	Example Programs.....	110
	Controller Fault Handler Program: MainRoutine Routine.....	110
	Controller Fault Handler Program: raC_Opr_LgxFault Routine.....	110
	GlobalFunctions Program: MajorFault Routine.....	111
	<b>Chapter 7</b>	
<b>Coordinated Control (CC) Control Strategies</b>	CS_CC Sheet.....	115
	PAI Input References.....	115
	PAI Outputs to CC Inputs.....	115
	Input References to CC.....	115



	CC Configuration Considerations .....	115
	CS_CC_HART Sheet .....	116
	CS_CC_EtherNetIP Sheet .....	117
	CS_CC_FF Sheet .....	118
	CS_CC_PA Sheet .....	118
	Control Variable Routines .....	119
	ACM Considerations for CC .....	119
	ACM-Based Parameters for a CC Instance .....	120
	Additional Sub-Objects for a CC Instance .....	122
	<b>Chapter 8</b>	
<b>Internal Model Control (IMC)</b>	CS_IMC Sheet .....	125
<b>Control Strategies</b>	PAI Input References .....	125
	PAI Outputs to IMC Inputs .....	125
	IMC Input Reference .....	125
	IMC Configuration Considerations .....	125
	CS_IMC_HART Sheet .....	126
	CS_IMC_EtherNetIP Sheet .....	127
	CS_IMC_FF Sheet .....	128
	CS_IMC_PA Sheet .....	128
	Control Variable Routines .....	129
	ACM Considerations for IMC .....	129
	ACM-Based Parameters for a IMC Instance .....	130
	Additional Sub-Objects for an IMC Instance .....	131
	<b>Chapter 9</b>	
<b>Modular Multivariable Control</b>	CS_MMC Sheet .....	135
<b>(MMC) Control Strategies</b>	PAI Input References .....	136
	PAI Outputs to MMC Inputs .....	136
	MMC Input References .....	136
	MMC Configuration Considerations .....	136
	CS_MMC_HART Sheet .....	137
	CS_MMC_EtherNetIP Sheet .....	138
	CS_MMC_FF Sheet .....	139
	CS_MMC_PA Sheet .....	140
	Control Variable Routines .....	141
	ACM Considerations for MMC .....	141
	ACM-Based Parameters for an MMC Instance .....	142
	Additional Sub-Objects for a MMC Instance .....	145
	<b>Chapter 10</b>	
<b>Process Analog Input (PAI)</b>	CS_PAI Sheet .....	148
<b>Control Strategies</b>	PAI Input References .....	148
	PAI Configuration Considerations .....	148
	CS_PAI_HART Sheet .....	149
	PAH Outputs to PAI Inputs .....	149
	PAH Status Outputs .....	149
	PAH Configuration Considerations .....	149

CS_PA1_EtherNetIP Sheet .....	151
raP_Dvc_EH_Flowmeter Outputs to PAI Inputs .....	151
raP_Dvc_EH_Flowmeter Status Outputs .....	151
CS_PA1_EtherNetIP_NoHB Sheet .....	153
raP_Dvc_EH_Heartbeat Outputs to raP_Dvc_EH_Flowmeter Inputs .....	153
raP_Dvc_EH_Heartbeat Configuration Considerations .....	153
CS_PA1_EtherNetIP_Sensor Sheet .....	154
raP_Dvc_EH_Sensor Outputs to PAI Input .....	154
raP_Dvc_EH_Sensor Configuration Considerations .....	154
CS_PA1_FF Sheet .....	155
raP_Dvc_AP_FFLink Outputs to PAI Inputs .....	155
raP_Dvc_AP_FFLink Configuration Considerations .....	155
CS_PA1_PA Sheet .....	156
raP_Dvc_AP_PALink Outputs to PAI Inputs .....	156
raP_Dvc_AP_PALink Configuration Considerations .....	156
ACM Considerations for PAI .....	156
ACM-Based Parameters for a PAI Instance .....	157
Additional Sub-Object for a PAI Instance .....	158

## Chapter 11

### Process Analog Dual Sensor Input (PAID) Control Strategies

CS_PAID Sheet .....	160
PAI Input References .....	160
PAI Outputs to PAID Inputs .....	161
PAID Outputs to PAI Inputs .....	161
PAID Configuration Considerations .....	161
CS_PAID_HART Sheet .....	162
CS_PAID_EtherNetIP Sheet .....	163
CS_PAID_EtherNetIP_NoHB Sheet .....	164
CS_PAID_FF Sheet .....	165
CS_PAID_PA Sheet .....	166
ACM Considerations for PAID .....	166
ACM-Based Parameters for a PAID Instance .....	167
Additional Sub-Object for a PAID Instance .....	168

## Chapter 12

### Process Analog Input Multi Sensor (PAIM) Control Strategies

CS_PAIM Sheet .....	170
PAI Input References .....	170
PAI Outputs to PAIM Inputs .....	171
PAIM Outputs to PAI Inputs .....	171
PAIM Configuration Considerations .....	171
CS_PAIM_HART Sheet .....	172
CS_PAIM_EtherNetIP Sheet .....	173
CS_PAIM_EtherNetIP_NoHB Sheet .....	174
CS_PAIM_FF Sheet .....	175
CS_PAIM_PA Sheet .....	176
ACM Considerations for PAIM .....	176
ACM-Based Parameters for a PAIM Instance .....	176
Additional Sub-Object for a PAIM Instance .....	178

<b>Process Analog Output (PAO) Control Strategies</b>	<b>Chapter 13</b>	
	CS_PAO Sheet	180
	PAO Input References	180
	PAO Output References	180
	PAO Configuration Considerations	181
	IO Faults Sheet	181
	Fault Input References	181
	Fault Output References	181
	CS_PAO_HART Sheet	182
	ACM Considerations for PAO	182
	ACM-Based Parameters for a PAO Instance	183
	Additional Sub-Objects for a PAO Instance	184
<b>Process Boolean Logic (PBL) Control Strategy</b>	<b>Chapter 14</b>	
	CS_PBL Sheet	188
	PBL Input References	188
	PBL Output Reference	188
	PBL Configuration Considerations	188
	ACM Considerations for a PBL Instance	188
<b>Process Discrete 2-, 3-, or 4- State Device (PD4SD) Control Strategy</b>	<b>Chapter 15</b>	
	CS_PD4SD Sheet	192
	PD4SD Input References	192
	PD4SD Output References	193
	PD4SD Configuration Considerations	193
	Permissive Sheet	193
	PPERM Input References	193
	PPERM Output References	193
	IO Fault Sheet	194
	Fault Input References	195
	Fault Output References	195
	ACM Considerations for a PD4SD Instance	195
	ACM-Based Parameters for a PD4SD Instance	196
	Additional Sub-Objects for a PD4SD Instance	196
<b>Process Deadband Controller (PDBC) Control Strategies</b>	<b>Chapter 16</b>	
	CS_PDBC Sheet	199
	PAI Input References	199
	PAI Outputs to PDBC Inputs	200
	PDBC Configuration Considerations	200
	CS_PDBC_HART Sheet	201
	CS_PDBC_EtherNetIP Sheet	201
	CS_PDBC_FF Sheet	202
	CS_PDBC_PA Sheet	202
	ACM Considerations for PDBC	203
	ACM-Based Parameters for a PDBC Instance	204

	Additional Sub-Object for a PDBC Instance .....	205
<b>Process Discrete Input (PDI) Control Strategy</b>	<b>Chapter 17</b>	
	CS_PDI Sheet .....	207
	PDI Input References .....	207
	PDI Configuration Considerations .....	207
	ACM Considerations for a PDI Instance .....	208
	ACM-Based Parameters for a PDI Instance .....	208
	Additional Sub-Object for a PDI Instance .....	208
<b>Process Discrete Output (PDO) Control Strategies</b>	<b>Chapter 18</b>	
	CS_PDO Sheet .....	210
	PDO Input References .....	210
	PDO Output References .....	210
	PDO Configuration Considerations .....	211
	Permissive Sheet .....	211
	PPERM Input References .....	211
	PPERM Output References .....	211
	IO Faults Sheet .....	212
	Fault Input References .....	212
	Fault Output References .....	212
	ACM Considerations for a PDO Instance .....	212
	ACM-Based Parameters for a PDO Instance .....	213
	Additional Sub-Objects for a PDO Instance .....	213
<b>Process Dosing Flow Meter (PDOSEFM) Control Strategies</b>	<b>Chapter 19</b>	
	CS_PDSEFM Sheet .....	216
	PAI Input References .....	216
	PAI Outputs to PDOSE Inputs .....	217
	PDOSE Output References .....	217
	PDOSE Configuration Considerations .....	217
	CS_PDSEFM_HART Sheet .....	218
	CS_PDSEFM_EtherNetIP Sheet .....	219
	CS_PDSEFM_EtherNetIP_ NoHB Sheet .....	220
	CS_PDSEFM_FF Sheet .....	220
	CS_PDSEFM_PA Sheet .....	221
	ACM Considerations for PDOSE .....	222
	ACM-Based Parameters for a PDOSE Instance .....	222
	Additional Sub-Object for a PDOSE Instance .....	225
<b>Process Dosing Weigh Scale (PDOSEWS) Control Strategy</b>	<b>Chapter 20</b>	
	CS_PDSEWS Sheet .....	228
	PAI Input References .....	228
	PAI Outputs to PDOSE Inputs .....	229
	PDOSE Output References .....	229
	PDOSE Configuration Considerations .....	229
	CS_PDSEWS_HART Sheet .....	230

	CS_PDSEWS_EtherNetIP Sheet .....	231
	CS_PDSEWS_EtherNetIP_	
	NoHB Sheet .....	231
	CS_PDSEWS_FF Sheet .....	232
	CS_PDSEWS_PA Sheet .....	232
	ACM Considerations for PDOSE .....	233
	ACM_Based Parameters for a PDOSE Instance .....	233
	Additional Sub-Object for a PDOSE Instance .....	236
	<b>Chapter 21</b>	
<b>Process Analog Fanout (PFO)</b>	CS_PFO Sheet .....	237
<b>Control Strategies</b>	PFO Input References .....	238
	PFO Output References .....	238
	PFO Configuration Considerations .....	238
	ACM Considerations for PFO .....	238
	ACM_Based Parameters for a PFO Instance .....	239
	Additional Sub-Object for a PFO Instance .....	239
	<b>Chapter 22</b>	
<b>Process High or Low Selector</b>	CS_PHLS Sheet .....	241
<b>(PHLS) Control Strategies</b>	PHLS Input References .....	242
	PHLS Output References .....	242
	PHLS Configuration Considerations .....	242
	ACM Considerations for PHLS .....	242
	ACM_Based Parameters for a PHLS Instance .....	243
	Additional Sub-Object for a PHLS Instance .....	243
	<b>Chapter 23</b>	
<b>Process Lead Lag Standby</b>	CS_PLLS_PMTR Sheet .....	246
<b>(PLLS) Control Strategy</b>	PLLS Output References .....	247
	PLLS Configuration Considerations .....	247
	Permissive Sheet .....	247
	PPERM Input References .....	247
	PPERM Output References .....	247
	CS_PLLS_PVSD Sheet .....	248
	PLLS Output References .....	248
	PLLS Configuration Considerations .....	249
	Permissive Sheet .....	249
	PPERM Input References .....	249
	PPERM Output References .....	249
	ACM Considerations for PLLS .....	249
	ACM_Based Parameters for a PLLS Instance .....	250
	Additional Sub-Objects for a PLLS Instance .....	250
	RefDevice Mapping .....	251



<b>Process Motor (PMTR) Control Strategies</b>	<b>Chapter 24</b>	
	CS_PMTR Sheet .....	255
	PMTR Input References .....	255
	PMTR Output References .....	256
	PMTR Configuration Considerations .....	256
	Permissive Sheet .....	257
	PPERM Input References .....	257
	PPERM Output References .....	257
	Motor Controller Device Objects .....	258
	Single Speed .....	258
	Single Speed with Energy Parameters .....	258
	Two Speed .....	259
	Two Speed with Energy Parameters .....	259
	Reversing .....	260
	Reversing with Energy Parameters .....	260
	ACM Considerations for a PMTR Instance .....	261
	ACM-Based Parameters for a PMTR Instance .....	262
	Additional Sub-Objects for a PMTR Instance .....	263
<b>Process n-Position (PNPOS) Control Strategy</b>	<b>Chapter 25</b>	
	CS_PNPOS Sheet .....	266
	PNPOS Input References .....	266
	PNPOS Output References .....	267
	PNPOS Configuration Considerations .....	267
	Permissive Sheet .....	267
	PPERM Input References .....	267
	PPERM Output References .....	268
	IO Faults Sheet .....	269
	Fault Input References .....	270
	Fault Output References .....	270
	ACM Considerations for a PNPOS Instance .....	270
	ACM-Based Parameters for a PNPOS Instance .....	271
	Additional Sub-Objects for a PNPOS Instance .....	271
<b>Process Proportional + Integral + Derivative (PPID) Basic Control Strategies</b>	<b>Chapter 26</b>	
	CS_PPID Sheet .....	274
	PAI Input References .....	274
	PAI Outputs to PPID Inputs .....	274
	PPID Input References .....	275
	PPID Output References .....	275
	PPID Configuration Considerations .....	275
	CS_PPID HART Sheet .....	276
	CS_PPID_EtherNetIP Sheet .....	277
	CS_PPID_EtherNetIP_NoHB Sheet .....	278
	CS_PPID_FF1 Sheet .....	279
	CS_PPID_PA Sheet .....	279
	IO Faults Sheet .....	280
	Fault Input References .....	280

Fault Output Reference .....	280
ACM Considerations for PPID.....	280
ACM-Based Parameters for a PPID Instance.....	280
Additional Sub-Objects for a PPID Instance.....	282

## Chapter 27

### Process Proportional + Integral + Derivative (PPID) with Cascade Control Strategies

PPID with Cascade Control Example.....	286
ROUTINE Inner Loop .....	288
CS_PPID_CASC - Inner Loop Sheet (XIC760) .....	288
CS_PPID_CASC - Inner Loop HART Sheet (XIC560).....	290
CS_PPID_CASC - Inner Loop EtherNet/IP Sheet (XIC660).....	291
CS_PPID_CASC - Inner Loop EtherNet/IP NoHB Sheet (XIC661) .....	292
CS_PPID_CASC - Inner Loop FF Sheet (XIC860) .....	293
CS_PPID_CASC - Inner Loop PA Sheet (XIC960) .....	294
Interlock Bank Inner Loop Sheet.....	295
IO Faults Inner Loop Sheet.....	295
ROUTINE Outer Loop.....	296
CS_PPID_CASC-Outer Loop Sheet (XIC770).....	296
CS_PPID_CASC - Outer Loop HART Sheet (XIC570).....	298
CS_PPID_CASC - Outer Loop EtherNet/IP Sheet (XIC670).....	299
CS_PPID_CASC - Outer Loop EtherNet/IP NoHB Sheet (XIC671) .....	300
CS_PPID_CASC - Outer Loop FF Sheet (XIC870).....	301
CS_PPID_CASC - Outer Loop PA Sheet (XIC970).....	302
IO Faults Outer Loop Sheet.....	302
ACM Considerations for PPID with Cascade Control.....	303
ACM-Based Parameters for a PPID Instance with Cascade Control .....	304
Additional Sub-Objects for a PPID with Cascade Control Instance.....	306

## Chapter 28

### Process Proportional + Integral + Derivative (PPID) Feedforward Control Strategies

PPID Feedforward Example.....	307
CS_PPID_FF Sheet.....	309
PAI Input References .....	309
PAI Outputs to PPID Inputs.....	310
PPID Input References .....	310
PPID Output References.....	310
PPID Configuration Considerations.....	311
CS_PPID_FF HART Sheet.....	311
CS_PPID_FF_EtherNetIP Sheet .....	312
CS_PPID_FF_EtherNetIP_NoHB Sheet.....	313
CS_PPID_FF_FF .....	314
CS_PPID_FF_PA .....	315
IO Faults Sheet .....	315
Fault Input References.....	315
Fault Output Reference .....	315
ACM Considerations for PPID with Feedforward Control .....	316
ACM-Based Parameters for a PPID Instance with Feedforward Control .....	317
Additional Sub-Objects for a PPID with Feedforward Control Instance .....	318

## Process Proportional + Integral + Derivative (PPID) Override Control Strategies

### Chapter 29

PPID Override Example.....	321
ROUTINE Override Low Select .....	324
CS_PHLS Sheet .....	324
ROUTINE Process Analog Output.....	325
CS_PAO Sheet.....	325
ROUTINE: Process Analog Input to Process PID .....	326
CS_PPID Sheet.....	326
CS_PPID HART Sheet.....	328
CS_PPID_EtherNetIP Sheet.....	329
CS_PPID_EtherNetIP_NoHB Sheet.....	329
CS_PPID_FF Sheet.....	330
CS_PPID_PA Sheet.....	331
IO Faults Sheet.....	331
ACM Considerations for PPID with Override Control.....	331
ACM-Based Parameters for a PPID Instance with Override Control .....	333
Additional Sub-Objects for a PPID with Override Control Instance.....	335

### Chapter 30

## Process Proportional + Integral + Derivative (PPID) Basic PPID with Process Analog Output (PAO) Control Strategies

ROUTINE PPID_PAO .....	339
CS_PPID_PAO Sheet (XC730).....	339
CS_PPID_PAO HART Sheet (XC530) .....	340
CS_PPID_PAO EtherNet/IP Sheet (XC630).....	341
CS_PPID_PAO EtherNet/IP NoHB Sheet (XC631).....	341
CS_PPID_PAO FF Sheet (XC830).....	342
CS_PPID_PAO PA Sheet (XC930).....	342
IO Faults Sheet.....	343
ROUTINE PPID_PAO_PID .....	344
CS_PPID_PAO_PID Sheet (XIC730) .....	344
CS_PPID_PAO_PID HART Sheet (XIC530).....	346
CS_PPID_PAO_PID EtherNet/IP Sheet (XIC630) .....	347
CS_PPID_PAO_PID EtherNet/IP NoHB (XIC631) .....	348
CS_PPID_PAO_PID FF Sheet (XIC830) .....	349
CS_PPID_PAO_PID Profibus PA Sheet (XIC930) .....	350
IO Faults Sheet.....	351
ACM Considerations for PPID with PAO Control.....	351
ACM-Based Parameters for a PPID Instance with PAO Control .....	352
Additional Sub-Objects for a PPID with PAO Instance.....	354

### Chapter 31

## Process Proportional + Integral + Derivative (PPID) Basic PPID with Process Variable Speed Drive (PVSD) Control Strategies

ROUTINE PPID_PVSD.....	356
CS_PPID_PVSD_PID Controller Sheet (XIC710).....	357
PAI Input References .....	357
PAI Outputs to PPID Inputs.....	358
PPID Input References .....	358
PPID Output Reference.....	358
PPID Configuration Considerations.....	358
CS_PPID_PVSD_HART_PID Controller Sheet (XIC520) .....	359

CS_PPID_EH_Promass100_PVSD_PID_Controller Sheet .....	360
CS_PPID_FFLink_PVSD_PID_Controller Sheet .....	360
CS_PPID_PALink_PVSD_PID_Controller Sheet .....	361
IO Faults Sheet .....	362
Fault Input References .....	362
Fault Output Reference .....	362
ACM Considerations for PPID with PVSD .....	362
Additional Sub-Objects for a PPID with PVSD Instance .....	365

## Chapter 32

### Process Proportional + Integral + Derivative (PPID) Ratio Control Strategies

PPID Ratio Example .....	368
CS_PPID_RATIO Sheet .....	369
PAI Input References .....	369
PAI Outputs to PPID Inputs .....	369
PPID Input References .....	370
PPID Output References .....	370
PPID Configuration Considerations .....	370
CS_PPID_RATIO HART Sheet .....	371
CS_PPID_RATIO_EtherNetIP Sheet .....	372
CS_PPID_RATIO_EtherNetIP_NoHB Sheet .....	373
CS_PPID_RATIO_FF .....	374
CS_PPID_RATIO_PA .....	375
IO Faults Sheet .....	376
Fault Input References .....	376
Fault Output Reference .....	376
ACM Considerations for PPID with Ratio Control .....	376
ACM-Based Parameters for a PPID Instance with Ratio Control .....	377
Additional Sub-Objects for a PPID with Ratio Control Instance .....	378

## Chapter 33

### Process Proportional + Integral + Derivative (PPID) Split Range Control Strategies

PPID Split Range Example .....	381
ROUTINE: Process Analog Output .....	383
CS_PAO Sheet .....	383
ROUTINE: Process Analog Input to Process PID with Fanout .....	384
CS_PPID Sheet .....	384
CS_PPID HART Sheet .....	386
CS_PPID_EtherNetIP Sheet .....	387
CS_PPID_FF Sheet .....	388
CS_PPID_PA Sheet .....	389
PFO Sheet .....	390
IO Faults Sheet .....	391
ACM Considerations for PPID with Split Range Control .....	391
ACM-Based Parameters for a PPID Instance with Split Range Control .....	392
Additional Sub-Objects for a PPID with Split Range Control Instance .....	394

<b>Chapter 34</b>	
<b>Process Pressure/Temperature Compensated Flow (PPTC) Control Strategies</b>	CS_PPTC Sheet ..... 399 PAI Input References ..... 400 PAI Outputs ..... 400 PPTC Outputs ..... 400 PPTC Configuration Considerations ..... 400 CS_PPTC_HART Sheet ..... 401 CS_PPTC_EtherNetIP Sheet ..... 403 CS_PPTC_FF Sheet ..... 405 CS_PPTC_PA Sheet ..... 406 ACM Considerations for PPTC ..... 407 ACM-Based Parameters for a PPTC Instance ..... 407 Additional Sub-Object for a PPTC Instance ..... 412
<b>Chapter 35</b>	
<b>Process Tank Strapping Table (PTST) Control Strategies</b>	CS_PTST Sheet ..... 415 PAI Input References ..... 415 PAI Outputs ..... 415 PTST Configuration Considerations ..... 415 CS_PTST_HART Sheet ..... 416 CS_PTST_FF Sheet ..... 417 CS_PTST_PA Sheet ..... 417 ACM Considerations for PTST ..... 417 ACM-Based Parameters for a PTST Instance ..... 418 Additional Sub-Object for a PTST Instance ..... 419
<b>Chapter 36</b>	
<b>Process Valve Hand Operated (PVLVHO) Control Strategy</b>	CS_PVLVHO Sheet ..... 422 PVLV Input References ..... 422 PVLV Output References ..... 423 PVLV Configuration Considerations ..... 423 PVLV Output References to PVLVS ..... 423 IO Faults Sheet ..... 423 Fault Input References ..... 423 Fault Output References ..... 424 ACM Considerations for a PVLV Instance ..... 425 ACM-Based Parameters for a PVLV Instance ..... 425 Additional Sub-Objects for a PVLV Instance ..... 427
<b>Chapter 37</b>	
<b>Process Valve Motor Operated (PVLVMO) Control Strategy</b>	CS_PVLVMO Sheet ..... 430 PVLV Input References ..... 430 PVLV Output References ..... 431 PVLV Configuration Considerations ..... 431 PVLV Output References to PVLVS ..... 431 Permissive Sheet ..... 431 PPERM Input References ..... 431 PPERM Output References ..... 432



	IO Faults Sheet .....	432
	Faults Input References .....	432
	Fault Output References .....	433
	ACM Considerations for a PVLV Instance .....	433
	ACM-Based Parameters for a PVLV Instance .....	433
	Additional Sub-Objects for a PVLV Instance .....	435
	<b>Chapter 38</b>	
<b>Process Mix Proof Valve (PVLVMP) Control Strategy</b>	CS_PVLVMP Sheet .....	438
	PVLVMP Input References .....	438
	PVLVMP Output References .....	439
	PVLVMP Configuration Considerations .....	439
	I/O Faults Sheet .....	440
	Fault Input References .....	441
	Fault Output References .....	441
	ACM Considerations for a PVLV Instance .....	442
	ACM-Based Parameters for a PVLV Instance .....	442
	Additional Sub-Objects for a PVLV Instance .....	444
	<b>Chapter 39</b>	
<b>Process Valve Solenoid Operated (PVLVSO) Control Strategy</b>	CS_PVLVSO_FailPos1 Sheet .....	448
	Position 2 Permissive Sheet .....	449
	CS_PVLVSO_FailPos2 Sheet .....	450
	Position 1 Permissive Sheet .....	451
	IO Faults Sheet-XV101 .....	452
	Fault Input References .....	452
	Fault Output References .....	453
	IO Faults Sheet-XV102 .....	453
	Fault Input References .....	453
	Fault Output References .....	453
	ACM Considerations for a PVLV Instance .....	455
	ACM-Based Parameters for a PVLV Instance .....	455
	Additional Sub-Objects for a PVLV Instance .....	457
	<b>Chapter 40</b>	
<b>Process Variable Speed Drive (PVSD) Control Strategies</b>	CS_PVSD Sheet .....	461
	PVSD Input References .....	461
	PVSD Output References .....	462
	PVSD Configuration Considerations .....	462
	Permissive Sheet .....	463
	PPERM Input References .....	463
	PPERM Output References .....	463
	Drive Device Objects .....	464
	Drive .....	464
	Drive with Energy Parameters .....	464
	ACM Considerations for a PVSD Instance .....	464
	ACM-Based Parameters for a PVSD Instance .....	465
	Additional Sub-Objects for a PVSD Instance .....	467

## Ramp/Soak (RMPS) Control Strategies

### Chapter 41

CS_RMPS Sheet .....	471
PAI Input References .....	471
PAI Outputs to RMPS Inputs .....	471
RMPS Output Reference .....	471
RMPS Configuration Considerations .....	471
CS_RMPS_HART Sheet .....	472
CS_RMPS_EtherNetIP Sheet .....	473
CS_RMPS_FF Sheet .....	473
CS_RMPS_PA Sheet .....	474
ACM Considerations for RMPS .....	474
ACM-Based Parameters for a RMPS Instance .....	475
Additional Sub-Objects for a RMPS Control Instance .....	476

## Organizational Control Strategies

### Chapter 42

Area Control Strategy .....	477
CS_raP_Opr_Area Sheet .....	477
CS_raP_Opr_Area Input Reference .....	478
CS_raP_Opr_Area Configuration Considerations .....	478
ACM Considerations for an Area Instance .....	478
ACM-Based Parameters for an Area Instance .....	479
Additional Sub-Objects for an Area Instance .....	479
Unit Control Strategy .....	479
CS_raP_Opr_Unit Sheet .....	480
CS_raP_Opr_Unit Input Reference .....	481
CS_raP_Opr_Unit Output Reference .....	481
CS_raP_Opr_Unit Configuration Considerations .....	481
ACM Considerations .....	481
Additional Sub-Objects for a Unit Instance .....	483
Generic Equipment Module (EMGEN) Control Strategy .....	483
Dispatch Routine .....	484
Dispatch EPGEN100 Sheet .....	485
Dispatch Permissive Sheet .....	486
State Routine Example .....	487
StateCall Routine Example .....	487
StateModel Routine Example .....	487
StateReady Map Routine Example .....	488
ACM Considerations for an EM Instance .....	488
ACM-Based Parameters for an EM Instance .....	488
Additional Sub-Objects for an EM Instance .....	490
Generic Equipment Phase (EPGEN) Control Strategy .....	490
Dispatch Routine .....	491
Dispatch EPGEN100 Sheet .....	492
Dispatch Permissive Sheet .....	493
PHASECommands Routine .....	493
ACM Considerations for an EP Instance .....	497
ACM-Based Parameters for an EP Instance .....	497
Additional Sub-Objects for an EP Instance .....	498
Extended Alarms Routine .....	498

---

raP_Opr_ExtddAlm Parameters .....	499
Parameters and Reports Routines .....	500
raP_Tec_ParRpt References .....	501
raP_Tec_ParRpt Configuration Considerations.....	501
CS_raP_Opr_EMGen or CS_raP_Opr_EPGen Parameters Routine .....	502



About This Publication

The Rockwell Automation® Process Objects Library includes PlantPAx® control strategies to help reuse logic to save development time. Use this reference manual for guidance on when and how to use each control strategy.

Download Firmware, AOP, EDS, and Other Files

Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at [rok.auto/pcdc](http://rok.auto/pcdc).

Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

The updates to this version of the publication apply to the 5.20 release of the PlantPAx process library. Screenshots are examples and might show previous versions, even though they apply to this release.

Topic	Page
Added new ACM Considerations chapter.	<a href="#">35</a>
Added ACM considerations section to every control strategy.	Throughout
Added PPID with VSD Chapter.	<a href="#">355</a>
Added parameter Connectiontype to HART Integration chapter.	<a href="#">67</a>
Added CS_raP_Opr_Unit and combined CS_raP_Opr_Area, CS_raP_Opr_EMGen, and CS_raP_Opr_EPGen into one chapter: Organizational Control Strategies.	<a href="#">477</a>

**Note:** The Totalizer (TOT) Control Strategy was removed from the 5.20 release of the library as it is an embedded process controller instruction. For more information, see the Advanced Process Control and Drives Instruction Manual, publication [1756-RM006](#) or the online help.



## Additional Resources

These documents contain additional information concerning related products from Rockwell Automation. You can view or download publications at [rok.auto/literature](http://rok.auto/literature).

Resource	Description
PlantPAx Distributed Control System Configuration and Implementation User Manual, publication <a href="#">PROCES-UM100</a>	Provides system guidelines and instructions to assist with the development of your PlantPAx system.
Rockwell Automation Library of Process Objects: HART Modules for PlantPAx DCS, publication <a href="#">PROCES-RM010</a>	Provides details on the integration of HART devices into a PlantPAx system or Integrated Architecture®
Rockwell Automation Library of Process Objects, publication <a href="#">PROCES-RM200</a>	Describes the Add-On Instructions, PlantPAx instructions, and associated faceplates that are available to develop applications.
Rockwell Automation Sequencer Object User Manual, publication <a href="#">PROCES-RM202</a>	Provides an overview of how to use the Rockwell Automation® Sequencer Object (raP_Opr_Seq).
Power Device Library Reference Manual, publication <a href="#">DEVICE-RM100</a>	Provides information on objects for discrete, velocity, motion, and PowerMonitor™ devices.
I/O Device Library Reference Manual, publication <a href="#">DEVICE-RM200</a>	Provides information on objects for Rockwell Automation 1756, 1769, 1734, 1794, 1738, 1732E, 1719, 5069, 5094 I/O modules, including pre-configured status and diagnostic faceplates.
Advanced Process Control and Drives and Phase and Sequence Instruction Manual, publication <a href="#">1756-RM006</a>	Provides details about the available General, Motion, Process, and Drives instruction set for a Logix-based controller.

## PlantPax Control Strategies

The PlantPax® control strategies are routines or programs that you import into your controller project. The PlantPax control strategies are Function Block Diagrams or Ladder Diagrams that include preconfigured process instructions that represent common control and equipment scenarios in process automation. The PlantPax control strategies have several preconfigured arrays and tags.

See the instruction online help for complete details on the instructions in the control strategies.

### Library Prerequisites

Download the latest versions of these libraries at the [Product Compatibility and Download Center \(PCDC\)](#).

- Power Device Library
- I/O Device Library

### How to use PlantPax Control Strategies

You can import the PlantPax control strategies into your project with Studio 5000 Logix Designer®, or with Application Code Manager (ACM) plug-ins within Studio 5000 Logix Designer.

Import Method	Considerations
Import using Studio 5000 Logix Designer	<ul style="list-style-type: none"> <li>• You can easily modify a source import file for each application:               <ol style="list-style-type: none"> <li>a. Import the standard routine.</li> <li>b. Modify the routine.</li> <li>c. Export the modified routine to a renamed control strategy for your application.</li> </ol> </li> <li>• You must import individual routines one at a time (even when a single control strategy is comprised of multiple routines).</li> <li>• You <b>can</b> add routines while you are online with the controller. For more information, see <a href="#">Import with Studio 5000 Logix Designer on page 22</a>.</li> </ul>
Import using ACM plug-ins in Studio 5000 Logix Designer	<ul style="list-style-type: none"> <li>• ACM process library includes a comprehensive set of PlantPax control strategies plug-ins.</li> <li>• You can enter multiple control strategies at once (even when there are multiple routines per control strategy)</li> <li>• You can configure faceplate navigation at import.</li> <li>• It is difficult to modify source routines.</li> <li>• You <b>cannot</b> use the plug-in feature while Online with the controller. For more information, see <a href="#">Import with Application Code Manager Plug-ins on page 24</a>.</li> </ul>

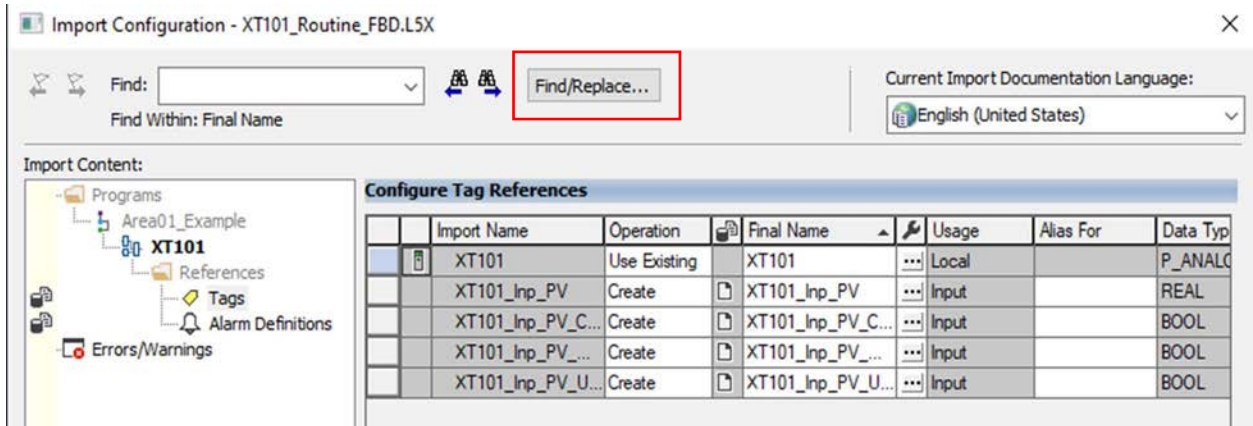
## Import with Studio 5000 Logix Designer

The PlantPAx control strategies are provided as folders that contain one or more routines that can be imported into an appropriate program.

For information on how to import routines and programs, see the Logix 5000® Controllers Import/Export Project Components Programming Manual, publication [1756-PM019](#).

When the Import Configuration window opens:

1. Select the Tags folder.  
All tags in the control strategy have a default prefix, such as XT101.
2. Use the Find/Replace button to rename the prefix to match your site's tag naming convention.



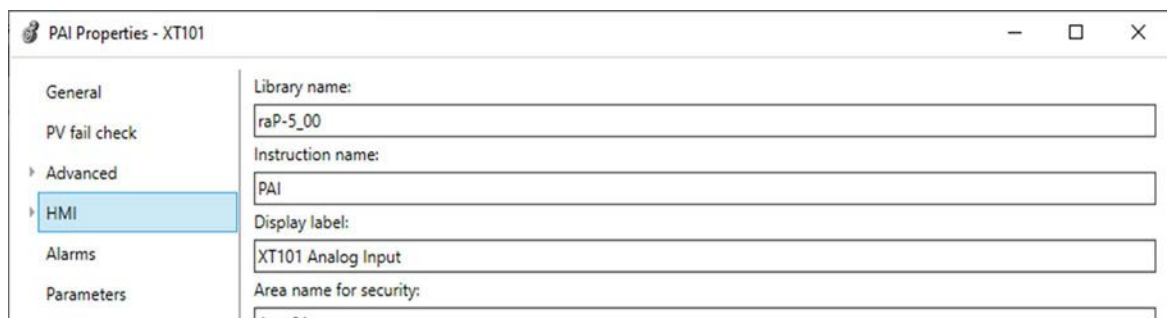
### Configure the HMI Display Label

To configure the appropriate display label in the HMI complete these steps.

1. Select the Properties dialog box for the instruction in the control strategy.

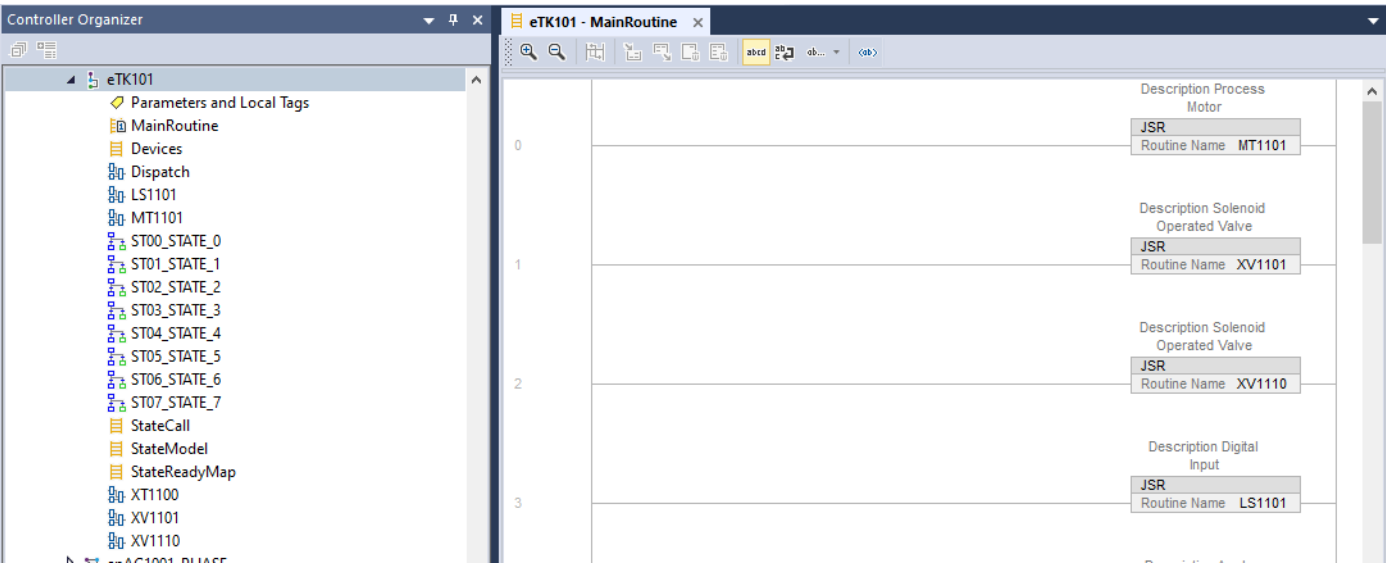


2. On the HMI page, edit the default display label to provide an appropriate label for the operator interface components.
  - As a best practice, use a consistent labeling method throughout all projects in the system. You could use the exact tag name, or use a more readable format.
  - If you do not use the exact tag name, the display label should generally align with ANSI/ISA-5.1-2022 naming standards.



Add Main Routine Code to Execute the Imported Control Strategies

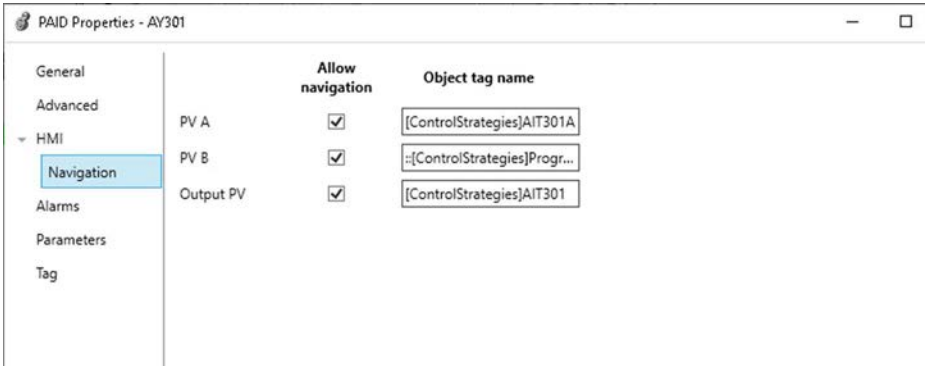
Add JSR instructions that reference the imported control strategy to the Main Routine to execute the new control strategy routines.



HMI Navigation

The process instructions in the PlantPax control strategies support HMI navigation to other instructions in the same control strategy. To leverage this capability, you only need to specify the appropriate controller-scoped or program-scoped tag.

On the process instruction, select Properties > HMI> Navigation and enter the tags for the control strategy objects that you want to allow navigation to.



This example shows both controller-scoped and program-scoped tags, but you can use either for each option. Use this syntax for each tag type.

Tag Type	Syntax	Example
Controller scope	[ TOPIC ]TagName	[ ControlStrategies ]AIT301A
Program scope	::[ TOPIC ]Program:ProgramName.TagName	::[ ControlStrategies ]Program:CS_PAID.AIT301B

## Import with Application Code Manager Plug-ins

The Application Code Manager (ACM) process library includes a comprehensive set of PlantPax control strategies plug-ins for you to use in your controller projects. Follow your project plan (the spreadsheet with your devices and tags) as you add PlantPax control strategies for devices (motors, valves, drives, and so forth) to the Studio 5000 Logix Designer application project file.

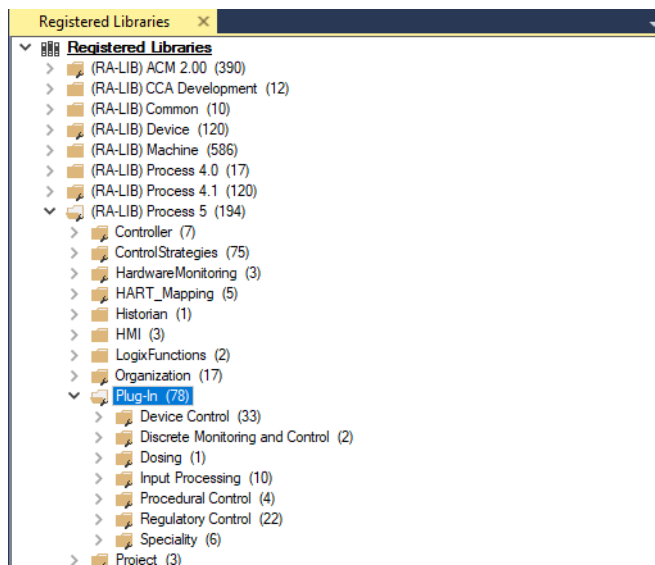
For more information, see the Application Code Manager User Manual, publication [LOGIX-UM003](#).

- 
- IMPORTANT**
- You can use ACM and ACM plug-ins to add PlantPax control strategies only when you are **offline** with the controller.
  - The Library Object Import Wizard can import one or more control strategies at a time.
  - When adding multiple PlantPax control strategies of the same type, rename each instance to a unique name.
- 

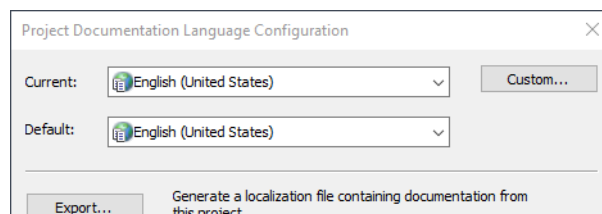
### Prerequisites

Before you can use plug-ins with the Import Library Objects in Studio 5000 Logix Designer, you must do the following:

1. Verify that the Application Code Manager is installed on the workstation that has Studio 5000 Logix Designer.
2. Verify that the Application Code Manager Process Library is registered in ACM.



3. In the Studio 5000 Logix Designer application, go to Tools > Documentation Languages and verify that the Project Documentation Language Configuration Default is set to English (United States).

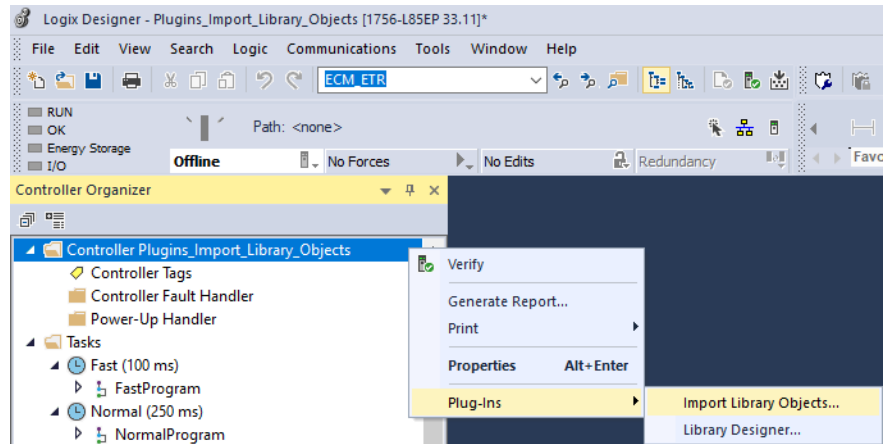




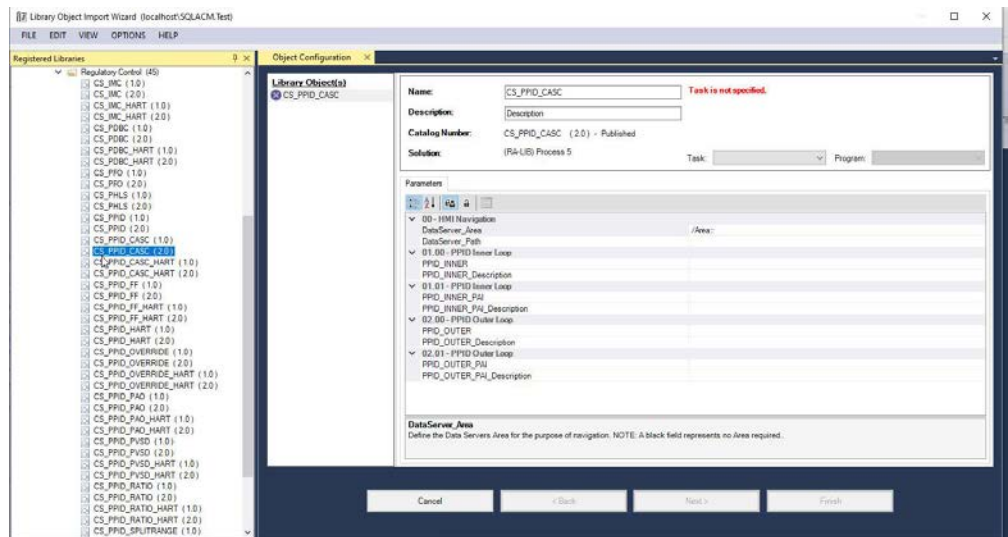
### Import Library Objects (Offline Only)

This example workflow shows how to use the Library Object Import Wizard to add two PlantPax control strategies (CS\_PPID\_CASC and CS\_PVLVSO) into the Logix Designer Project (ACD) file.

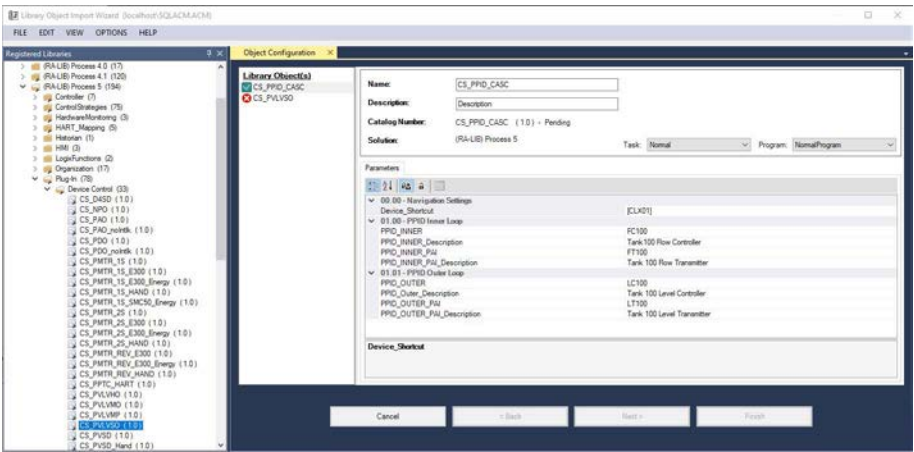
1. Open a Studio 5000 Logix Designer Project (.ACD) file.
2. Right-click on the Controller, and navigate to Plug-Ins > Import Library Objects... to launch the Library Object Import Wizard.



3. In the Library Object Import Wizard, navigate to (RA-LIB) Process 5 > Plug-In > Regulatory Control.
4. Double-click CS\_PPID\_CASC to add it to the Library Objects.



5. Under Library Objects, click CS\_PPID\_CASC and configure the Task and Program.

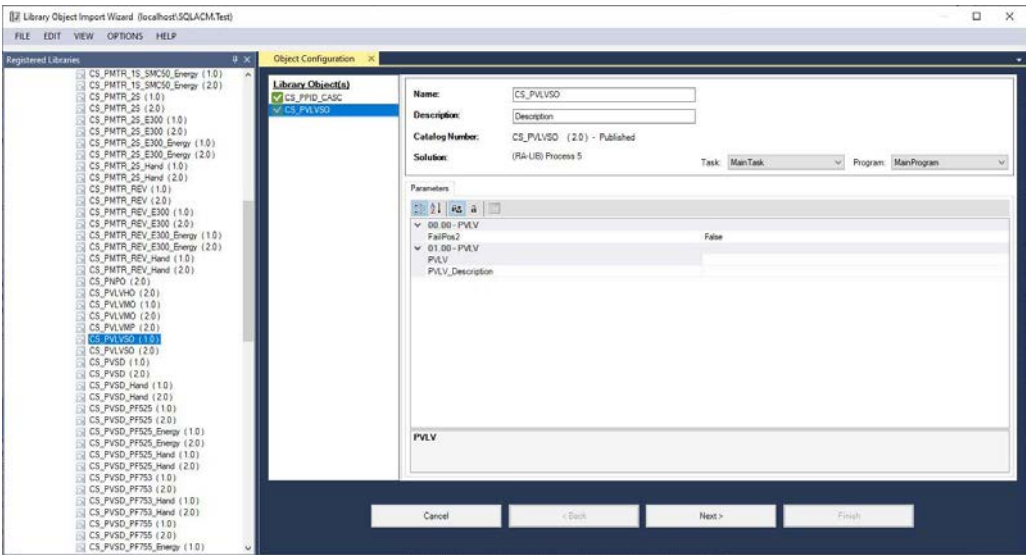


6. Configure these required parameters for the CS\_PPID\_CASC control strategies.

ACM Parameter	Description	Example
Device_Shortcut	Example: [TOPIC] or /Area/DATA:.[TOPIC]	[CLX01]
PPID_INNER	Inner Loop Controller Tag	FC100
PPID_INNER_Description	Inner Loop Controller Tag Description	Tank 100 Flow Controller
PPID_INNER_PA1	Inner Loop Analog Input Tag	FT100
PPID_INNER_PA1_Description	Inner Loop Analog Input Tag Description	Tank 100 Flow Transmitter
PPID_OUTER	Outer Loop Controller Tag	LC100
PPID_OUTER_Description	Outer Loop Controller Tag Description	Tank 100 Level Controller
PPID_OUTER_PA1	Outer Loop Analog Input Tag	LT100
PPID_OUTER_PA1_Description	Outer Loop Analog Input Tag Description	Tank 100 Level Transmitter

7. In the Library Object Import Wizard, navigate to (RA-LIB) Process 5 > Plug-In > Device Control.

8. Double-click CS\_PVLVSO to add it to the Library Objects.



9. Under Library Objects, click CS\_PVLVSO and configure the Task and Program.

The Object Configuration dialog for CS\_PVLVSO shows the following settings:

- Name:** CS\_PVLVSO
- Description:** Description
- Catalog Number:** CS\_PVLVSO (1.0) - Pending
- Solution:** (RA-LIB) Process 5
- Task:** Normal
- Program:** NormalProgram

**Parameters:**

Parameter	Value
01.00 - PVID	
PVLVSO	XV100
PVLVSO_Description	Tank 100 Feed Valve

**PVLVSO\_Description**

Buttons: Cancel, < Back, Next >, Finish

11. Configure these required parameters for the CS\_PVLVSO control strategies.

ACM Parameter	Description	Example
PVLVSO	Valve Tag	XV100
PVLVSO_Description	Valve Tag Description	Tank 100 Feed Valve

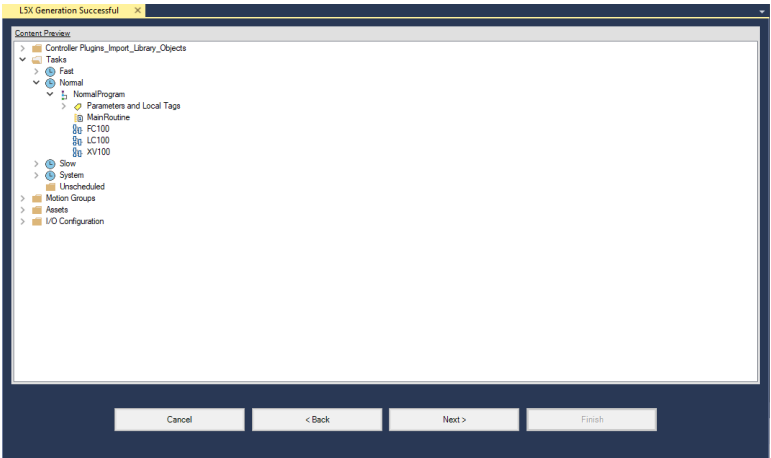
12. Click Next.
13. Review the Merge Actions window, and click Next.

The Merge Actions dialog displays a list of actions to be merged into the project. The columns are Category, Name, Action, Task Name, Program Name, and Info.

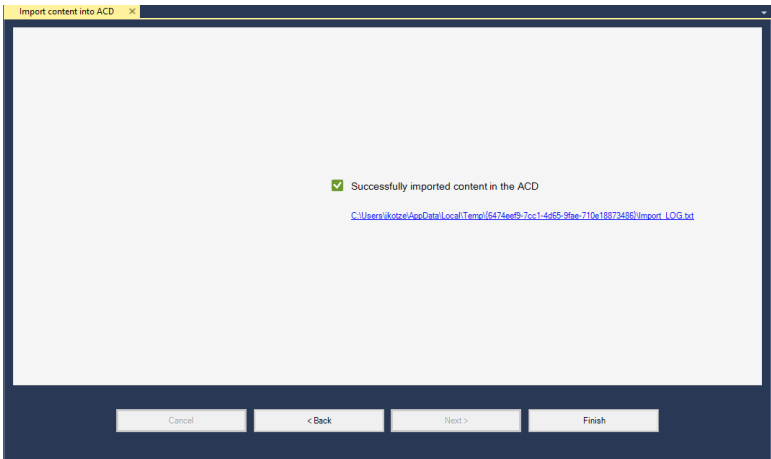
Category	Name	Action	Task Name	Program Name	Info
DatatypeAlarmDefinition	P_ANALOG_INPUT	Use Existing			
DatatypeNamDefinition	P_PID	Use Existing			
DatatypeNamDefinition	P_VALVE_DISCRETE	Use Existing			
FBD Sheet	FC100 (Sheet 1)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 10)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 2)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 3)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 4)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 5)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 6)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 7)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 8)	Add	Normal	NormalProgram	
FBD Sheet	FC100 (Sheet 9)	Add	Normal	NormalProgram	
FBD Sheet	LC100 (Sheet 1)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 1)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 10)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 11)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 2)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 3)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 4)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 5)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 6)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 7)	Add	Normal	NormalProgram	
FBD Sheet	XV100 (Sheet 8)	Add	Normal	NormalProgram	

Buttons: Cancel, < Back, Next >, Finish

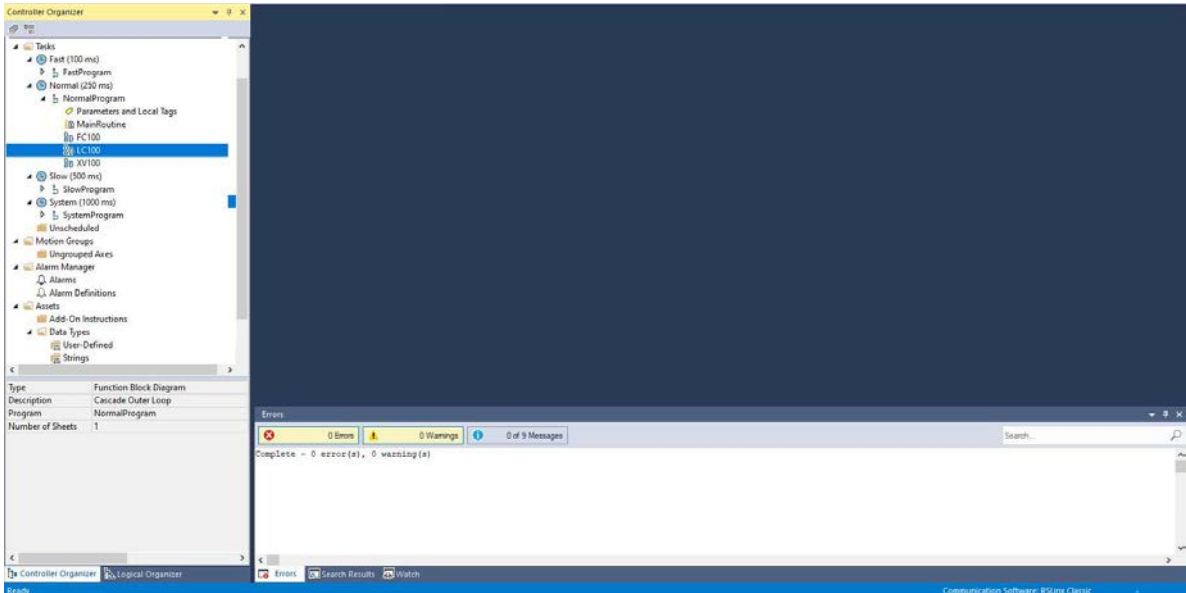
14. Review the L5X Generation Successful window (expand tasks and programs), and click Next.



15. On the Import content into ACD window, verify that the content was imported successfully.



16. Click Finish.
17. In the Studio 5000 Logix Designer project, use the Verify Controller feature to confirm that the control strategies were added to the Logix Designer Project without creating additional errors.



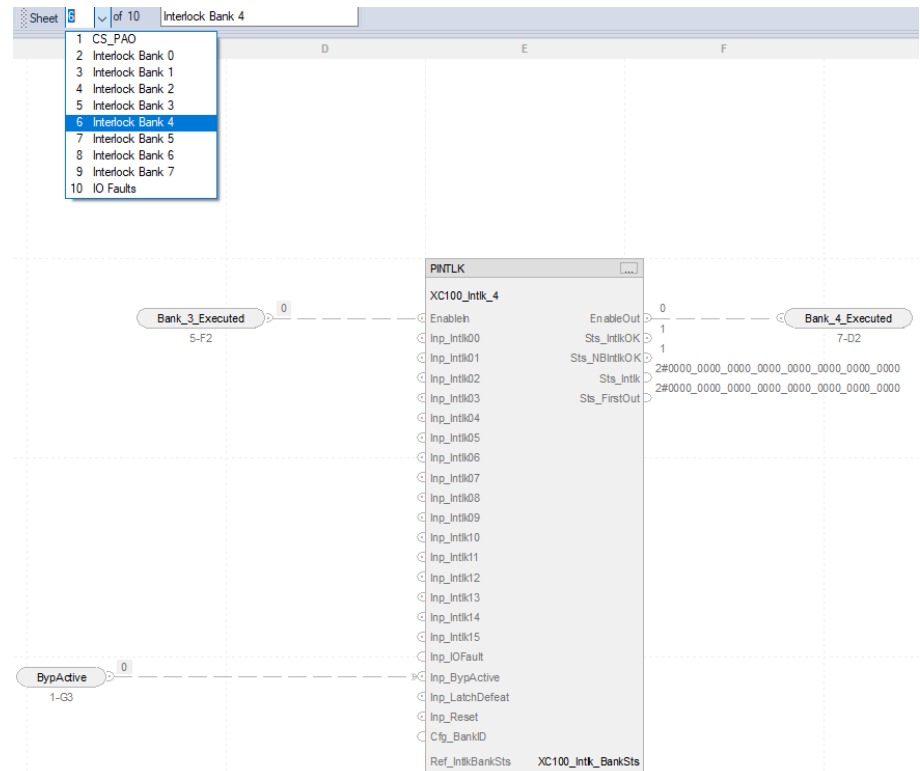
## Interlock Options

The Process Interlocks (PINTLK) instruction prevents equipment from starting or being energized. Interlocks are always evaluated to de-energize equipment. For permissive conditions that must be made true to start the equipment, but are ignored once the equipment is running, use the Process Permissive (PPERM) instruction.

In each PlantPAx control strategy that has interlocks, there are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default.

If your project runs into memory constraints, you can remove unused code, such as unused interlock banks. Remove the interlock banks in order of the last to the first bank.

If you edit or add interlock sheets, make sure the PINTLK Cfg\_BankID value matches the number of the interlock bank sheet. For example, Interlock Bank Sheet 4 has a PINTLK instruction where the Cfg\_BankID is also 4.



ACM creates the interlock banks sheets that you need based on your settings when you create your application.

Interlock Sheet



PINTLK Input Reference

Parameter	Description
BypActive	Input connection from CS_PPID_RATIO sheet

PINTLK Output Reference

Parameter	Description
Bank_0_Executed	Interlock sheet has been evaluated (0 = bank number)

PINTLK Configuration Considerations

Operand	Type	Description
PlantPAx control	P_INTERLOCK	Instance of data structure (backing tag) required for proper operation of instruction
Ref_IntlkBankSts	P_INTERLOCK_BANK_STATUS	Reference interlock bank status

For more information, see the online help for the PINTLK instruction.

I/O Connections

The PlantPAx control strategies have preconfigured program connections for the input and output values for the process instruction in the control strategies. These input and output values are program-scoped tags in the Parameters and Local Tags for the control strategy (not controller-scoped tags).

For example, in the PA0 control strategy, the output CV is a program connection to a channel on the module.

Scope: CS_PA0		Show: All Tags		Enter Name Filter				
Name	Value	Force Mask	Style	Data Type	Description	Constant	Connections	
XC100_Inp_ClosedLS_ChFit	0		Decimal	BOOL	TagDescript - Clos...	<input type="checkbox"/>		
XC100_Inp_ClosedLS_ModFit	1		Decimal	BOOL	1 = This or parent...	<input type="checkbox"/>		
XC100_Inp_OpenLS_ChFit	0		Decimal	BOOL	TagDescript - Ope...	<input type="checkbox"/>		
XC100_Inp_OpenLS_ModFit	1		Decimal	BOOL	1 = This or parent...	<input type="checkbox"/>		
XC100_Inp_PosFdbk	0.0		Float	REAL	TagDescript - Inp...	<input type="checkbox"/>		
XC100_Inp_PosFdbk_ChFit	0		Decimal	BOOL	TagDescript - Tie...	<input type="checkbox"/>		
XC100_Inp_PosFdbk_ModFault	0		Decimal	BOOL	1 = This or parent...	<input type="checkbox"/>		
XC100_Intlk_BankSts	(...)	(...)		P INTERLOCK BA...	TagDescript - Inte...	<input type="checkbox"/>		
XC100_Out_CV	0.0		Float	REAL	TagDescript - Con...	<input type="checkbox"/>	Local:5:0.Ch0Data	
XC100_PSet_CV	0.0		Float	REAL		<input type="checkbox"/>		

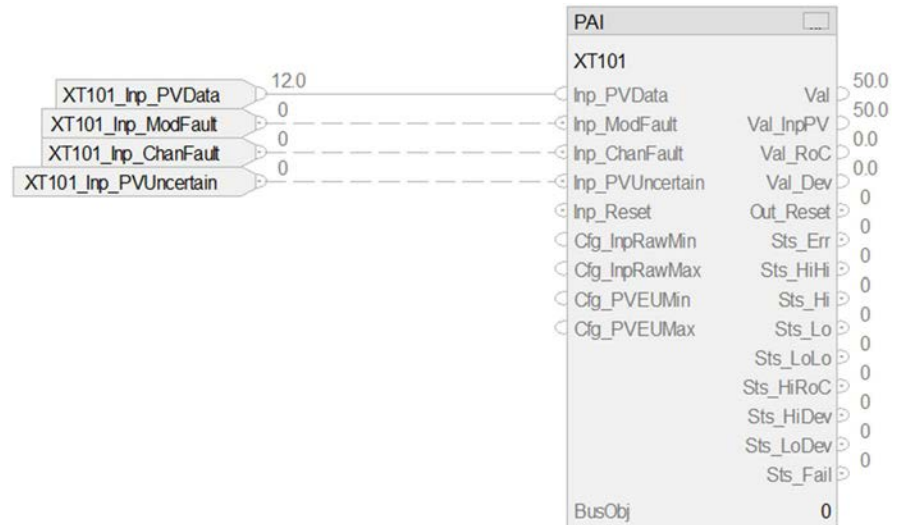
## Map Device Tags to Input Data

In each PlantPAx control strategy, inputs to the main instructions are preconfigured to map to similar locations for input modules.

For example, a PAI control strategy for this 1756-IF16 analog input module in slot 2:

- [2] 1756-IF16 Local\_02
- [3] 1756-L85EP RA\_LIB\_CS\_5\_00\_03
- [4] 1756-IF16 Local\_04
- [5] 1756-OF8 Local\_05
- [6] 1756-IB16 Local\_06
- [7] 1756-OB32 Local\_07

Has this logic:



And the inputs map as follows:

Input	Description
XT101_Inp_PVData	Process variable input (program-scoped tag) Source: sensor or input Program connection to Local:2:I.Ch0Data
XT101_Inp_ModFault	Controller-scoped tag Local_02.Sts_IOFault output from raP_Dvc_LgxModuleSts block for Local_02
XT101_Inp_ChanFault	Controller-scoped tag Local:2:I.Ch0Fault directly from AB:1756-IF16.Float_No_Alm:I:0 module tag
XT101_Inp_PVUncertain	Controller-scoped tag Local_02.Sts_AnyChanUncertain output from raP_Dvc_LgxModuleSts block for Local_02

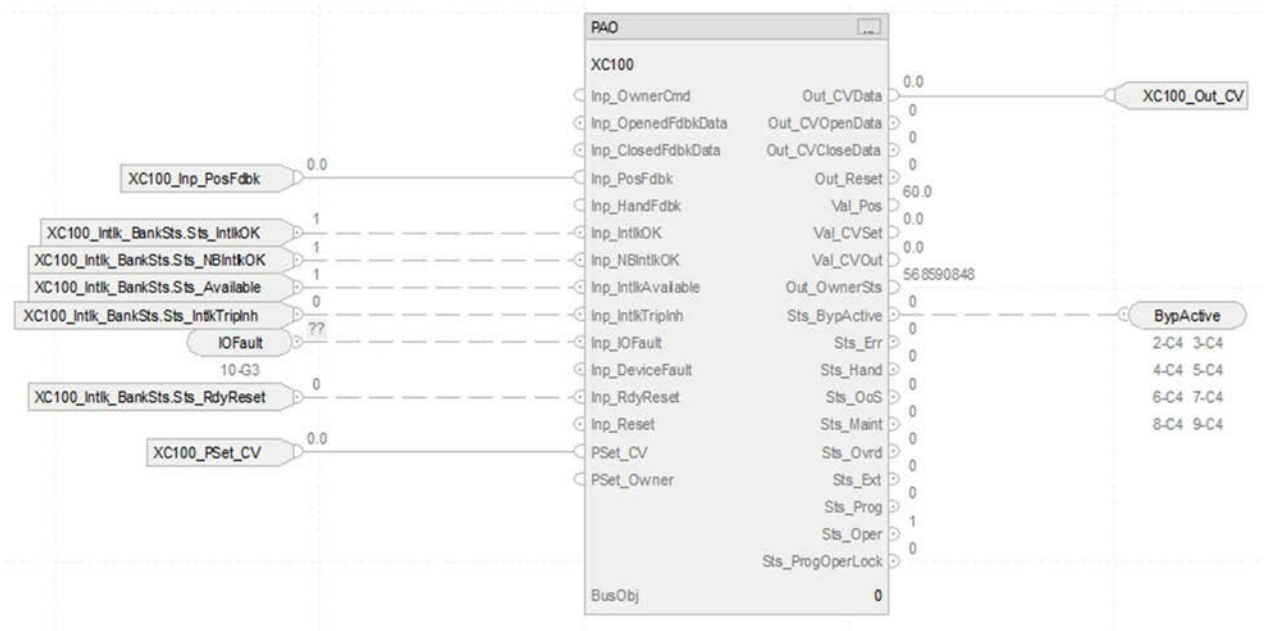
### Map Tags to Output Data

In each PlantPAx control strategy, the output from the main instructions is preconfigured to map to similar locations for output modules.

For example, a PAO control strategy for this 1756-OF8 analog output module:

- [2] 1756-IF16 Local\_02
- [3] 1756-L85EP RA\_LIB\_CS\_5\_00\_03
- [4] 1756-IF16 Local\_04
- [5] 1756-OF8 Local\_05
- [6] 1756-IB16 Local\_06
- [7] 1756-OB32 Local\_07

Has this logic:



And the output maps as follows:

Input	Description
XC100_Out_CV	Control variable output (program-scoped tag) Program connection to Local:5:0.Ch0Data



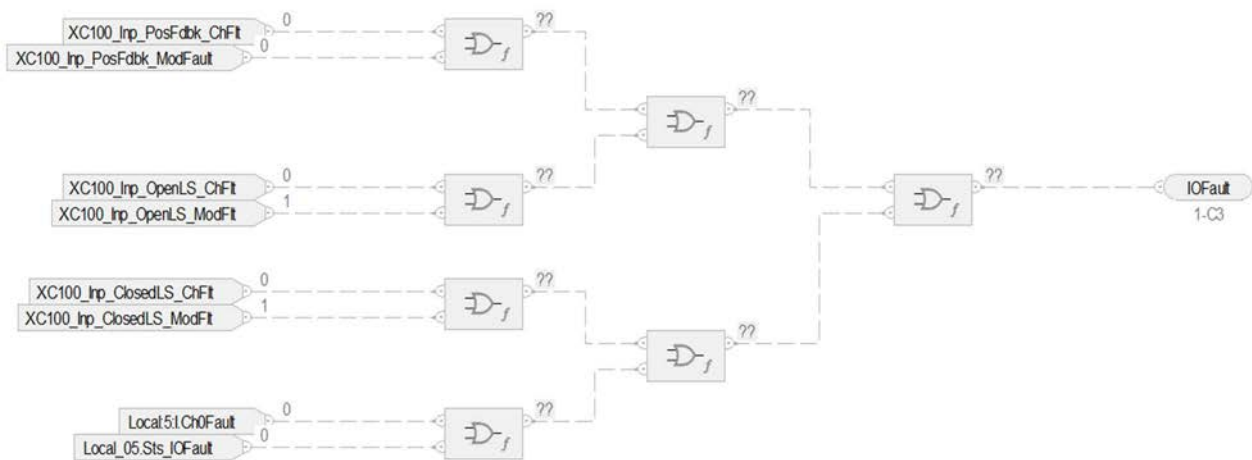
## Map I/O Faults

Fault data for output modules all wire to the IOFault reference on the associated I/O Fault sheet in the PlantPAx control strategy.

For example, a PA0 control strategy for this 1756-OF8 analog input module:

- [2] 1756-IF16 Local\_02
- [3] 1756-L85EP RA\_LIB\_CS\_5\_00\_03
- [4] 1756-IF16 Local\_04
- [5] 1756-OF8 Local\_05
- [6] 1756-IB16 Local\_06
- [7] 1756-OB32 Local\_07

Has this logic:



And has these fault tags:

Input	Description
XC100_Inp_PosFdbk_ChFault	Program-scoped tag
XC100_Inp_PosFdbk_ModFault	Program-scoped tag
XC100_Inp_OpenLS_ChFit	Program-scoped tag
XC100_Inp_OpenLS_ModFit	Program-scoped tag
XC100_Inp_ClosedLS_ChFit	Program-scoped tag
XC100_Inp_ClosedLS_ModFit	Program-scoped tag
Local5:I.Ch0Fault	Controller-scoped tag directly from Local:5:I.Ch0Fault within AB:1756_OF8_Float:I:0 module tag
Local_05.Sts.IOFault	Controller-scoped tag Local_05.Sts.IOFault From raP_Dvc_LgxModuleSts block for Local_05

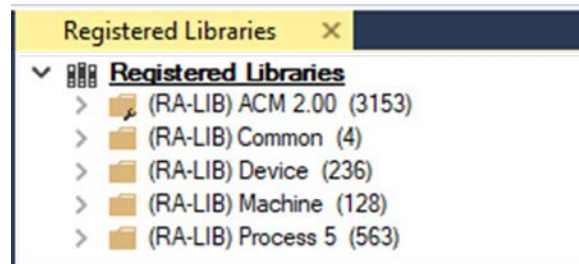
The program-scoped tags are preconfigured in the PlantPAx control strategy and must be mapped to the appropriate I/O points.

**Notes:**

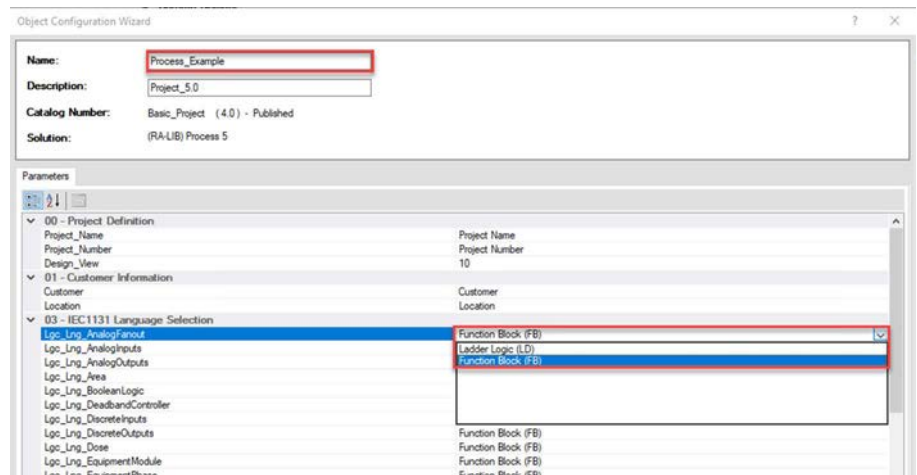
## ACM Considerations

ACM provides the ability to configure many parameters for the process controller, I/O processing, device control options, and individual control strategies. These considerations focus on those parameters that affect the final controller code, rather than configuration of the instances that you generate from ACM.

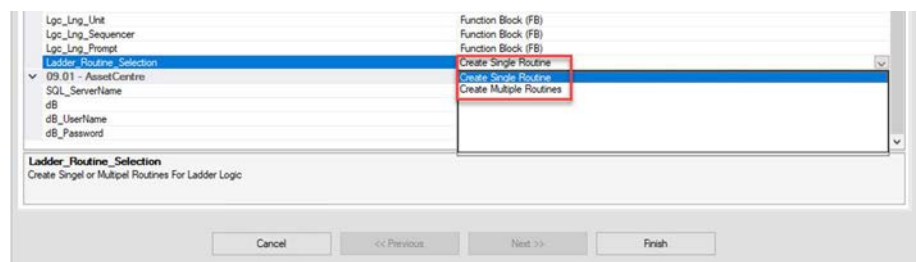
To use ACM to develop a process controller application, make sure that you have the required libraries registered in ACM.



When you create the project, the control strategies default to Function Block code.



If you select Ladder Logic, you also need to select whether the control modules will be instantiated in a single routine for each instruction type (create single routine) or for each instruction (Create Multiple Routines).



## Process Controller

You must select a process controller to use the PlantPax® process objects and corresponding control strategies.

Configure the parameters that affect instantiation of code. Changes to these parameters affect visibility for the parameters of all control strategies that are added to this controller or add code to the resulting ACD file.

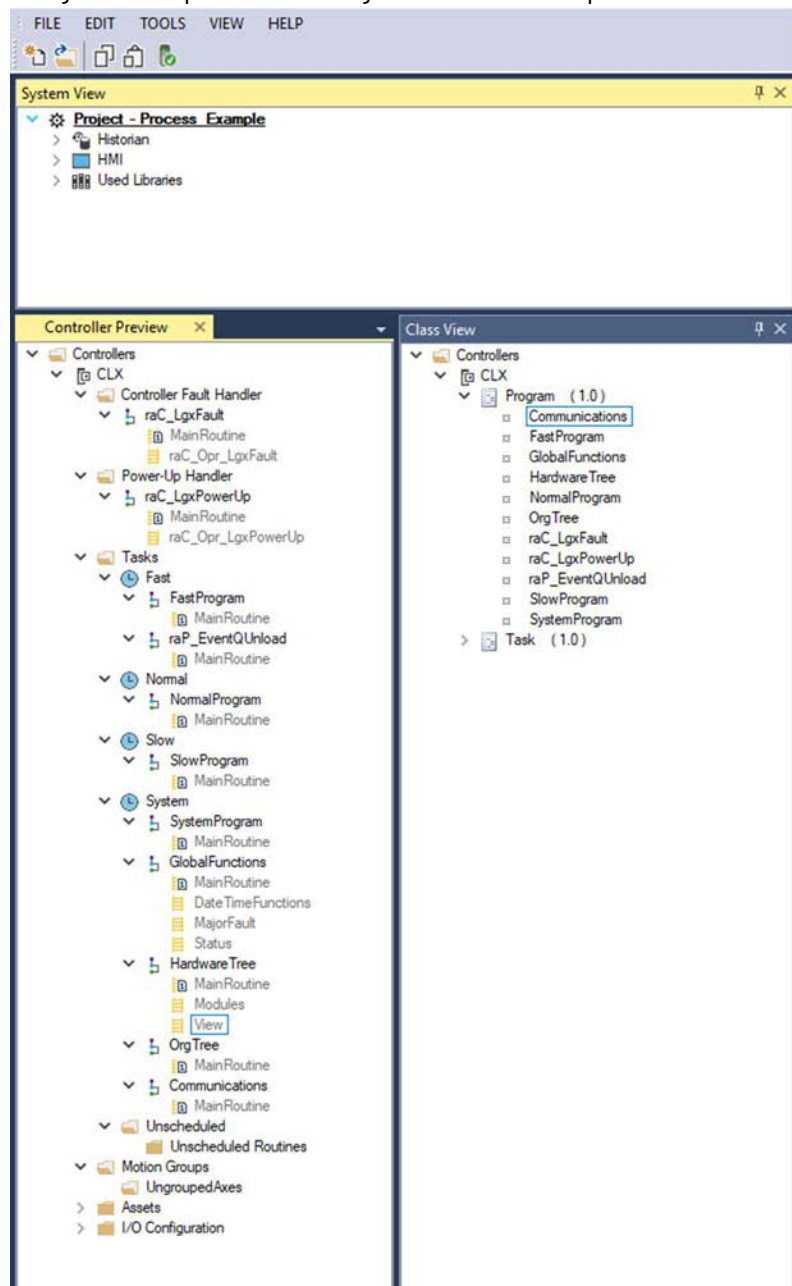
Parameter	When True
Has_ChangeDetect	Creates an instance of raP_Dvc_LgxChangeDet instruction and associated alarm. Adds an instance of the raP_Dvc_LgxChangeDet instruction to the HardwareTree program's MainRoutine.
Has_Redundancy	Creates an instance of raP_Dvc_LgxRedun instruction and associated alarm. Makes the configuration of the secondary controller not ready alarm available. Adds an instance of the raP_Dvc_LgxRedun instruction to the HardwareTree program's MainRoutine.
Has_TaskMonitor	Creates instances of the raP_Dvc_LgxTaskMon instruction in each task. Adds an instance of the raP_Dvc_LgxTaskMon for each task to the MainRoutine of the matching name program of every task.
Has_OOAP	If true, [Use_OOAP] parameter is visible for each control strategy added to the project. If [Use_OOAP] is true, code to use arbitration and ownership through the organizational bus will be created.
Has_HART	Creates a routine in the system task for HART module code. If Has_TaskMonitor=True, setting Has_HART=True creates a Hart_Modules routine. Add HART objects and associated code in that routine.
Has_EventLogging	Adds event logging via Historian or FactoryTalk® Edge Gateway™ in the controller. The Edge Gateway selection requires additional code in the controller for event triggers. This selection creates: <ul style="list-style-type: none"> <li>• Event routine the same program as the control strategy.</li> <li>• Event tab in the parameter configuration for the control strategy</li> </ul> Historian reads and logs data from configured points directly in the controller.
Cfg_IncludeSystemTag	Creates a controller scope system tag that is used for enumerations of EM, EP, and other objects.

## ACM-Based Parameters for a Process Controller

These parameters affect code generation from ACM:

Parameter	Visible When	Details
<b>04 - Operations</b>		
Has_TaskMonitor	Always	Default=True Required to be set for a process application.
Has_OOAP	always	Set to use the bus for ownership and arbitration
Has_EventLogging	always	Set to add event logging instructions to each object to track programmatic events
EventLogging_Style	Has_EventLogging=True	Select the type of event logs: <ul style="list-style-type: none"> <li>• Historian</li> <li>• EdgeGateway (default)</li> </ul>
LGXEVENT_DelimeterOption	Has_EventLogging=True EventLoggingStyle=EdgeGateway	Select the delimiter type for the record: <ul style="list-style-type: none"> <li>• Character (default)</li> <li>• Length</li> </ul>
LBSMEVENT_DelimeterOption	Has_EventLogging=True EventLoggingStyle=EdgeGateway	Select the delimiter type for the record: <ul style="list-style-type: none"> <li>• Character (default)</li> <li>• Length</li> </ul>
RACEVENT_DelimeterOption	Has_EventLogging=True EventLoggingStyle=EdgeGateway	Select the delimiter type for the record: <ul style="list-style-type: none"> <li>• Character (default)</li> <li>• Length</li> </ul>
UserDefinedEVENT_DelimeterOption	Has_EventLogging=True EventLoggingStyle=EdgeGateway	Select the delimiter type for the record: <ul style="list-style-type: none"> <li>• Character (default)</li> <li>• Length</li> </ul>
<b>06 - I/O</b>		
IO_Map_Strategy	always	Select the I/O mapping strategy: <ul style="list-style-type: none"> <li>• 0=Standard mapping</li> <li>• 1=Use aliases</li> <li>• 2=Use I/O mapping tags in mapping routines</li> <li>• 3=Use I/O mapping tags and diagnostics in mapping routines</li> <li>• 4=Map I/O directly in mapping routines</li> <li>• 5=Use program connections</li> </ul>
Skip_IO	always	Select whether to generate ACM code with or without all I/O references: <ul style="list-style-type: none"> <li>• 0=I/O references present</li> <li>• 1=Create code without I/O references</li> </ul>
Skip_UnUsedIO	always	Select whether to generate ACM code with or without all unused I/O references: <ul style="list-style-type: none"> <li>• 0=Program parameters are present for all unused I/O points</li> <li>• 1=Create code without unused I/O points</li> </ul>
<b>09 - FactoryTalk Innovation Suite</b>		
Has_FTIS	always	Set if the application is integrated with a FactoryTalk Innovation Suite project

The following is an example of the resulting ACM structure with a process controller object.



## I/O Mapping

The `IO_Map_Strategy` parameter in the `Process_Controller` Object determines the I/O mapping strategy for all control modules in the ACM project.

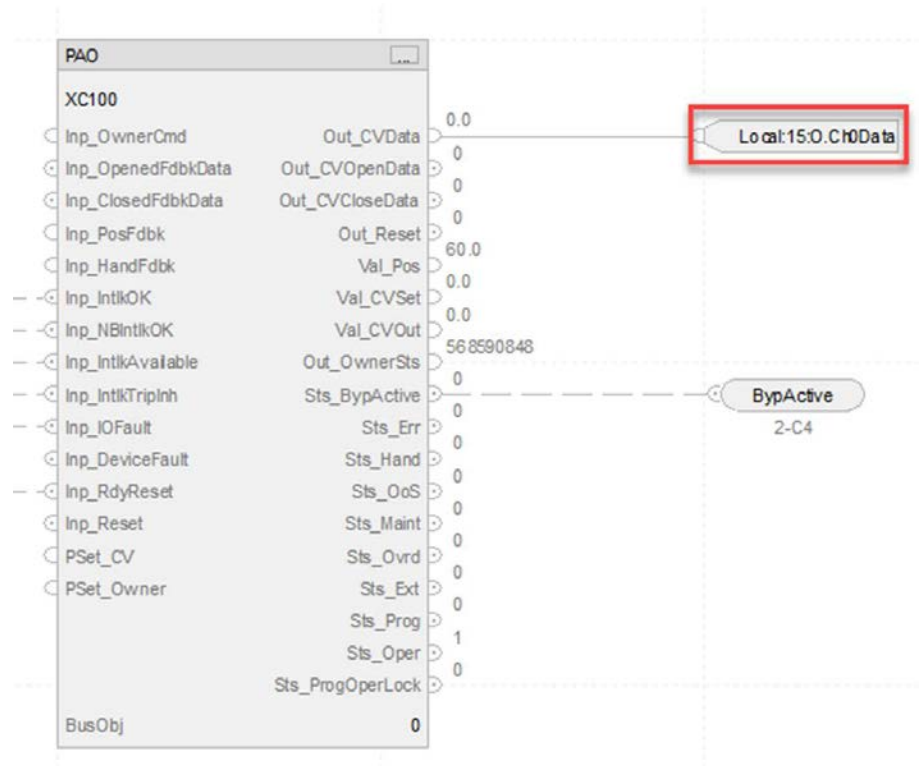
The available strategy types are:

- 0 = Standard Mapping in ACM
- 1 = Use Aliases for I/O
- 2 = Use I/O Mapping Tags in Mapping Routines
- 3 = Use I/O Mapping Tags and Diagnostics in Mapping Routines
- 4 = Map I/O Directly in Mapping Routines
- 5 = Use Program Connections

## I/O Map Strategy = 0

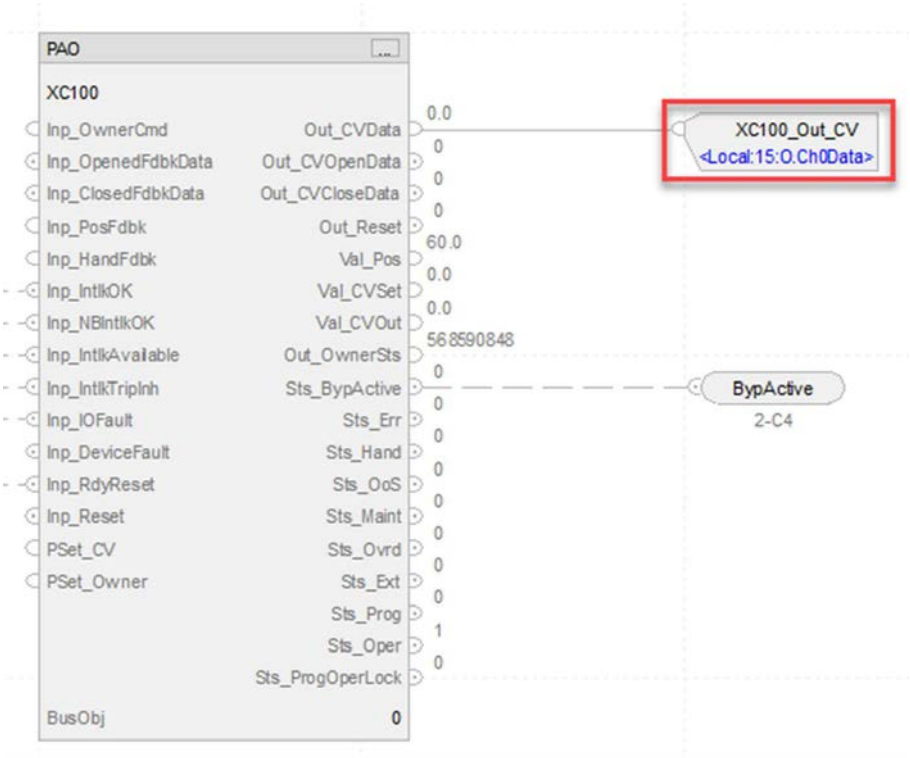
IO\_Map\_Strategy=0 specifies 'Standard Mapping in ACM.' Standard mapping uses a direct reference to the I/O module address.

The I/O module address is pinned directly to the corresponding Input or output parameter in the control module routine.



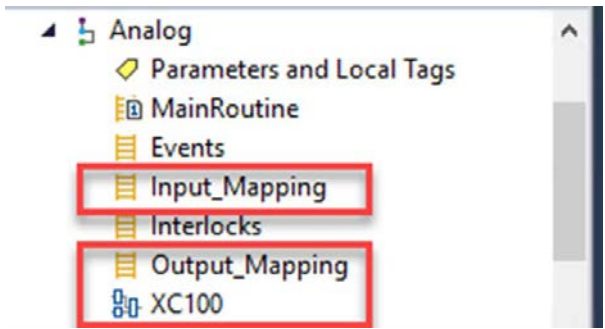
### IO\_Map\_Strategy = 1

IO\_Mapping\_Strategy=1 specifies 'Use Aliases for IO.' I/O tags are assigned alias tags according to the control module tag name and the parameter that it is pinned to. The alias of the tag references the I\O module tag.



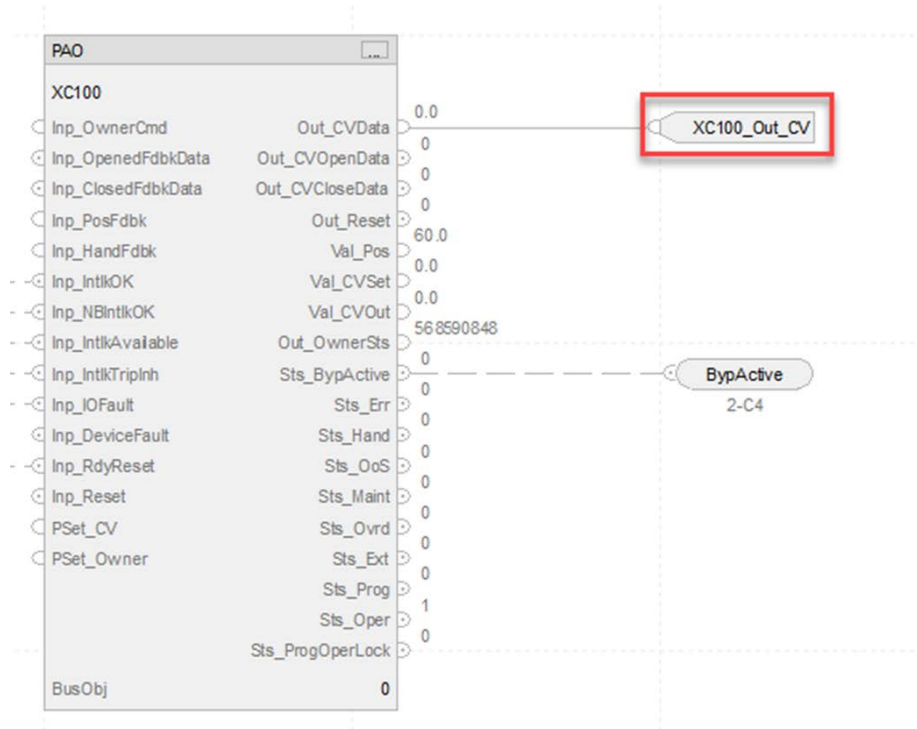
### IO\_Map\_Strategy = 2

IO\_Mapping\_Strategy=2 specifies 'Use IO Mapping Tags in Mapping Routines.' This mapping strategy creates a routine for the Control Module, Inputs, and Outputs. An intermediate tag is created and pinned to the control module and then is written to or from I\O module tags in the mapping routines.





In this example, the OutCVData parameter is pinned to an intermediate tag XC100\_Out\_CV.

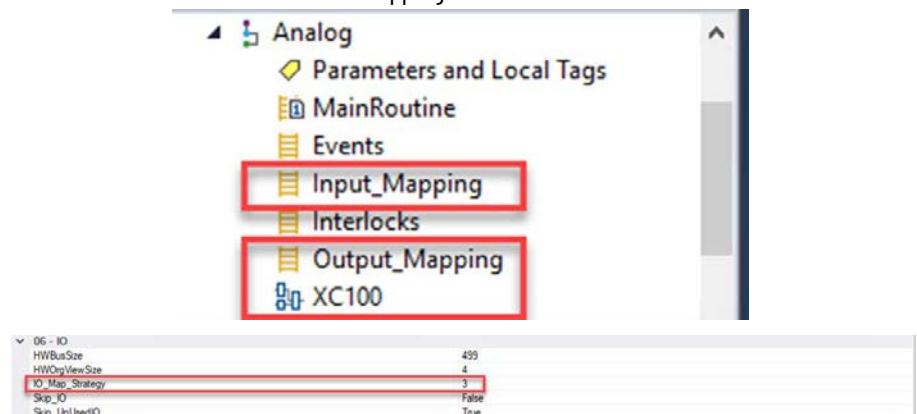


The intermediate tag's value (XC100\_Out\_CV) is moved into the I/O module address in the mapping routine.

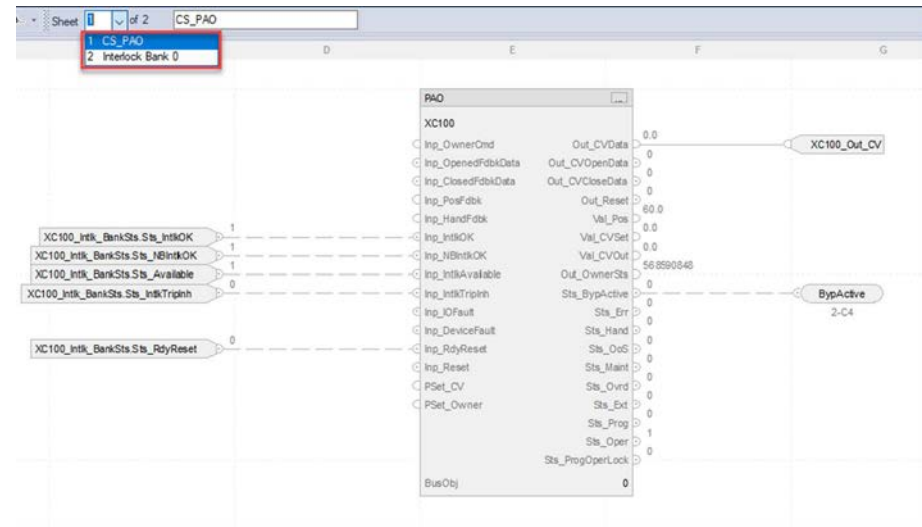


## IO\_Map\_Strategy = 3

IO\_Mapping\_Srtategy=3 specifies 'Use I/O Mapping Tags and Diagnostics in Mapping Routines.' This mapping strategy creates a routine with diagnostics for the Control Module, Inputs, and Outputs. An intermediate tag is created and pinned to the control module and then is written to or from I/O module addresses in the mapping routines.



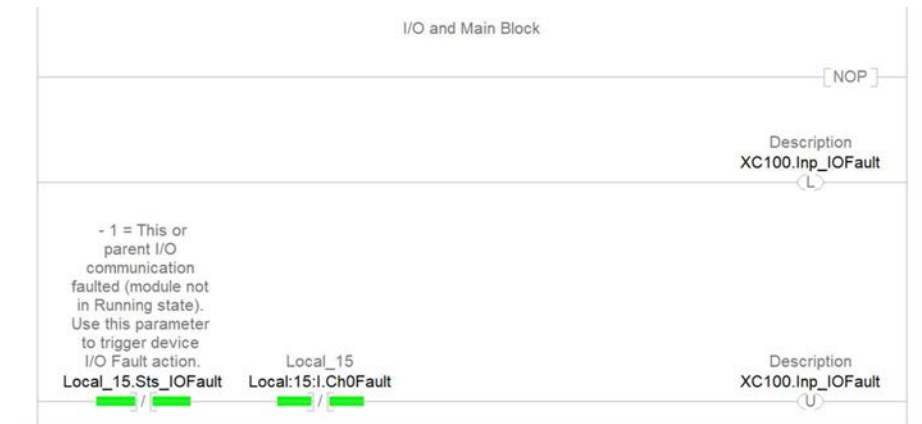
In this example, the OutCVData parameter is pinned to an intermediate tag XC100\_Out\_CV. For all other I/O mapping strategies, there is a separate page for I/O fault mapping. For this mapping strategy, the I/O fault code is moved to the Input and Output Mapping Routine.



The intermediate tag's value (XC100\_Out\_CV) is moved into the I/O module address in the mapping routine.

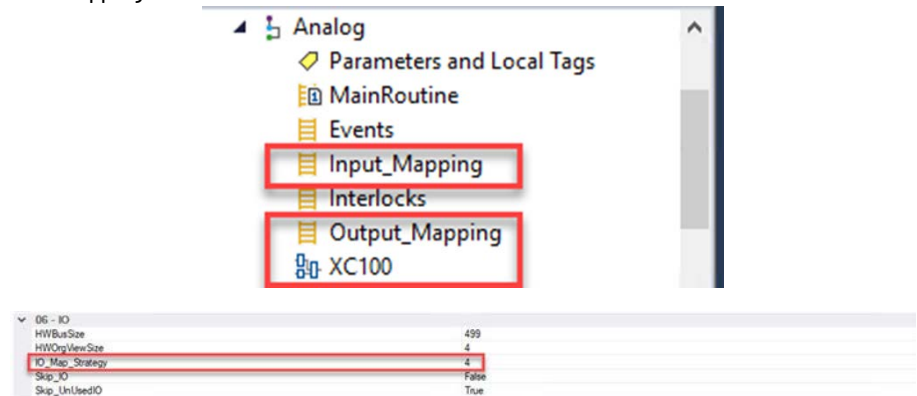


The I/O fault input parameter of the control module is set with the following code in the mapping routine.

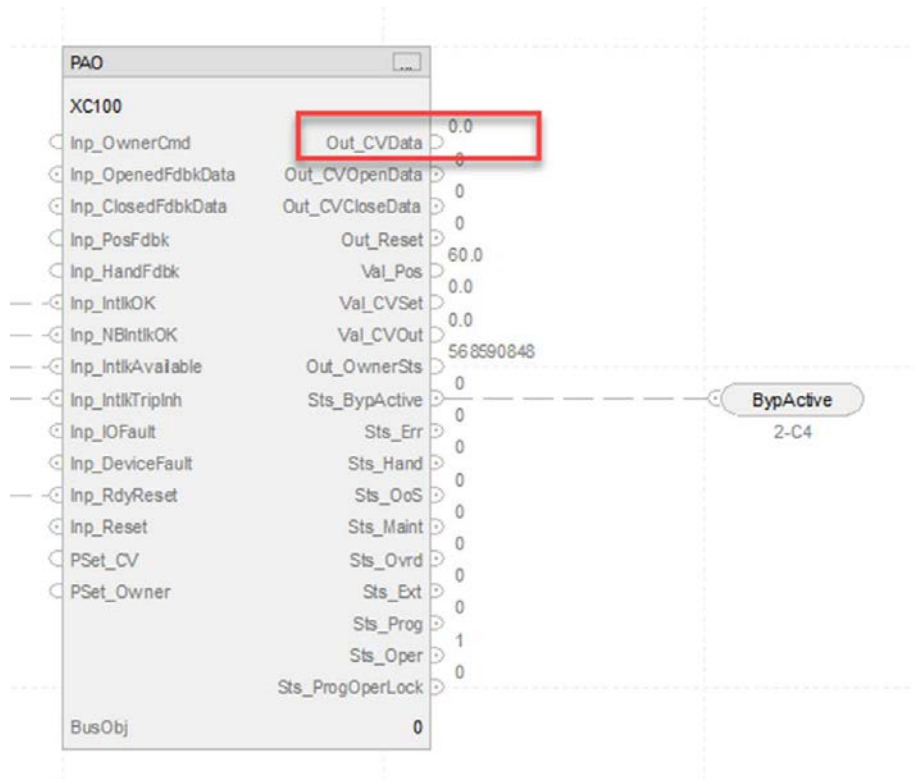


## IO\_Map\_Strategy = 4

IO\_Mapping\_Strategy=4 specifies 'Map IO Directly in Mapping Routines.' This mapping strategy creates a routine for the Control Module, Inputs, and Outputs. A direct reference from the Control Module's input and output parameters is written to and from the I/O module addresses in the mapping routines.



In this example, the OutCVDData parameter is not pinned to any tag. This connection is coded in the Output Mapping routine.



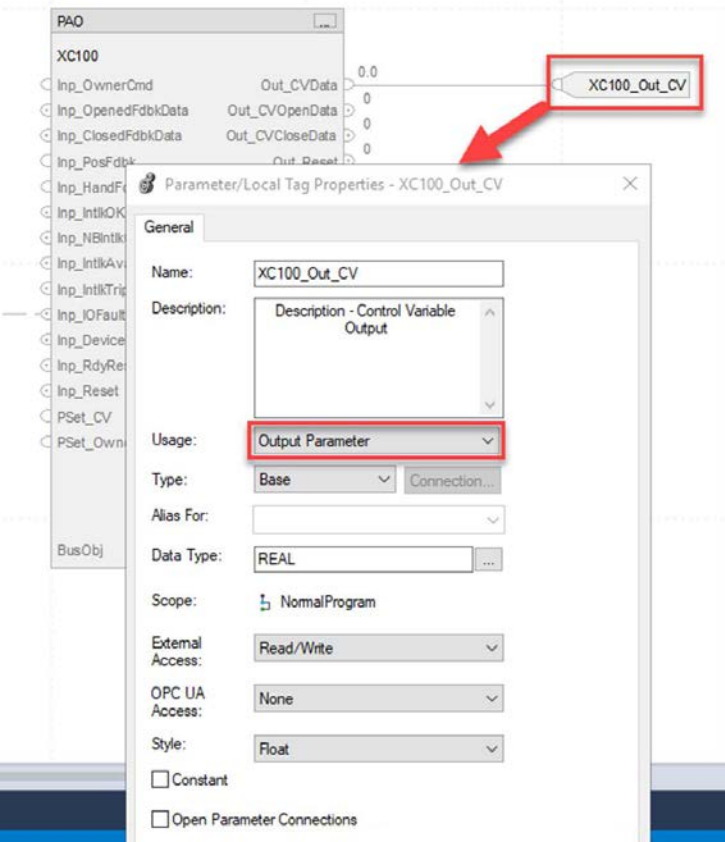
The OutCVDData parameter is moved into the I/O module address in the Output mapping routine.



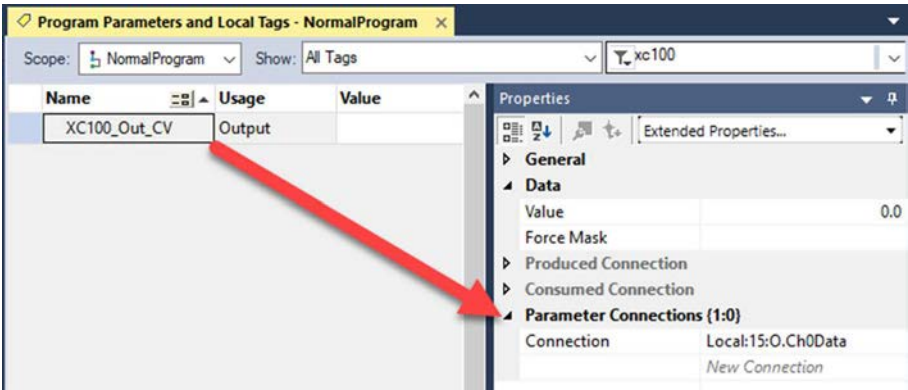
## IO\_Map\_Strategy = 5

IO\_Mapping\_Strategy=5 specifies 'Use Program Connections.' This mapping strategy links I/O tags to input or output local parameter connections. A local parameter connection tag is created for each I/O point in the control module. The maximum number of local parameters that are allowed per program is 250

In this example, the XC100\_Out\_CV tag is as an output parameter.



The property panel for XC100\_Out\_CV displays the parameter connection.

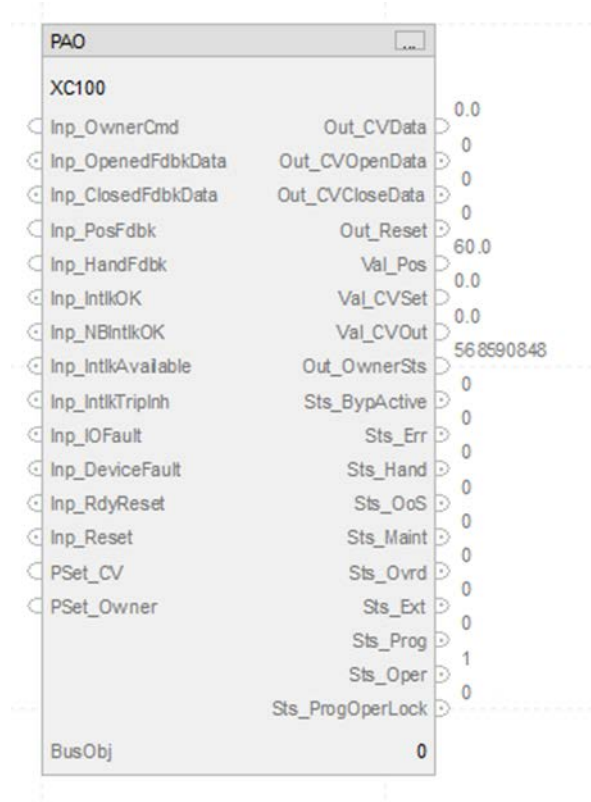


## Skip\_IO and Skip\_UnusedIO

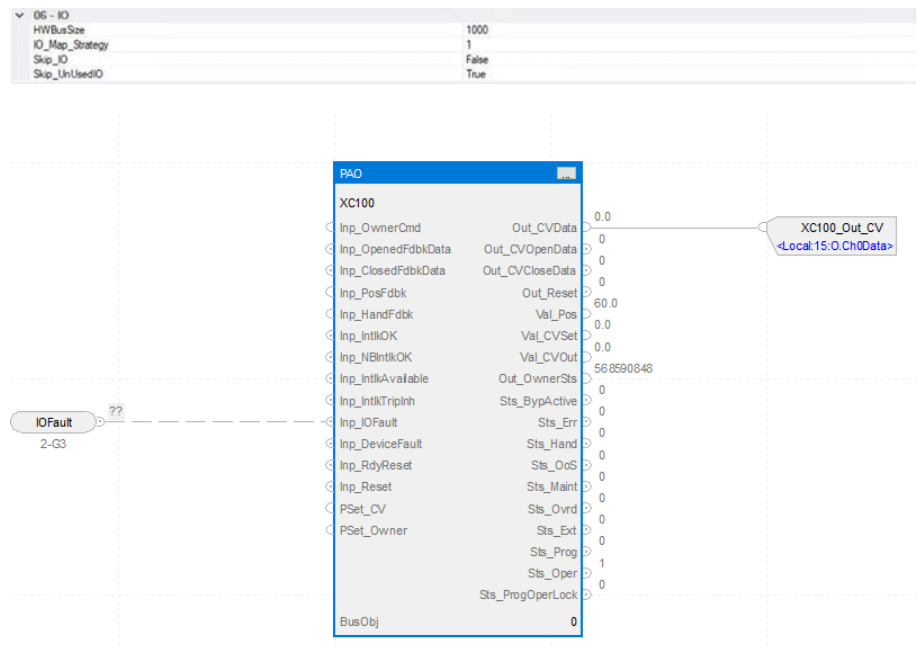
The Skip\_IO and Skip\_UnusedIO parameters provide the flexibility to create a program if I/O assignments are not yet defined. In these examples, the Skip\_IO and Skip\_UnusedIO parameters use an IO\_Map\_Strategy=1 for aliased tags.

If Skip\_IO=True, the control module is created without any I/O mapping code or I/O tags. In addition, no program parameters are created for the possible unused I/O modules.

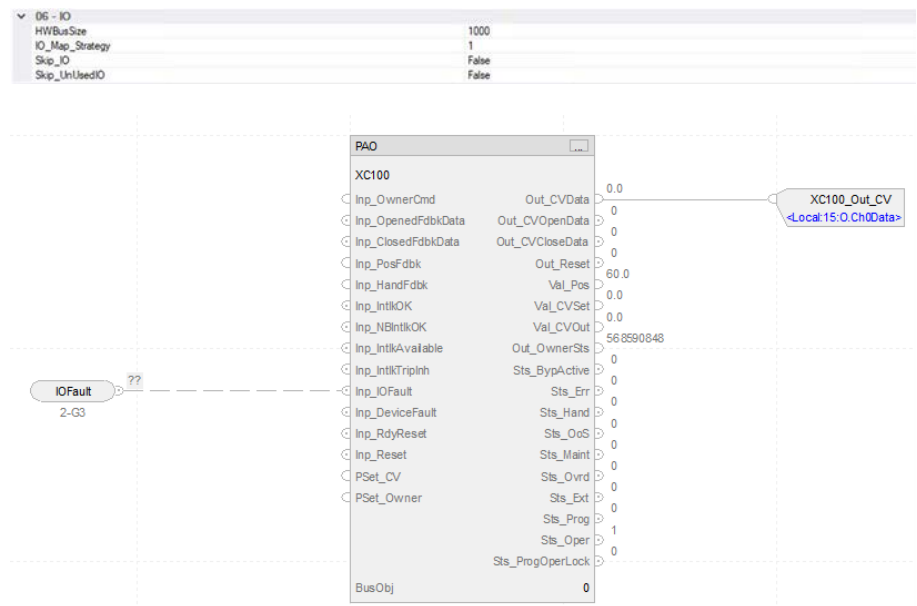
06 - IO	
HWBusSize	1000
IO_Map_Strategy	1
Skip_IO	True
Skip_UnusedIO	True



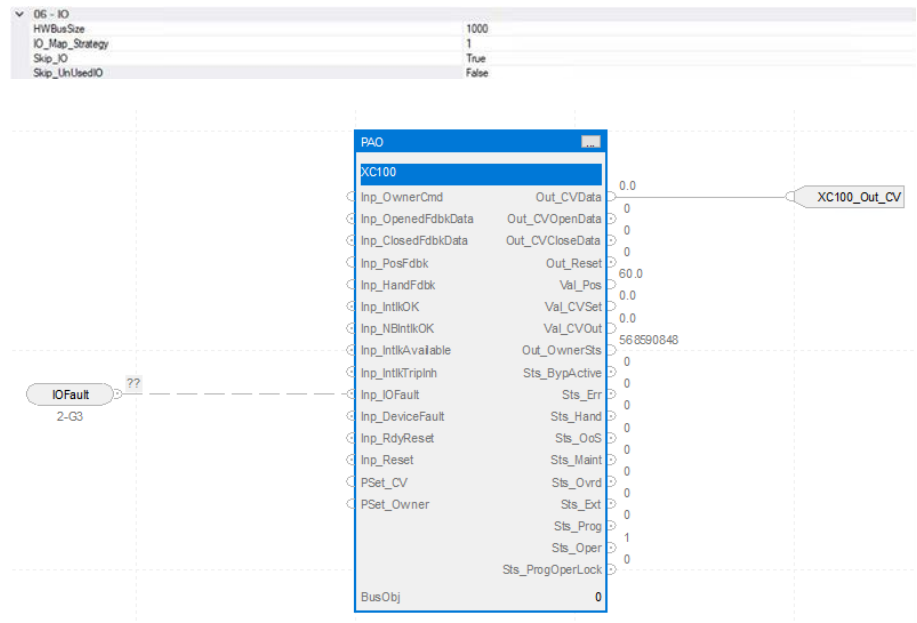
If Skip\_I0=False, the control module is created with I/O mapping code. Alias tags are created and pinned to the control module for all the assigned I/O modules. If SkipUnusedI0=True, no program parameters are created for the possible unused I/O modules.



If SkipUnusedIO=False, a local parameter tag is created for each possible I/O connection. Make sure you have fewer than 250 local parameters per program, or the ACD file will not compile from ACM.



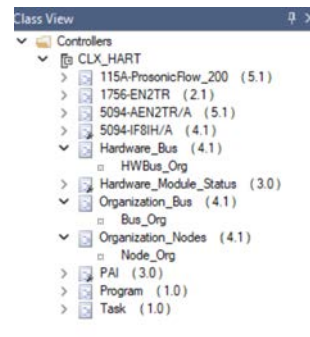
Name	Usage	Alias For	Base Tag	Data Type
XC100_Inp_ClosedLS_ChFit	Input			BOOL
XC100_Inp_ClosedLS_ModFit	Input			BOOL
XC100_Inp_OpenLS_ChFit	Input			BOOL
XC100_Inp_OpenLS_ModFit	Input			BOOL
XC100_Inp_PosFdbk_ChFit	Input			BOOL
XC100_Inp_PosFdbk_ModFit	Input			BOOL
XC100_Out_CV	Local	Local:15:O.Ch0Data(C)	Local:15:O.Ch0Data(C)	REAL



XC100_Inp_ClosedLS_ChFit	Input		BOOL
XC100_Inp_ClosedLS_ModFit	Input		BOOL
XC100_Inp_OpenLS_ChFit	Input		BOOL
XC100_Inp_OpenLS_ModFit	Input		BOOL
XC100_Inp_PosFdbk_ChFit	Input		BOOL
XC100_Inp_PosFdbk_ModFit	Input		BOOL
XC100_Out_CV	Local		REAL
XC100_Out_CV_ChFit	Input		BOOL
XC100_Out_CV_ModFit	Input		BOOL

## Ownership/Arbitration

To use ownership and arbitration in your project, you must add these objects to the System task: Hardware\_Bus, Hardware\_Module\_Status, Organization Bus, and Organization\_Nodes.

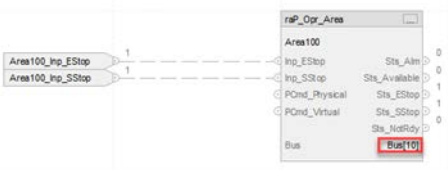


## Organization Bus

The size of the Organization Bus should reflect the size of the system. Each Area, Unit, Equipment Module, and Control Module uses one connection to the bus array.

Parameter	When True
OrgViewSize	This is a Bus_Org parameter. For PlantPAx system release 5.00.06 and later, the bus array supports as many as 1500 elements. For earlier PlantPAx system releases, the bus array supports as many as 500 elements.
NodeSize	This is a Bus_Org parameter. Specify a node array size for the system. Typically, the node array should be twice the size of the bus array.

The following parameters can be used on the controller and control strategy objects to configure the organization bus.

Parameter	When True
Has_OOAP	This is a controller parameter; See <a href="#">Process Controller on page 36</a> Set to use arbitration and ownership through the organizational bus
Use_OOAP	Set to use the bus for ownership and arbitration.
Use_ArbitrationQ	Set to use the raP_Opr_arbitrationQ instruction for ownership queuing.
Bus_Instance	Set to link to a bus array instance. This is unique for each device. For example: 

## Hardware Bus and Hardware Module Status

Add a Hardware\_Module\_Status instance for each device in the Hardware Bus.

Parameter	Description
Module	Link to the device. The name of the object changes to reflect the selected device.
Bus_Instance	Link to the instance in the Hardware Bus. There should be one hardware monitoring status object for each device in the I/O tree and 1 per task in the controller. Each instance of the object should be linked to a unique bus reference.

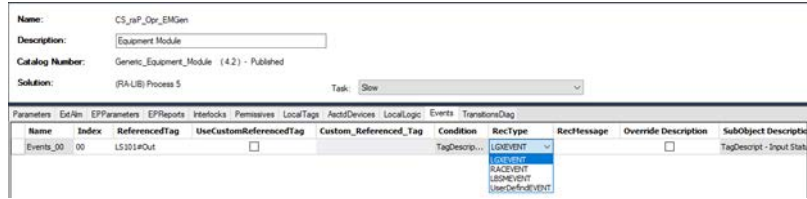


## Sub-Objects

Many objects have sub-objects that appear as tabs on the configuration dialog. Each control strategy refers to the sub-objects it uses.

### Event Logging

If a control strategy is configured for Has\_EventLogging=True on the Process\_Controller object, the Event tab lets you define events to log.



Parameter	Available When	Details
Index	always	Determines the event number
ReferencedTag	UseCustomReferencedTag=False	Link to the tag that triggers the event. The resulting code contains an XIC of the linked tag. In this example, LS101.Out is the trigger event tag.
UseCustomReferencedTag	always	Set to enable custom neutral next entry for the event trigger tag.
Custom_Referenced_Tag	UseCustomReferencedTag=True	Enter the custom neutral text for the trigger tag. For example, to have LS101.Out as the text, enter 'XIC(LS101.Out)'
RecType	Always	Select the event record type: <ul style="list-style-type: none"> <li>• LGXEVENT</li> <li>• RACEVENT</li> <li>• LBSMEVENT,</li> <li>• UserDefinedEVENT</li> </ul>

### Interlocks

The Interlock sub-object lets you create instances of inputs into the PPERM instruction. Verify the Cfg\_HasIntlkObj parameter (or similar parameter) for the control strategy is set to true.



Parameter	Available When	Details
Index	always	Determines the Interlock input number. ACM generates new PINTLK instruction for every 32 interlock sub-objects. For example, Index 0 is pinned to Inp_intlk00 of the first interlock instance (Intlk_0) and index 32 is pinned to Inp_intlk00 of the second interlock instance (Intlk_1)
ReferencedTag	UseCustomReferencedTag=False	Link to the tag that triggers the interlock. The resulting code includes an XIC of the linked tag. In this example, LS101.Out is the interlock input trigger.

Parameter	Available When	Details
UseIOFault	always	Set this parameter if the ReferencedTag parameter is a control module and you want to monitor the Sts.IOFault of the control module as an interlock.
UseCustomReferencedTag	always	Set this parameter to enable custom neutral text entry for the interlock input
Custom_Referenced_Tag	UseCustomReferencedTag=True	Enter the custom neutral text for the interlock input tag. For example, to have LS101.Out as the text, enter 'XIC{LS101.Out}' into the parameter

Permissives

The permissive sub-object lets you create instances of inputs into the PPERM instruction. Verify the Cfg\_HasPermObj parameter (or similar parameter) for the control strategy is set to true.

Name:CS\_xsl\_Opr\_EMGen

Description:Equipment Module

Catalog Number:Generic\_Equipment\_Module (4.2) - Published

Solution:(RA-LB) Process 5

Task:Slow

Parameters

ExtNm

EPParameters

EPReports

Interlocks

Permissives

LocalTags

AcctDevices

LocalLogic

Events

Transitions

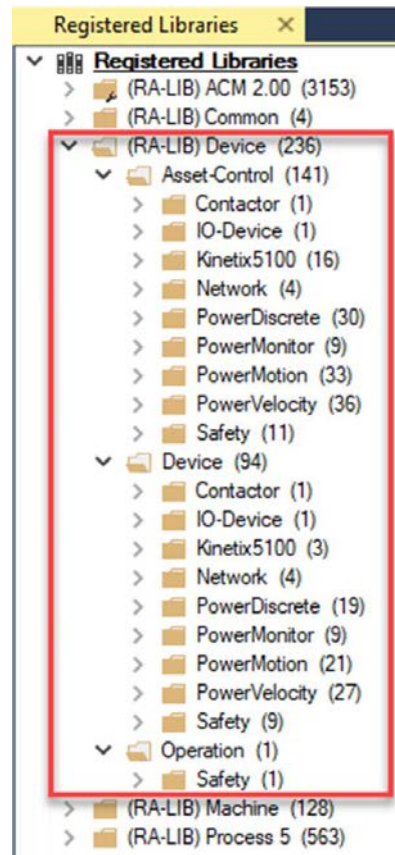
Diag

Name	Index	ReferencedTag	UseIOFault	UseCustomReferencedTag	Custom_Referenced_Tag	Condition	Custom_Description	Bypassable	Invert	Latched	Sto
Inlk_00	00	LS101#Out	<input checked="" type="checkbox"/>	<input type="checkbox"/>		TagDescr...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

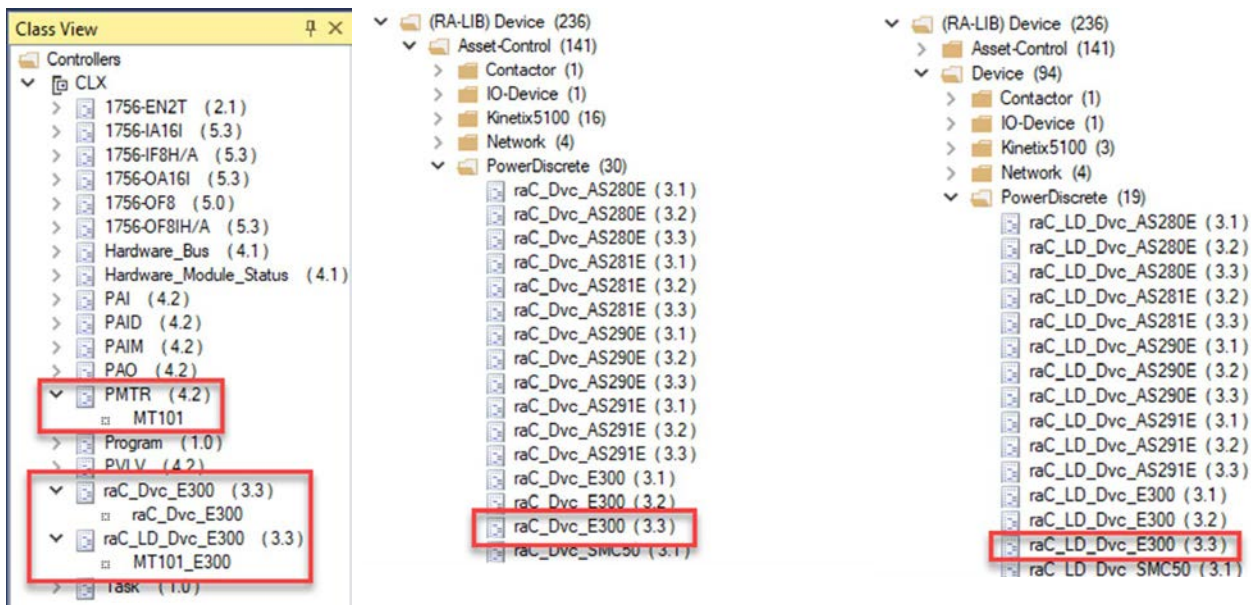
Parameter	Available When	Details
Index	always	Determines the permissive input number. For example, Index 0 will be pinned to Inp_Perm00
ReferencedTag	UseCustomReferencedTag=False	Link to the tag that triggers the permissive. The resulting code contains an XIC of the linked tag. In this example, LS101.Out is the permissive input.
UseCustomReferencedTag	always	Set this parameter to enable custom neutral text entry for the permissive input
Custom_Referenced_Tag	UseCustomReferencedTag=True	Enter the custom neutral text for the permissive input tag. For example, to have LS101.Out as the text, enter 'XIC{LS101.Out}' into the parameter

## Device Object [Cfg\_HasDvcObj]

The PMTR, PVSD, and PPID\_PVSD control strategies use additional libraries for associated devices. You need to register these libraries in ACM.




For example, a PMTR instruction references a linked library file for an E300 device. For each linked library to the motor, you need an asset control object and a device object.



Create the raC\_Dvc\_E300 object first. Leave the ObjectInterfaceDatatype parameter blank because it is automatically populated when you link to the raC\_LD\_Dvc\_Object object.

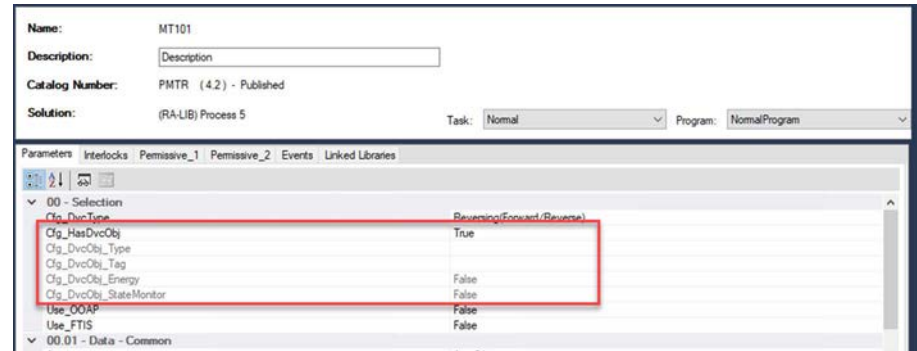
Before you add the raC\_LD\_Dvc\_E300 object, configure the parent EtherNet/IP™ communication module in the I/O tree. When you add device object, it shows under the communication module.

Configure the scope of the tags and the ObjectInterfaceDatatype to your application needs. After you configure the parameters, link the device object to the raC\_Dvc\_E300 object on the Linked Libraries tab.

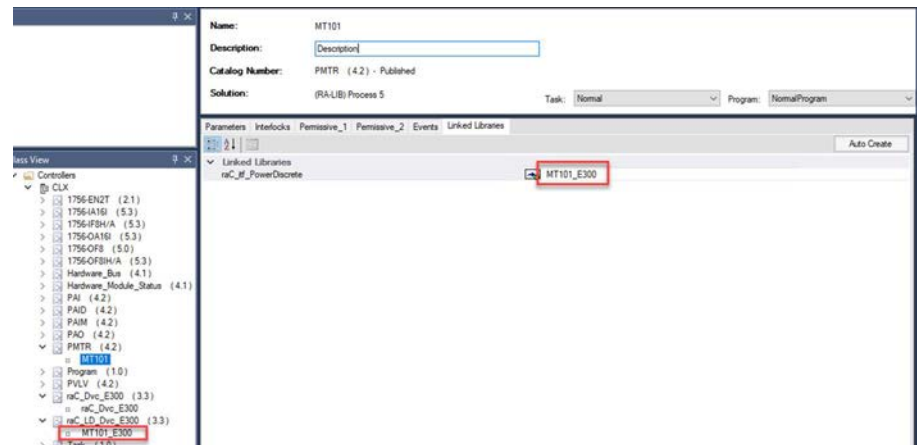
 Make sure that the IP address for the E300 is not a duplicate of any other device in the project or ACM will not add the device when it generates the ACD file.

Click the ellipsis and select the desired raC\_Dvc\_E300 asset library object.

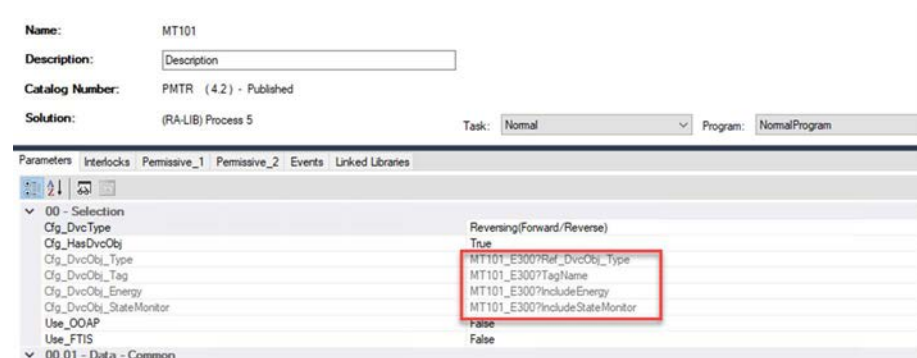
On the PMTR object, set Cfg\_HasDvcObj=True for the grayed-out parameters and the Linked Libraries tab to become visible.



Link the raC\_LD\_Dvc.E300 object corresponding to the motor.



After the library link is connected, additional parameters populate automatically.



## Extended Alarms

The extended alarm sub-object creates instances of the raP\_Opr\_ExtddAlm instruction. The parent object supports as many as 32 extended alarms. The related code is placed in the A04\_ExtddAlarms routine.

Name:CS\_raP\_Opr\_EMGen

Description:Equipment Module

Catalog Number:Generic\_Equipment\_Module (4.2) - Published

Solution:(RA-LIB) Process 5

Task:Slow

Parameters	ExtAlm	EPParameters	EPReports	Interlocks	Permissives	LocalTags	AcctsDevices	LocalLogic	Events	TransitionsDiag
Name	Index	Message	AckReqd	ResetReqd	Severity	ReferencedTag	UseCustomReferencedTag	Custom_Referenced_Tag	Condition	TagDescr...
EM_00	00	Insert extended message here.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	501	LS101.Out	<input type="checkbox"/>			TagDescr...

Parameter	Available When	Details
Index	always	Determines the alarm number. For example, Index 0 creates the EM_ExtddAlm_00 tag for an Equipment module.
ReferencedTag	UseCustomReferencedTag=False	Link to the tag that triggers the alarm. The resulting code contains an XIC of the linked tag. In this example, LS101.Out is the alarm input.
UseCustomReferencedTag	always	Set this parameter to enable custom neutral text entry for the alarm input
Custom_Referenced_Tag	UseCustomReferencedTag=True	Enter the custom neutral text for the alarm input tag. For example, to have LS101.Out as the text, enter 'XIC(LS101.Out)' into the parameter

## Parameters

The parameter sub-object lets you create instances of the raP\_Tec\_ParRpt instruction. The parent objects supports as many as 256 parameters. The related code is placed in the A07\_Parameters routine.

Name:CS\_raP\_Opr\_EMGen

Description:Equipment Module

Catalog Number:Generic\_Equipment\_Module (4.2) - Published

Solution:(RA-LIB) Process 5

Task:Slow

Parameters	ExtAlm	EPParameters	EPReports	Interlocks	Permissives	LocalTags	AcctsDevices	LocalLogic	Events	TransitionsDiag
Name	Index	ParamType	Param_Name	Param_DvcDescrpt	Param_Public	Param_DcmIPics	Param_EU	Param_Default	ParamR_Cfg_Default	ParamS_Cfg_Def
Par_00	00	Enum	Task	0	<input type="checkbox"/>	0	%	0	1.0	Null
Par_01	01	Integer	Duration	0	<input type="checkbox"/>	0	%	0	1.0	Null
Par_02	02	Real	TempSP	0	<input type="checkbox"/>	0	C	0	1.0	Null
Par_03	03	String	ID	0	<input type="checkbox"/>	0	%	0	1.0	Null

Parameter	Available When	Details
Index	always	Determines the parameter number. For example, Index 0 creates the EM_PAR_00 tag for an Equipment module.
ParamType	always	Select a datatype for the parameter: Integer, Real, String or Enumeration
Param_Name	always	Creates a tag (of ParamType) with the name of this parameter's value. This tag is linked to the input parameter of the corresponding data type on the raP_Tec_ParRpt in code
Param_Public	always	Set to make the Param_Name tag a public parameter

## Reports

The reports sub-object lets you create instances of the raP\_Tec\_ParRpt instruction. The parent object supports as many as 256 Reports. The resulting code is placed in the A08\_Reports routine.

Name	Index	ReportType	Report_Name	Report_Descrpt	Report_Public	Report_Default	ReportR_Cfg_Default	ReportS_Cfg_Default	ReportD_ConfRcs	Req
Rep_00	00	Enum	ExitStatus	0	<input type="checkbox"/>	0	0	Null	0	%
Rep_01	01	Integer	FaultCode	0	<input type="checkbox"/>	0	0	Null	0	%
Rep_02	02	Real	ActualTemp	0	<input type="checkbox"/>	0	0	Null	0	%
Rep_03	03	String	ID	0	<input type="checkbox"/>	0	0	Null	0	%

Parameter	Available When	Details
Index	always	Determines the report number. For example, Index 0 creates the EM_RPT_00 tag for an Equipment module.
ParamType	always	Select a datatype for the report: Integer, Real, String or Enumeration
Param_Name	always	Creates a tag (of ParamType) with the name of this parameter's value. This tag is linked to the input parameter of the corresponding data type on the raP_Tec_ParRpt in code
Param_Public	always	Set to make the Param_Name tag a public parameter

## EM and EP Additional Sub-Objects

The EM and EP Objects have additional sub-objects.

### Local Tags

The local tags sub-object creates locally-scoped tags in the program.

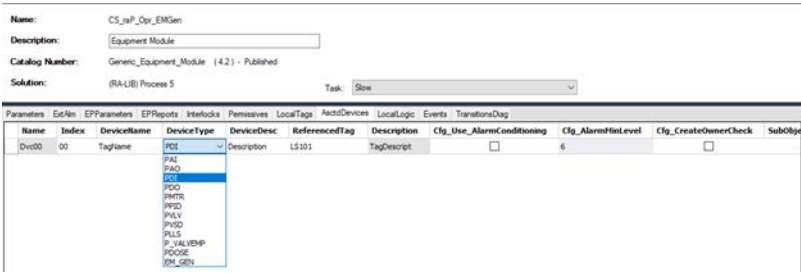
Name	Index	LocalTagType	LocalTagName	LocalTagDescription	Timer_Preset	String_Default	Bool_Default	Real_Default	DINT_Default	INT_Default	SubObjec
LocalTag	00	BOOL			0		<input type="checkbox"/>	0	0	0	

Parameter	Available When	Details
LocalTagName	always	Tag name
LocalTagType	always	Select a datatype for the report: BOOL, INT, DINT, REAL, STRING, TIMER
LocalTagDescription	always	Tag description

## Associated Devices

The associated devices sub-object creates associated control modules for use with the program. Status code monitoring, ready status, and maintenance mode are added for each device. Ownership status and alarm status are optional for each device

Set BuildArbtrnLogic=True to create manual arbitration logic for each device.



Parameter	Available When	Details
DeviceName	always	Name of locally-scoped tag aliased to ReferencedTag parameter
DeviceType	always	Select a datatype for the associated device: PAI, PAO, PDI, PDO, PMTR, PPID, PVLV, PVSD, PLLS, P_VALVEMP, PDOSE, EM_GEN
DeviceDesc	always	Tag description
Cfg_Use_AlarmConditioning	always	Set to add code to monitor the alarm status of the associated device
Cfg_CreateOwnerCheck	always	Set to add code to monitor the ownership status of the associated device

## Local Logic

The local logic sub-object creates custom rungs of ladder logic. The resulting code is placed in the A10\_LocalLogic routine.

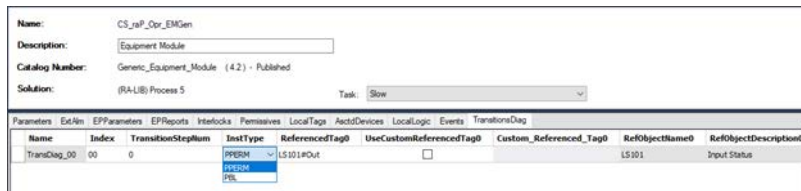


Parameter	Available When	Details
Index	always	Determines the rung number where the logic will be created
LocalLogic_Code	always	Custom neutral text logic input. For example, XIC(Tag1)OTE(Tag2)
LocalLogic_Desc	always	Rung Comment for resulting ladder code



## Transition Diagnostics

The transition diagnostic sub-object links a permissive or PBL instruction to the transition of a state. You can have as many as 16 sub-objects using a permissive block or 8 using the PBL instruction.



Parameter	Available When	Details
Index	always	Determines the input number. For example, Index 00 is pinned to Inp_Perm00 of the permissive if the InstType is PPERM
Transition Step Number	always	Set equal to the value of the step number of the associated PPERM or PBL.
ReferencedTagX X = index value	UseCustomReferencedTag=False	Link to the tag that triggers the PPERM or PBL input. The resulting code contains an XIC of the linked tag. In this example, LS101.Out is the permissive input.
UseCustomReferencedTagX X = index value	always	Set this parameter to enable custom neutral text entry for the input.
Custom_Referenced_Tag X = index value	UseCustomReferencedTag=True	Enter the custom neutral text for the permissive input tag. For example, to have LS101.Out as the text, enter 'XIC(LS101.Out)' into the parameter

## Statistics Objects

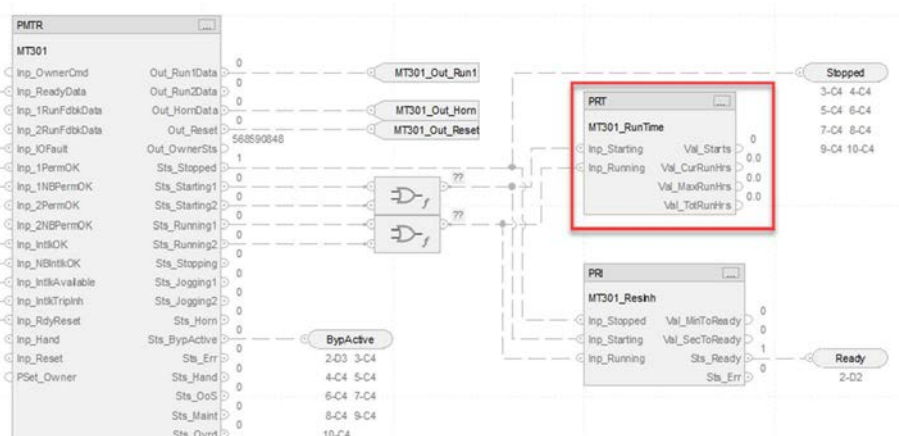
These statistics objects are available in the specified control strategies.

### Runtime Object

PMTR and PVSD both support a runtime object that tracks statistics of the motor or variable speed drive. For example, the object can track the number of starts and total amount of running time. To create an instance of the runtime object, set the Cfg\_HasRunTimeObj=True (available in the 01 - Options section).



This example shows a motor control strategy configured to use a runtime object.

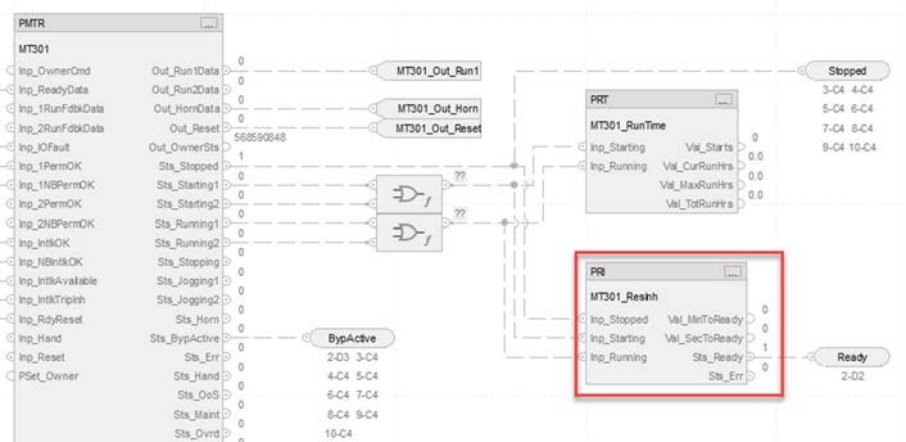


## Restart Inhibit

PMTR and PVSD both support a restart inhibit object that determines how many times a device can start in a given period of time. For example, it is common to prevent a medium voltage motor from starting more than three times in an hour to prevent over-heating of the start contact. To create an instance of the restart inhibit object, set Cfg\_HasResinhObj=True (available in the 01 - Options section). The restart inhibit Sts\_Ready wire connector is pinned to input 0 of the permissive block by default.

01 - Options	
Cfg_HasIntrkObj	True
Cfg_Has1PermObj	True
Cfg_HasResinhObj	True
Cfg_HasRunTimeObj	True
Cfg_HasHand	False

This example shows a motor control strategy configured to use a restart inhibit object.

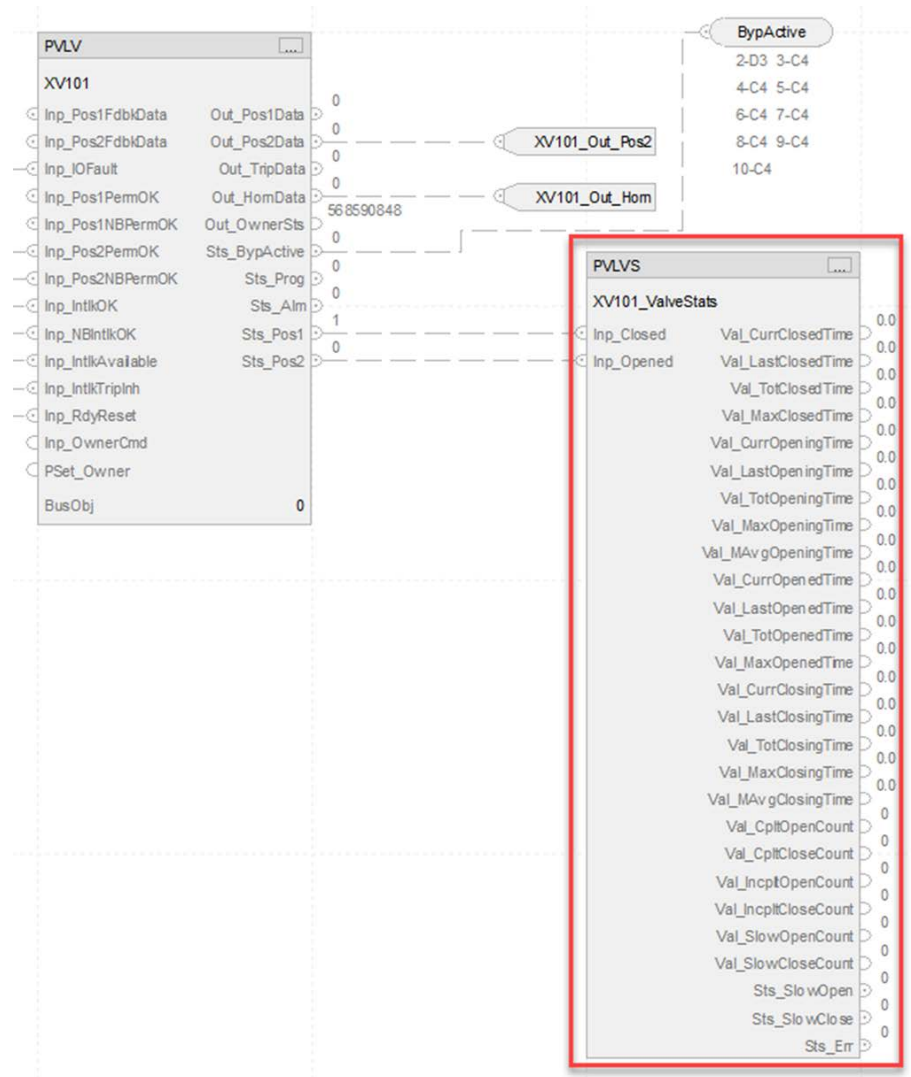


## Valve Statistics

PVLV supports a valve statistics object that tracks data about the valve. For example, the object can track long a valve takes to open after an open command. To create an instance of the valve statistics object, set Cfg\_HasStatsObj=True (available in the 02 - Device Configuration section).

02 - Device Configuration	
Cfg_FalPos2	False
Cfg_FdbkFail	True
Cfg_MtrnOut	False
Cfg_MtrnOutAin	False
Cfg_HasPulse	False
Cfg_CompletePulse	False
Cfg_HasPulseToState	False
Cfg_HasMaxOvr	False
Cfg_HasStatsObj	True
Cfg_CoastToLS	True
Cfg_OperPos1Prio	False
Cfg_OCmdResets	False
Cfg_XCmdResets	False
Cfg_OvrdPermIntrk	False
Cfg_ICmdPos2AsLevel	False
Cfg_ShedOnIOFault	True
Cfg_ShedOnFullStall	True
Cfg_ShedOnLossPos1	False
Cfg_ShedOnLossPos2	False
Cfg_ShedOnTransitStall	True
Cfg_Pos1Dly	2.0
Cfg_Pos2Dly	2.0
Cfg_Pos1PulseTime	5.0
Cfg_Pos2PulseTime	5.0
Cfg_OutPulseTime	5.0
Cfg_StartHornTime	0.0
Cfg_VirtualFdbkTime	3.0
Cfg_FalTime	3.0
Cfg_ShedOnFal	True
Cfg_AllowDisable	True
Cfg_AllowShelve	True

This example of a valve control strategy is configured to use a valve statistics object.



## ACM Considerations for a Prompt Instance

There are several instructions that utilize the Prompt instruction. Configure the Prompt parameters to define operator interaction that can be used within an associated control strategy.

### ACM-Based Parameters for a Prompt Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See Controller Parameters on pg #
Cfg_HasConnection	always	Set if the Prompt object has an associated sequencer, equipment module, equipment phase, or unit object
Cfg_ConnectionType	Cfg_HasConnection=True	Select the type of connection: Sequencer Equipment Unit
<b>04 - Alarm Configuration</b>		
Cfg_HasAlertTimeOutAlm	always	If Cfg_HasAlertTimeOutAlm=True, ACM displays section 4.01 - Alert Time Out Alarm with additional parameters

### Additional Sub-Objects for an Area Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Prompt Instructions	Configure messages and labels associated with the prompt object. This sub-object does not affect the structure of the resulting code. All parameters in this sub-object can be configured with an OPC read/write spreadsheet or PlantPAx Tool after code generation.
Events	Configure an event to monitor for the control strategy See Event Logging on pg#

## HART Integration

### HART Data

The PlantPax® control strategies that use HART data use a Process Analog HART (PAH) instruction to provide input to a Process Analog Input (PAI) instruction. For more information, see the PAI Control Strategy on [page 149](#).



The examples in this chapter use Application Code Manager (ACM) to enable more efficient project development with libraries of reusable code. Application Code Manager creates modular objects with customizable configuration parameters using the reusable content. Application Code Manager can also create the associated visualization, historical, and alarming elements for a project.

#### PAH Configuration Considerations

Operand	Type	Description
PlantPax control	P_ANALOG_HART	<ul style="list-style-type: none"> <li>Instance of data structure (backing tag) required for proper operation of instruction.</li> </ul>
Ref_HARTData	PAX_HART_DEVICE:I:0	<ul style="list-style-type: none"> <li>Required data type: HART data from the I/O module assembly.</li> <li>Select the HART device in your Controller Organizer. The device must support the PAXDevice data type: IOTreeObject:I.PAXDevice</li> </ul>
Ref_DiagTable	P_HART_CODE_DESC_STATUS[2]	<ul style="list-style-type: none"> <li>Lookup table for diagnostic bit number (to message and status).</li> <li>Select the correct table for your HART device; see table below.</li> </ul>
Ref_UnitsTable	RAC_CODE_DESCRIPTION[2]	<ul style="list-style-type: none"> <li>Lookup table for units of measure code (to units text). Select _HART_EUTable_Generic.</li> </ul>

## Fully Integrated HART with FLEX 5000 I/O

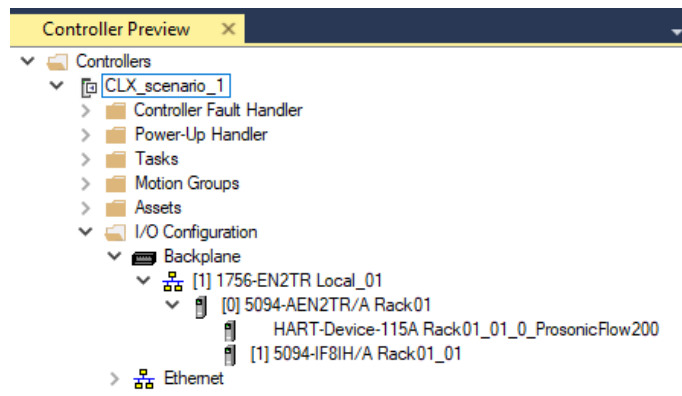
Highly-integrated HART uses a PlantPAx data type in the process controller for use with FLEX 5000® modules:

- Configuration of devices within the I/O Configuration tree (no Add-On Instruction needed)
- Device diagnostics automatically propagate to the controller project

### Integrate FLEX 5000 I/O with HART Device Using PAH and PAI Instructions

In this example, the ACM project contains:

- ControlLogix Process controller
- 1756-EN2TR communication module
- 5094-AEN2TR communication module for FLEX 5000 I/O connectivity
- 5094-IF8IH HART analog input module with an Endress+Hauser
- ProsonicFlow 200 instrument connected to Channel 0




---

**IMPORTANT** When you add multiple EtherNet/IP™ communication modules to an ACM project, remember to enter a unique IP address for each module.

---

1. Configure the process controller parameters that are required for your application, and set Has\_HART to True.

The screenshot displays the configuration window for the **CLX\_scenario\_1** controller. The **Class View** on the left shows the hierarchy: **Controllers** > **CLX\_scenario\_1**. The main panel shows the following details:

- Name:** CLX\_scenario\_1
- Description:** Description
- Catalog Number:** Process\_Controller (4.0) - Published
- Solution:** (RA-LIB) Process 5

The **Parameters** section is expanded, showing a list of parameters organized into categories:

- 01 - Controller**
  - ChassisName: Local
  - Slot: 0
  - Size: 17
  - SoftwareRevision: 36
  - ProcessorType: 1756-L85EP
  - PlantPaxTaskingModel\_Enabled: False
- 02 - HMI**
  - Area: /Area::
  - Path: [shortcut]
  - Has\_IsPositioned: False
  - AreaPath: /Area::[shortcut]
  - AreaPathME: [shortcut]
- 03 - Historian**
  - HistorianMachineName:
  - HistorianMachineID:
  - FTVAppName:
  - HistorianPath: Application/Area:RSLink Enterprise:[shortcut]
  - FTLDInterfaceNo: 1
- 04 - Operations**
  - Has\_Redundant: False
  - Has\_ChangeDetect: False
  - Has\_TaskMonitor: True
  - Has\_OOAP: False
  - Has\_HART: True** (highlighted)
  - Has\_EventLogging: False
  - Cfg\_IncludeSystemTag: True
- 05 - Alarm Configuration**
  - AlarmClass: 0
  - Cfg\_HasMajorFaultAlm: True
  - Cfg\_HasTaskMonAlm: True
- 05.03 - Major Fault Alarm**
  - MajorFaultAlarmCommand:
  - Cfg\_MajorFaultAckReqd: True
  - Cfg\_MajorFaultResetReqd: False
  - Cfg\_MajorFaultSeverity: 1000
  - Cfg\_MajorFaultMaxShelfDuration: 480
  - Cfg\_MajorFaultAlarmGroup: [dropdown arrow]
  - Cfg\_MajorFaultShelfDuration: 0

2. When you add the HART instrument, configure the ParentModule Parameter to the 1756-IF8IH module in Rack01.

The screenshot displays the configuration window for the **Rack01\_01\_0\_ProsonicFlow200** module. The **Class View** on the left shows the hierarchy: **Controllers** > **CLX\_scenario\_1** > **Rack01\_01** > **115A-ProsonicFlow\_200** > **Rack01\_01\_0\_ProsonicFlow200**. The main panel shows the following details:

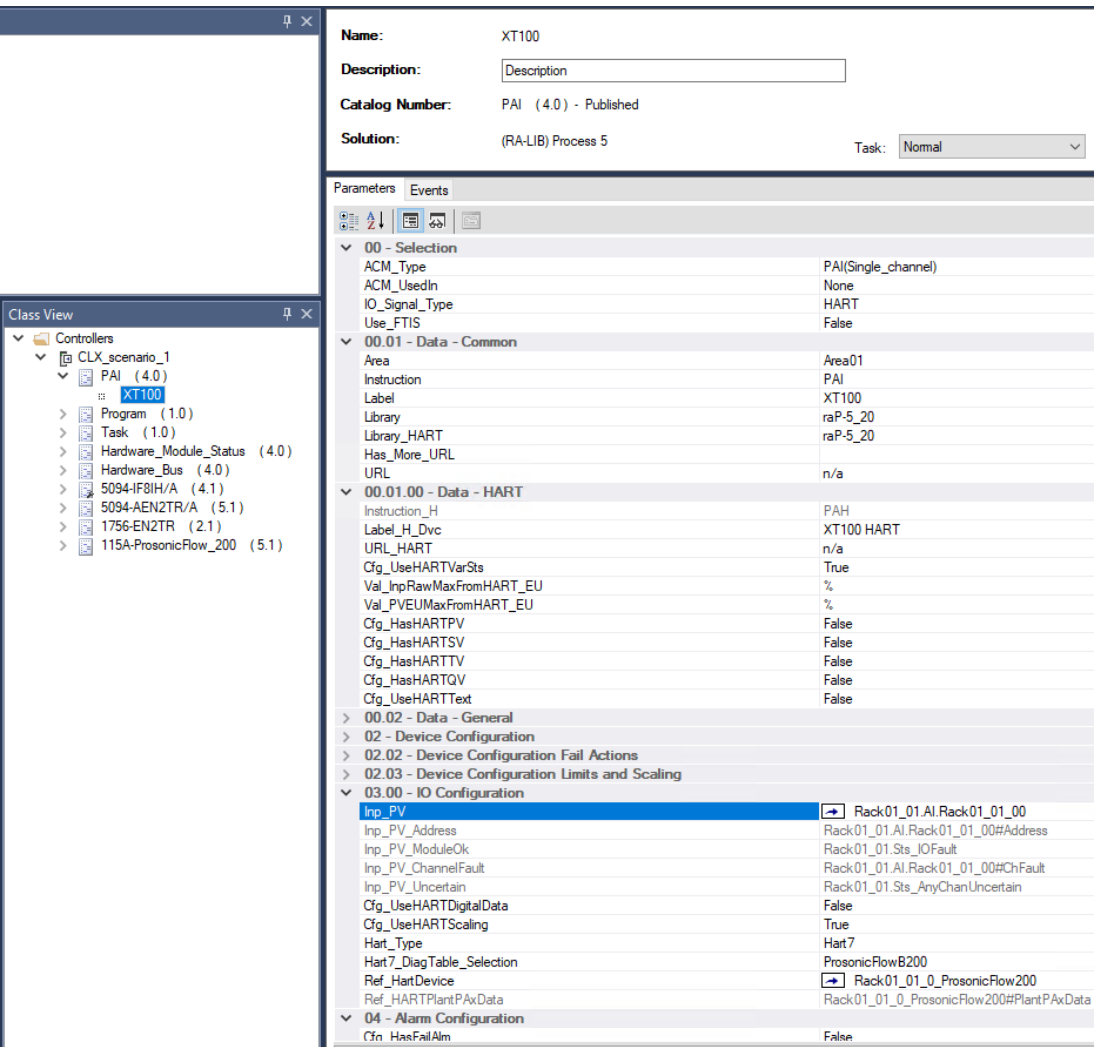
- Name:** Rack01\_01\_0\_ProsonicFlow200
- Description:** ProsonicFlow\_200
- Catalog Number:** 115A-ProsonicFlow\_200 (5.1) - Published
- Solution:** (RA-LIB) ACM 2.00

The **Parameters** section is expanded, showing the **Module Configuration** parameters:

- ParentModule:** Rack01\_01 (selected via dropdown arrow)
- Channel:** 0
- ChassisName:** Rack01\_01\_0\_ProsonicFlow200
- RPI:** 500

3. From the Process library > Control Strategies > Input Processing folder, add a PAI instance for the analog input module and configure the Task and Program.

**IMPORTANT** You must create an individual PAI instance for each input module in your application.



4. Configure these parameters.

ACM Parameter	Description/Value
IO_Signal_Type	HART
Inp_PV	Connect to the channel of the I/O module that is connected to the instrument.
Cfg_ UseHARTDigitalData	Not applicable, leave at default value.
Cfg_UseHARTScaling	Set to True if you want to connect the scaling parameters from the PAH module.
Hart_Type	Select the HART protocol revision (Generic, Hart, Hart5, Hart6 or Hart7).
Hart7_DiagTable_Selection	Select the relevant Diag Table value for the instrument.
Ref_HartDevice	Connect to the instrument.



5. From the Process library > Organization > Bus folder, add a Hardware\_Bus object.

The screenshot displays the software interface for configuring the **HWBus\_Org** object. On the left, the **Class View** pane shows the project hierarchy under **Controllers** > **CLX\_scenario\_1** > **Hardware\_Bus (4.0)**, with **HWBus\_Org** selected. The main pane shows the object's properties:

- Name:** HWBus\_Org
- Description:** Description
- Catalog Number:** Hardware\_Bus (4.0) - Published
- Solution:** (RA-LIB) Process 5
- Task:** System

The **Parameters** tab is active, showing a tree structure of parameters:

- 00.00 - Org**
  - HWOrgViewSize: 4
- 00.01 - Org Scan Data - Common**
  - Scan\_Library: raP-5\_20
  - Scan\_Instruction: raP\_Opr\_OrgScan
  - Scan\_Label: HWOrgScan
  - Scan\_Area: Area01
- 00.02 - Org View Data - Common**
  - View\_Library: raP-5\_20
  - View\_Instruction: raP\_Opr\_OrgView
  - View\_Area: Area01
  - View\_Label: HWOrgScan
  - View\_Area\_01: Area01
  - View\_Label\_01: OrgView
  - View\_Area\_02: Area01
  - View\_Label\_02: OrgView
  - View\_Area\_03: Area01
  - View\_Label\_03: OrgView

6. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object for each module in the project. Give each object a unique instance on the hardware bus.

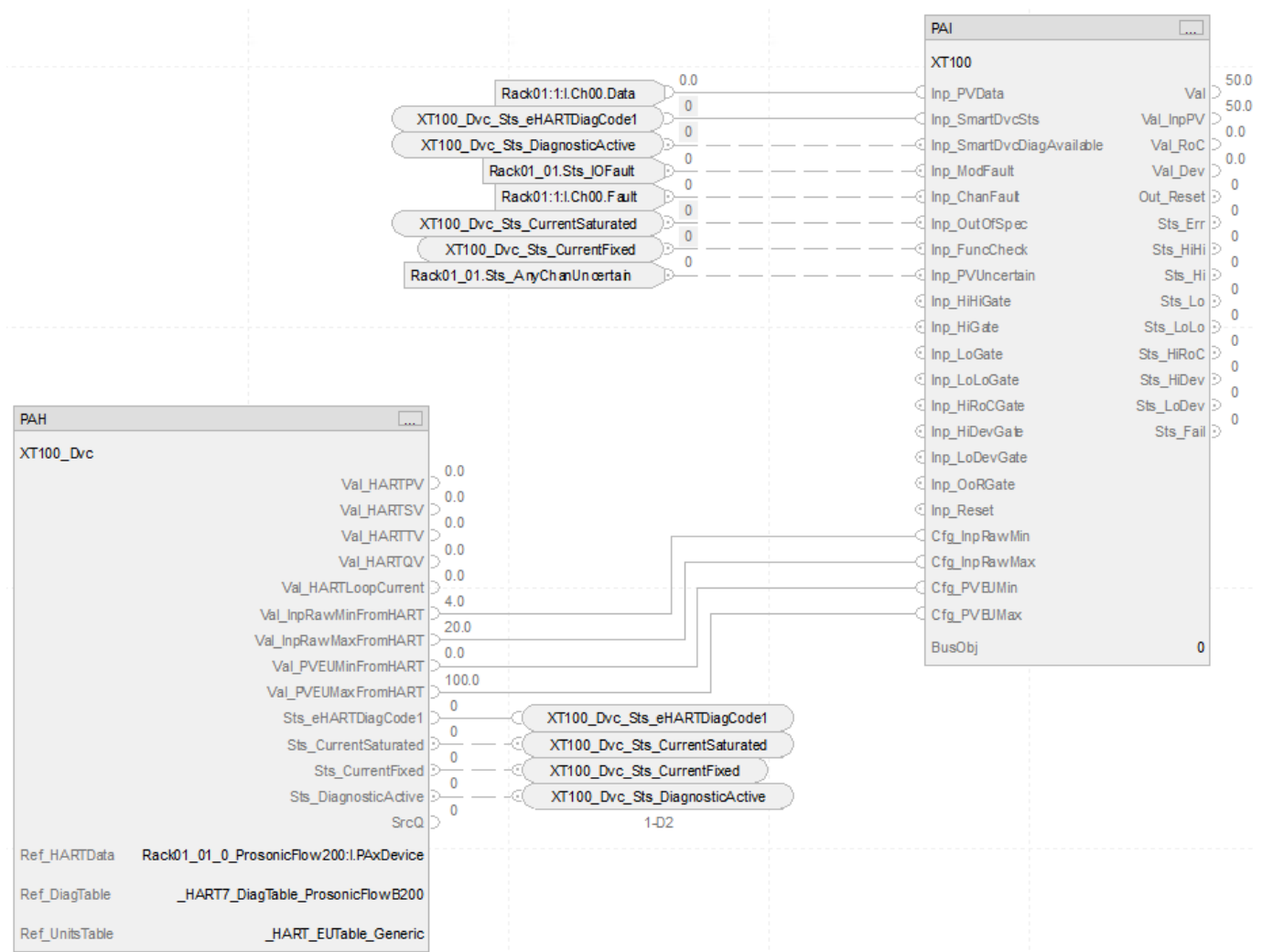
The screenshot displays the software interface for configuring the **HWMS\_Local\_01** object. On the left, the **Class View** pane shows the project hierarchy under **Controllers** > **CLX\_scenario\_1** > **Hardware\_Module\_Status (4.0)**, with **HWMS\_Local\_01** selected. The main pane shows the object's properties:

- Name:** HWMS\_Local\_01
- Description:** This instruction checks the I/O connection status of the given
- Catalog Number:** Hardware\_Module\_Status (4.0) - Published
- Solution:** (RA-LIB) Process 5

The **Parameters** tab is active, showing a tree structure of parameters:

- 00.01 - Data - Common**
  - Area: Area01
  - Instruction: raP\_Dvc\_LgxModuleSts
  - Label: Module Status
  - Library: raP-5\_20
- 01 - Options**
  - Module: Local\_01
  - Bus\_Instance: HWBus\_Org.Bus.Cmd\_1
- 01.01 - Options**
  - SetNumberOfChannels: False
  - ParentModuleName: Local\_1
  - ModuleCatNum: Local\_01#CatNum
  - ModuleSlot: Local\_01?Slot
- 04 - Alarm Configuration**
  - AlarmClass: 0
- 04.01 - Module Fault Alarm**
  - ModuleFaultAlarmCommand: NavToDisplay [ControlStrategies] x "Faceplate" "/RP"
  - Cfg\_ModuleFaultAckReqd: True
  - Cfg\_ModuleFaultResetReqd: False
  - Cfg\_ModuleFaultSeverity: 1000
  - Cfg\_ModuleFaultAlarmGroup:
  - Cfg\_ModuleFaultMaxShelfDuration: 480
  - Cfg\_ModuleFaultShelfDuration: 0
  - Cfg\_ModuleFaultAlarmSetoperations: True
  - Cfg\_ModuleFaultAlarmSetrollupcount: True
  - Cfg\_ModuleFaultDeadband: 0.0
  - Cfg\_ModuleFaultOffDelay: 0
  - Cfg\_ModuleFaultOnDelay: 0

7. Generate the controller ACD file.

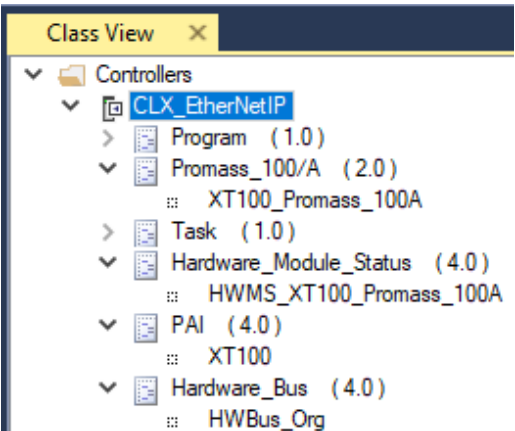


## Integrate FLEX 5000 with HART Device via PV, SV, TV, or QV Values

In this example, the ACM project contains:

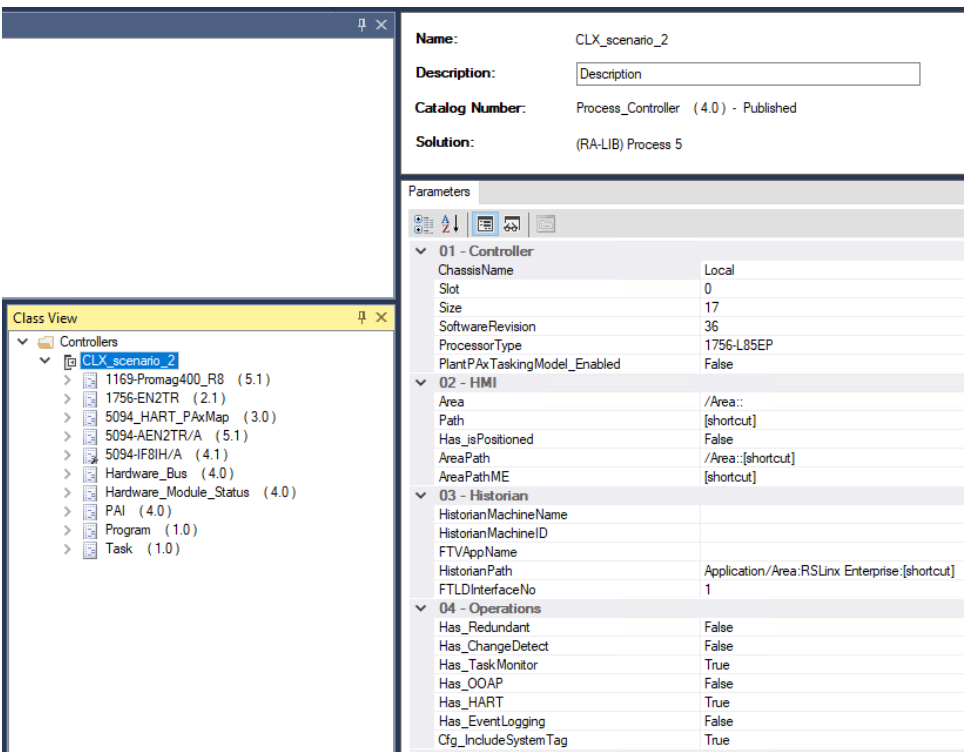
- ControlLogix Process controller
- 1756-EN2TR communication module
- 5094-AEN2TR communication module for FLEX 5000 I/O connectivity
- 5094-IF8IH HART analog input module with an Endress+Hauser
- Promag 400 revision 8 instrument connected to Channel 7

**IMPORTANT** When you add multiple EtherNet/IP communication modules to an ACM project, remember to enter a unique IP address for each module.



Add the devices to the ACM project and configure parameters as needed.

1. Configure the process controller parameters required for your application, and set Has\_HART to True.



2. When you add the HART instrument, configure the ParentModule Parameter to the 1756-IF8IH module in Rack01\_01. Also, change the Connectiontype to PlantPaxData.

The screenshot displays the software interface for configuring a HART instrument. On the left, the 'Class View' pane shows a tree structure under 'Controllers' > 'CLX\_scenario\_2' > '1169-Promag400\_R8 (5.2)'. The 'Rack01\_01\_0\_Promag400R8' module is selected. On the right, the 'Parameters' pane shows the configuration for this module. The 'Name' is 'Rack01\_01\_0\_Promag400R8', 'Description' is 'Promag400\_R8', 'Catalog Number' is '1169-Promag400\_R8 (5.2) - Published', and 'Solution' is '(RA-LIB) ACM 2.00'. Under the 'General' tab, the 'Connectiontype' is set to 'PlantPaxData'. Under the 'Module Configuration' tab, the 'ParentModule' is set to 'Rack01\_01', 'Channel' is '0', 'ChassisName' is 'Rack01\_01\_0\_Promag400R8', and 'RPI' is '500'.

3. From Process library > HART\_Mapping > HART IO Card Mapping, create an instance of the 5094\_HART\_PAxMap and connect to the Promag 400 revision 8 instrument. You must have an instance of the library for each extra signal you use (PV, SV, TV, or QV).

The screenshot displays the software interface for configuring a HART IO Card Mapping module. On the left, the 'Class View' pane shows a tree structure under 'Controllers' > 'CLX\_scenario\_2' > '5094\_HART\_PAxMap (3.0)'. The 'Rack01\_01\_0\_Promag400R8\_HART\_Map' module is selected. On the right, the 'Parameters' pane shows the configuration for this module. The 'Name' is 'Rack01\_01\_0\_Promag400R8\_HART\_Map', 'Description' is 'HART advanced data mapping to 5x PAI object.', 'Catalog Number' is '5094\_HART\_PAxMap (3.0) - Published', and 'Solution' is '(RA-LIB) Process 5'. The 'Task' is set to 'Normal' and the 'Program' is 'NormalProgram'. Under the 'Misc' tab, the 'Ref\_Module' is set to 'Rack01\_01\_0\_Promag400R8'.

4. From the Process Library > Organization > Bus folder, add a Hardware\_Bus object.

The screenshot displays the configuration for the **HWBus\_Org** object. The **Class View** on the left shows the project hierarchy under **Controllers** > **CLX\_scenario\_2**, with **Hardware\_Bus (4.0)** > **HWBus\_Org** selected. The main configuration pane shows the following details:

- Name:** HWBus\_Org
- Description:** [Empty text box]
- Catalog Number:** Hardware\_Bus (4.0) - Published
- Solution:** (RA-LIB) Process 5
- Task:** System

The **Parameters** tab is active, showing a table of parameters:

Parameter Name	Value
00.00 - Org	
HWOrgViewSize	4
00.01 - Org Scan Data - Common	
Scan_Library	raP-5_20
Scan_Instruction	raP_Opr_OrgScan
Scan_Label	HWOrgScan
Scan_Area	Area01
00.02 - Org View Data - Common	
View_Library	raP-5_20
View_Instruction	raP_Opr_OrgView
View_Area	Area01
View_Label	HWOrgScan
View_Area_01	Area01
View_Label_01	OrgView
View_Area_02	Area01
View_Label_02	OrgView
View_Area_03	Area01
View_Label_03	OrgView

5. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object for each module in the project. Give each object a unique instance on the hardware bus.

The screenshot displays the configuration for the **HWMS\_Rack01\_01** object. The **Class View** on the left shows the project hierarchy under **Controllers** > **CLX\_scenario\_2**, with **Hardware\_Module\_Status (3.0)** > **HWMS\_Rack01\_01** selected. The main configuration pane shows the following details:

- Name:** HWMS\_Rack01\_01
- Description:** This instruction checks the I/O connection status
- Catalog Number:** Hardware\_Module\_Status (3.0) - Pending
- Solution:** (RA-LIB) Process 5

The **Parameters** tab is active, showing a table of parameters:

Parameter Name	Value
00.01 - Data - Common	
01 - Options	
Module	Rack01_01
Bus_Instance	
04 - Alarm Configuration	
AlarmClass	0
04.01 - Module Fault Alarm	
ModuleFaultAlarmCommand	NavToDisplay [ControlStrategies] * "Faceplate" "RP"
Clg_ModuleFaultAckReqd	True
Clg_ModuleFaultResetReqd	False
Clg_ModuleFaultSeverity	1000
Clg_ModuleFaultAlarmGroup	
Clg_ModuleFaultMaxShelfDuration	400
Clg_ModuleFaultShelfDuration	0
Clg_ModuleFaultAlarmSetOperations	True
Clg_ModuleFaultAlarmSetRollupcount	True
Clg_ModuleFaultDeadband	0.0
Clg_ModuleFaultOffDelay	0
Clg_ModuleFaultOnDelay	0
04.02 - Any Channel Fault Alarm	
AnyChanFaultAlarmCommand	NavToDisplay [ControlStrategies] * "Faceplate" "RP"

6. From the Process library > Control Strategies > Input Processing folder, add a PAI instance for the analog input module.

Class View

Controllers

CLX\_scenario\_2

1169-Promag400\_R8 (5.1)

1756-EN2TR (2.1)

5094\_HART\_PaXMap (3.0)

5094-AEN2TR/A (5.1)

5094-IF8IH/A (4.1)

Hardware\_Bus (4.0)

Hardware\_Module\_Status (4.0)

PAI (4.0)

**XT100**

Program (1.0)

Task (1.0)

Name: XT100

Description:

Catalog Number: PAI (4.0) - Published

Solution: (RA-LIB) Process 5

Task: Normal

Program: NormalProgram

Parameters

00 - Selection

ACM\_Type PAI(Single\_channel)

ACM\_UsedIn None

IO\_Signal\_Type HART

Use\_FTIS False

00.01 - Data - Common

Area Area01

Instruction PAI

Label XT100

Library raP-5\_20

Library\_HART raP-5\_20

Has\_More\_URL n/a

URL

00.01.00 - Data - HART

Instruction\_H PAH

Label\_H\_Dvc XT100 HART

URL\_HART n/a

Cfg\_UseHARTVarSts True

Val\_InpRawMaxFromHART\_EU %

Val\_PVEUMaxFromHART\_EU %

Cfg\_HasHARTPV False

Cfg\_HasHARTSV True

Cfg\_HasHARTTV False

Cfg\_HasHARTQV False

Val\_HARTSV\_Label SV

Cfg\_UseHARTText False

Val\_HARTSV\_EU %

00.02 - Data - General

02 - Device Configuration

02.02 - Device Configuration Fail Actions

02.03 - Device Configuration Limits and Scaling

03.00 - IO Configuration

Inp\_PV Rack01\_01\_0\_Promag400R8\_HART\_Map.SV.Rack01\_01\_0\_Promag400R8\_SV

Inp\_PV\_Address Rack01\_01\_0\_Promag400R8\_HART\_Map.SV.Rack01\_01\_0\_Promag400R8\_SV#Address

Inp\_PV\_ModuleOk Rack01\_01\_0\_Promag400R8\_HART\_Map.Sts\_IOFault

Inp\_PV\_ChannelFault Rack01\_01\_0\_Promag400R8\_HART\_Map.SV.Rack01\_01\_0\_Promag400R8\_SV#ChFault

Inp\_PV\_Uncertain Rack01\_01\_0\_Promag400R8\_HART\_Map.Sts\_AnyChanUncertain

Cfg\_UseHARTDigitalData True

Ref\_HARTModule Rack01\_01\_0\_Promag400R8

7. Configure these parameters in the I/O Configuration section.

ACM Parameter	Description/Value
Task Program	Assign a Task and Program for the PAI control strategy.
IO_Signal_Type	HART
Cfg_HasHARTxV	Set the referenced input (PV, SV, TV, QV) to true as needed.
Inp_PV	Set this reference to the PV, SV, TV, or QV of the 50.94_HART_MapIo object that was created for the HART device
Cfg_UseHARTDigitalData	Set to True.
Ref_HARTModule	Set this reference to the 5094 module that is connected to the instrument.

This example shows the SV value as the selection for the Inp\_PV connection.

Select a Reference

Project1 - ReferenceManual\_HART

History

HMI

Controllers

CLX\_scenario\_2

1169-Promag400\_R8 (5.1)

1756-EN2TR (2.1)

5094\_HART\_PaXMap (3.0)

**5094-AEN2TR/A (5.1)**

5094-IF8IH/A (4.1)

Rack01\_01

Hardware\_Bus (4.0)

Hardware\_Module\_Status (4.0)

PAI (4.0)

Program (1.0)

Task (1.0)

Controller

Type

Object Name

Object Description

CLX\_scenario\_2

HART\_Mapping

Rack01\_01\_0\_Promag400R8\_HART\_Map

HART advanced data mapping to 5x PAI object.

Object References

Parameters

PV

SV

TV

QV

Name

Referenced By

Referenced By Description

Channel

SubQ

Rack01\_01\_0\_Promag400R8\_SV

XT100.Inp\_PV

Description

0

SubObject References

SubParameters

Name

Scope

Val

Description

DType

CLXDes

Address

SV

Rack01\_01\_0\_Promag400R8.HARTDevice.SV.Ch.Data

String

true

ChFault

SV

Rack01\_01\_0\_Promag400R8.HARTDevice.SV.Ch.Fault

String

true

Cancel

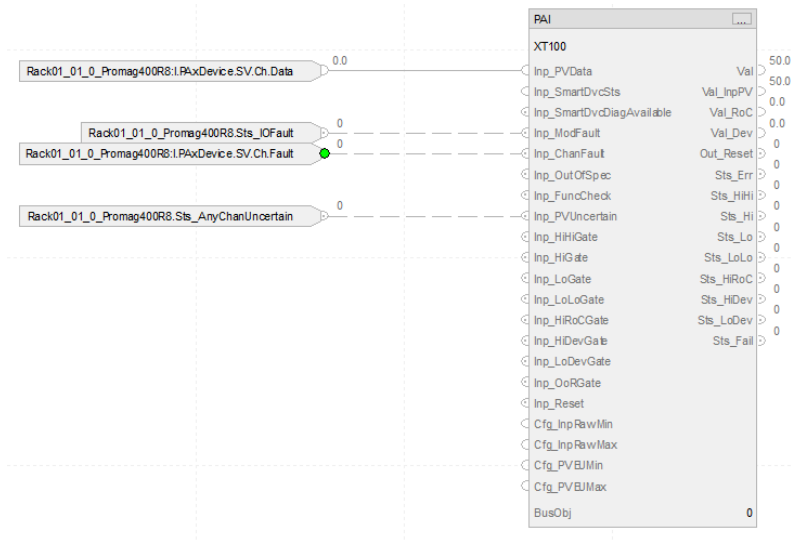
<< Previous

Next >>

Finish

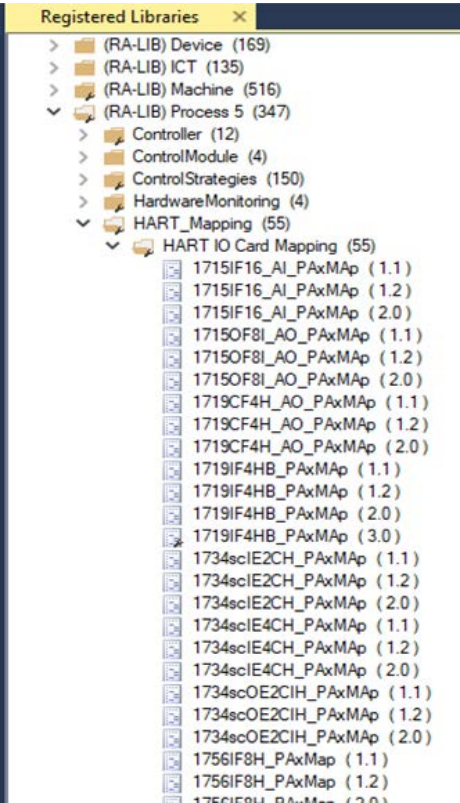
Selected Item: Rack01\_01\_0\_Promag400R8\_SV

8. Generate the controller ACD file.



# Integrate Other HART Modules with the Process Controller

HART modules for other I/O platforms must be used with PlantPax 5.0 instructions in a different way than FLEX 5000 modules. There is a HART-mapping ACM library for each HART I/O module in the HART\_Mapping > HART IO Card Mapping folder.



Each HART-mapping library has these features.

- For each I/O Module, you can connect to a HART device that is connected to each channel.
- The HART device information is mapped into a standard data type `PAX_HART_DEVICE::I:0`

➤ Rack1794_02_HART_Map_HARTDevice0	(...)	(...)	PAX_HART_DEVICE1:0
Rack1794_02_HART_Map_HARTDevice0.RunMode	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.ConnectionFaulted	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.DiagnosticActive	0	Decimal	BOOL
➤ Rack1794_02_HART_Map_HARTDevice0.DiagnosticSequenceCount	0	Decimal	SINT
Rack1794_02_HART_Map_HARTDevice0.CurrentSaturated	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.CurrentFixed	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.MoreStatusAvailable	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.CurrentMismatch	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.ConfigurationChanged	0	Decimal	BOOL
Rack1794_02_HART_Map_HARTDevice0.Malfunction	0	Decimal	BOOL
➤ Rack1794_02_HART_Map_HARTDevice0.LoopCurrent	(...)	(...)	CHANNEL_AI1:0
➤ Rack1794_02_HART_Map_HARTDevice0.PV	(...)	(...)	CHANNEL_AI_HART1:0
➤ Rack1794_02_HART_Map_HARTDevice0.SV	(...)	(...)	CHANNEL_AI_HART1:0
➤ Rack1794_02_HART_Map_HARTDevice0.TV	(...)	(...)	CHANNEL_AI_HART1:0
➤ Rack1794_02_HART_Map_HARTDevice0.QV	(...)	(...)	CHANNEL_AI_HART1:0
➤ Rack1794_02_HART_Map_HARTDevice0.Static	(...)	(...)	AB5000_HART_Static_Struct1:0
Rack1794_02_HART_Map_HARTDevice0.ChDataAtSignal4	0.0	Float	REAL
Rack1794_02_HART_Map_HARTDevice0.ChDataAtSignal20	0.0	Float	REAL

- For each channel, you can map any of the HART Digital Variables (PV, SV, TV, QV) to a PAI module.
- For each Channel of the HART module, you can connect to either the Device, PV, SV, TV, and QV (displayed as sub-objects for each mapping library).

**Name:** Rack\_1719\_01\_HART\_Map

**Description:**

**Catalog Number:** SSB\_1719-CF4H/A\_wMap (1.2) - Pending

**Solution:** (SSB) Process 4.0

---

Parameters    Device    TV    PV    QV    SV

- Misc**
  - Unicast
  - ACM\_Type
  - Cfg\_CH1\_UseHART
  - Cfg\_CH2\_UseHART**
  - Cfg\_CH3\_UseHART
  - Cfg\_CH4\_UseHART
  - Ref\_Module
  - Ref\_Module\_Chassis
  - Ref\_Module\_Slot
- Module Configuration**
  - Slot
  - RPI

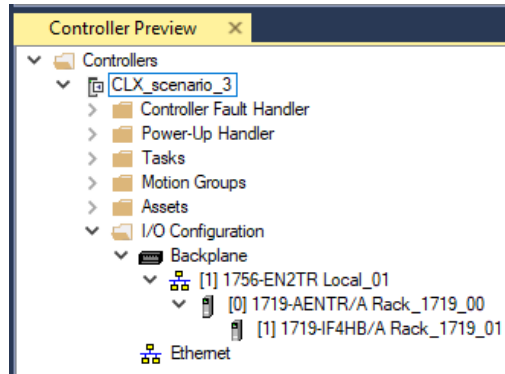
Unicast  
 Analog Input  
 True  
 True  
 False  
 False  
 Rack\_1719\_01  
 Rack\_1719\_01?ChassisName  
 Rack\_1719\_01?Slot  
 1  
 150



## Map HART Device to PAH from Non-FLEX 5000 I/O

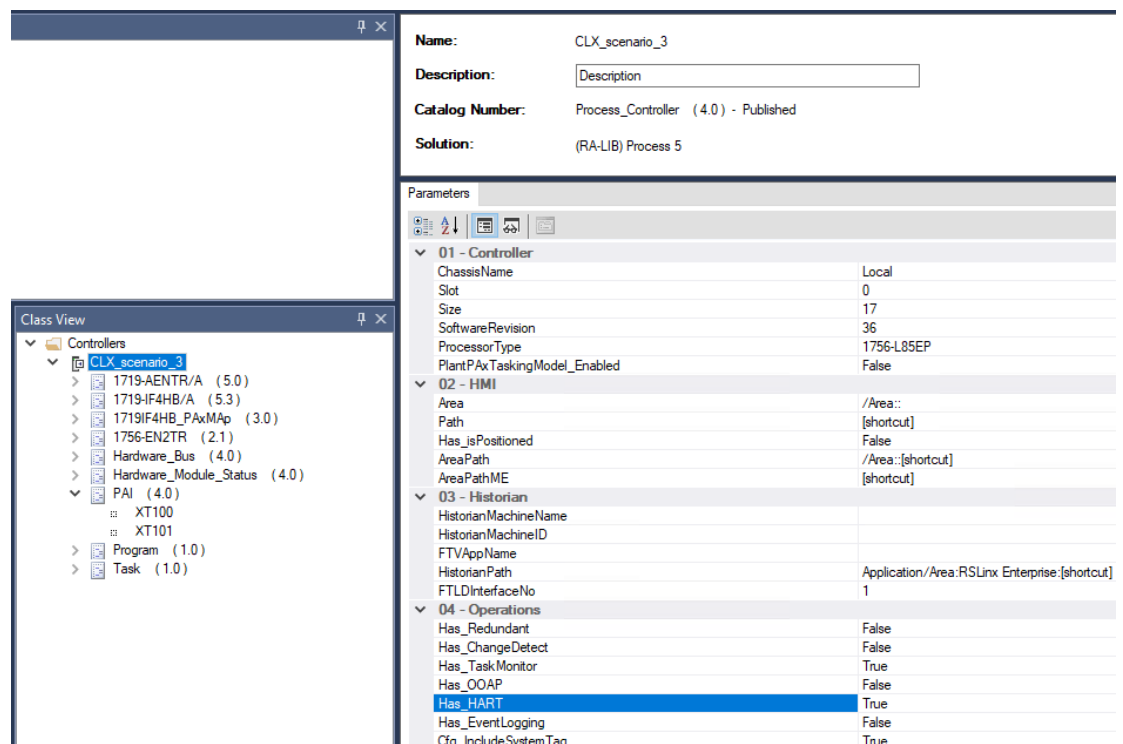
In this example, the ACM project contains:

- ControlLogix Process controller
- 1756-EN2TR communication module
- 1719-AENTR communication module connected to a 1719-IF4HB HART module
- Endress+Hauser
- ProsonicFlow 200 instrument connected to channel 1 of the 1719-IF4HB module
- Endress+Hauser
- Promag revision 9 instrument connected to channel 4 of the 1719-IF4HB module



**IMPORTANT** When you add the 1719-AENTR module to the ACM project, specify a unique rack name and IP address for the module.

1. Configure the process controller for parameters you need for your application, and set `Has_HART` to `True`.



2. From Process library > HART\_Mapping > HART IO Card Mapping, create an instance of the 1719-IF4HB\_PAxMap and connect to the 1719-IF4HB module.

**Name:** Rack\_1719\_00\_01\_HART\_Map

**Description:** HART data mapping of legacy IO to 5 x PAI object.

**Catalog Number:** 1719IF4HB\_PAxMap (3.0) - Published

**Solution:** (RA-LIB) Process 5

**Task:** Normal

**Program:** NormalProgram

**Parameters** | Device | SV | PV | TV | QV

**Module Configuration**

RPI	150
ChassisName	Rck_1719
Unicast	Unicast
Cfg_CH1_UseHART	True
Cfg_CH2_UseHART	False
Cfg_CH3_UseHART	False
Cfg_CH4_UseHART	True
Ref_Module	Rack_1719_00_01
Ref_Module_Chassis	Rack_1719_00_01?ChassisName
Ref_Module_Slot	Rack_1719_00_01?Slot

**Class View**

- Controllers
  - CLX\_scenario\_3
    - 1719-AENTR/A (5.0)
    - 1719-IF4HB/A (5.3)
      - 1719IF4HB\_PAxMap (3.0)
        - Rack\_1719\_00\_01\_HART\_Map**
    - 1756-EN2TR (2.1)
    - Hardware\_Bus (4.0)
    - Hardware\_Module\_Status (4.0)
    - PAI (4.0)
      - XT100
      - XT101
    - Program (1.0)
    - Task (1.0)

3. Configure these parameters.

ACM Parameter	Description
Cfg_CH1_UseHART	Set to True if a HART device is connected to this channel.
Cfg_CH2_UseHART	Set to True if a HART device is connected to this channel.
Cfg_CH3_UseHART	Set to True if a HART device is connected to this channel.
Cfg_CH4_UseHART	Set to True if a HART device is connected to this channel.
Ref_Module	Select the HART I/O module.

4. Change the 1719-IF4HB ChassisName to the 1719-AENTR name.

**Name:** Rack\_1719\_00\_01

**Description:** 1719 4 Channel HART Analog Input Wide

**Catalog Number:** 1719-IF4HB/A (5.3) - Published

**Solution:** (RA-LIB) ACM 2.00

**Parameters** | AI

**Channel Descriptions**

Ch1Description	Input Ch1
Ch2Description	Input Ch2
Ch3Description	Input Ch3
Ch4Description	Input Ch4

**General**

FPLaunch_Button_Label	Rack_1719_01
-----------------------	--------------

**HMI Configuration**

NavigationButton	GraphicalButton
SEAssocDisplay	SEAssocDisplay
MEAssocDisplay	MEAssocDisplay

**Module Configuration**

Slot	1
RPI	150
ChassisName	Rack_1719_00

**Class View**

- Controllers
  - CLX\_scenario\_3
    - 1719-AENTR/A (5.0)
    - 1719-IF4HB/A (5.3)
      - 1719IF4HB\_PAxMap (3.0)
        - Rack\_1719\_00\_01**
    - 1756-EN2TR (2.1)
    - Hardware\_Bus (4.0)
    - Hardware\_Module\_Status (4.0)
    - PAI (4.0)
    - Program (1.0)
    - Task (1.0)

5. From the Process library > Organization > Bus folder, add a Hardware\_Bus object.

The screenshot shows the configuration for the **HWBus\_Org** object. The **Class View** on the left shows the object is added under **Hardware\_Bus (4.0)**. The main configuration pane shows the following details:

- Name:** HWBus\_Org
- Description:** [Empty field]
- Catalog Number:** Hardware\_Bus (4.0) - Published
- Solution:** (RA-LIB) Process 5
- Task:** System

The **Parameters** tab is active, showing a tree structure of parameters:

- 00.00 - Org**
  - HWOrgViewSize: 4
- 00.01 - Org Scan Data - Common**
  - Scan\_Library: raP-5\_20
  - Scan\_Instruction: raP\_Opr\_OrgScan
  - Scan\_Label: HWOrgScan
  - Scan\_Area: Area01
- 00.02 - Org View Data - Common**
  - View\_Library: raP-5\_20
  - View\_Instruction: raP\_Opr\_OrgView
  - View\_Area: Area01
  - View\_Label: HWOrgScan
  - View\_Area\_01: Area01
  - View\_Label\_01: OrgView
  - View\_Area\_02: Area01
  - View\_Label\_02: OrgView
  - View\_Area\_03: Area01
  - View\_Label\_03: OrgView

6. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object for each module in the project. Give each object a unique instance on the hardware bus.

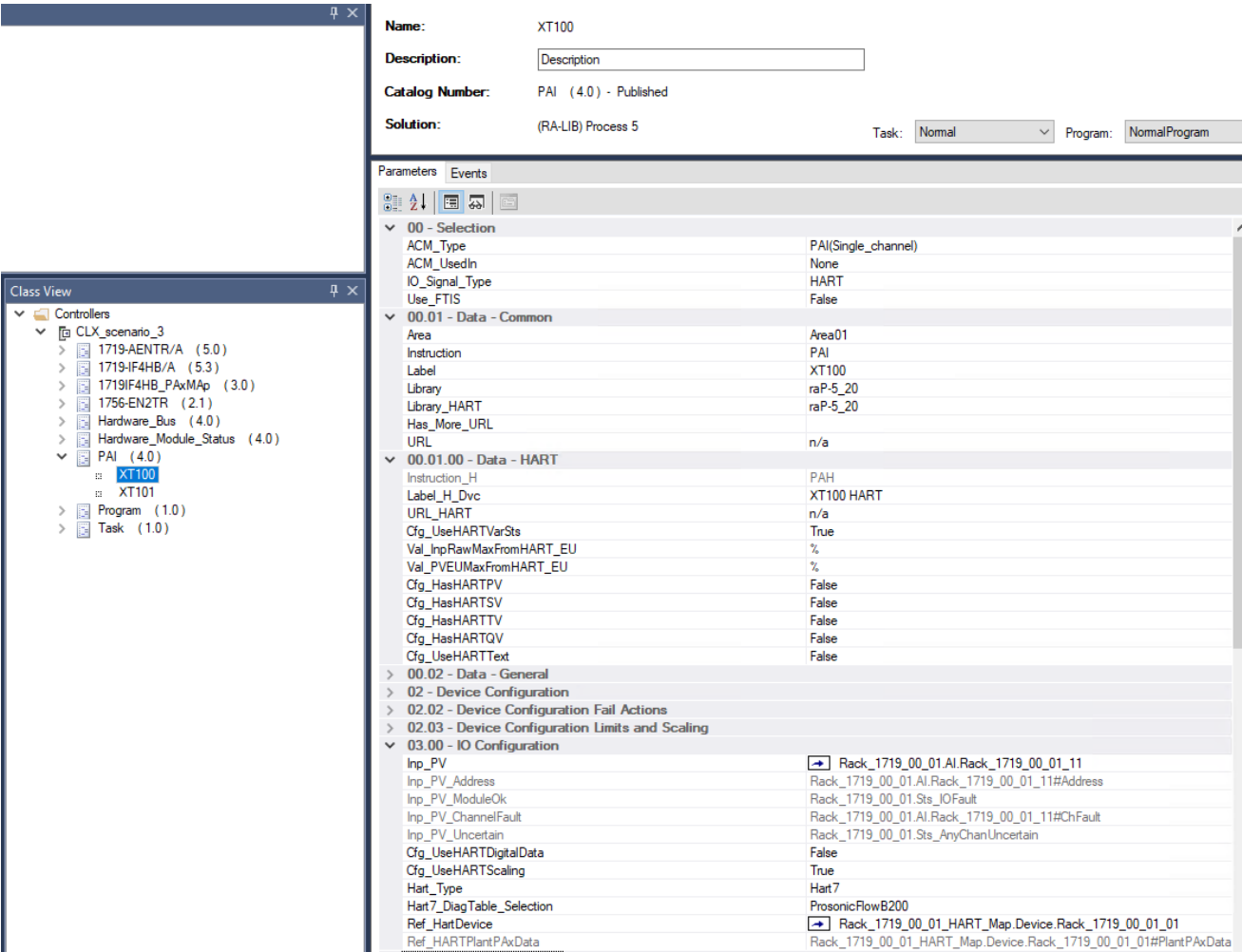
The screenshot shows the configuration for the **HWMS\_Local\_01** object. The **Class View** on the left shows the object is added under **Hardware\_Module\_Status (4.0)**. The main configuration pane shows the following details:

- Name:** HWMS\_Local\_01
- Description:** This instruction checks the I/O connection status of the given
- Catalog Number:** Hardware\_Module\_Status (4.0) - Published
- Solution:** (RA-LIB) Process 5

The **Parameters** tab is active, showing a tree structure of parameters:

- 00.01 - Data - Common**
  - Area: Area01
  - Instruction: raP\_Dvc\_LgxModuleSts
  - Label: Module Status
  - Library: raP-5\_20
- 01 - Options**
  - Module: Local\_01
  - Bus\_Instance: HWBus\_Org.Bus.Cmd\_1
- 01.01 - Options**
  - SetNumberOfChannels: False
  - ParentModuleName: Local\_1
  - ModuleCatNum: Local\_01HCatNum
  - ModuleSlot: Local\_01Slot
- 04 - Alarm Configuration**
  - AlarmClass: 0
- 04.01 - Module Fault Alarm**
  - ModuleFaultAlarmCommand: Nav ToDisplay [ControlStrategies] x "F
  - Cfg\_ModuleFaultAckReqd: True
  - Cfg\_ModuleFaultResetReqd: False
  - Cfg\_ModuleFaultSeverity: 1000
  - Cfg\_ModuleFaultAlarmGroup: [Empty field]
  - Cfg\_ModuleFaultMaxShelfDuration: 480

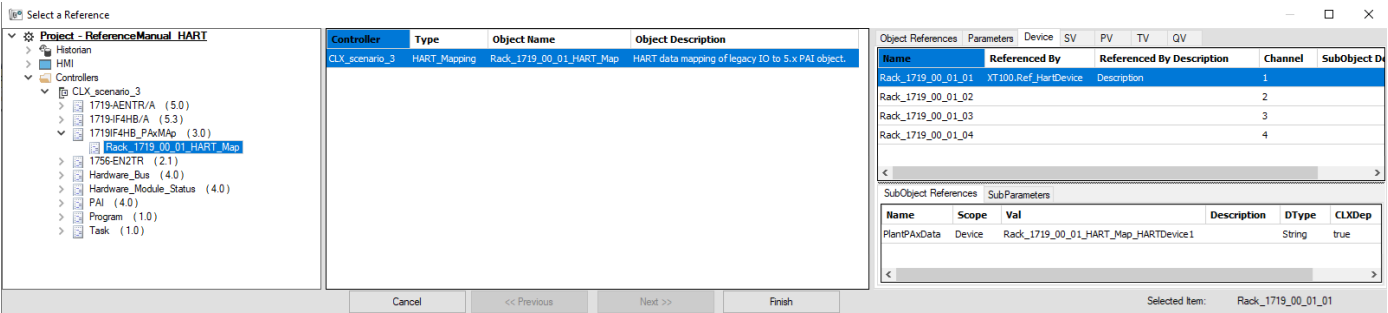
7. From the Process library > Control Strategies > Input Processing folder, add a PAI instance for each instrument.



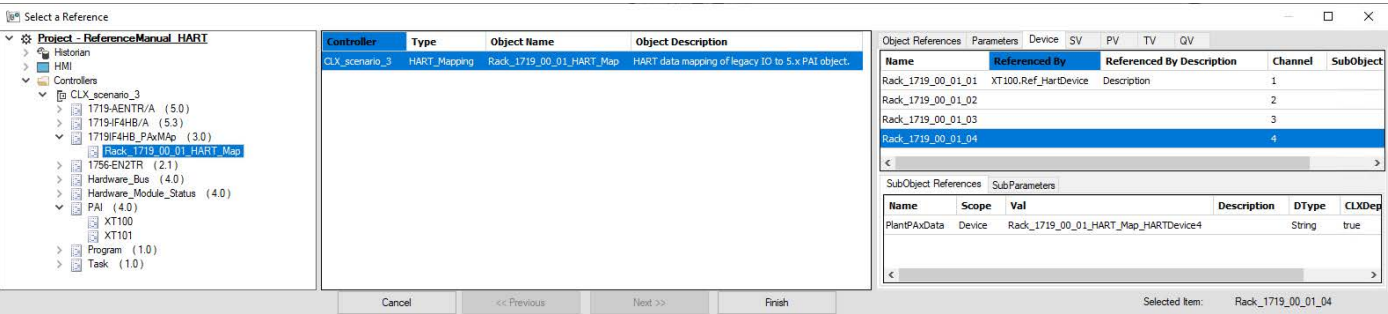
8. Configure these parameters in the I/O Configuration section.

ACM Parameter	Description/Value
Task Program	Assign a Task and Program for each PAI control strategy.
IO_Signal	HART
Cfg_HasHARTxV	Set the referenced input (PV, SV, TV, QV) to true as needed.
Inp_PV	Connect to the channel of the I/O module that is connected to the instrument.
Cfg_UseHARTDigitalData	Not applicable, leave at default value.
Hart_Type	Select the HART protocol revision (Generic, Hart, Hart5, Hart6 or Hart7).
Cfg_UseHARTScaling	Set to False.
Hart7_DiagTable_Selection	Select the relevant Diag Table value for the instrument.
Ref_HartDevice	Connect to the mapping library in ACM and on the Device tab select the correct channel.

This example of the first PAI instance (XT100) shows the Ref\_HartDevice for the Prosonic 200 instrument connected to channel 1 of the 1719-IF4HB module.

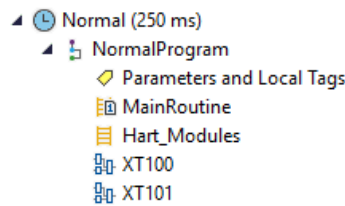


This example of the second PAI instance (XT101) shows the Promag revision 9 instrument connected to channel 4 of the 1719-IF48H module.



9. Generate the controller ACD file.

The controller code contains a routine for each HART instrument and a Hart\_Modules routine.

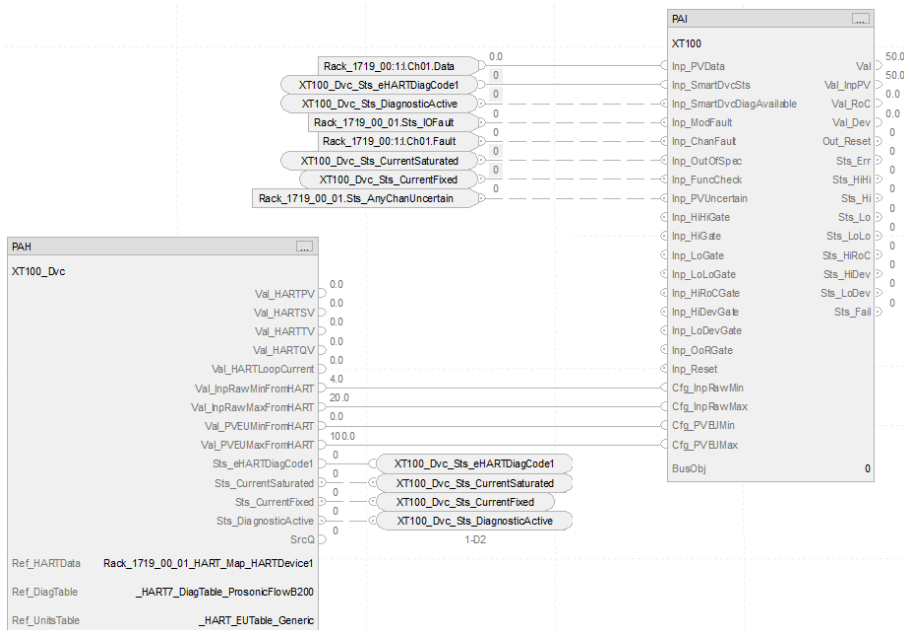


These actions occur in the Hart\_Modules routine.

- An Add-On Instruction is inserted which gets the data from the HART Module (in this case I\_1718\_1719\_AI4H).
- The HART data is mapped into an array for each device. (Rack\_1719\_01\_HART\_Map\_HARTCH1 and Rack\_1719\_01\_HART\_Map\_HARTCH1 CH4).
- Each element of the array is mapped into a tag of type PAX\_HART\_DEVICE:I:0 via the Add-On Instruction rap\_Tec\_HARTChanData\_to\_PAH.



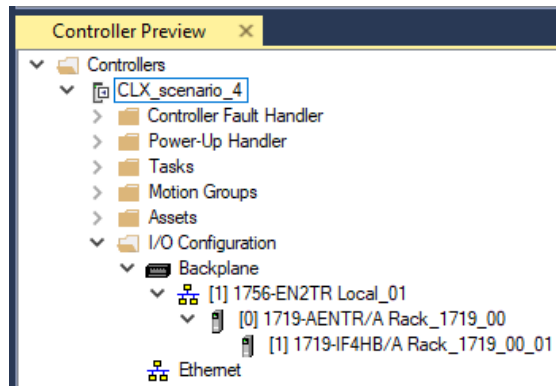
The instrument logic has no connection between the PAH module and the PAI module as the ACM parameter Cfg\_UseHARTScaling is set to False.



## Map HART Device Digital Data to PAI from Non-FLEX 5000 I/O

In this example, the ACM project contains:

- ControlLogix Process controller
- 1756-EN2TR communication module
- 1719-AENTR communication module connected to a 1719-IF4HB HART module
- Endress+Hauser
- ProsonicFlow 200 instrument connected to channel 2 of the 1719-IF4HB module



Map one of the HART digital signals (PV, SV, TV, QV) to a PAI Module. The TV of the HART device that is connected to Channel 2 of the 1719-IF4HB is connected to the PAI module.

**IMPORTANT** When you add the 1719-AENTR module to the ACM project, specify a unique rack name and IP address for the module.

Add the devices to the ACM project and configure the parameters as needed.

1. Configure the process controller for parameters that are required for your application, and set Has\_HART to True.

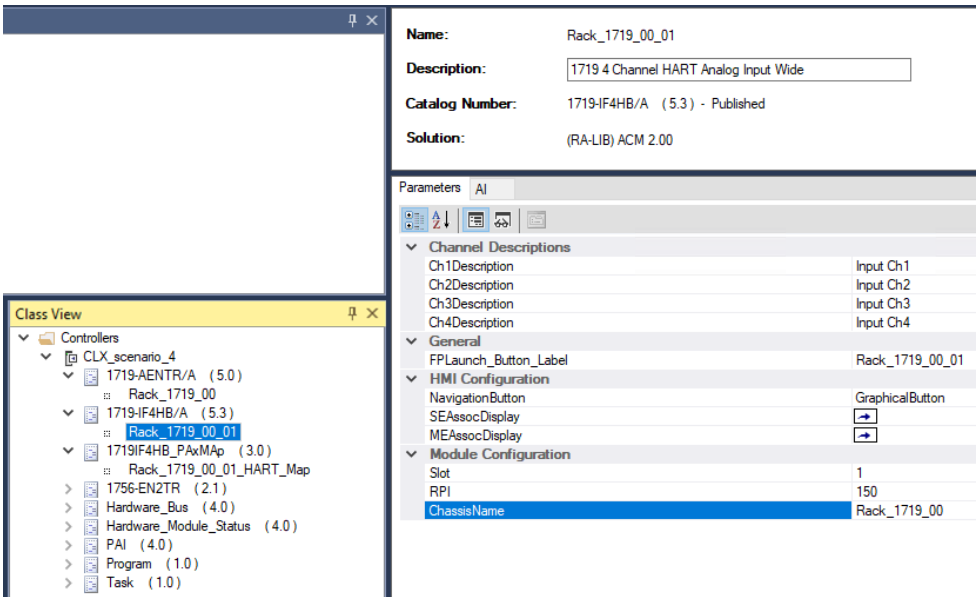
**Class View**

- Controllers
  - CLX\_scenario\_4
    - 1719-AENTR/A (5.0)
    - 1719-IF4HB/A (5.3)
    - 1719-IF4HB\_PaMap (3.0)
    - 1756-EN2TR (2.1)
    - Hardware\_Bus (4.0)
    - Hardware\_Module\_Status (4.0)
    - PAI (4.0)
    - Program (1.0)
    - Task (1.0)

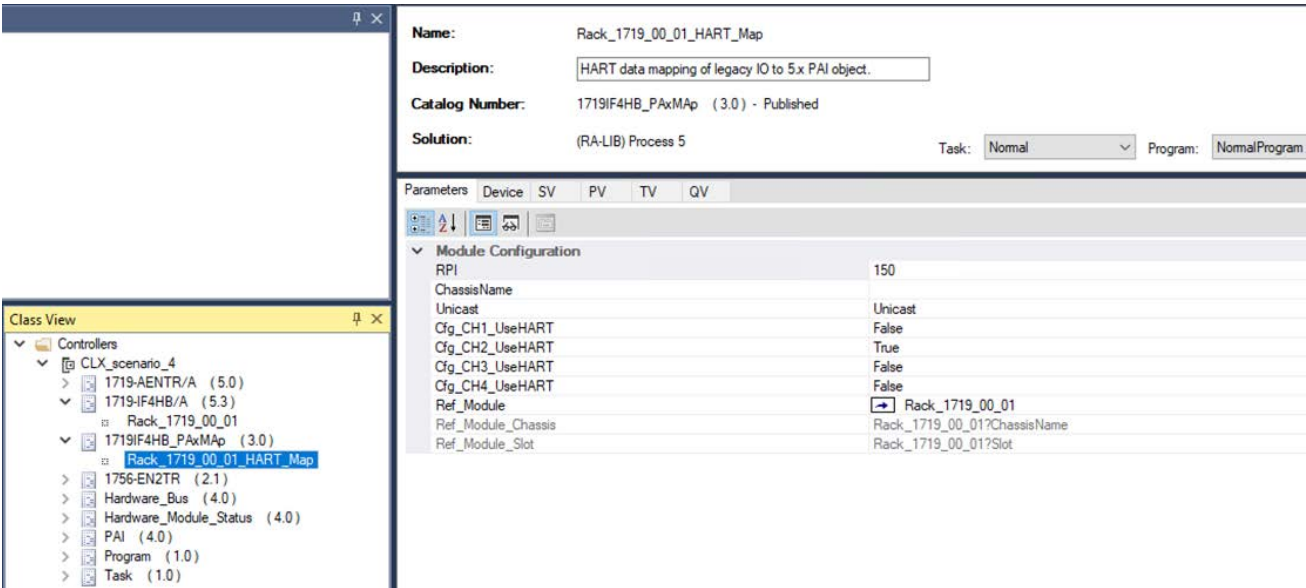
**Parameters**

Section	Parameter	Value
01 - Controller	ChassisName	Local
	Slot	0
	Size	17
	SoftwareRevision	36
	ProcessorType	1756-L85EP
	PlantPaxTaskingModel_Enabled	False
02 - HMI	Area	/Area::
	Path	[shortcut]
	Has_IsPositioned	False
	AreaPath	/Area::[shortcut]
	AreaPathME	[shortcut]
03 - Historian	HistorianMachineName	
	HistorianMachineID	
	FTVAppName	
	HistorianPath	Application/Area:RSLink Enterprise:[shortcut]
	FTLInterfaceNo	1
04 - Operations	Has_Redundant	False
	Has_ChangeDetect	False
	Has_TaskMonitor	True
	Has_OOAP	False
	Has_HART	True
	Has_EventLogging	False
	Cfg_IncludeSystem Tag	True

2. Change the 1719-IF4HB ChassisName to the 1719-AENTR name.



3. From Process library > HART\_Mapping > HART IO Card Mapping, create an instance of the 1719-IF4HB\_PAxMap and connect to the 1719-IF4HB module.
- You must have an instance of the library for each extra signal you use (PV, SV, TV, or QV).



4. Configure these parameters.

ACM Parameter	Usage
Cfg_CH1_UseHART	Set to True if a HART device is connected to this channel
Cfg_CH2_UseHART	Set to True if a HART device is connected to this channel
Cfg_CH3_UseHART	Set to True if a HART device is connected to this channel
Cfg_CH4_UseHART	Set to True if a HART device is connected to this channel
Ref_Module	Select the HART I/O module



- From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object for each module in the project. Give each object a unique instance on the hardware bus.

The screenshot displays the Rockwell Automation software interface with three main panels:

- Class View (Left):** Shows a hierarchical tree of objects. Under 'Controllers' > 'CLX\_scenario\_4', the 'Hardware\_Module\_Status (4.0)' object is expanded, showing its instance 'HWMS\_Local\_01'.
- Properties Window (Top Right):** Displays the configuration for 'HWMS\_Local\_01'.
  - Name:** HWMS\_Local\_01
  - Description:** This instruction checks the I/O connection status of the given
  - Catalog Number:** Hardware\_Module\_Status (4.0) - Published
  - Solution:** (RA-LIB) Process 5
- Parameters Window (Bottom Right):** Shows a list of parameters for the 'Hardware\_Module\_Status' object.
  - 00.01 - Data - Common:**
    - Area: Area01
    - Instruction: raP\_Dvc\_LgxModuleSts
    - Label: Module Status
    - Library: raP-5\_20
  - 01 - Options:**
    - Module: Local\_01
    - Bus\_Instance: HWBus\_Org.Bus.Cmd\_1
  - 01.01 - Options:**
    - SetNumberOfChannels: False
    - ParentModuleName: Local\_1
    - ModuleCatNum: Local\_01#CatNum
    - ModuleSlot: Local\_01?Slot
  - 04 - Alarm Configuration:**
    - AlarmClass: 0
  - 04.01 - Module Fault Alarm:**
    - ModuleFaultAlarmCommand: NavToDisplay [ControlStrategies] x "Faceplate" "/RP"
    - Cfg\_ModuleFaultAckReqd: True
    - Cfg\_ModuleFaultResetReqd: False
    - Cfg\_ModuleFaultSeverity: 1000
    - Cfg\_ModuleFaultAlarmGroup: [Selection Icon]
    - Cfg\_ModuleFaultMaxShelfDuration: 480
    - Cfg\_ModuleFaultShelveDuration: 0
    - Cfg\_ModuleFaultAlarmSetoperations: True
    - Cfg\_ModuleFaultAlarmSetrollupcount: True
    - Cfg\_ModuleFaultDeadband: 0.0
    - Cfg\_ModuleFaultOffDelay: 0
    - Cfg\_ModuleFaultOnDelay: 0

6. From the Process library > Control Strategies > Input Processing folder, add a PAI instance for each instrument and configure these parameters in the I/O Configuration section.

Class View

Controllers

CLX\_scenario\_4

1719-AENTR/A (5.0)

1719-IF4HB/A (5.3)

1719IF4HB\_PAxMap (3.0)

1756-EN2TR (2.1)

Hardware\_Bus (4.0)

Hardware\_Module\_Status (4.0)

PAI (4.0)

XT100

Program (1.0)

Task (1.0)

Name: XT100

Description: Description

Catalog Number: PAI (4.0) - Published

Solution: (RA-LIB) Process 5

Task: Normal

Program: NormalProgram

Parameters

Events

00 - Selection

ACM\_Type PAI(Single\_channel)

ACM\_UsedIn None

IO\_Signal\_Type HART

Use\_FTIS False

00.01 - Data - Common

Area Area01

Instruction PAI

Label XT100

Library raP-5\_20

Library\_HART raP-5\_20

Has\_More\_URL n/a

URL n/a

00.01.00 - Data - HART

Instruction\_H PAH

Label\_H\_Dvc XT100 HART

URL\_HART n/a

Cfg\_UseHARTVarSts True

Val\_InpRawMaxFromHART\_EU %

Val\_PVEUMaxFromHART\_EU %

Cfg\_HasHARTPV False

Cfg\_HasHARTSV False

Cfg\_HasHARTTV True

Cfg\_HasHARTQV False

Val\_HARTTV\_Label TV

Cfg\_UseHARTText False

Val\_HARTTV\_EU %

00.02 - Data - General

02 - Device Configuration

02.02 - Device Configuration Fail Actions

02.03 - Device Configuration Limits and Scaling

03.00 - IO Configuration

Inp\_PV Rack\_1719\_00\_01\_HART\_Map.TV.Rack\_1719\_00\_01\_02

Inp\_PV\_Address Rack\_1719\_00\_01\_HART\_Map.TV.Rack\_1719\_00\_01\_02#Address

Inp\_PV\_ModuleOk Rack\_1719\_00\_01\_HART\_Map.Sts\_IOFault

Inp\_PV\_ChannelFault Rack\_1719\_00\_01\_HART\_Map.TV.Rack\_1719\_00\_01\_02#ChFault

Inp\_PV\_Uncertain Rack\_1719\_00\_01\_HART\_Map.Sts\_AnyChanUncertain

Cfg\_UseHARTDigitalData True

Ref\_HARTModule Rack\_1719\_00\_01

7. Configure these parameters.

ACM Parameter	Usage
Task	Assign a Task and Program for the PAI control strategy.
Program	
IO_Signal	HART
Inp_PV	Set this reference to the PV, SV, TV, or QV of the HART_MapIO object that was created for the HART device
Cfg_ UseHARTDigitalData	Set to True
Ref_HARTModule	Set this reference to the HART I/O module to which the instrument is connected.

This example shows the TV value as the selection for the Inp\_PV connection.

Select a Reference

Project - ReferenceManual\_HART

Historian

HMI

Controllers

CLX\_scenario\_4

1719-AENTR/A (5.0)

1719-IF4HB/A (5.3)

Rack\_1719\_00\_01

1719IF4HB\_PAxMap (3.0)

Rack\_1719\_00\_01\_HART\_Map

1756-EN2TR (2.1)

Hardware\_Bus (4.0)

Hardware\_Module\_Status (4.0)

PAI (4.0)

Program (1.0)

Task (1.0)

Controller

Type

Object Name

Object Description

CLX\_scenario\_4

HART\_Mapping

Rack\_1719\_00\_01\_HART\_Map

HART data mapping of legacy IO to 5.x PAI object.

Object References

Parameters

Device

SV

PV

TV

QV

Name

Referenced By

Referenced By Description

Channel

SubObject

Rack\_1719\_00\_01\_01

1

Rack\_1719\_00\_01\_02

XT100.Inp\_PV

Description

2

Rack\_1719\_00\_01\_03

3

Rack\_1719\_00\_01\_04

4

SubObject References

SubParameters

Name

Scope

Val

Description

DType

Address

TV

Rack\_1719\_00\_01\_HART\_Map\_HARTDevice2.TV.Ch.Data

String

ChFault

TV

Rack\_1719\_00\_01\_HART\_Map\_HARTDevice2.TV.Ch.Fault

String

Cancel

<< Previous

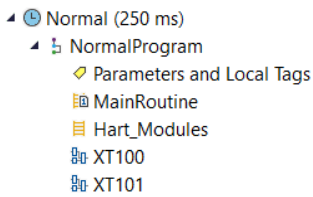
Next >>

Finish

Selected Item: Rack\_1719\_00\_01\_02

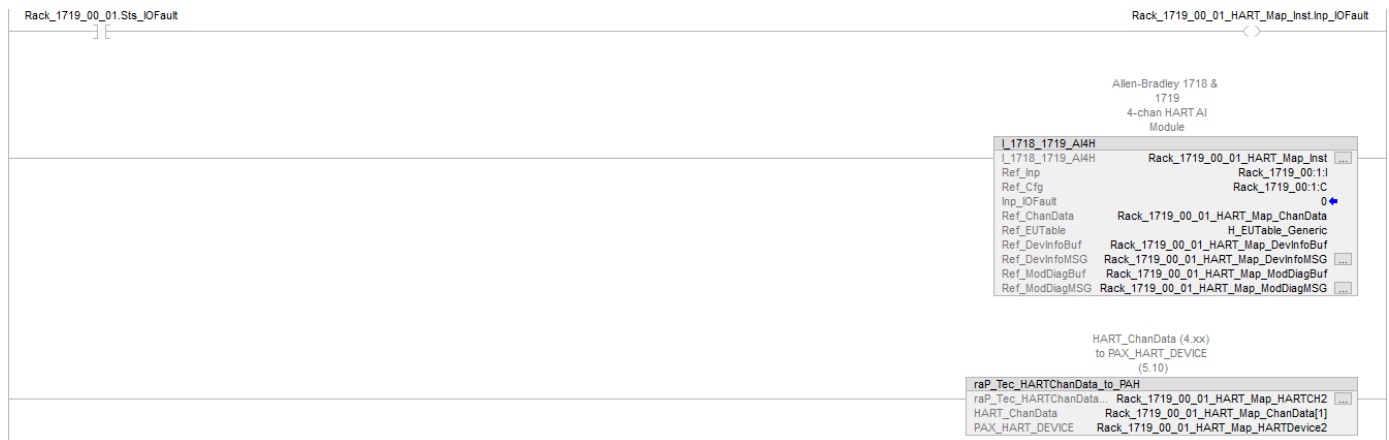
8. Generate the controller ACD file.

The controller code contains a routine for each HART instrument and a Hart\_Modules routine.

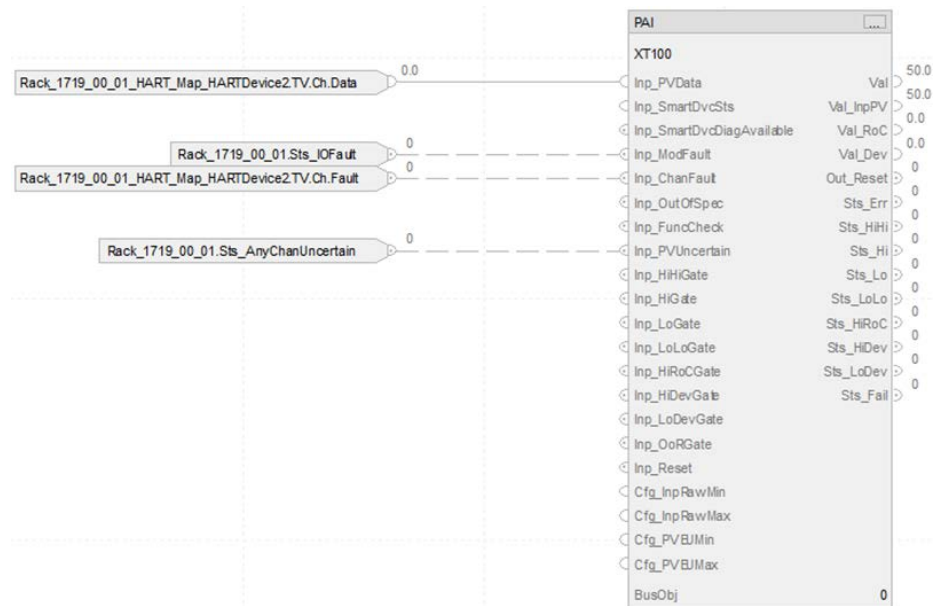


These actions occur in the Hart\_Modules routine.

- An Add-On Instruction is inserted which gets the data from the HART Module (in this case I\_1718\_1719\_AI4H).
- The HART data is mapped into an array for the device (Rack\_1719\_01\_HART\_Map\_HARTCH2).
- Each element of the array is mapped into a tag of type PAX\_HART\_DEVICE:I:0 via the Add-On Instruction rap\_Tec\_HARTChanData\_to\_PAH.



The instrument logic has no PAH module.



**Notes:**

## EtherNet/IP Integration

### EtherNet/IP Data

The PlantPax® control strategies that use EtherNet/IP™ data use Add-On Instructions for Endress+Hauser devices to provide input to a Process Analog Input (PAI) instruction. For more information, see the PAI Control Strategy on [page 151](#).

#### EtherNet/IP Module Configuration Considerations

Configuration Type	Operand	Type	Description
Module	PlantPax control	Flowmeter/sensor AOIs: <ul style="list-style-type: none"> <li>• raP_Dvc_EH_Promag100_FW2, for Promag 100</li> <li>• raP_Dvc_EH_Promag300_500, for Promag 300 and Promag 500</li> <li>• raP_Dvc_EH_Promag400_FW3, for Promag 400 V01</li> <li>• raP_Dvc_EH_Promag400V02_Rev4, for Promag 400 V02</li> <li>• raP_Dvc_EH_Promag53_FW1, for Promag 53</li> <li>• raP_Dvc_EH_Promass100_FW3, for Promass 100)</li> <li>• raP_Dvc_EH_Promass300_500, for Promass 300 and Promass 500</li> <li>• raP_Dvc_EH_Promass83_FW2, for Promass 83</li> <li>• raP_Dvc_EH_Sensor, for Liquiline CM44x sensors</li> </ul>	Instance of device required for proper operation of control strategy.
	Ref_Inp	EH:xxx_:I1:0	Required data type from the device, where xxx = device type.
	Ref_Out	EH:xxx_:100:01:0	Required data type from the device, where xxx = device type.
	Ref_Cfg	EH:xxx_:100_Rev2:C:0	Required data type from the device, where xxx = device type.
	Ref_MeterData	raP_UDT_Dvc_EH_Flowmeter_Data	Passes information from module to raP_Dvc_EH_Flowmeter
	Ref_EUTable	RAC_CODE_DESCRIPTION[897]	Table of engineering unit codes Specific in-out parameters to sensor
	Ref_DiagTable	RAC_CODE_DESCRIPTION[317]	Table to reference diagnostic codes Specific in-out parameters to sensor
Flowmeter	PlantPax control	raP_Dvc_EH_Flowmeter	Works in conjunction with EH module AOI. Receives data from Ref_MeterData tag. Optionally connected with a raP_Dvc_EH_Heartbeat AOI.
	Ref_MeterData	raP_UDT_Dvc_EH_Flowmeter_Data	Receives information from module AOI
	Ref_EUTable	RAC_CODE_DESCRIPTION[897]	Table of engineering unit codes
	Ref_DiagTable	RAC_CODE_DESCRIPTION[99]	Table to reference diagnostic codes Specific to device being used
Heartbeat	PlantPax control	raP_Dvc_EH_Heartbeat	Provides option to do extensive self-test of device
	Ref_SendMSG	MESSAGE	Sent message to device
	Ref_SendData	DINT[9]	Sent data to device
	Ref_ReadMSG	MESSAGE	Received message from device
	Ref_ReadData	DINT[2]	Received data from device
	Ref_HeartbeatSeq	raP_UDT_Dvc_EH_Heartbeat_Seq	Sequence of steps for the test to run Specific to device being used

# EtherNet/IP Instructions

The PlantPAX control strategies include EtherNet/IP options that support the EtherNet/IP devices available from Endress+Hauser. These options are available:

- EtherNet/IP control strategies use both the raP\_Dvc\_EH\_Flowmeter instruction and the raP\_Dvc\_EH\_Heartbeat instruction for Promag or Promass flowmeters; the heartbeat instruction provides diagnostics.
- EtherNet/IP NoHB control strategies use the raP\_Dvc\_EH\_Flowmeter instruction for Promag or Promass flowmeters.
- EtherNet/IP Sensor control strategies use the raP\_Dvc\_EH\_Sensor instruction for Liquiline analyzers.

For more information on these instructions, see Endress+Hauser EtherNet/IP Instrumentation for PlantPAX DCS Reference Manual, [PROCES-RM212A](#).

## Flowmeter Example - raP\_Dvc\_EH\_Flowmeter

Control strategies with the raP\_Dvc\_EH\_Flowmeter instruction integrate Endress+Hauser Promag and Promass flowmeters into a PlantPAX system.

- Promass meters measure flow using a Coriolis flow sensor and provide mass flow, volumetric flow, density, and other variables depending on the model chosen.
- Endress+Hauser Promag meters use a magnetic flow sensor and provide volumetric flow, mass flow, and other variable, depending on the model chosen.

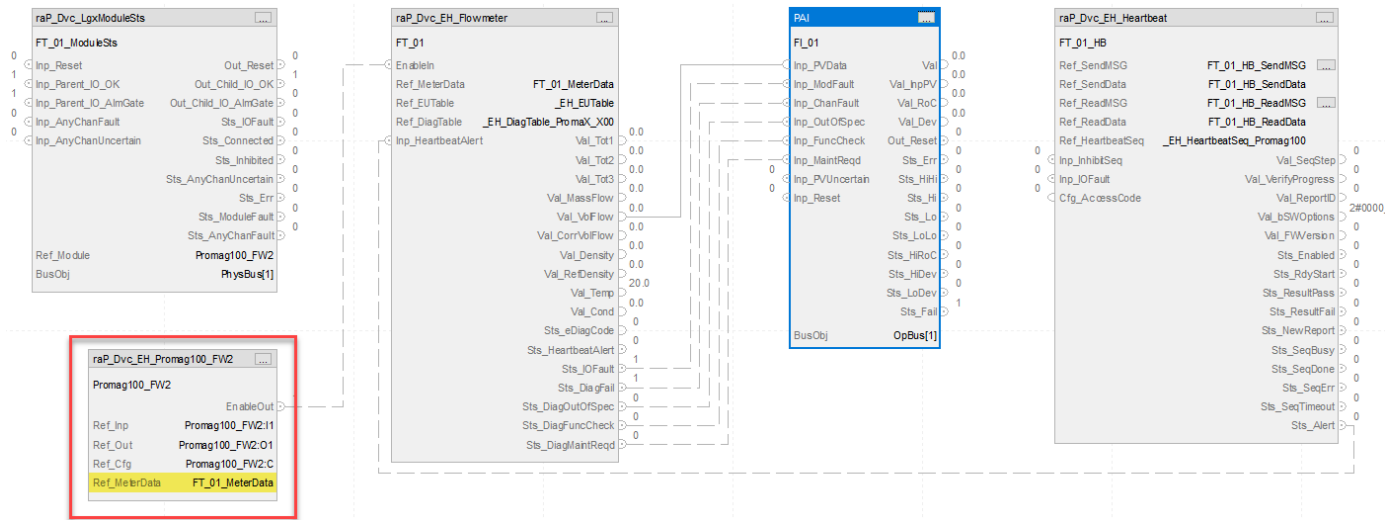
Both types of flowmeters also offer three independent totalizers as well, each with the ability to be cleared from the faceplate or from user logic. Measured variables and their status are provided to a PAI instruction for display on the HMI and for generation of alarms.

These are the primary instructions in this control strategy.

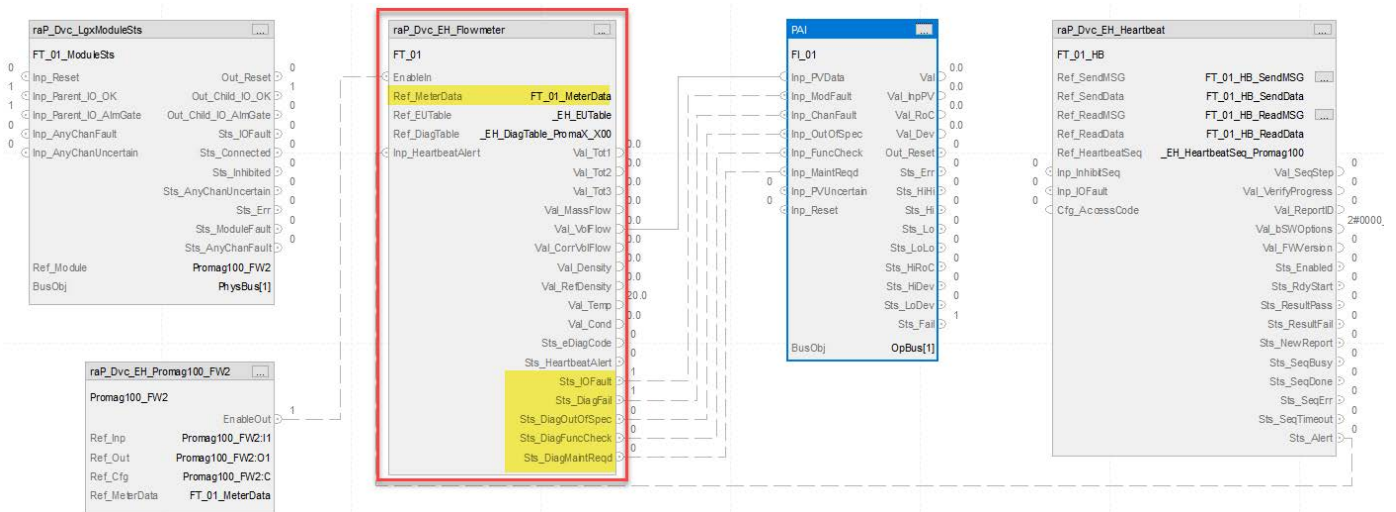
Instruction Type	Description
Model-specific instructions, where xxx = device type. <ul style="list-style-type: none"><li>• raP_Dvc_EH_Promagxxx</li><li>• raP_Dvc_EH_Promassxxx</li></ul>	Provides data from the flowmeter to the user program in a common MeterData structure. This data structure provides meter capabilities, analog data, status, units of measure, and totalizer data for the meter, and controls for the meter's totalizers. Model-specific instructions gather and distribute the MeterData but do not have any faceplate or other user interface.
The raP_Dvc_EH_Flowmeter instruction	Consolidates the meter data in a common format. Has a faceplate that shows the data from the meter and accepts commands to control the totalizers.
One or more PAI instructions	Provides a single analog value for display on the HMI. Has a faceplate for setting alarm thresholds and displaying analog alarms.

The flow of information through the blocks is:

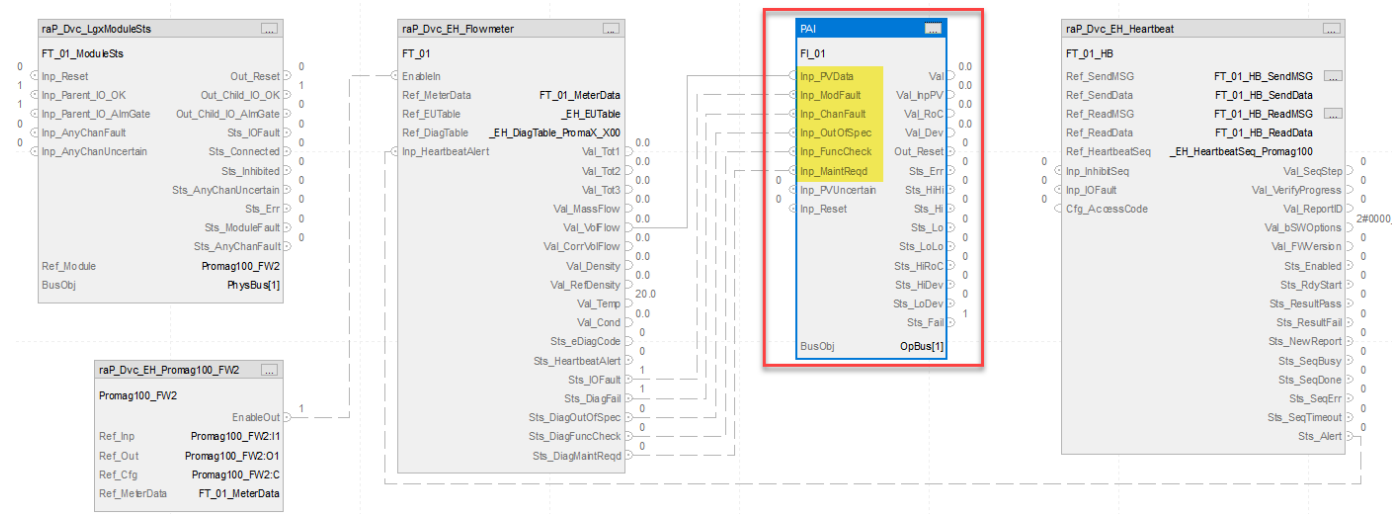
1. To interface to the flowmeter, the model-specific Add-On Instruction uses InOut parameters for the device assembly tags for Input, Configuration, and Output. It provides data in a common format to a MeterData tag through an additional InOut parameter.



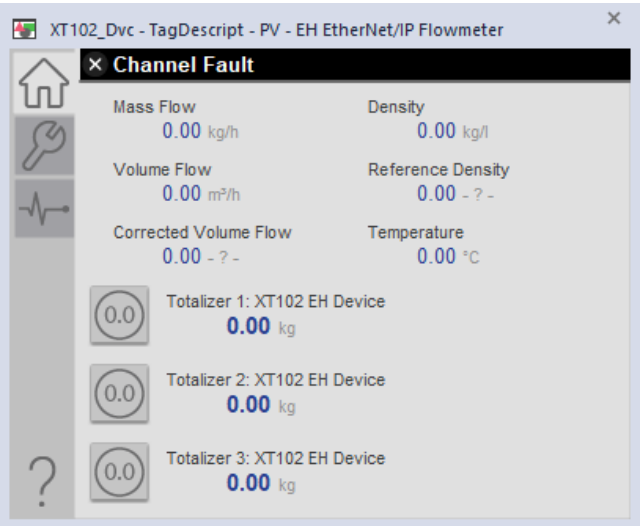
2. The raP\_Dvc\_EH\_Flowmeter tag gets data from the MeterData tag and provides several Values (for example, Val\_MassFlow) with units and status.



3. A display or alarm variable is wired to a PAI instruction, which provides those services.



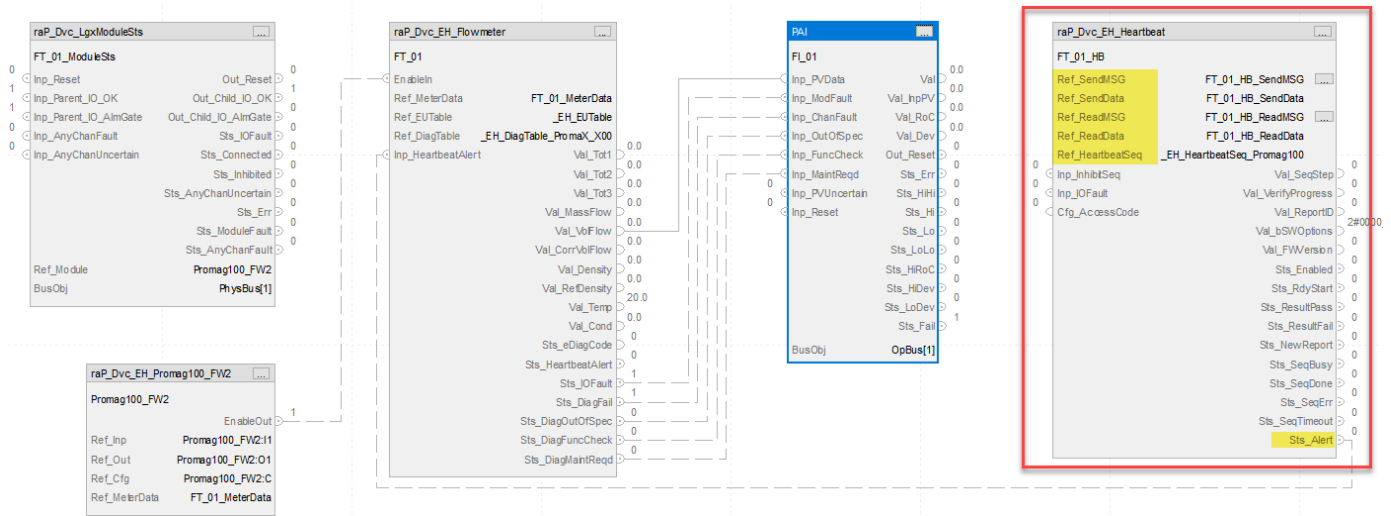
For detailed meter status and diagnostics, click the analog display object to open the PAI faceplate, then click the Smart Device button  to open the Flowmeter faceplate.




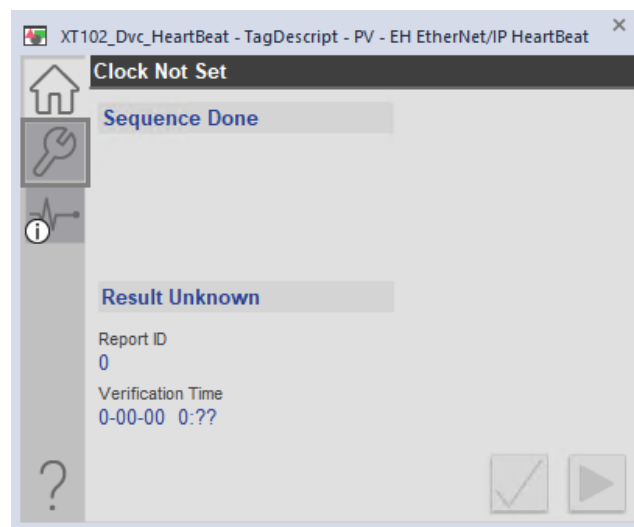


## Heartbeat Example - raP\_Dvc\_EH\_Heartbeat

Many Endress+Hauser flowmeters provide Heartbeat Verification technology, which can perform an extensive self-test of the device. From the operator faceplate, you can initiate the Heartbeat Verification, see the test progress and the overall pass/fail result. A detailed report of the test is then available from the on-device web page.



To open the Heartbeat faceplate, click the analog display object to open the PAI faceplate, then click the Heartbeat button .



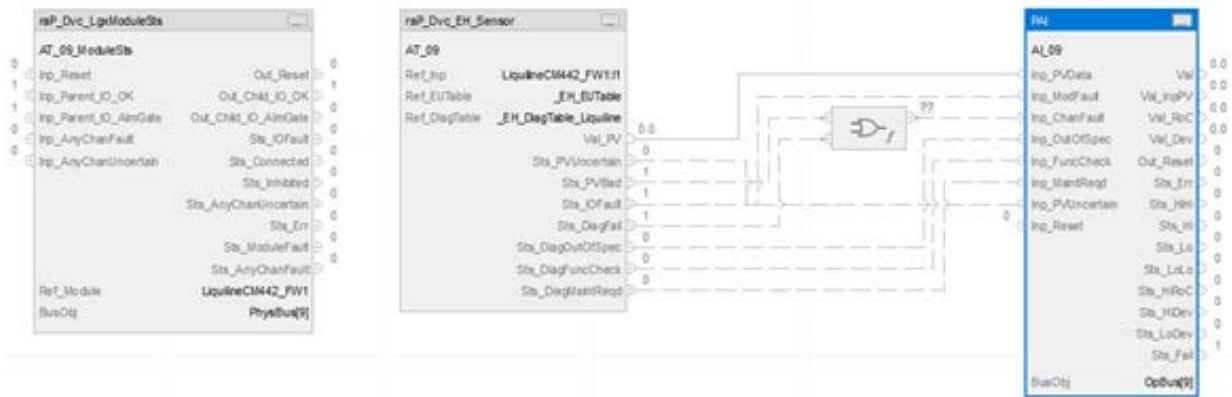
On the Heartbeat faceplate, enter the appropriate data for the test, initiate the test, monitor progress of the test, and view the final overall test result.


Control logic can also connect to the **raP\_Dvc\_EH\_Heartbeat** instruction and can disable the button that initiates the Heartbeat Verification test while the process is running.

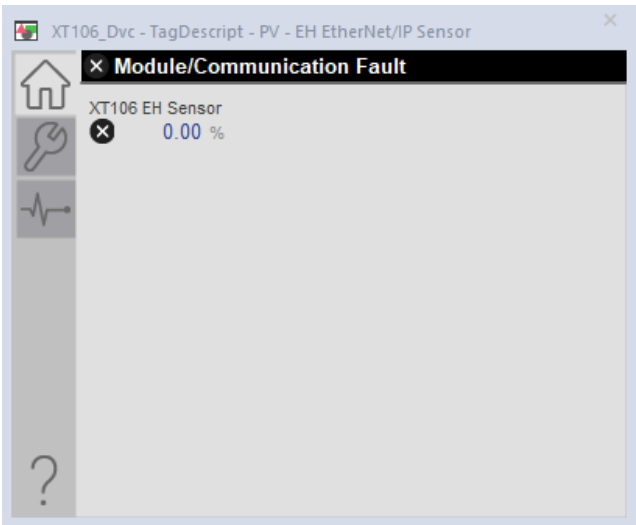
The **raP\_Dvc\_EH\_Heartbeat** instruction timestamps the test using the controller clock. For a successful test, the controller time must be set to a date later than January 1, 2000.

### Sensor Example - raP\_Dvc\_EH\_Sensor

The raP\_Dvc\_EH\_Sensor instruction is used with the Endress+Hauser Liquiline CM442, Liquiline CM444, or Liquiline CM448 analyzer to monitor an analysis probe for a variable such as pH, oxidation-reduction potential (ORP), or dissolve oxygen (DO). The raP\_Dvc\_EH\_Sensor instruction selects one the 16 configurable analog input variables in the analyzer and presents the value, status, and units for use by a PAI instruction. The PAI instruction provides filtering and value, rate of change, and deviation alarms.



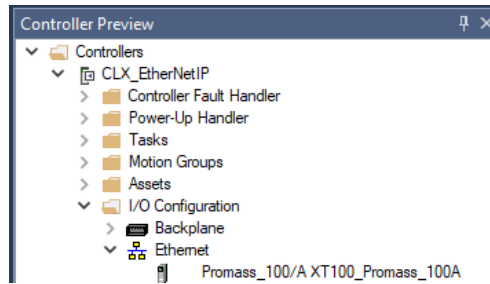
To view detailed status and diagnostics for the device, click the analog display object to open the PAI faceplate, then click the Smart Device button  to open the Sensor faceplate.



## Integrate an EtherNet/IP E+H Flowmeter Device

In this example, the ACM project contains:

- ControlLogix Process controller
- Promass 100 flowmeter



**IMPORTANT** When you add multiple modules to an ACM project, remember to enter a unique IP address for each module.

1. Configure the process controller parameters that are required for your application.

<b>Name:</b>	CLX_EtherNetIP
<b>Description:</b>	<input type="text" value="Description"/>
<b>Catalog Number:</b>	Process_Controller (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5

Parameters	
<div> <div>01 - Controller</div> <div> <div>ChassisName</div> <div>Local</div> </div> <div> <div>Slot</div> <div>0</div> </div> <div> <div>Size</div> <div>17</div> </div> <div> <div>SoftwareRevision</div> <div>36</div> </div> <div> <div>ProcessorType</div> <div>1756-L85EP</div> </div> <div> <div>PlantPAxTaskingModel_Enabled</div> <div>False</div> </div> </div>	
<div> <div>02 - HMI</div> <div> <div>Area</div> <div>/Area::</div> </div> <div> <div>Path</div> <div>[shortcut]</div> </div> <div> <div>Has_IsPositioned</div> <div>False</div> </div> <div> <div>AreaPath</div> <div>/Area::[shortcut]</div> </div> <div> <div>AreaPathME</div> <div>[shortcut]</div> </div> </div>	
<div> <div>03 - Historian</div> <div> <div>HistorianMachineName</div> <div></div> </div> <div> <div>HistorianMachineID</div> <div></div> </div> <div> <div>FTVAppName</div> <div></div> </div> <div> <div>HistorianPath</div> <div>Application/Area:RSLink Enterprise:[shortcut]</div> </div> <div> <div>FTLInterfaceNo</div> <div>1</div> </div> </div>	
<div> <div>04 - Operations</div> <div> <div>Has_Redundant</div> <div>False</div> </div> <div> <div>Has_ChangeDetect</div> <div>False</div> </div> <div> <div>Has_TaskMonitor</div> <div>True</div> </div> <div> <div>Has_OOAP</div> <div>False</div> </div> <div> <div>Has_HART</div> <div>False</div> </div> <div> <div>Has_EventLogging</div> <div>False</div> </div> <div> <div>Cfg_IncludeSystemTag</div> <div>True</div> </div> </div>	
<div> <div>05 - Alarm Configuration</div> <div> <div>AlarmClass</div> <div>0</div> </div> <div> <div>Cfg_HasMajorFaultAlm</div> <div>True</div> </div> <div> <div>Cfg_HasTaskMonAlm</div> <div>True</div> </div> </div>	
<div> <div>05.03 - Major Fault Alarm</div> <div> <div>MajorFaultAlarmCommand</div> <div></div> </div> <div> <div>Cfg_MajorFaultAckReqd</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultResetReqd</div> <div>False</div> </div> <div> <div>Cfg_MajorFaultSeverity</div> <div>1000</div> </div> <div> <div>Cfg_MajorFaultMaxShelfDuration</div> <div>480</div> </div> <div> <div>Cfg_MajorFaultAlarmGroup</div> <div><input type="button" value="→"/></div> </div> <div> <div>Cfg_MajorFaultShelveDuration</div> <div>0</div> </div> <div> <div>Cfg_MajorFaultAlarmSetoperations</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultAlarmSetrollupcount</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultDeadband</div> <div>0.0</div> </div> <div> <div>Cfg_MajorFaultOffDelay</div> <div>0</div> </div> </div>	

2. In the Controller Preview, add the EtherNet/IP device under the Ethernet network.

**IMPORTANT** You can only add devices in the Controller Preview.  
You can only delete devices in the Class View.

- a. Configure the chassis name, the IP address, the RPI, and if the device has Heartbeat. The chassis name should match the name of the PAI instance to which the device is connected.

Name:	XT100_Promass_100A
Description:	E+H EtherNet/IP Promass 100
Catalog Number:	Promass_100/A (2.0) - Published
Solution:	(RA-LIB) Process 5

Parameters

Module Configuration

ChassisName	XT100
Address	192.168.1.0
RPI	20.0
Cfg_HasHB	True

3. From the Process library > Control Strategies > Input Processing folder, add a PAI instance in the Class View for the analog input module.

**IMPORTANT**

- You must create an individual PAI instance for each input module in your application.
- Some parameters are only visible when certain parameters are set or populated. For example, if Cfg\_HasHB is True or False in the module instance, or if the Ref\_EtherNetIPModule has been populated in the PAI instance.

Name:	XT100
Description:	Description
Catalog Number:	PAI (4.0) - Published
Solution:	(RA-LIB) Process 5
Task:	Normal
Program:	Nom

Parameters    Events

00 - Selection

ACM_Type	PAI(Single_channel)
ACM_UsedIn	None
IO_Signal_Type	EH_EtherNetIP
Use_FTIS	False

00.01 - Data - Common

Area	Area01
Instruction	PAI
Label	XT100
Library	raP-5_20
Library_EH	raP-5_20
Has_More_URL	
URL	n/a

00.01.01 - Data - EH

Label_EH_FT	XT100 EH Flowmeter
Label_EH_Dvc	XT100 EH Device
Label_EH_HB	XT100 EH Heartbeat
EH_Dvc_EU	g/s
URL_EH_FT	n/a
URL_EH_Dvc	n/a
URL_EH_HB	n/a

00.02 - Data - General

02 - Device Configuration

02.02 - Device Configuration Fail Actions

02.03 - Device Configuration Limits and Scaling

03.00 - IO Configuration

Ref_EtherNetIPModule	XT100_Promass_100A
----------------------	--------------------

03.00.01 - IO Configuration EH Flowmeter

Cfg_FT_HasMoreOnli	False
--------------------	-------

a. Configure these parameters.

ACM Parameter	Description
Task Program	Assign a Task and Program for the PAI control strategy.
IO_Signal_Type	EH_EtherNetIP
Label_EH_FT	Label for EH block
Label_EH_Dvc	Label for EH flowmeter block
Label_EH_HB	Label for EH heartbeat block
EH_Dvc_EU	EH flowmeter engineering units
URL_EH_FT	Help URL for EH block
URL_EH_Dvc	Help URL for EH flowmeter block
URL_EH_HB	Help URL for EH heartbeat block
Ref_EtherNetIPModule	Connect to the EH device (PAI name should be in EH device name)
Cfg_FT_HasMoreObj	EH Flowmeter config: 1 = Tells HMI an object with more information is available, enable navigation
Cfg_FT_Has_More_URL	EH Flowmeter config: Has more URL
Cfg_FT_CnfrmReqd	EH Flowmeter config: 0 = None, 1 = Command confirmation required, 2 = Performer e-signature required, 3 = Performer and approver e-signature required
Cfg_HB_Has_More_URL	EH Heartbeat config: Has more URL
Cfg_HB_HasMoreObj	EH Heartbeat config: 1 = Tells HMI an object with more information is available, enable navigation
Cfg_HB_CnfrmReqd	EH Heartbeat config: 0 = None, 1 = Command confirmation required, 2 = Performer e-signature required, 3 = Performer and approver e-signature required
Cfg_AccessCode	EH Heartbeat config: Device access code
Cfg_StepTime	EH Heartbeat config: How often to trigger a message during sequence (seconds). Valid 0.1 to 10.0
Cfg_SeqTimeout	EH Heartbeat config: Maximum time to allow sequence to complete (seconds). Valid 60.0 to 3600.0
Cfg_sCustomer	EH Heartbeat config: Customer description, text sent to device for report
Cfg_sLocation	EH Heartbeat config: Location description, test sent to device for report

4. From the Process library > Organization > Bus folder, add a Hardware\_Bus object to the Class View.

**Name:** HWBus\_Org

**Description:**

**Catalog Number:** Hardware\_Bus (4.0) - Published

**Solution:** (RA-LIB) Process 5 Task:

---

Parameters Bus View\_Assignment

00.00 - Org  
HWOrgViewSize 4

00.01 - Org Scan Data - Common

Scan_Library	raP-5_20
Scan_Instruction	raP_Opr_OrgScan
Scan_Label	HWOrgScan
Scan_Area	Area01

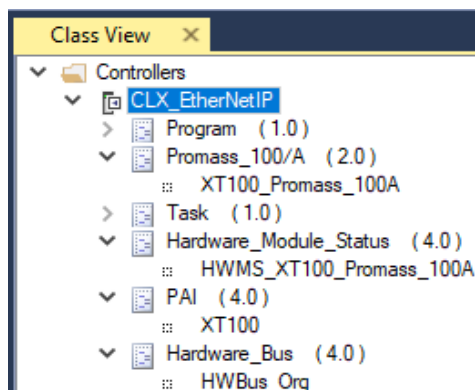
00.02 - Org View Data - Common

View_Library	raP-5_20
View_Instruction	raP_Opr_OrgView
View_Area	Area01
View_Label	HWOrgScan
View_Area_01	Area01
View_Label_01	OrgView
View_Area_02	Area01
View_Label_02	OrgView
View_Area_03	Area01
View_Label_03	OrgView

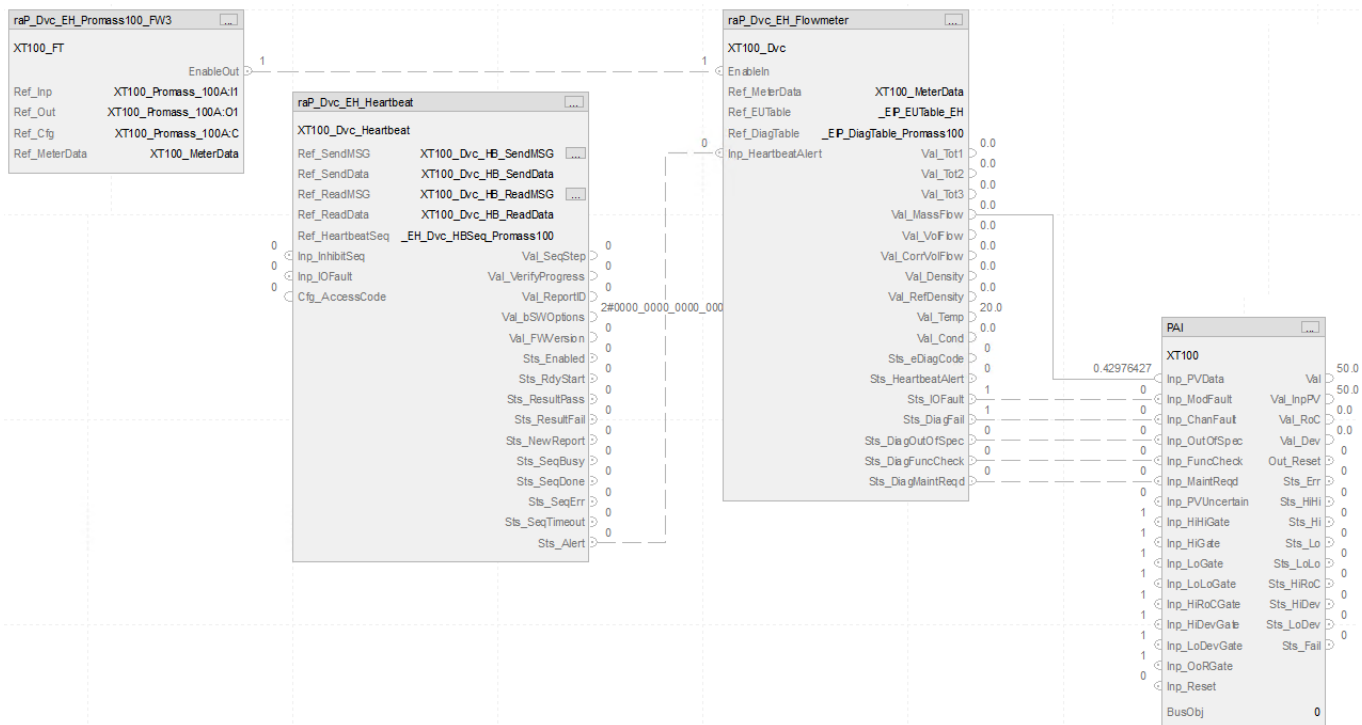
5. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object to the Class View.
  - a. Configure the Module parameter for the Promass 100 module.
  - b. Assign a unique Bus\_Instance on the Hardware\_Bus.

<b>Name:</b>	HWMS_XT100_Promass_100A
<b>Description:</b>	This instruction checks the I/O connection status of the given
<b>Catalog Number:</b>	Hardware_Module_Status (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5
<b>Parameters</b>	
<div> <div>00.01 - Data - Common</div> <div> <div>Area</div> <div>Area01</div> </div> <div> <div>Instruction</div> <div>raP_Dvc_LgxModuleSts</div> </div> <div> <div>Label</div> <div>Module Status</div> </div> <div> <div>Library</div> <div>raP-5_20</div> </div> </div>	
<div> <div>01 - Options</div> <div> <div>Module</div> <div>XT100_Promass_100A</div> </div> <div> <div>Bus_Instance</div> <div>HWBus_Org.Bus.Cmd_1</div> </div> </div>	

The Class View contains these objects:



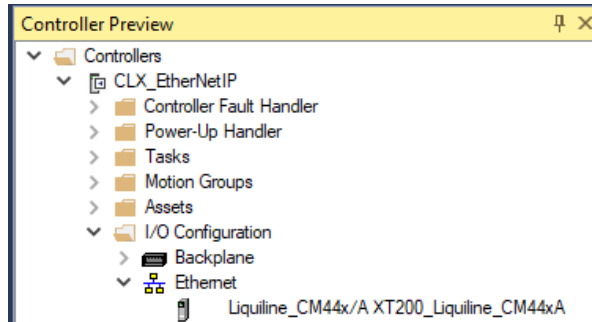
When you have added all of the devices, generate the controller ACD file.



## Integrate an EtherNet/IP E+H Sensor Device

In this example, the ACM project contains:

- ControlLogix Process controller
- Liquiline CM44x sensor



1. Configure the process controller parameters that are required for your application.

<b>Name:</b>	CLX_EtherNetIP
<b>Description:</b>	<input type="text" value="Description"/>
<b>Catalog Number:</b>	Process_Controller (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5

Parameters	
<div> <div>01 - Controller</div> <div> <div>ChassisName</div> <div>Local</div> </div> <div> <div>Slot</div> <div>0</div> </div> <div> <div>Size</div> <div>17</div> </div> <div> <div>SoftwareRevision</div> <div>36</div> </div> <div> <div>ProcessorType</div> <div>1756-L85EP</div> </div> <div> <div>PlantPAxTaskingModel_Enabled</div> <div>False</div> </div> </div>	
<div> <div>02 - HMI</div> <div> <div>Area</div> <div>/Area::</div> </div> <div> <div>Path</div> <div>[shortcut]</div> </div> <div> <div>Has_isPositioned</div> <div>False</div> </div> <div> <div>AreaPath</div> <div>/Area::[shortcut]</div> </div> <div> <div>AreaPathME</div> <div>[shortcut]</div> </div> </div>	
<div> <div>03 - Historian</div> <div> <div>HistorianMachineName</div> <div></div> </div> <div> <div>HistorianMachineID</div> <div></div> </div> <div> <div>FTVAppName</div> <div></div> </div> <div> <div>HistorianPath</div> <div>Application/Area:RSLink Enterprise:[shortcut]</div> </div> <div> <div>FTLDInterfaceNo</div> <div>1</div> </div> </div>	
<div> <div>04 - Operations</div> <div> <div>Has_Redundant</div> <div>False</div> </div> <div> <div>Has_ChangeDetect</div> <div>False</div> </div> <div> <div>Has_TaskMonitor</div> <div>True</div> </div> <div> <div>Has_OOAP</div> <div>False</div> </div> <div> <div>Has_HART</div> <div>False</div> </div> <div> <div>Has_EventLogging</div> <div>False</div> </div> <div> <div>Cfg_IncludeSystem Tag</div> <div>True</div> </div> </div>	
<div> <div>05 - Alarm Configuration</div> <div> <div>AlarmClass</div> <div>0</div> </div> <div> <div>Cfg_HasMajorFaultAlm</div> <div>True</div> </div> <div> <div>Cfg_HasTaskMonAlm</div> <div>True</div> </div> </div>	
<div> <div>05.03 - Major Fault Alarm</div> <div> <div>MajorFaultAlarmCommand</div> <div></div> </div> <div> <div>Cfg_MajorFaultAckReqd</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultResetReqd</div> <div>False</div> </div> <div> <div>Cfg_MajorFaultSeverity</div> <div>1000</div> </div> <div> <div>Cfg_MajorFaultMaxShelfDuration</div> <div>480</div> </div> <div> <div>Cfg_MajorFaultAlarmGroup</div> <div></div> </div> <div> <div>Cfg_MajorFaultShelfDuration</div> <div>0</div> </div> <div> <div>Cfg_MajorFaultAlarmSetoperations</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultAlarmSetrollupcount</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultDeadband</div> <div>0.0</div> </div> <div> <div>Cfg_MajorFaultOffDelay</div> <div>0</div> </div> </div>	

2. Add the EtherNet/IP device in the Controller Preview under the Ethernet network.

**IMPORTANT** You can only add devices in the Controller Preview.  
You can only delete devices in the Class View.

- a. Configure the chassis name, the IP address, and the RPI. The chassis name should match the name of the PAI instance to which the device is connected.

Name:	XT200_Liquiline_CM44xA
Description:	E+H EtherNet/IP Liquiline_CM44x
Catalog Number:	Liquiline_CM44x/A (2.0) - Published
Solution:	(RA-LIB) Process 5

Parameters

Module Configuration

ChassisName	XT200
Address	192.168.1.0
RPI	100.0

3. From the Process library > Control Strategies > Input Processing folder, add a PAI instance in the Class View for the analog input module.

- IMPORTANT**
- You must create an individual PAI instance for each input module in your application.
  - Some parameters are only visible when certain parameters are set or populated. For example, if Cfg\_HasHB is True or False in the module instance, or if the Ref\_EtherNetIPModule has been populated in the PAI instance.

Name:	XT200
Description:	Description
Catalog Number:	PAI (4.0) - Published
Solution:	(RA-LIB) Process 5
Task:	Normal
Program:	NormalPr

Parameters    Events

00 - Selection

ACM_Type	PAI(Single_channel)
ACM_UsedIn	None
IO_Signal_Type	EH_EtherNetIP
Use_FTIS	False

00.01 - Data - Common

Area	Area01
Instruction	PAI
Label	XT200
Library	raP-5_20
Library_EH	raP-5_20
Has_More_URL	
URL	n/a

00.01.01 - Data - EH

Label_EH_Sensor	XT200 EH Sensor
EH_Sensor_EU	%
URL_EH_Sensor	n/a

00.02 - Data - General

02 - Device Configuration

02.02 - Device Configuration Fail Actions

02.03 - Device Configuration Limits and Scaling

03.00 - IO Configuration

Ref_EtherNetIPModule	XT200_Liquiline_CM44xA
----------------------	------------------------

03.00.00 - IO Configuration EH Sensor

Cfg_UseDvcEUText	False
Cfg_Sensor_HasMoreObj	False
Cfg_Sensor_Has_More_URL	False
Cfg_Chan	1
Cfg_PVInpNum	1
Cfg_SVInpNum	0
Cfg_TVInpNum	0



a. Configure these parameters.

ACM Parameter	Description
Task Program	Assign a Task and Program for the PAI control strategy.
IO_Signal_Type	EH_EtherNetIP
Label_EH_Sensor	Label for EH block
EH_Sensor_EU	EH block engineering units
URL_EH_Sensor	Help URL for EH block
Ref_EtherNetIPModule	Connect to the EH device (PAI name should be in EH device name)
Cfg_UseDvcEUText	EH Sensor config: 1 = Use engineering units text based on lookup of EU codes from device, 0 = use user-entered engineering units text
Cfg_Sensor_HasMoreObj	EH Sensor config: 1 = Tells HMI an object with more information is available, enable navigation
Cfg_Sensor_Has_More_URL	EH Sensor config: Has more URL
Cfg_Chan	EH Sensor config: Channel in CM44x to which the sensor/probe is connected. Valid = 1 to 8
Cfg_PVInpNum	EH Sensor config: Assigned analog input in CM44x to use for PV. Valid = 1 to 16
Cfg_SVInpNum	EH Sensor config: Assigned analog input in CM44x to use for SV. Valid = 1 to 16, 0 = SV not used
Cfg_TVInpNum	EH Sensor config: Assigned analog input in CM44x to use for TV. Valid = 1 to 16, 0 = TV not used
Cfg_FVInpNum	EH Sensor config: Assigned analog input in CM44x to use for FV. Valid = 1 to 16, 0 = FV not used
Cfg_sFVEU	EH Sensor config: Engineering units for FV display on HMI
Cfg_sPVEU	EH Sensor config: Engineering units for PV display on HMI
Cfg_sSVEU	EH Sensor config: Engineering units for SV display on HMI
Cfg_sTVEU	EH Sensor config: Engineering units for TV display on HMI

4. From the Process library > Organization > Bus folder, add a Hardware\_Bus object to the Class View.

**Name:** HWBus\_Org

**Description:**

**Catalog Number:** Hardware\_Bus (4.0) - Published

**Solution:** (RA-LIB) Process 5 Task:

---

Parameters Bus View\_Assignment

00.00 - Org  
HWOrgViewSize 4

00.01 - Org Scan Data - Common

Scan_Library	raP-5_20
Scan_Instruction	raP_Opr_OrgScan
Scan_Label	HWOrgScan
Scan_Area	Area01

00.02 - Org View Data - Common

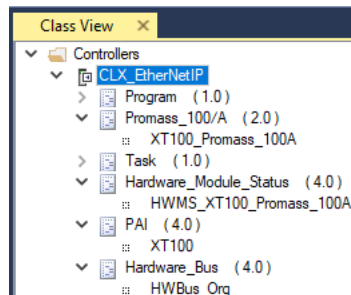
View_Library	raP-5_20
View_Instruction	raP_Opr_OrgView
View_Area	Area01
View_Label	HWOrgScan
View_Area_01	Area01
View_Label_01	OrgView
View_Area_02	Area01
View_Label_02	OrgView
View_Area_03	Area01
View_Label_03	OrgView

5. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object to the Class View.
  - a. Configure the Module parameter for the Promass 100 module.
  - b. Assign a unique Bus\_Instance on the Hardware\_Bus.

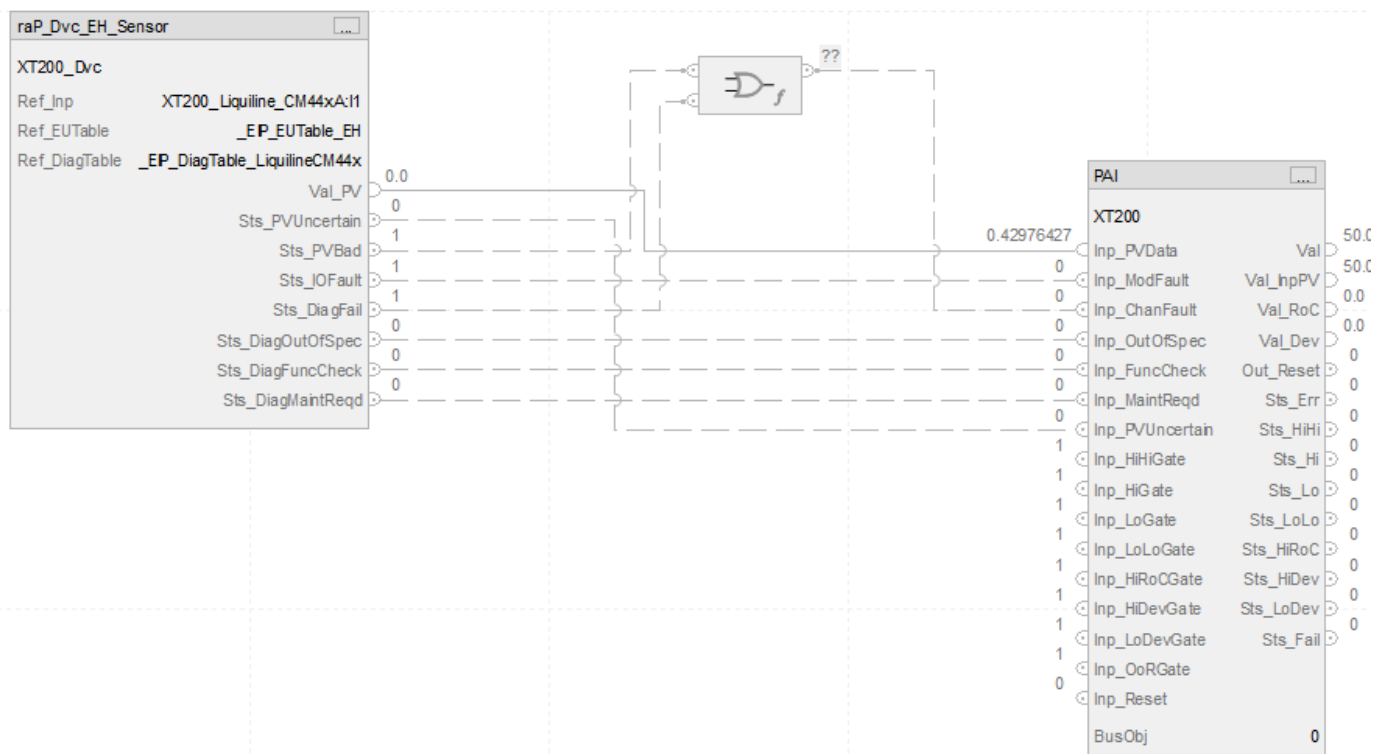
<b>Name:</b>	HWMS_XT200_Liquiline_CM44xA
<b>Description:</b>	This instruction checks the I/O connection status of the given
<b>Catalog Number:</b>	Hardware_Module_Status (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5

Parameters	
<div> <div>00.01 - Data - Common</div> <div> <div>Area</div> <div>Area01</div> </div> <div> <div>Instruction</div> <div>raP_Dvc_LgxModuleSts</div> </div> <div> <div>Label</div> <div>Module Status</div> </div> <div> <div>Library</div> <div>raP-5_20</div> </div> </div>	
<div> <div>01 - Options</div> <div> <div>Module</div> <div>XT200_Liquiline_CM44xA</div> </div> <div> <div>Bus_Instance</div> <div>HWBus_Org.Bus.Cmd_1</div> </div> </div>	

The Class View contains these objects:



When you finish adding devices, generate the controller ACD file.



# FOUNDATION Fieldbus and Profibus PA Integration

## FOUNDATION Fieldbus and Profibus PA Instructions

The PlantPAX® control strategies include these FOUNDATION Fieldbus (FF) and Profibus PA (PA) options.

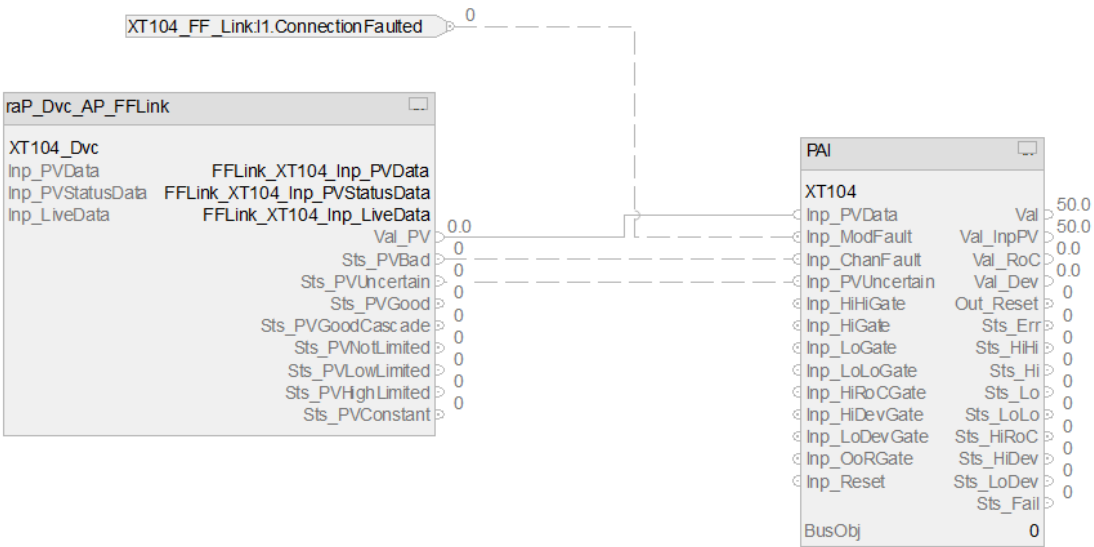
- FOUNDATION Fieldbus control strategies use the raP\_Dvc\_AP\_FFLink instruction to integrate FOUNDATION Fieldbus devices.
- Profibus PA control strategies use the raP\_Dvc\_AP\_PALink instruction to integrate Profibus PA devices

For more information, see Profibus PA and FOUNDATION Fieldbus Linking Devices in a PlantPAX Distributed Control System Reference Manual, publication [PROCES-RM213](#).

### Foundation Fieldbus Link Example - raP\_Dvc\_AP\_FFLink

Control strategies with the raP\_Dvc\_AP\_FFLink instruction integrate FOUNDATION Fieldbus devices into a PlantPAX system. This instruction transfers data from one FOUNDATION Fieldbus analog PV, mapping the REAL PV directly and mapping the PV status to a set of status bits suitable for use with the PAI instruction.

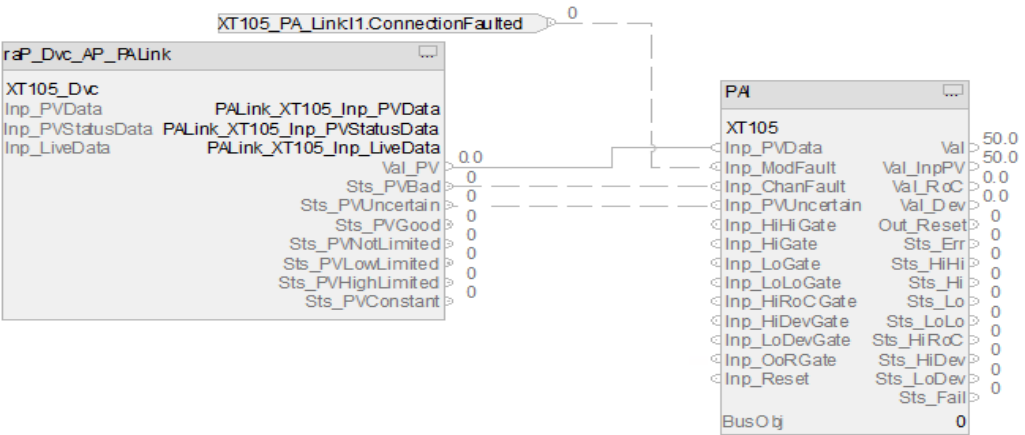
This instruction has no prescan, EnableInFalse, or postscan logic. It is intended to be executed always true in logic every scan. The instruction can be used in Ladder Diagram, Function Block Diagram, or Structured Text logic.



### Profibus PA Link Example - raP\_Dvc\_AP\_PALink

Control strategies with the raP\_Dvc\_AP\_PALink instruction integrate Profibus PA devices into a PlantPax system. This instruction transfers data from one FOUNDATION Fieldbus analog PV, mapping the REAL PV directly and mapping the PV status to a set of status bits suitable for use with the PAI instruction.

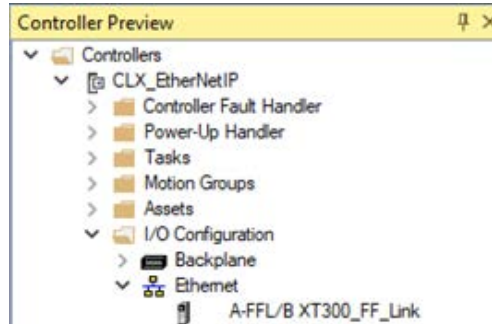
This instruction has no prescan, EnableInFalse, or postscan logic. It is intended to be executed always true in logic every scan. The instruction can be used in Ladder Diagram, Function Block Diagram, or Structured Text logic.



## Integrate a FOUNDATION Fieldbus Device

In this example, the ACM project contains:

- ControlLogix Process controller
- Aparian FOUNDATION Fieldbus Linking device



**IMPORTANT** When you add multiple modules to an ACM project, remember to enter a unique IP address for each module.

1. Configure the process controller parameters that are required for your application.

<b>Name:</b>	CLX_FF
<b>Description:</b>	Description
<b>Catalog Number:</b>	Process_Controller (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5

Parameters	
<div> <div>01 - Controller</div> <div> <div>ChassisName</div> <div>Local</div> </div> <div> <div>Slot</div> <div>0</div> </div> <div> <div>Size</div> <div>17</div> </div> <div> <div>SoftwareRevision</div> <div>36</div> </div> <div> <div>ProcessorType</div> <div>1756-L85EP</div> </div> <div> <div>PlantPaxTaskingModel_Enabled</div> <div>False</div> </div> </div>	
<div> <div>02 - HMI</div> <div> <div>Area</div> <div>/Area::</div> </div> <div> <div>Path</div> <div>[shortcut]</div> </div> <div> <div>Has_IsPositioned</div> <div>False</div> </div> <div> <div>AreaPath</div> <div>/Area::[shortcut]</div> </div> <div> <div>AreaPathME</div> <div>[shortcut]</div> </div> </div>	
<div> <div>03 - Historian</div> <div> <div>HistorianMachineName</div> <div></div> </div> <div> <div>HistorianMachineID</div> <div></div> </div> <div> <div>FTVAppName</div> <div></div> </div> <div> <div>HistorianPath</div> <div>Application/Area:RSLink Enterprise:[shortcut]</div> </div> <div> <div>FTLDInterfaceNo</div> <div>1</div> </div> </div>	
<div> <div>04 - Operations</div> <div> <div>Has_Redundant</div> <div>False</div> </div> <div> <div>Has_ChangeDetect</div> <div>False</div> </div> <div> <div>Has_TaskMonitor</div> <div>True</div> </div> <div> <div>Has_OOAP</div> <div>False</div> </div> <div> <div>Has_HART</div> <div>False</div> </div> <div> <div>Has_EventLogging</div> <div>False</div> </div> <div> <div>Cfg_IncludeSystemTag</div> <div>True</div> </div> </div>	
<div> <div>05 - Alarm Configuration</div> <div> <div>AlarmClass</div> <div>0</div> </div> <div> <div>Cfg_HasMajorFaultAlm</div> <div>True</div> </div> <div> <div>Cfg_HasTaskMonAlm</div> <div>True</div> </div> </div>	
<div> <div>05.03 - Major Fault Alarm</div> <div> <div>MajorFaultAlarmCommand</div> <div></div> </div> <div> <div>Cfg_MajorFaultAckReqd</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultResetReqd</div> <div>False</div> </div> <div> <div>Cfg_MajorFaultSeverity</div> <div>1000</div> </div> <div> <div>Cfg_MajorFaultMaxShelfDuration</div> <div>480</div> </div> </div>	

2. In the Controller Preview, add the device under the Ethernet network.

**IMPORTANT**

You can only add devices in the Controller Preview.  
You can only delete devices in the Class View.

3. Configure the chassis name, the IP address, and the RPI. The chassis name should match the name of the PAI instance to which the device is connected.

**Name:**
XT300\_FF\_Link

**Description:**
Aparian Foundation Fieldbus Link A-FFL/B

**Catalog Number:**
FF\_Link/B (2.0) - Published

**Solution:**
(RA-LIB) Process 5

Parameters

2

Module Configuration

ChassisName
XT300

Address
192.168.1.0

RPI
100.0

4. From the Process library > Control Strategies > Input Processing folder, add a PAI instance in the Class View for the analog input module.

**IMPORTANT**

- You must create an individual PAI instance for each input module in your application.
- Some parameters are only visible when certain parameters are populated. For example, if the Ref\_FF\_Module has been configured in the PAI instance.

**Name:**
XT300

**Description:**
Description

**Catalog Number:**
PAI (4.0) - Published

**Solution:**
(RA-LIB) Process 5

Task:
Normal

Program:
NormalProgram

Parameters
Events

2

00 - Selection

ACM\_Type
PAI(Single\_channel)

ACM\_UsedIn
None

IO\_Signal\_Type
FF

Use\_FTIS
False

00.01 - Data - Common

Area
Area01

Instruction
PAI

Label
XT300

Library
raP-5\_20

Library\_FFPA
raP-5\_20

Has\_More\_URL

URL
n/a

00.01.02 - Data - FF

Label\_FF
XT300 FF

FF\_EU
%

URL\_FF
n/a

00.02 - Data - General

02 - Device Configuration

02.02 - Device Configuration Fail Actions

02.03 - Device Configuration Limits and Scaling

03.00 - IO Configuration

Ref\_FF\_Module
XT300\_FF\_Link

03.00.03 - IO Configuration FF

Cfg\_FF\_HasMoreObj
False

04 - Alarm Configuration

06 - HMI Configuration

06.01 - HMI 2nd Language

08 - Field Device

Historian Configuration

102

Rockwell Automation Publication PROCES-RM201C-EN-P - October 2025

a. Configure these parameters.

ACM Parameter	Description
Task Program	Assign a Task and Program for the PAI control strategy
IO_Signal_Type	FF
Label_FF	Label for Fieldbus device
FF_EU	Fieldbus engineering units
URL_FF	Help URL for Fieldbus
Ref_FF_Module	Connect to the FF device (PAI name should be in FF device name)
Cfg_FT_HasMoreObj	FF config: 1 = Tells HMI an object with more information is available, enable navigation

5. From the Process library > Organization > Bus folder, add a Hardware\_Bus object.

The screenshot shows the configuration window for the **HWBus\_Org** object. The **Name** is set to **HWBus\_Org**. The **Description** field is empty. The **Catalog Number** is **Hardware\_Bus (4.0) - Published**. The **Solution** is **(RA-LIB) Process 5**. The **Task** is set to **System**. Below these fields are three tabs: **Parameters**, **Bus**, and **View\_Assignment**. The **Parameters** tab is active, showing a tree structure of parameters:

- 00.00 - Org**
  - HWOrgViewSize**: 4
- 00.01 - Org Scan Data - Common**
  - Scan\_Library**: raP-5\_20
  - Scan\_Instruction**: raP\_Opr\_OrgScan
  - Scan\_Label**: HWOrgScan
  - Scan\_Area**: Area01
- 00.02 - Org View Data - Common**
  - View\_Library**: raP-5\_20
  - View\_Instruction**: raP\_Opr\_OrgView
  - View\_Area**: Area01
  - View\_Label**: HWOrgScan
  - View\_Area\_01**: Area01
  - View\_Label\_01**: OrgView
  - View\_Area\_02**: Area01
  - View\_Label\_02**: OrgView
  - View\_Area\_03**: Area01
  - View\_Label\_03**: OrgView

6. From the Process library > Organization > Bus folder, add a Hardware\_Bus object to the Class View.

a. Configure the Module parameter for the module.

b. Assign a unique Bus\_Instance on the Hardware\_Bus.

Name:

HWMS\_XT300\_FF\_Link

Description:

This instruction checks the I/O connection status of the given

Catalog Number:

Hardware\_Module\_Status (4.0) - Published

Solution:

(RA-LIB) Process 5

Parameters

00.01 - Data - Common

Area

Area01

Instruction

raP\_Dvc\_LgxModuleSts

Label

Module Status

Library

raP-5\_20

01 - Options

Module

XT300\_FF\_Link

Bus\_Instance

HWBus\_Org.Bus.Cmd\_1

Generate the controller ACD file.

The diagram illustrates the connection between a controller (raP\_Dvc\_AP\_FFLink) and a PAI (XT300) module. The controller outputs Val\_PV (0.0) and Sts\_PVUncertain (0) to the PAI inputs. The PAI outputs Val (50.0) and Sts\_Fail (0) to the controller. A connection faulted signal (XT300\_FF\_Link11.ConnectionFaulted) is also shown.

Controller (raP_Dvc_AP_FFLink)	PAI (XT300)
Inp_PVData	Inp_PVData
Inp_PVStatusData	Inp_PVStatusData
Inp_LiveData	Inp_LiveData
Val_PV	Val
Sts_PVBad	Sts_PVBad
Sts_PVUncertain	Sts_PVUncertain
Sts_PVGood	Sts_PVGood
Sts_PVGood Cascade	Sts_PVGood Cascade
Sts_PVNotLimited	Sts_PVNotLimited
Sts_PVLowLimited	Sts_PVLowLimited
Sts_PVHighLimited	Sts_PVHighLimited
Sts_PVConstant	Sts_PVConstant

104

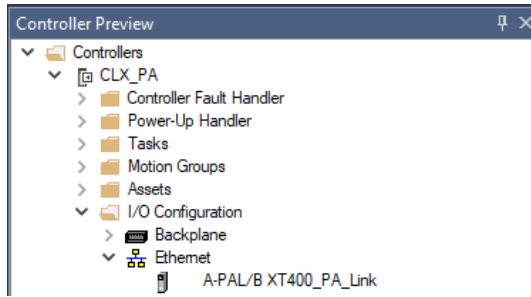
Rockwell Automation Publication PROCES-RM201C-EN-P - October 2025



## Integrate a Profibus PA Device

In this example, the ACM project contains:

- ControlLogix Process controller
- Aparian Profibus PA Fieldbus Linking device



**IMPORTANT** When you add multiple modules to an ACM project, remember to enter a unique IP address for each module.

1. Configure the process controller parameters that are required for your application.

<b>Name:</b>	CLX_PA
<b>Description:</b>	Description
<b>Catalog Number:</b>	Process_Controller (4.0) - Published
<b>Solution:</b>	(RA-LIB) Process 5

Parameters	
<div> <div>01 - Controller</div> <div> <div>ChassisName</div> <div>Local</div> </div> <div> <div>Slot</div> <div>0</div> </div> <div> <div>Size</div> <div>17</div> </div> <div> <div>SoftwareRevision</div> <div>36</div> </div> <div> <div>ProcessorType</div> <div>1756-L85EP</div> </div> <div> <div>PlantPAxTaskingModel_Enabled</div> <div>False</div> </div> </div>	
<div> <div>02 - HMI</div> <div> <div>Area</div> <div>/Area::</div> </div> <div> <div>Path</div> <div>[shortcut]</div> </div> <div> <div>Has_isPositioned</div> <div>False</div> </div> <div> <div>AreaPath</div> <div>/Area::[shortcut]</div> </div> <div> <div>AreaPathME</div> <div>[shortcut]</div> </div> </div>	
<div> <div>03 - Historian</div> <div> <div>HistorianMachineName</div> <div></div> </div> <div> <div>HistorianMachineID</div> <div></div> </div> <div> <div>FTVAppName</div> <div></div> </div> <div> <div>HistorianPath</div> <div>Application/Area:RSLink Enterprise:[shortcut]</div> </div> <div> <div>FTLInterfaceNo</div> <div>1</div> </div> </div>	
<div> <div>04 - Operations</div> <div> <div>Has_Redundant</div> <div>False</div> </div> <div> <div>Has_ChangeDetect</div> <div>False</div> </div> <div> <div>Has_TaskMonitor</div> <div>True</div> </div> <div> <div>Has_OOAP</div> <div>False</div> </div> <div> <div>Has_HART</div> <div>False</div> </div> <div> <div>Has_EventLogging</div> <div>False</div> </div> <div> <div>Cfg_IncludeSystemTag</div> <div>True</div> </div> </div>	
<div> <div>05 - Alarm Configuration</div> <div> <div>AlarmClass</div> <div>0</div> </div> <div> <div>Cfg_HasMajorFaultAlm</div> <div>True</div> </div> <div> <div>Cfg_HasTaskMonAlm</div> <div>True</div> </div> </div>	
<div> <div>05.03 - Major Fault Alarm</div> <div> <div>MajorFaultAlarmCommand</div> <div></div> </div> <div> <div>Cfg_MajorFaultAckReqd</div> <div>True</div> </div> <div> <div>Cfg_MajorFaultResetReqd</div> <div>False</div> </div> <div> <div>Cfg_MajorFaultSeverity</div> <div>1000</div> </div> <div> <div>Cfg_MajorFaultMaxShelfDuration</div> <div>480</div> </div> <div> <div>Cfg_MajorFaultAlarmGroup</div> <div>→</div> </div> <div> <div>Cfg_MajorFaultShelfDuration</div> <div>0</div> </div> <div> <div>Cfg_MajorFaultAlarmSetpoints</div> <div>True</div> </div> </div>	



a. Configure these parameters.

ACM Parameter	Description
Task Program	Assign a Task and Program for the PAI control strategy
IO_Signal_Type	PA
Label_FF	Label for Profibus PA device
PA_EU	Profibus PA engineering units
URL_PA	Help URL for Profibus PA
Ref_PA_Module	Connect to the Profibus PA device (PAI name should be in PA device name)
Cfg_PA_HasMoreObj	PA config: 1 = Tells HMI an object with more information is available, enable navigation

4. From the Process library > Organization > Bus folder, add a Hardware\_Bus object to the Class View.

**Name:** HWBus\_Org

**Description:**

**Catalog Number:** Hardware\_Bus (4.0) - Published

**Solution:** (RA-LIB) Process 5 Task:

---

Parameters Bus View\_Assignment

- ▼ 00.00 - Org
  - HWOrgViewSize 4
- ▼ 00.01 - Org Scan Data - Common
  - Scan\_Library raP-5\_20
  - Scan\_Instruction raP\_Opr\_OrgScan
  - Scan\_Label HWOrgScan
  - Scan\_Area Area01
- ▼ 00.02 - Org View Data - Common
  - View\_Library raP-5\_20
  - View\_Instruction raP\_Opr\_OrgView
  - View\_Area Area01
  - View\_Label HWOrgScan
  - View\_Area\_01 Area01
  - View\_Label\_01 OrgView
  - View\_Area\_02 Area01
  - View\_Label\_02 OrgView
  - View\_Area\_03 Area01
  - View\_Label\_03 OrgView

5. From the Process library > Hardware Monitoring > Specialty folder, add a Hardware\_Module\_Status object to the Class View.

- a. Configure the Module parameter for the module.
- b. Assign a unique Bus\_Instance on the Hardware\_Bus.

**Name:** HWMS\_XT400\_PA\_Link

**Description:**

**Catalog Number:** Hardware\_Module\_Status (4.0) - Published

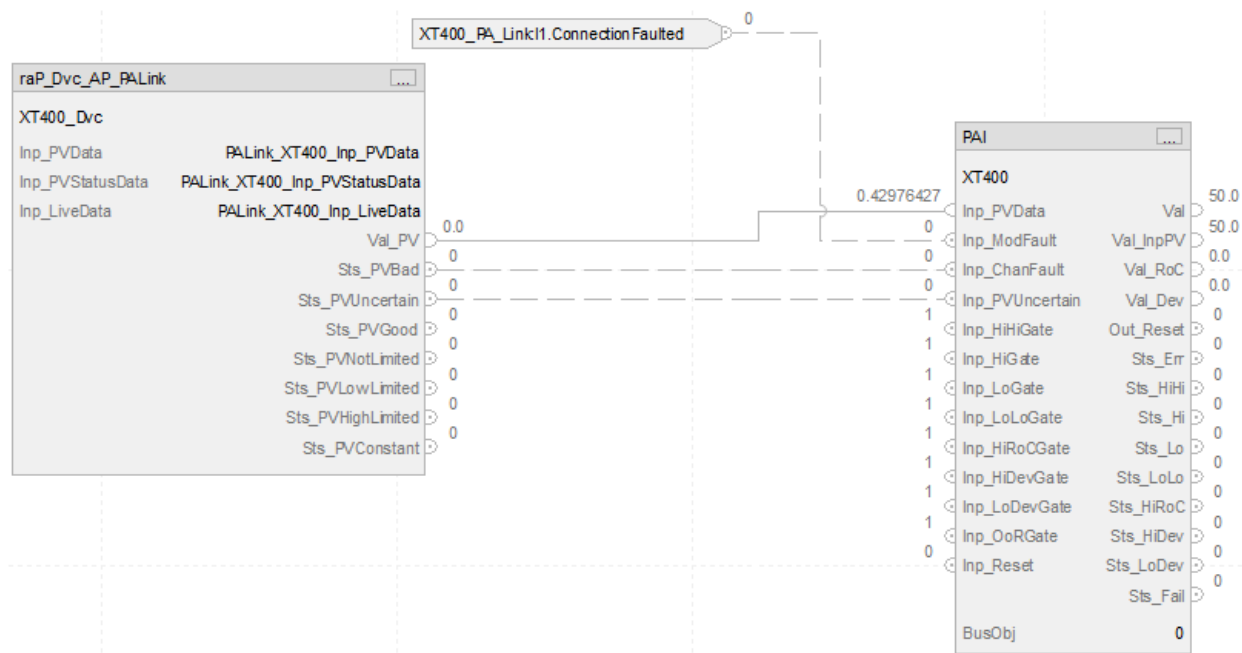
**Solution:** (RA-LIB) Process 5

---

Parameters

- ▼ 00.01 - Data - Common
  - Area Area01
  - Instruction raP\_Dvc\_LgxModuleSts
  - Label Module\_Status
  - Library raP-5\_20
- ▼ 01 - Options
  - Module  XT400\_PA\_Link
  - Bus\_Instance  HWBus\_Org.Bus.Cmd\_1

6. Generate the controller ACD file.



## Controller Fault Handler Control Strategy

If a fault condition occurs that prevents an instruction from running, the instruction aborts, and the controller reports a major fault. A major fault halts logic execution and the controller switches to faulted mode (the OK status indicator flashes red). Depending on the application, you may not want all major faults to shut down the system. If you do not want all major faults to shut down the system, create a fault routine to clear the fault and let the application continue to run. The process of resuming execution after the fault clears is known as fault recovery.

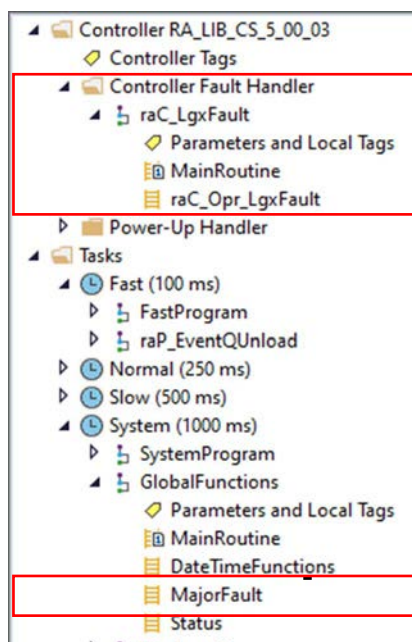
The Controller Fault Handler control strategy has the following functionality:

- Get fault information for the active fault
- Generate an alarm that a major fault occurred
- Record the last 10 major faults that occurred containing a time stamp
- Configuration to automatically clear major fault to help prevent controller from faulting (use selectively)

The Controller Fault Handler control strategy requires:

- a raC\_LgxFault **program** (with raC\_Opr\_LgxFlt and MainRoutine **routines**) in the Controller Fault Handler folder,
- and a MajorFault **routine** in the GlobalFunctions Program.

Import the **routines** into the correct program.



# Example Programs

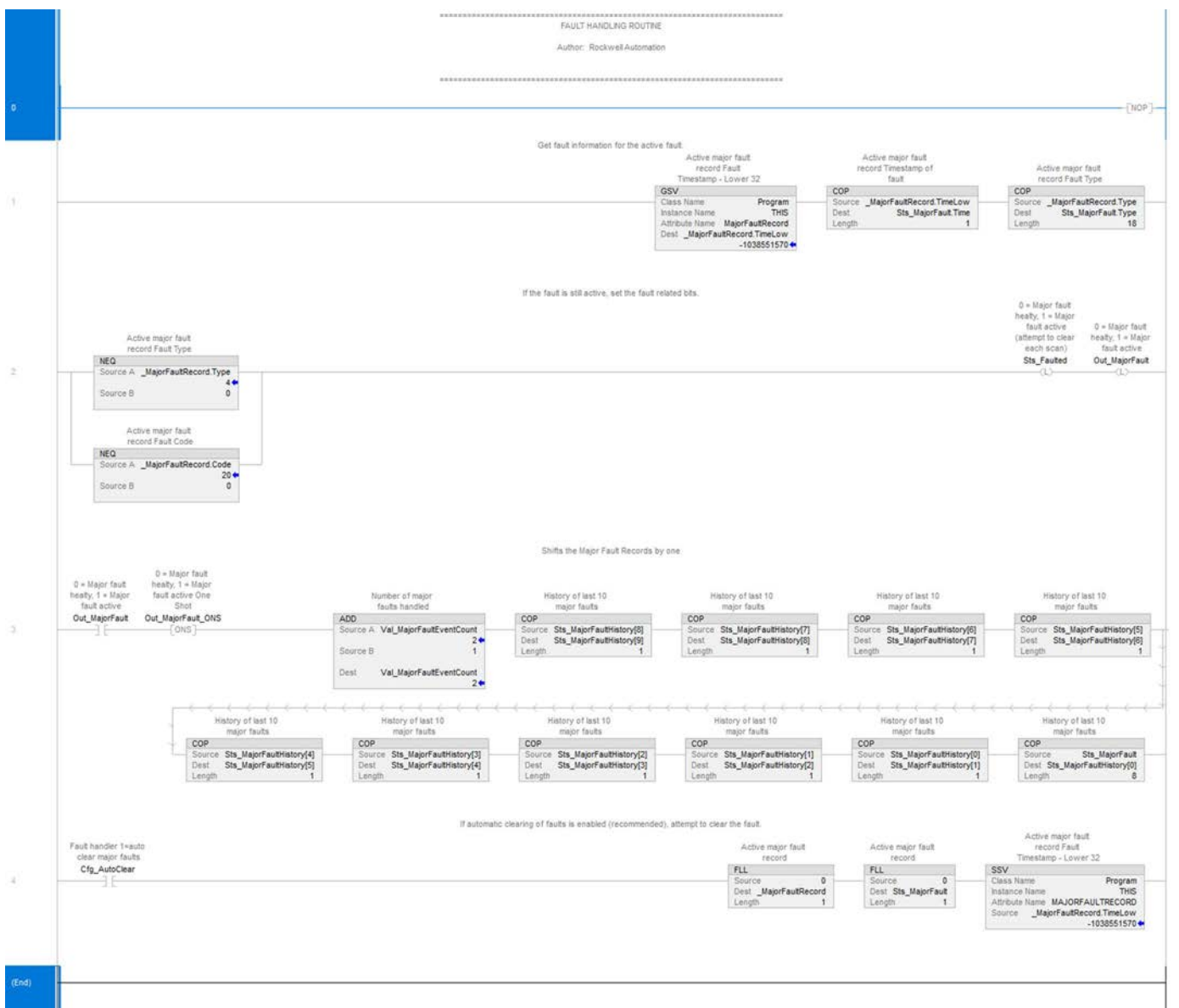
In a system that uses recipe numbers as indirect addresses, a mistyped number could produce a major fault, such as type 4, code 20.

To keep the entire system from shutting down, a fault routine clears any type 4, code 20, major faults.

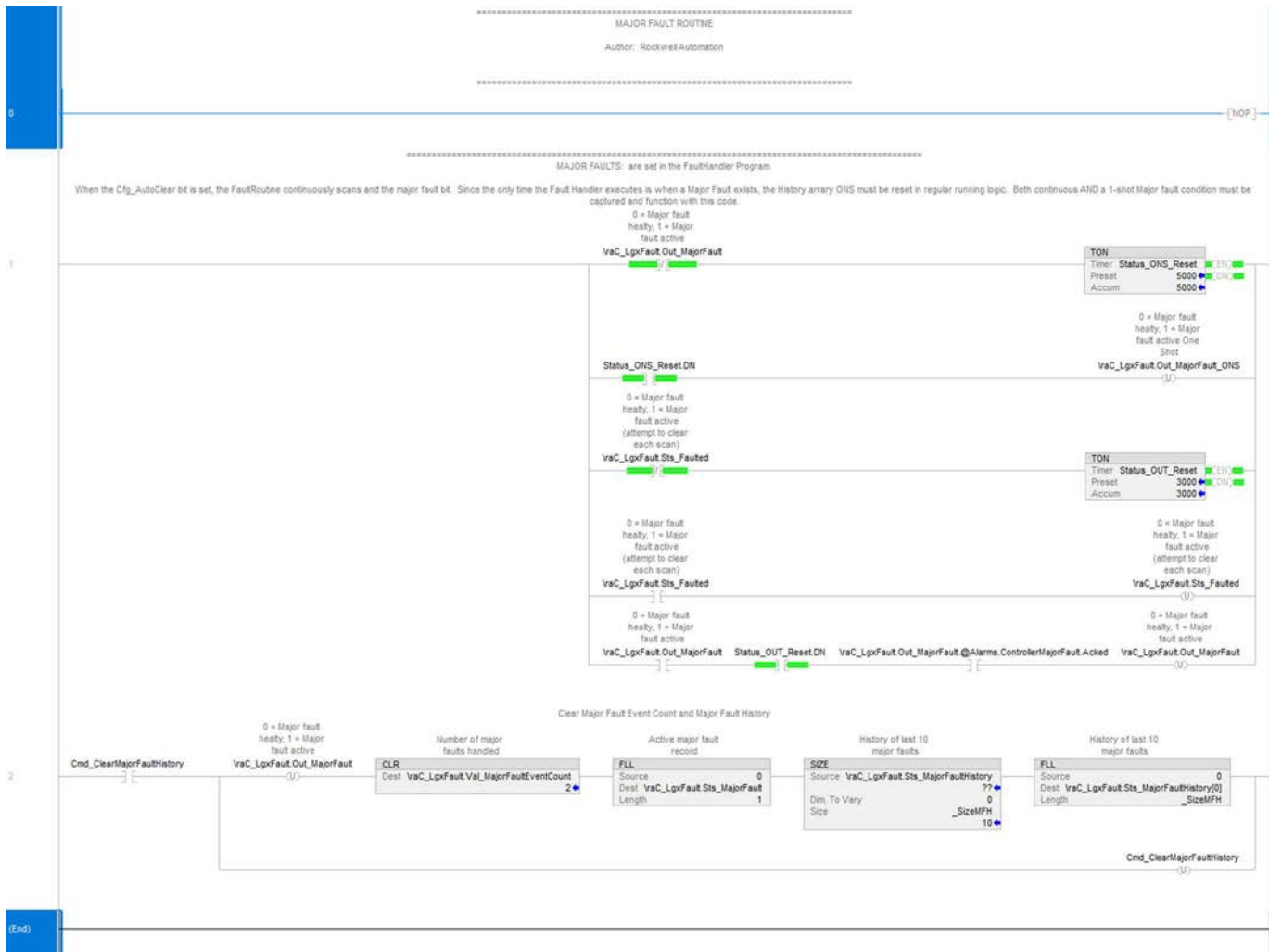
## Controller Fault Handler Program: MainRoutine Routine



## Controller Fault Handler Program: raC\_Opr\_LgxFault Routine



## GlobalFunctions Program: MajorFault Routine



For more information on handling controller faults, see the Logix 5000® Controllers Major, Minor, and I/O Faults Programming Manual, publication [1756-PM014](#).

**Notes:**



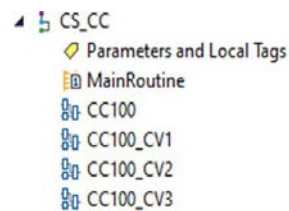
## Coordinated Control (CC) Control Strategies

Use the CC control strategy to control one process variable by manipulating up to three different control variables. Any of the three outputs can be used as an input to create feed forward action in the controller. The CC instruction calculates the control variables (CV1, CV2, and CV3) in the auto mode based on the PV - SP deviation, internal models, and tuning.

The CC control strategy is a model-based instruction, where as many as three models can be configured to relate the output of each CV to the single PV. Each model is a first order plus delay (FOPD) response, which is more effective than PID controllers for controlling processes with long deadtimes. The CC control strategy coordinates the action of the CVs to limit interactions among the CVs.

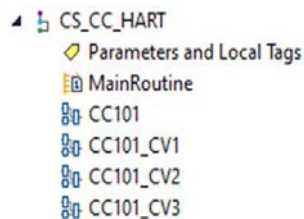
The CC control strategy is available as four routines in the process library:

Routine	Description
CC100	Coordinated Control instruction.
CC100_CV1 CC100_CV2 CC100_CV3	Control variable routines.



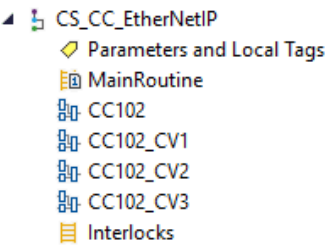
The CC HART control strategy is available as four routines in the process library:

Routine	Description
CC101	Coordinated Control instruction with HART input in the CC101 routine.
CC101_CV1 CC101_CV2 CC101_CV3	Control variable routines.



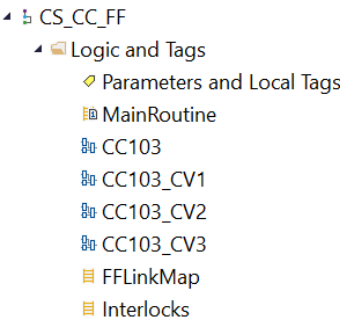
The CC EtherNetIP control strategy is available as four routines in the process library:

Routine	Description
CC102	Coordinated Control instruction with EtherNetIP input in the CC102 routine.
CC102_CV1 CC102_CV2 CC102_CV3	Control variable routines.



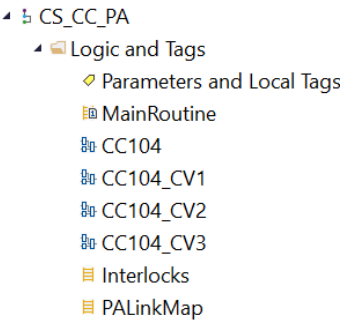
The CC FOUNDATION Fieldbus control strategy is available as four routines in the process library:

Routine	Description
CC103	Coordinated Control instruction with FOUNDATION Fieldbus input in the CC103 routine.
CC103_CV1 CC103_CV2 CC103_CV3	Control variable routines.



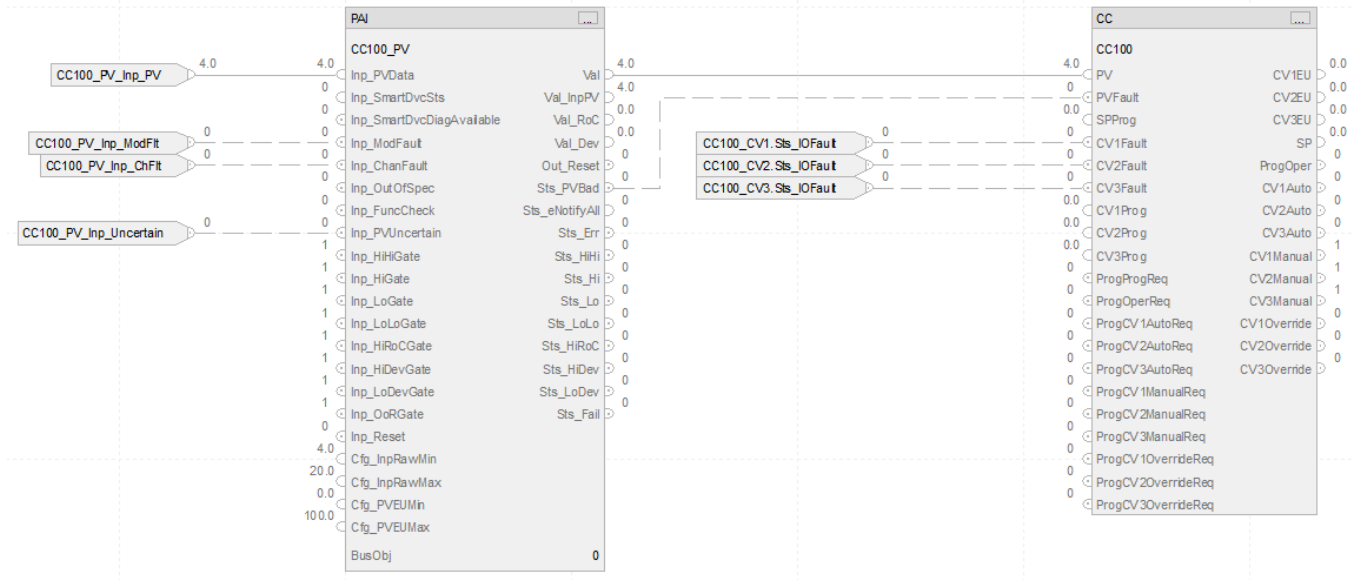
The CC Profibus PA control strategy is available as four routines in the process library:

Routine	Description
CC104	Coordinated Control instruction with Profibus PA input in the CC104 routine.
CC104_CV1 CC104_CV2 CC104_CV3	Control variable routines.



Import the appropriate control strategy as a **program** in your controller project.

# CS\_CC Sheet



## PAI Input References

See [CS\\_PA1 Sheet on page 148](#) for details.

- Substitute CC100 for XT101

## PAI Outputs to CC Inputs

Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)  Source: Analog input channel or upstream REAL tag representing position feedback
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad

## Input References to CC

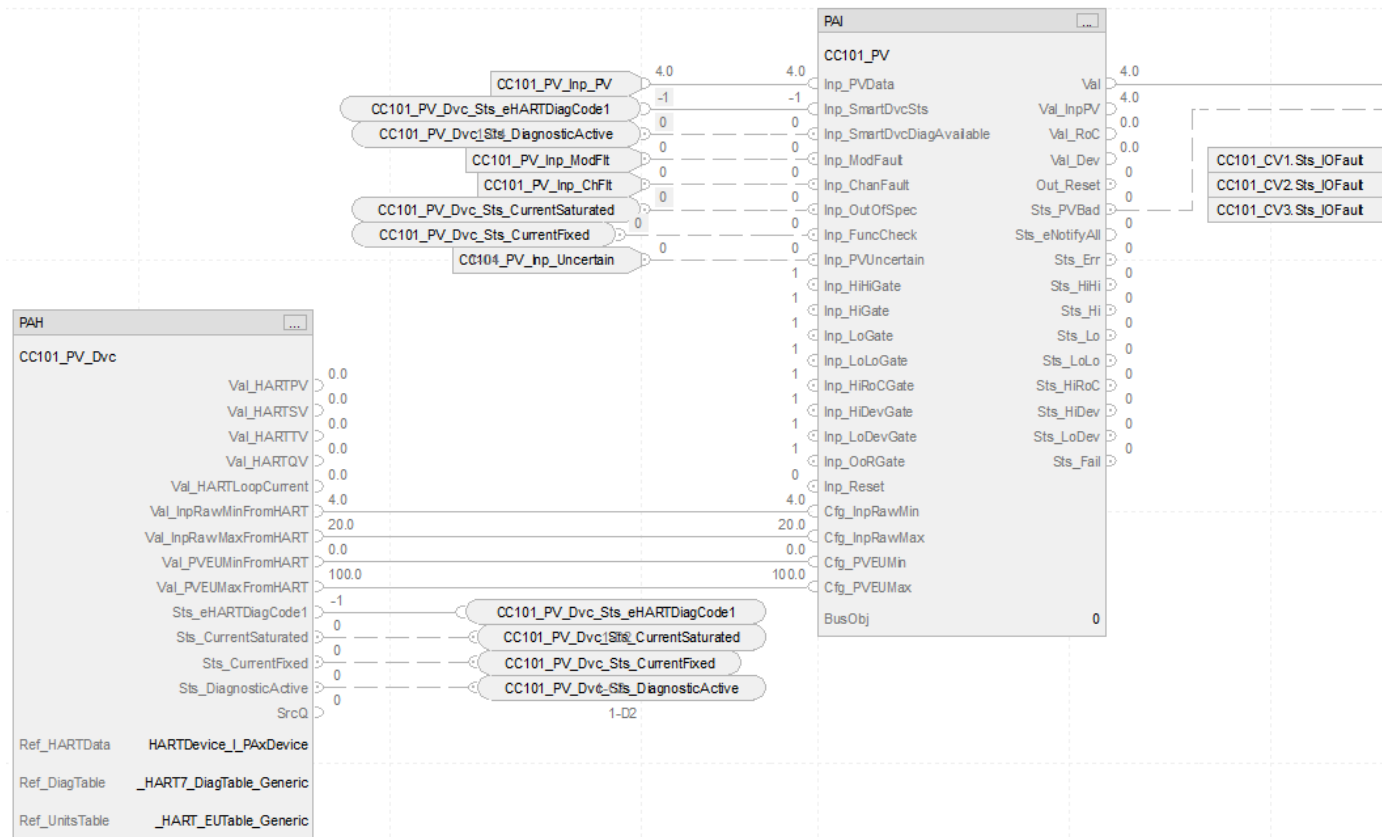
Parameter	Description
CC_100_CV1.Sts.IOFault	Control variable 1 fault input <ul style="list-style-type: none"> <li>• If CV1EU controls an analog output, then CV1Fault will normally come from the analog output's fault status.</li> <li>• If CV1Fault is TRUE, it indicates an error on the output module, set bit in Status.</li> </ul>
CC_100_CV2.Sts.IOFault	Control variable 2 fault input <ul style="list-style-type: none"> <li>• If CV2EU controls an analog output, then CV2Fault will normally come from the analog output's fault status.</li> <li>• If CV2Fault is TRUE, it indicates an error on the output module, set bit in Status.</li> </ul>
CC_100_CV3.Sts.IOFault	Control variable 3 fault input <ul style="list-style-type: none"> <li>• If CV3EU controls an analog output, then CV3Fault will normally come from the analog output's fault status.</li> <li>• If CV3Fault is TRUE, it indicates an error on the output module, set bit in Status.</li> </ul>

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## CC Configuration Considerations

Operand	Type	Description
CC tag	COORDINATED_CONTROL	Instance of data structure (backing tag) required for proper operation of instruction.

CS\_CC\_HART Sheet

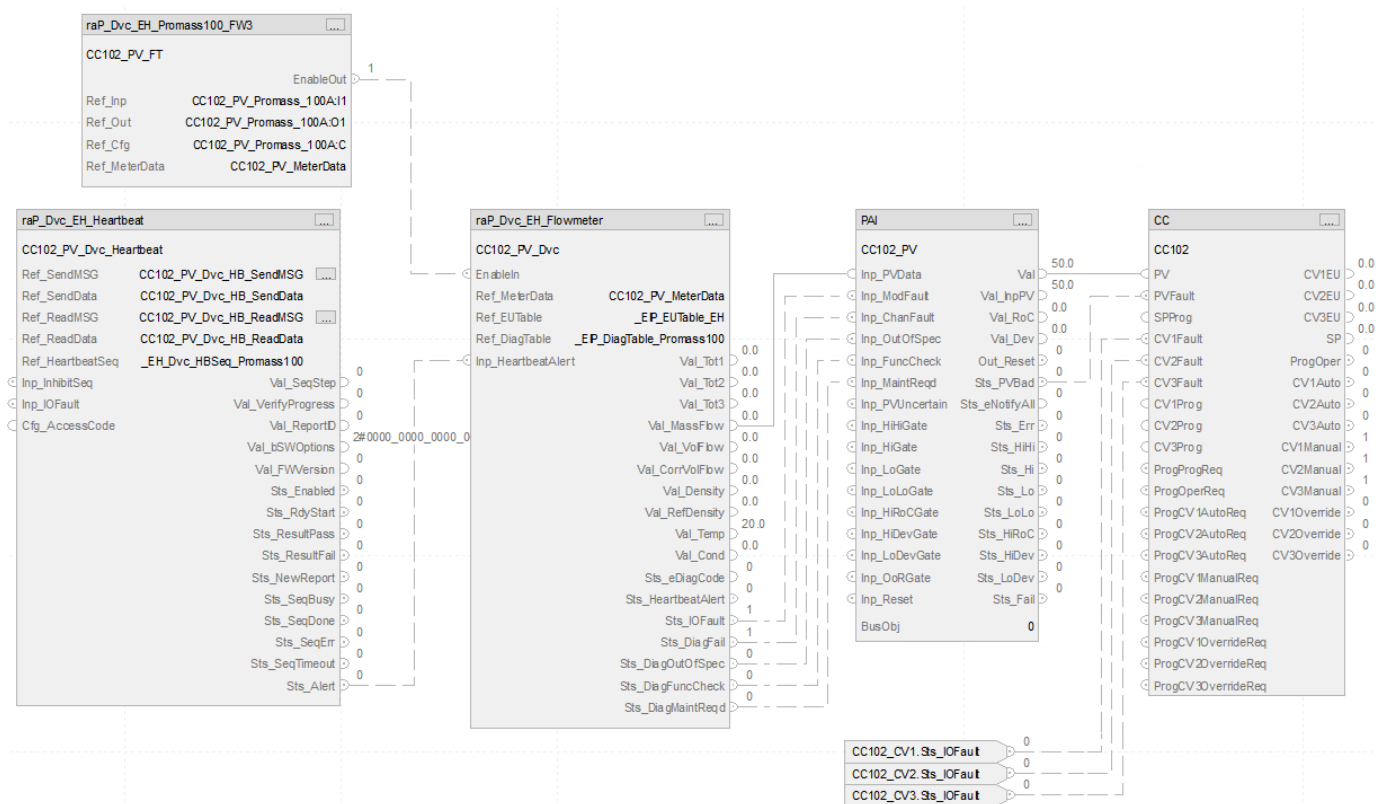


The CS\_CC\_Hart control strategy operates the same as the CS\_CC control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute CC101 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_CC\_EtherNetIP Sheet

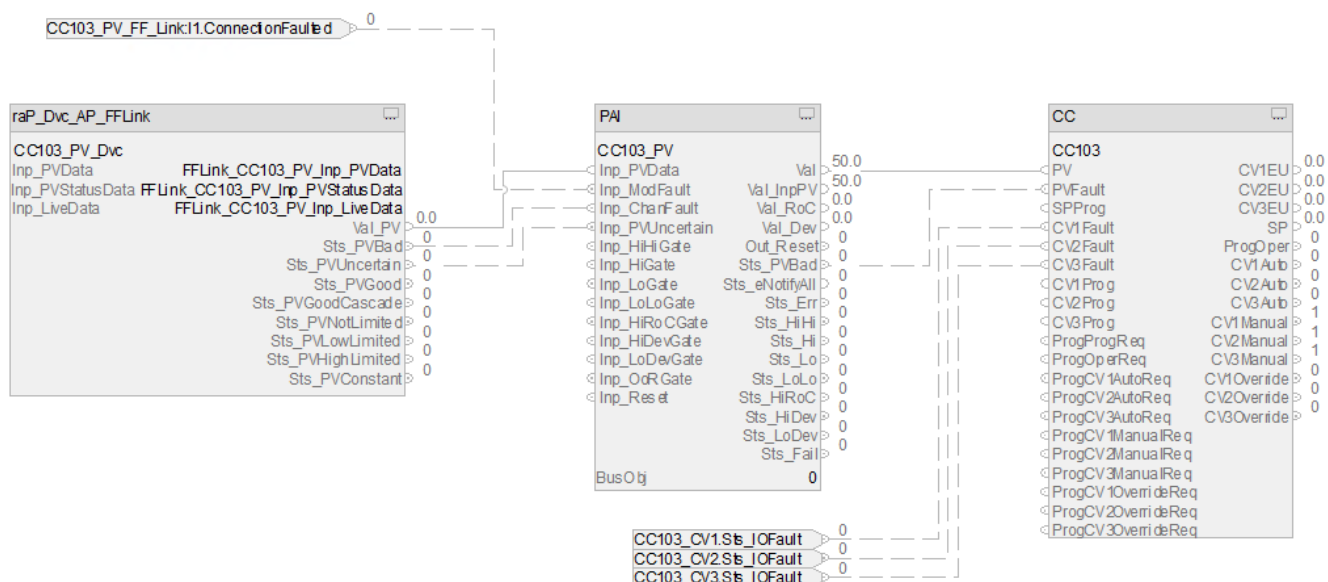


The CS\_CC\_EtherNetIP control strategy operates the same as the CS\_CC control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute CC102 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_CC\_FF Sheet

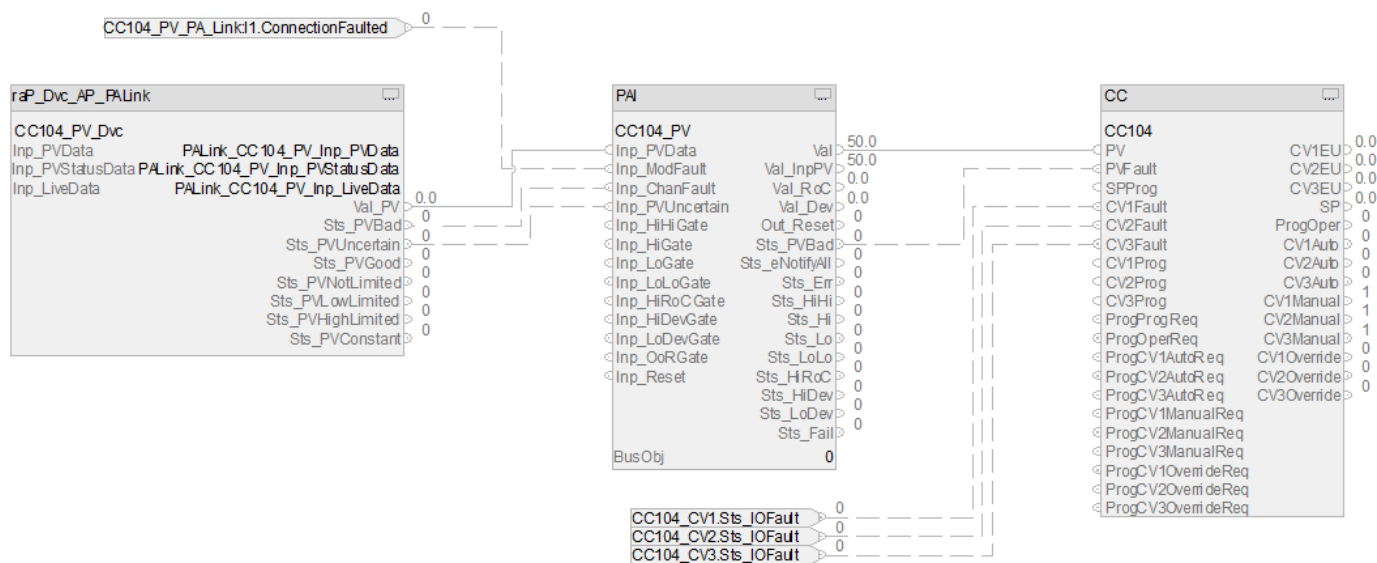


The CS\_CC\_FF control strategy operates the same as the CS\_CC control strategy but relies on FOUNDATION Fieldbus input data.

- For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute CC103 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_CC\_PA Sheet



The CS\_CC\_PA control strategy operates the same as the CS\_CC control strategy but relies on Profibus PA input data.

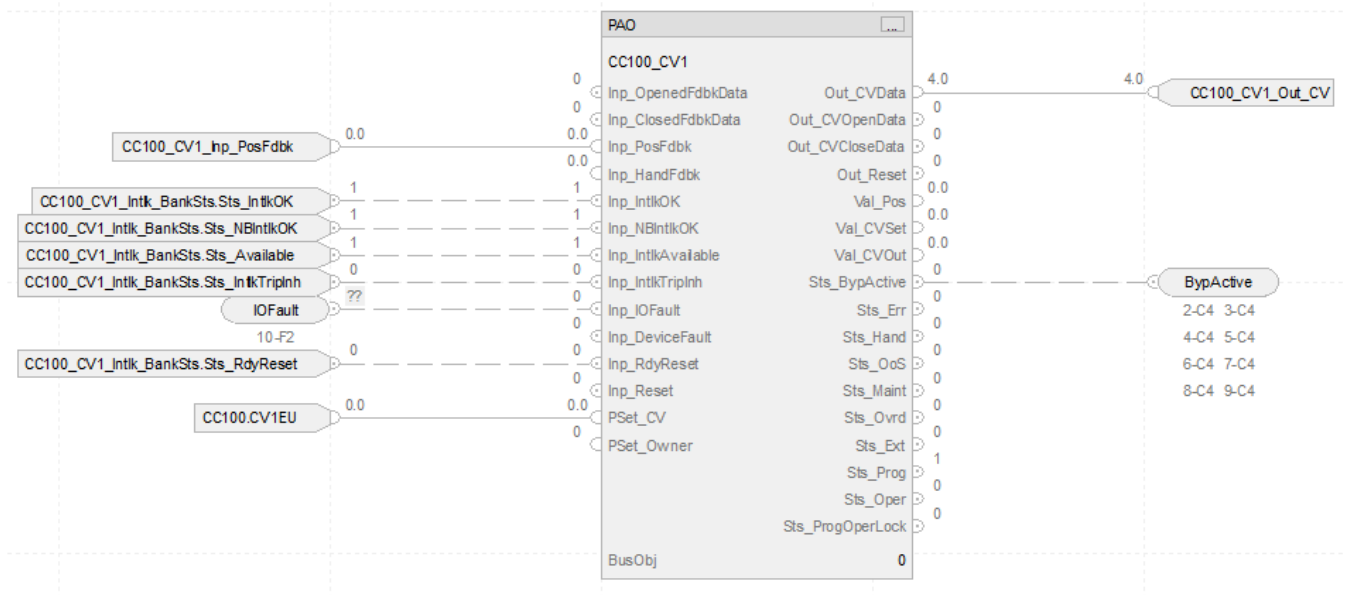
- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute CC104 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## Control Variable Routines

This control variable routines include the PAO control strategy, with an additional input reference.

- For PAO configuration considerations, and input and output references, see [CS\\_PAO Sheet on page 180](#).
- The routines also include these PAO input references:



Parameter	Description
CC100.CV1EU	<p>Scaled control variable output for CV1. Scaled by using CV1EUMax and CV1EUMin, where CV1EUMax corresponds to 100% and CV1EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV1EU = (CV1 * CV1EUSpan / 100) + CV1EUMin$ <p>CV1EU span calculation: <math>CV1EUSpan = (CV1EUMax - CV1EUMin)</math></p>
CC100.CV2EU	<p>Scaled control variable output for CV2. Scaled by using CV2EUMax and CV2EUMin, where CV2EUMax corresponds to 100% and CV2EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV2EU = (CV2 * CV2EUSpan / 100) + CV2EUMin$ <p>CV2EU span calculation: <math>CV2EUSpan = (CV2EUMax - CV2EUMin)</math></p>
CC100.CV3EU	<p>Scaled control variable output for CV3. Scaled by using CV3EUMax and CV3EUMin, where CV3EUMax corresponds to 100% and CV3EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV3EU = (CV3 * CV3EUSpan / 100) + CV3EUMin$ <p>CV3EU span calculation: <math>CV3EUSpan = (CV3EUMax - CV3EUMin)</math></p>

## ACM Considerations for CC

Configure these parameters first because they affect the visibility of the remaining parameters in the CC object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for a CC Instance

Parameter	Visible When	Details
00 - Selection		
PAI_Type	always	Important: Select this parameter first as the option affects the remaining parameters. Define the PAI type: <ul style="list-style-type: none"><li>PAI(Single_channel),</li><li>PAID(Dual_channel),</li><li>PAIM(Multi_channel), or</li><li>External PAI(Single_channel)</li></ul>
Localize_PAO	always	Set to use a local routine in the program
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
01 - Options		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
02 - Device Configuration		
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
03.00 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PAI_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV input reference
	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=HART	
03.00.10 - Ref PAI Alarm Configuration		



Parameter	Visible When	Details
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_HiGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_LoGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.
Ref_OoRGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference.

**03.11 - IO Configuration**

PA01_RefTag	always	Link to first analog output reference
Cfg_HasCV1IntlkObj	Localize_PA0=True	Set if the analog output reference has an interlock CV
Bus_Instance_CV1	PA01_RefTag is linked to an analog output reference. Localize_PA0=True Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk1	Localize_PA0=True	Link to input position feedback
Out_CV1	Localize_PA0=True	Link to the first output CV reference

**03.12 - IO Configuration**

PA02_RefTag	always	Link to second analog output reference
Cfg_HasCV2IntlkObj	Localize_PA0=True	Set if the analog output reference has an interlock CV
Bus_Instance_CV2	PA02_RefTag is linked to an analog output reference. Localize_PA0=True Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk2	Localize_PA0=True	Link to input position feedback
Out_CV2	Localize_PA0=True	Link to the second output CV reference

**03.13 - IO Configuration**

PA03_RefTag	always	Link to third analog output reference
Cfg_HasCV3IntlkObj	Localize_PA0=True	Set if the analog output reference has an interlock CV
Bus_Instance_CV3	PA03_RefTag is linked to an analog output reference. Localize_PA0=True Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk3	Localize_PA0=True	Link to input position feedback
Out_CV3	Localize_PA0=True	Link to the third output CV reference

## Additional Sub-Objects for a CC Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CV1Intlk CV2Intlk CV3Intlk	Configure an interlock for the CV instance. See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy. See <a href="#">Event Logging on page 49</a>
CV1Events CV2Events CV3Events	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>

## Internal Model Control (IMC) Control Strategies

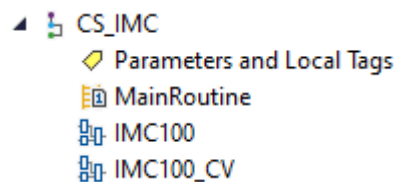
Use the IMC control strategy to control a single process variable by manipulating a single control-variable output. This control strategy performs an algorithm where the actual error signal is compared against that of an internal first-order lag plus deadtime model of the process. The IMC instruction calculates the control variable output (CV) in the Auto mode based on the PV - SP deviation, internal model, and tuning. IMC is a model-based instructions that is more effective than PID control for processes with long deadtimes.

The following IMC control strategies are available as routines in the process library:

- CS\_IMC
- CS\_IMC\_HART
- CS\_IMC\_EtherNetIP
- CS\_IMC\_FF
- CS\_IMC\_PA

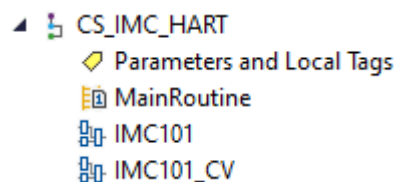
The IMC control strategy is available as two routines in the process library:

Routine	Description
IMC100	Internal Model Control instruction.
IMC100_CV1	Control variable routine.



The IMC HART control strategy is available as two routines in the process library:

Routine	Description
IMC101	Internal Model Control instruction with HART input.
IMC101_CV1	Control variable routine.



The IMC EtherNet/IP control strategy is available as two routines in the process library:

Routine	Description
IMC102	Internal Model Control instruction with EtherNet/IP input.
IMC102_CV	Control variable routine.

CS\_IMC\_EtherNetIP

Parameters and Local Tags

MainRoutine

IMC102

IMC102\_CV

Interlocks

The IMC FOUNDATION Fieldbus control strategy is available as two routines in the process library:

Routine	Description
IMC103	Internal Model Control instruction with FOUNDATION Fieldbus input.
IMC103_CV	Control variable routine.

CS\_IMC\_FF

Logic and Tags

Parameters and Local Tags

MainRoutine

FFLinkMap

IMC103

IMC103\_CV

Interlocks

The IMC Profibus PA control strategy is available as two routines in the process library:

Routine	Description
IMC104	Internal Model Control instruction with Profibus PA input.
IMC104_CV	Control variable routine.

CS\_IMC\_PA

Logic and Tags

Parameters and Local Tags

MainRoutine

IMC104

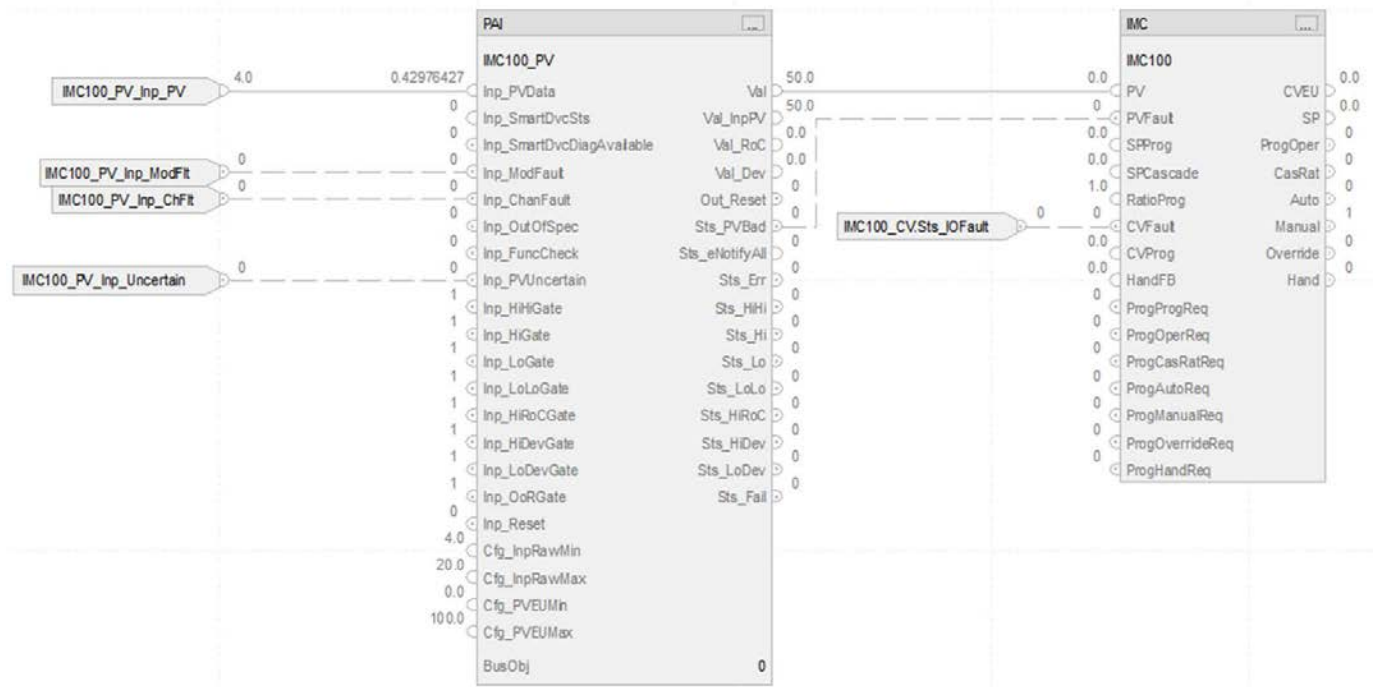
IMC104\_CV

Interlocks

PALinkMap

Import the appropriate control strategy as a **program** in your controller project.

## CS\_IMC Sheet



### PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

- Substitute IMC100 for XT101

### PAI Outputs to IMC Inputs

Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)  Source: Analog input channel or upstream REAL tag that represents position feedback
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad

### IMC Input Reference

Parameter	Description
IMC_100.CV.Sts.IOFault	Control variable fault input <ul style="list-style-type: none"> <li>If CVEU controls an analog output, then CVFault normally comes from the analog output's fault status.</li> <li>If CVFault is TRUE, it indicates an error on the output module, set bit in Status.</li> </ul>

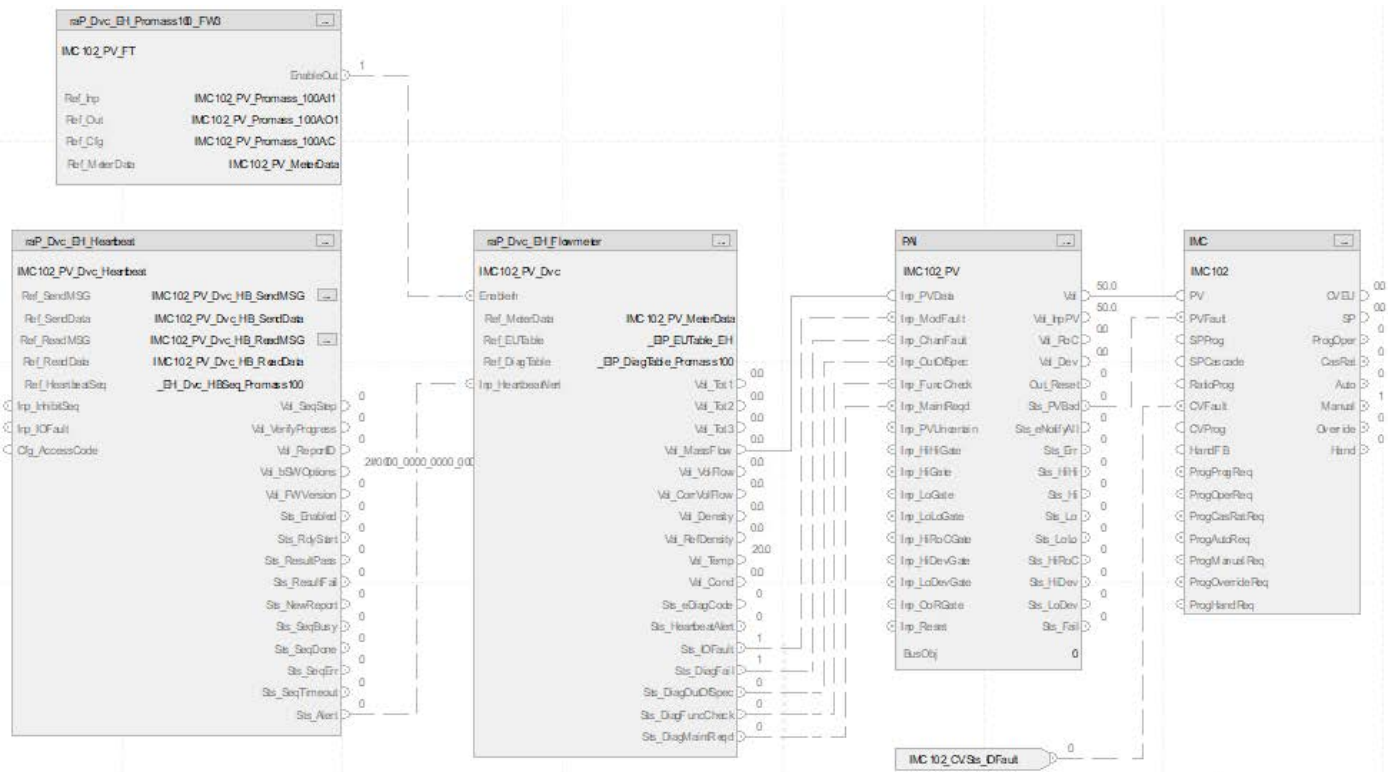
For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

### IMC Configuration Considerations

Operand	Type	Description
IMC tag	Internal Model Control	Instance of data structure (backing tag) required for proper operation of instruction



## CS\_IMC\_EtherNetIP Sheet

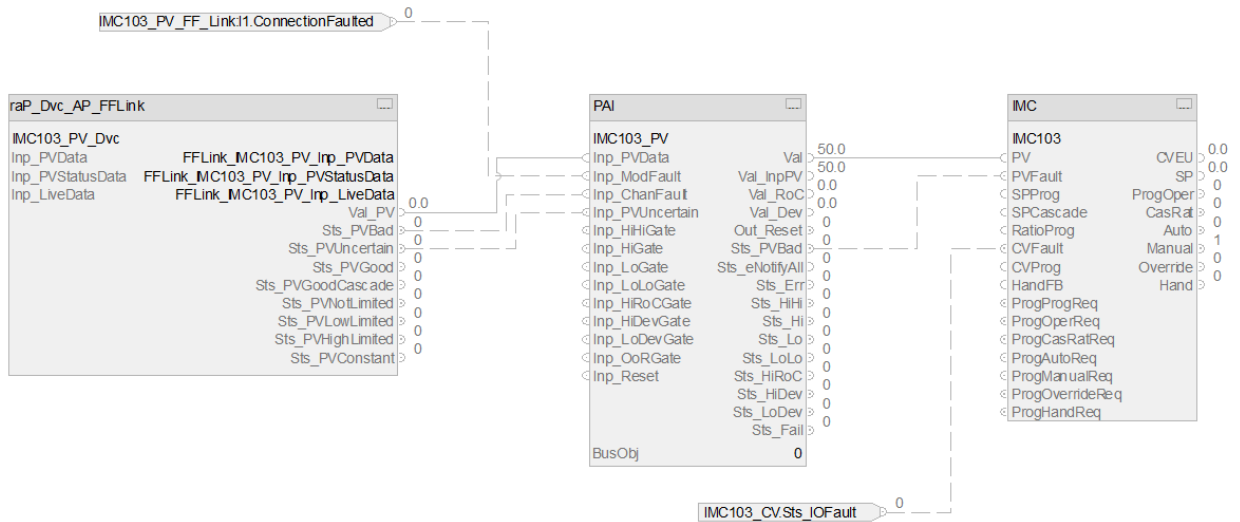


The CS\_IMC\_EtherNetIP control strategy operates the same as the CS\_IMC control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute IMC102 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_IMC\_FF Sheet



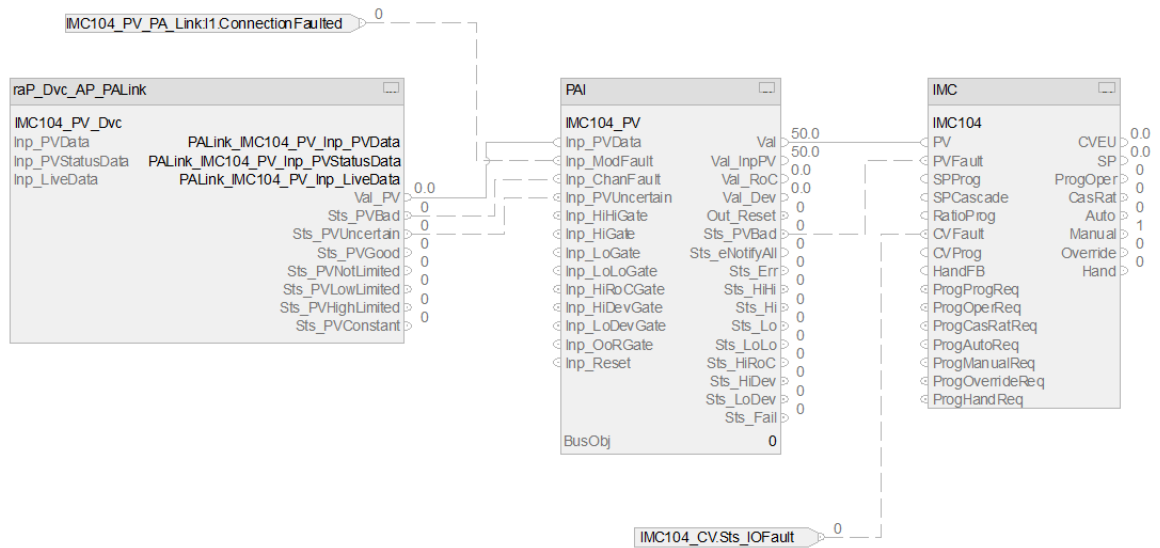
The CS\_IMC\_FOUNDATION Fieldbus control strategy operates the same as the CS\_IMC control strategy but relies on FOUNDATION Fieldbus input data.

For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).

- Substitute IMC103 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_IMC\_PA Sheet



The CS\_IMC\_PA control strategy operates the same as the CS\_IMC control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute IMC104 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

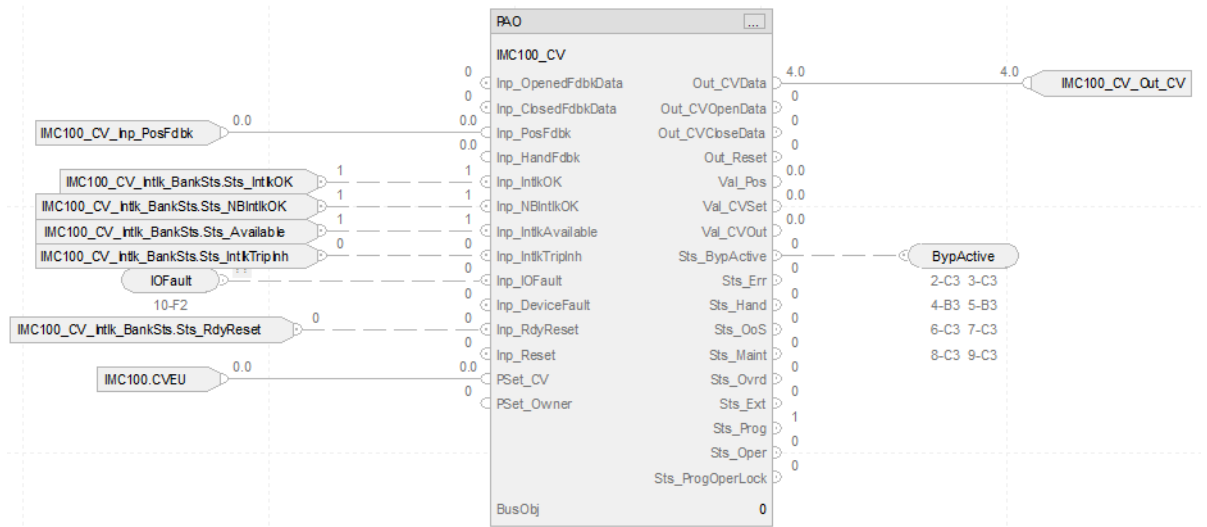


## Control Variable Routines

This control variable routines include the PAO control strategy, with an additional input reference.

- For PAO configuration considerations, and input and output references, see [CS\\_PAO Sheet on page 180](#) for details.
- The routine also includes this PAO input reference:

Parameter	Description
IMC100.CVEU	Scaled control variable output for CV1. Scaled by using CV1EUMax and CV1EUMin, where CV1EUMax corresponds to 100% and CV1EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.  $CV1EU = (CV1 * CV1EUSpan / 100) + CV1EUMin$ $CV1EU \text{ span calculation: } CV1EUSpan = (CV1EUMax - CV1EUMin)$



## ACM Considerations for IMC

Configure these parameters first because they affect the visibility of the remaining parameters in the IMC object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for a IMC Instance

Parameter	Visible When	Details
00 - Selection		
PAI_Type	always	Important: Select this parameter first as the option affects the remaining parameters. Define the PAI type: <ul style="list-style-type: none"><li>PAI(Single_channel),</li><li>PAID(Dual_channel),</li><li>PAIM(Multi_channel), or</li><li>External PAI(Single_channel)</li></ul>
Localize_PA0	always	Set to use a local routine in the program
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
01 - Options		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object.
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
03.00 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PAI_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV input reference
	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel) IO_SignalType=HART	
03.00.10 - Ref PAI Alarm Configuration		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_HiGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)\	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference

### 03.11 - IO Configuration

PAO_RefTag	always	Link to the analog output reference
Cfg_HasCVIntlkObj	Localize_PAO=True	Set if the analog output reference has an interlock CV
UseResetWireConnectors	Cfg_HasCVIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance_CV	PAO_RefTag is linked to an analog out reference. Has_OOAP=True (controller parameter) Use_OOAP=True Localize_PAO=True	Link to a bus array instance. This should be unique for each device.
Inp_PosFdbk	Localize_PAO=True	Link to the input position feedback reference
Out_CV	Localize_PAO=True	Link to the output CV reference

## Additional Sub-Objects for an IMC Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CVIntlk	Configure an interlock for the CV instance. See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CVEvents	Configure an event to monitor for the CV instance. See <a href="#">Event Logging on page 49</a>

**Notes:**

## Modular Multivariable Control (MMC) Control Strategies

Use the MMC control strategy to control two process variables to their setpoints using as many as three control variables. The MMC instruction calculates the control variables (CV1, CV2, and CV3) in the auto mode based on the PV1 - SP1, PV2 - SP2 deviation, internal model, and tuning.

The MMC controller is a model-based instruction, where you can configure as many as six models to relate the output of each CV to the two PVs. Each model is a first order plus delay (FOPD) response, which is more effective than PID controllers at controlling processes with long deadtimes. The MMC control strategy coordinates the actions of the CVs to limit interactions among the CVs and control the 2 PVs to their respective setpoints.

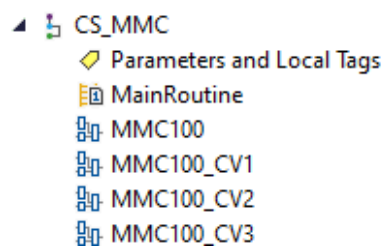
The following MMC control strategies are available as routines in the process library:

- CS\_MMC
- CS\_MMC\_HART
- CS\_MMC\_EtherNetIP
- CS\_MMC\_FF
- CS\_MMC\_PA

Import the appropriate control strategy as a **program** in your controller project.

The MMC control strategy is available as four routines in the process library:

Routine	Description
MMC100	Modular Multivariable Control instruction.
MMC100_CV1 MMC100_CV2 MMC100_CV3	Control variable routines.



The MMC HART control strategy is available as four routines in the process library:

Routine	Description
MMC101	Modular Multivariable Control instruction with HART input in the CC101 routine.
MMC101_CV1 MMC101_CV2 MMC101_CV3	Control variable routines.

- CS\_MMC\_HART
  - Parameters and Local Tags
  - MainRoutine
  - MMC101
  - MMC101\_CV1
  - MMC101\_CV2
  - MMC101\_CV3

The MMC EtherNet/IP control strategy is available as four routines in the process library:

Routine	Description
MMC102	Modular Multivariable Control instruction.
MMC102_CV1 MMC102_CV2 MMC102_CV3	Control variable routines.

- CS\_MMC\_EtherNetIP
  - Parameters and Local Tags
  - MainRoutine
  - Interlocks
  - MMC102
  - MMC102\_CV1
  - MMC102\_CV2
  - MMC102\_CV3

The MMC FOUNDATION Fieldbus control strategy is available as four routines in the process library:

Routine	Description
MMC103	Modular Multivariable Control instruction.
MMC103_CV1 MMC103_CV2 MMC103_CV3	Control variable routines.

- CS\_MMC\_FF
  - Logic and Tags
    - Parameters and Local Tags
  - MainRoutine
  - FFLinkMap
  - Interlocks
  - MMC103
  - MMC103\_CV1
  - MMC103\_CV2
  - MMC103\_CV3

The MMC Profibus PA control strategy is available as four routines in the process library:

Routine	Description
MMC104	Modular Multivariable Control instruction.
MMC104_CV1 MMC104_CV2 MMC104_CV3	Control variable routines.

#### CS\_MMC\_PA

##### Logic and Tags

##### Parameters and Local Tags

##### MainRoutine

##### Interlocks

##### MMC104

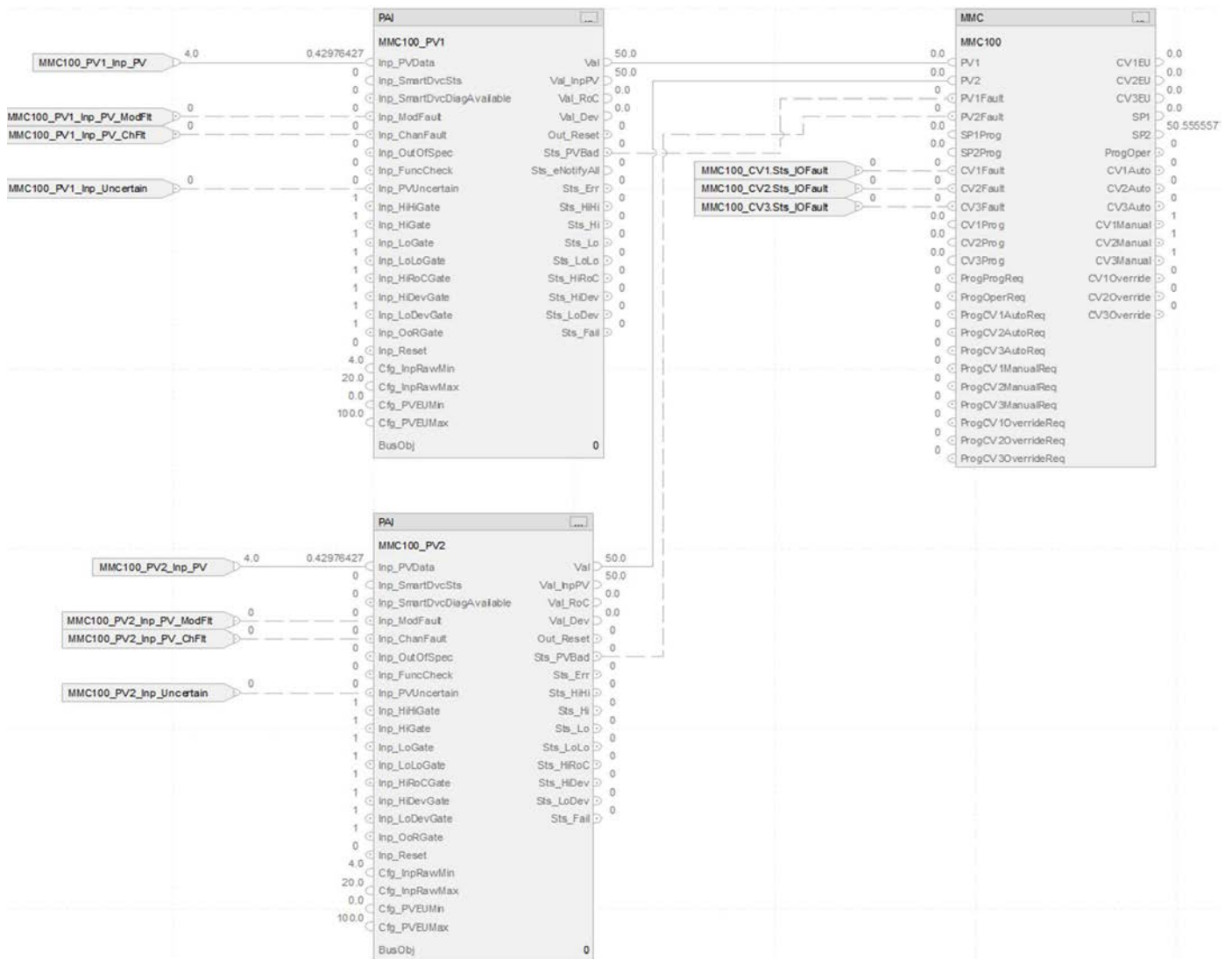
##### MMC104\_CV1

##### MMC104\_CV2

##### MMC104\_CV3

##### PALinkMap

## CS\_MMC Sheet



## PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

- Substitute MMC100\_PV1 for the first instance of XT101
- Substitute MMC100\_PV2 for the second instance of XT101

## PAI Outputs to MMC Inputs

One instance for PV1 and a second instance for PV2.

Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)  Source: Analog input channel or upstream REAL tag that represents position feedback
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad

## MMC Input References

Parameter	Description
MMC_100_CV1.Sts_IOFault	Control variable 1 fault input If CV1EU controls an analog output, then CV1Fault normally comes from the analog output's fault status. If CV1Fault is TRUE, it indicates an error on the output module, set bit in Status.
MMC_100_CV2.Sts_IOFault	Control variable 2 fault input If CV2EU controls an analog output, then CV2Fault normally comes from the analog output's fault status. If CV2Fault is TRUE, it indicates an error on the output module, set bit in Status.
MMC_100_CV3.Sts_IOFault	Control variable 3 fault input If CV3EU controls an analog output, then CV3Fault normally comes from the analog output's fault status. If CV3Fault is TRUE, it indicates an error on the output module, set bit in Status.

For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

## MMC Configuration Considerations

Operand	Type	Description
MMC tag	MODULAR MULTIVARIABLE CONTROL	Instance of data structure (backing tag) required for proper operation of instruction



## CS\_MMC\_HART Sheet

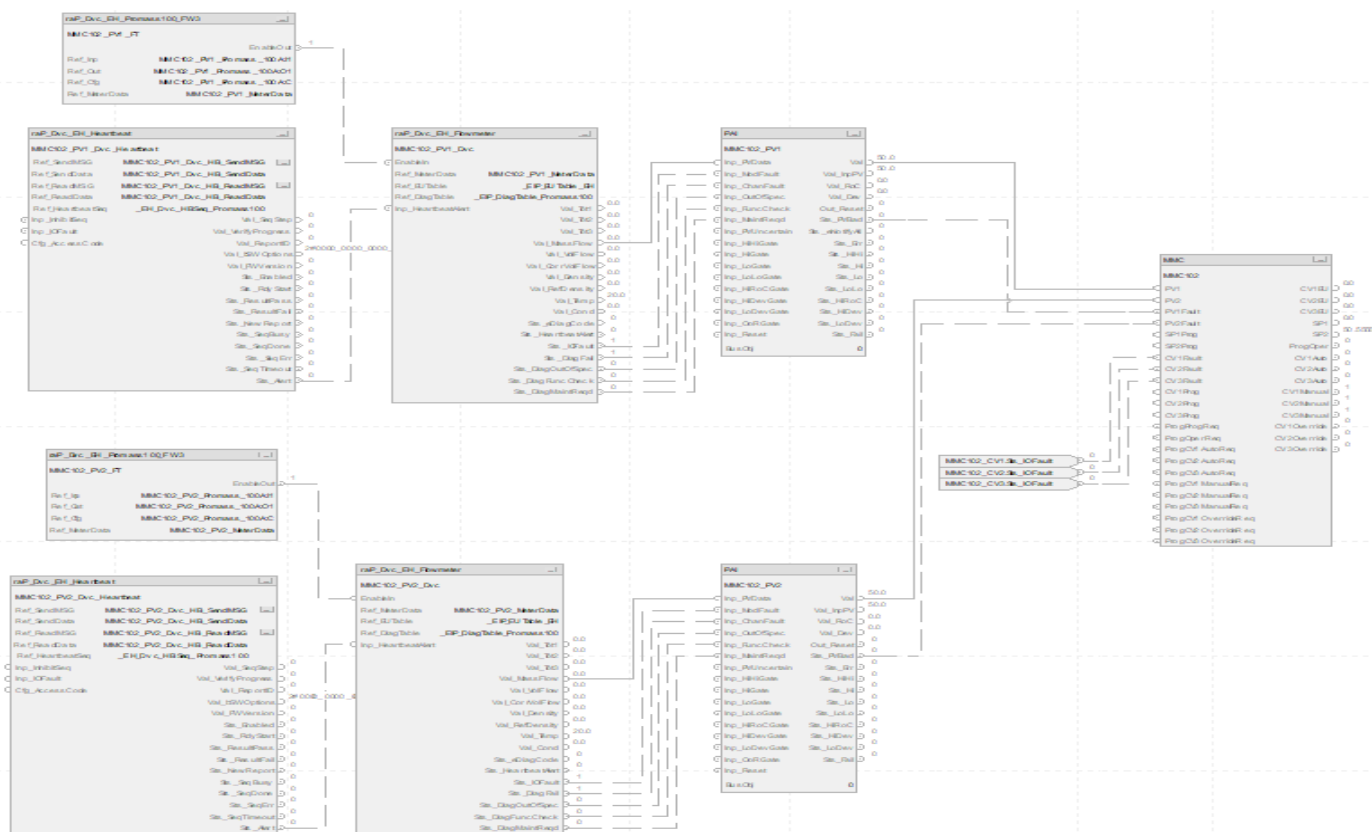


The CS\_MMC\_Hart control strategy operates the same as the CS\_MMC control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
  - Substitute MMC101\_PV1 for the first instance of XT100
  - Substitute MMC101\_PV2 for the second instance of XT100

For more information, see [HART Integration on page 61](#).

## CS\_MMC\_EtherNetIP Sheet

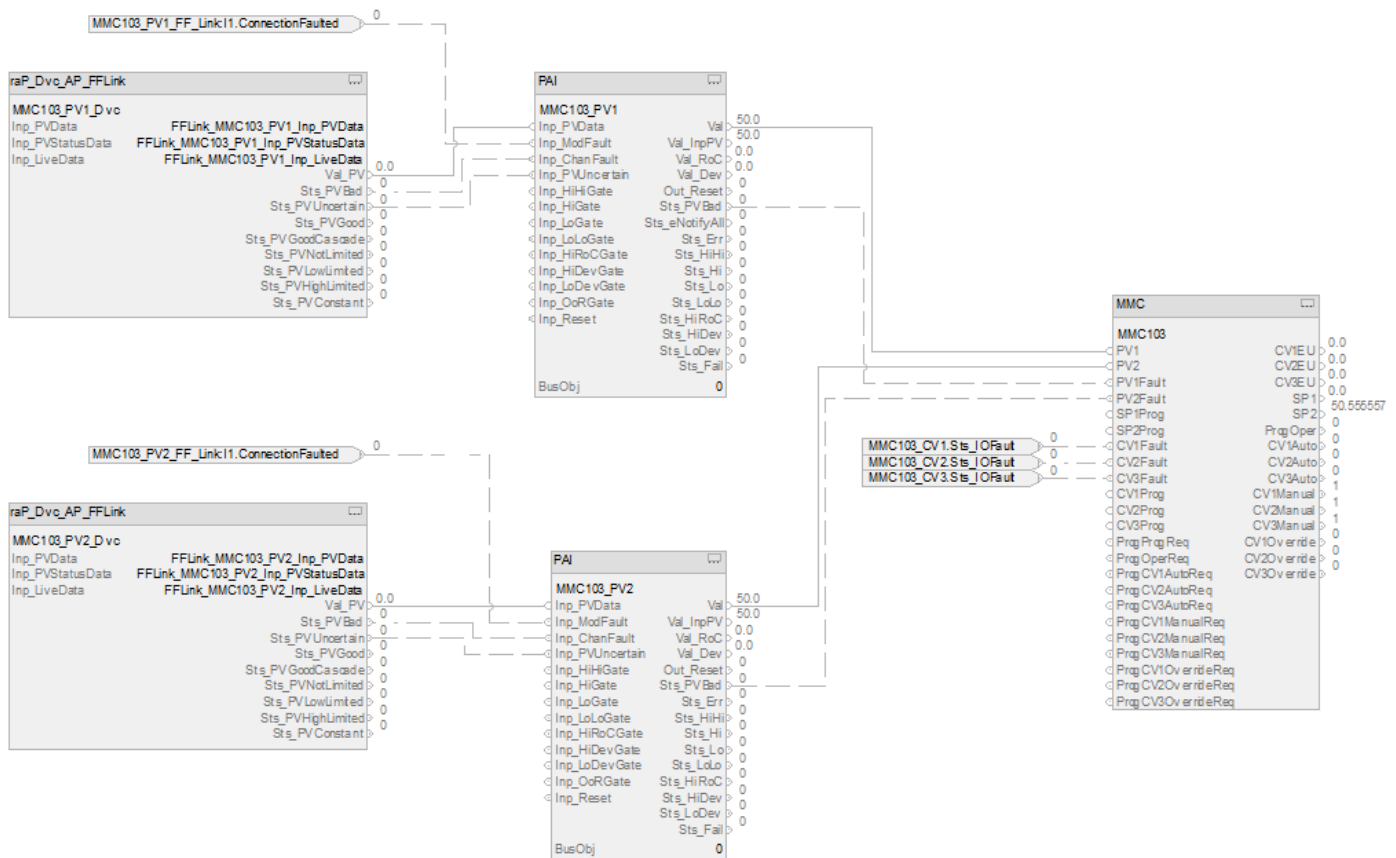


The CS\_MMC\_EtherNetIP control strategy operates the same as the CS\_MMC control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, [Sheet on page 151](#).
  - Substitute MMC102\_PV1 for the first instance of XT100
  - Substitute MMC102\_PV2 for the second instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_MMC\_FF Sheet

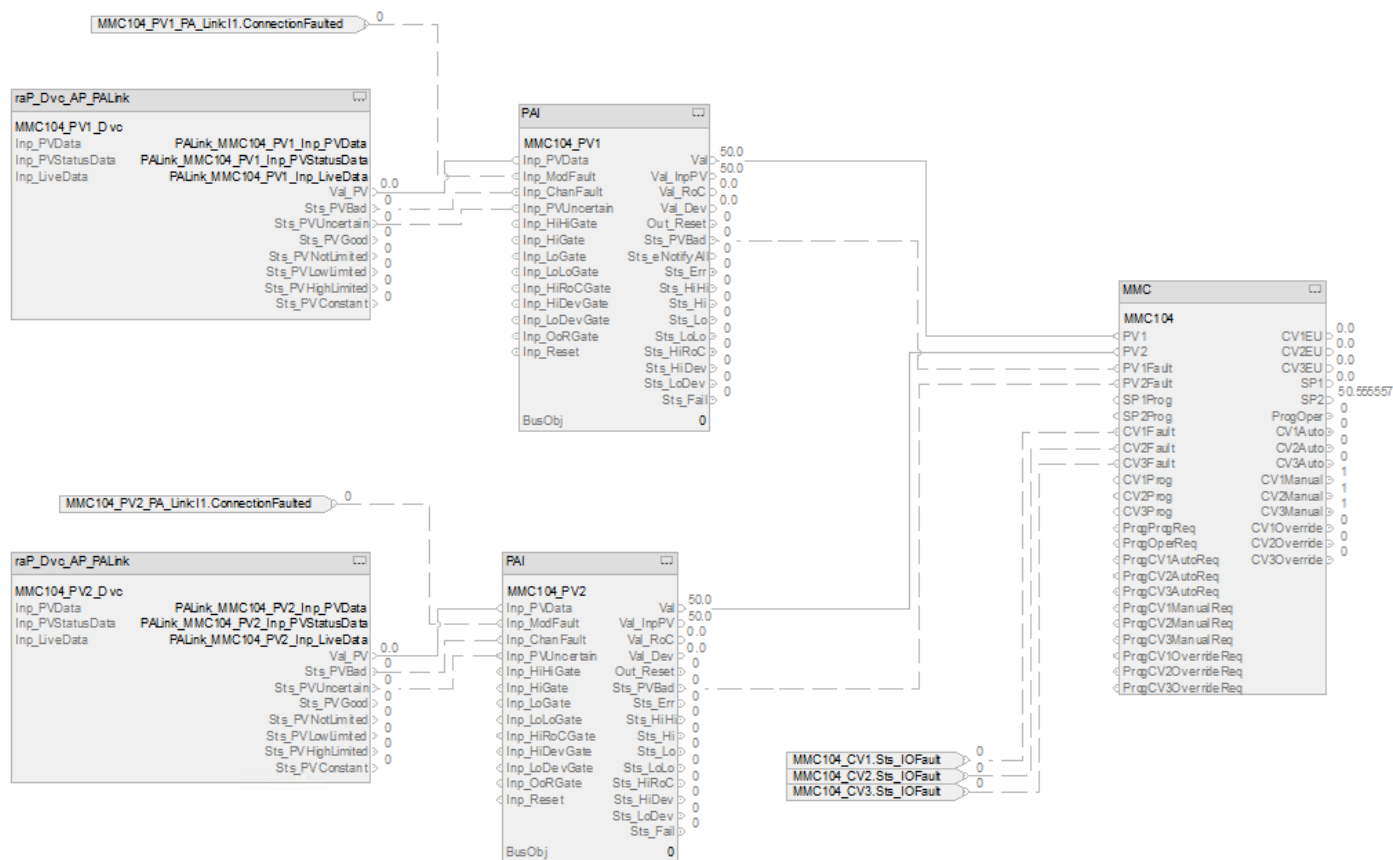


The CS\_MMC\_FF control strategy operates the same as the CS\_MMC control strategy but relies on FOUNDATION Fieldbus input data.

- For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
  - Substitute MMC103\_PV1 for the first instance of XT100
  - Substitute MMC103\_PV2 for the second instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_MMC\_PA Sheet



The CS\_MMC\_PA control strategy operates the same as the CS\_MMC control strategy but relies on PA data.

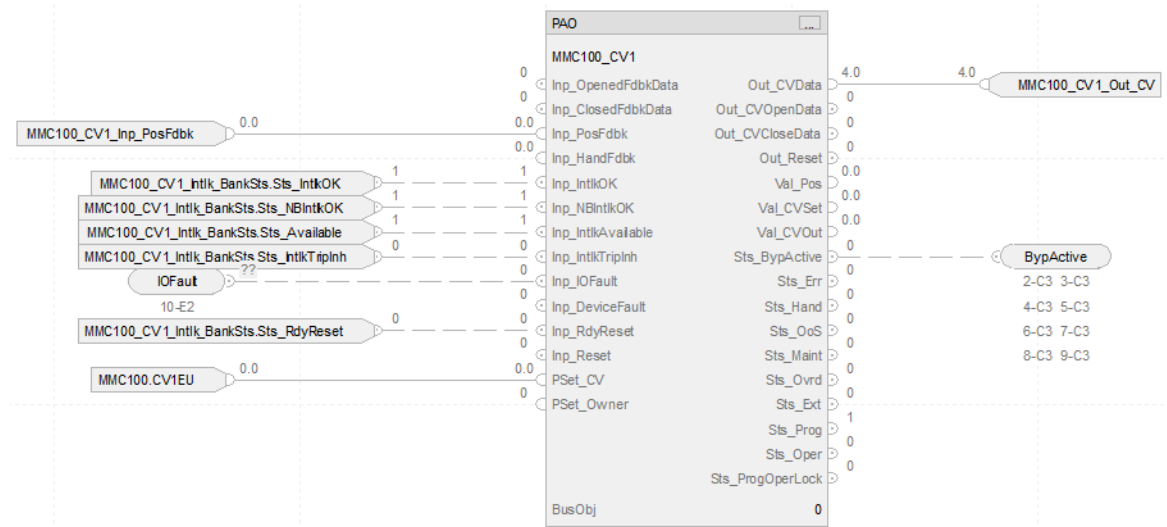
- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
  - Substitute MMC104\_PV1 for the first instance of XT100
  - Substitute MMC104\_PV2 for the second instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## Control Variable Routines

This control variable routines include the PAO control strategy, with an additional input reference.

- For PAO configuration considerations, and input and output references, see [CS\\_PA0 Sheet on page 180](#).
- The routines also include these PAO input references:



Parameter	Description
MMC100.CV1EU	<p>Scaled control variable output for CV1. Scaled by using CV1EUMax and CV1EUMin, where CV1EUMax corresponds to 100% and CV1EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV1EU = (CV1 * CV1EUSpan / 100) + CV1EUMin$ <p>CV1EU span calculation: <math>CV1EUSpan = (CV1EUMax - CV1EUMin)</math></p>
MMC100.CV2EU	<p>Scaled control variable output for CV2. Scaled by using CV2EUMax and CV2EUMin, where CV2EUMax corresponds to 100% and CV2EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV2EU = (CV2 * CV2EUSpan / 100) + CV2EUMin$ <p>CV2EU span calculation: <math>CV2EUSpan = (CV2EUMax - CV2EUMin)</math></p>
MMC100.CV3EU	<p>Scaled control variable output for CV3. Scaled by using CV3EUMax and CV3EUMin, where CV3EUMax corresponds to 100% and CV3EUMin corresponds to 0%. This output is typically used to control an analog output module or a secondary loop.</p> $CV3EU = (CV3 * CV3EUSpan / 100) + CV3EUMin$ <p>CV3EU span calculation: <math>CV3EUSpan = (CV3EUMax - CV3EUMin)</math></p>

## ACM Considerations for MMC

Configure these parameters first because they affect the visibility of the remaining parameters in the MMC object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for an MMC Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	Important: Select this parameter first as the option affects the remaining parameters.  Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
Localize_PA0	always	Set to use a local routine in the program
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_HasHARTPV1	IO_Signal_Type=HART	Set if there is a HART device connected to PV1
Cfg_HasHARTPV2	IO_Signal_Type=HART	Set if there is a HART device connected to PV2
Cfg_HasEHPV1	IO_Signal_Type=EH_EtherNetIP	Set if there is an E+H EtherNet/IP device connected to PV1
Cfg_HasEHPV2	IO_Signal_Type=EH_EtherNetIP	Set if there is an E+H EtherNet/IP device connected to PV2
Cfg_HasFFPV1	IO_Signal_Type=FF	Set if there is a FOUNDATION Fieldbus device connected to PV1
Cfg_HasFFPV2	IO_Signal_Type=FF	Set if there is a FOUNDATION Fieldbus device connected to PV2
Cfg_HasPAPV1	IO_Signal_Type=PA	Set if there is a Profibus PA device connected to PV1
Cfg_HasPAPV2	IO_Signal_Type=PA	Set if there is a Profibus PA device connected to PV2
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	Cfg_HasHARTPV1=True IO_Signal_Type=HART Cfg_HasHARTPV2=True IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	Cfg_HasHARTPV1=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False Cfg_HasHARTPV2=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	Cfg_HasHARTPV1=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False Cfg_HasHARTPV2=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_PV1_HartDevice	Cfg_HasHARTPV1=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object for PV1. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_PV2_HartDevice	Cfg_HasHARTPV2=True IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object for PV2. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModulePV1	Cfg_HasEHPV1=True IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object for PV1. See (RA-LIB) Process > Module > Endress+Hauser for available objects

Parameter	Visible When	Details
Ref_EtherNetIPModulePV2	Cfg_HasEHPV2=False IO_SignalType=EH_EthernetIP	Link to the E+H EtherNet/IP device object for PV2. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_PV1_Module	Cfg_HasFFPV1=True IO_SignalType=FF	Link to the FOUNDATION Fieldbus device object for PV1. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_FF_PV2_Module	Cfg_HasFFPV2=True IO_SignalType=FF	Link to the FOUNDATION Fieldbus device object for PV2. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_PV1_Module	Cfg_HasPAPV1=True IO_SignalType=PA	Link to the Profibus PA device object for PV1. See (RA-LIB) Process > Module > Profibus PA for available objects
Ref_PA_PV2_Module	Cfg_HasPAPV2=True IO_SignalType=PA	Link to the Profibus PA device object for PV2. See (RA-LIB) Process > Module > Profibus PA for available objects

## 02 - Device Configuration

UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
------------------------	----------------------	---

### 03.00 - IO Configuration

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

PAI_RefTag1	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_RefTag1	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag1	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV2 input reference
	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=HART	

### 03.00.10 - Ref PAI Alarm Configuration

Ref_HiHiGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGatePV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel).	Link to the gate reference
Ref_LoGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_OoRGatePV1	PAI_RefTag1 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference

**03.02 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Inp_PV2	PAI_RefTag2 is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV2 input reference
	PAI_RefTag2 is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=HART	
PAI_RefTag2	PAI_Type=PAI(Single_channel) PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID_RefTag2	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag2	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference

**03.02.10 - Ref PAI Alarm Configuration**

Ref_HiHiGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGatePV2	PAI_RefTag2 is linked to an analog input reference.	Link to the gate reference
Ref_LoGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGatePV2	PAI_RefTag2 is linked to an analog input reference. PAI_Type=PAI(Single_channel)	Link to the gate reference

**03.11 - IO Configuration**

PA01_RefTag	always	Link to first analog output reference
Cfg_HasCV1IntlkObj	Localize_PA0=True	Set if the first analog output reference has an interlock CV
Bus_Instance_CV1	PA01_RefTag is linked to an analog out reference. Has_O0AP=True (controller parameter) Use_O0AP=True Localize_PA0=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk1	Localize_PA0=True	Link to the first input position feedback reference
Out_CV1	Localize_PA0=True	Link to the first output CV reference

**03.12 - IO Configuration**

PA02_RefTag	always	Link to second analog output reference
Cfg_HasCV2IntlkObj	Localize_PA0=True	Set if the second analog output reference has an interlock CV



Parameter	Visible When	Details
Bus_Instance_CV2	PA02_RefTag is linked to an analog out reference. Has_OOAP=True (controller parameter) Use_OOAP=True Localize_PA0=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk2	Localize_PA0=True	Link to the second input position feedback reference
Out_CV2	Localize_PA0=True	Link to the second output CV reference
<b>03.13 - IO Configuration</b>		
PA03_RefTag	always	Link to third analog output reference
Cfg_HasCV3IntlkObj	Localize_PA0=True	Set if the third analog output reference has an interlock CV
Bus_Instance_CV3	PA03_RefTag is linked to an analog out reference. Has_OOAP=True (controller parameter) Use_OOAP=True Localize_PA0=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk3	Localize_PA0=True	Link to the third input position feedback reference
Out_CV3	Localize_PA0=True	Link to the third output CV reference

## Additional Sub-Objects for a MMC Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CV1Intlk	Configure an interlock for the CV instance. See <a href="#">Interlocks on page 49</a>
CV2Intlk	
CV3Intlk	
Events	Configure an event to monitor for the control strategy. See <a href="#">Event Logging on page 49</a>
CV1Events	
CV2Events	
CV3Events	Configure an event to monitor for the CV instance. See <a href="#">Event Logging on page 49</a>

**Notes:**

## Process Analog Input (PAI) Control Strategies

Use a PAI control strategy to monitor an analog input and check for alarm conditions. The PAI control strategy that is included with the library download processes a signal from a channel of an analog input module, but it can be used to process any analog (REAL) signal.

The following PAI control strategies are available as routines in the process library:

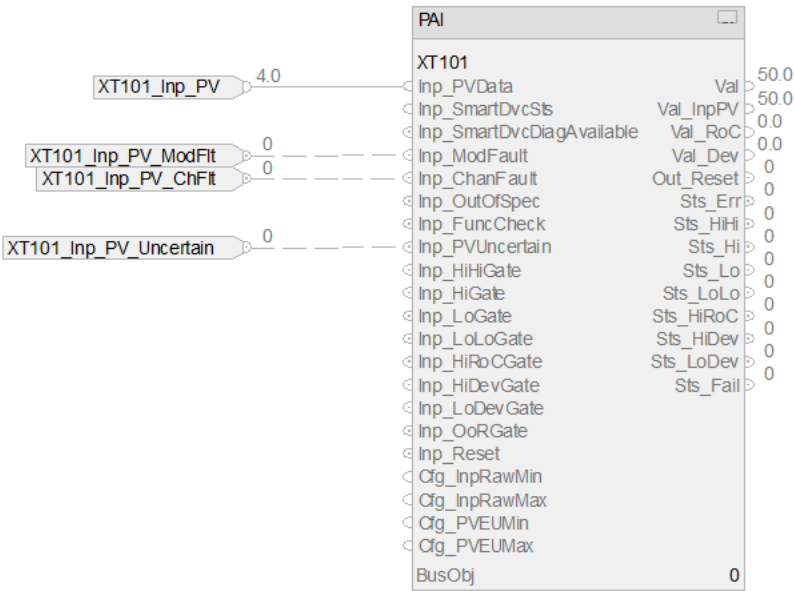
- CS\_PAI
- CS\_PAI\_HART
- CS\_PAI\_EtherNetIP
- CS\_PAI\_EtherNetIP\_NoHB
- CS\_PAI\_EtherNetIP\_Sensor
- CS\_PAI\_FF
- CS\_PAI\_PA

Import the appropriate control strategy as a **routine** in your controller project.

Each PAI control strategy contains one Function Block sheet:

Sheet	Description
CS_PAI	Process Analog Input instruction
CS_PAI_HART	Process Analog Input instruction with HART input
CS_PAI_EtherNetIP	Process Analog Input instruction with EtherNetIP input
CS_PAI_EtherNetIP_NoHB	Process Analog Input instruction with No HB EtherNetIP input
CS_PAI_EtherNetIP_Sensor	Process Analog Input instruction with EtherNetIP Sensor input
CS_PAI_FF	Process Analog Input instruction with FOUNDATION Fieldbus input
CS_PAI_PA	Process Analog Input instruction with PA input

CS\_PAI Sheet



PAI Input References

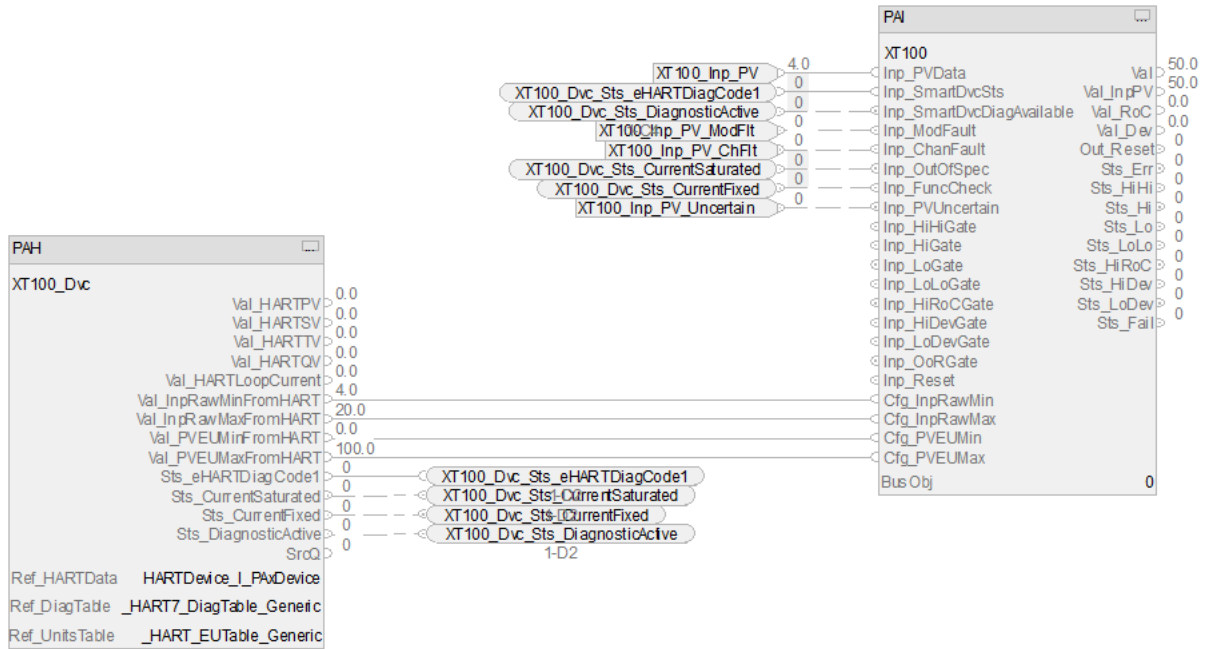
Input	Description
XT101_Inp_PVData	Process variable input Source: sensor or input
XT101_Inp_ModFault	Process variable input module fault 1 = I/O module failure or module communication status bad 0 = OK
XT101_Inp_ChanFault	Process variable input channel fault 1 = I/O channel fault or failure 0 = OK
XT101_Inp_PVUncertain	Process variable input uncertain Indicates the channel data accuracy is undetermined 1 = The channel data is uncertain This input sets Sts.PVUncertain if not in Virtual

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

PAI Configuration Considerations

Operand	Type	Description
PlantPax® control	P_ANALOG_INPUT	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"><li>0 if not using organization</li><li>Bus[x].Obj when using organization</li></ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PAH\_HART Sheet



The CS\_PAH\_HART control strategy operates the same as the CS\_PAH control strategy but relies on HART input data. Substitute XT100 for XT101.

### PAH Outputs to PAI Inputs

Output	Description
Val_InpRawMinFromHART	Analog input unscaled signal minimum from HART module (in module units).
Val_InpRawMaxFromHART	Analog input unscaled signal maximum from HART module (in module units).
Val_PVEUMinFromHART	Analog input scaled range minimum from HART device (in engineering units).
Val_PVEUMaxFromHART	Analog input scaled range maximum from HART device (in engineering units).

### PAH Status Outputs

Output	Description
Sts_eHARTDiagCode1	HART Diagnostic Code #1 (bit number in command 48, 255 = none)
Sts_CurrentSaturated	HART reports analog current is limited
Sts_CurrentFixed	Loop Current set to fixed value via HART command
Sts_DiagnosticsActive	HART data input diagnostic active

### PAH Configuration Considerations

Operand	Type	Description
PlantPAX control	P_ANALOG_HART	Instance of data structure (backing tag) required for proper operation of instruction.
Ref_HARTData	PAX_HART_DEVICE:I:0	Required data type HART data from I/O module assembly Select the HART device in your Controller Organizer; the device must support the PAXDevice data type: IOTreeObject:I.PAXDevice
Ref_DiagTable	P_HART_CODE_D ESC_STATUS[2]	Lookup table for diagnostic bit number (to message and status) Select the correct table for your HART device; see table below
Ref_UnitsTable	RAC_CODE_DESC RIPTION[2]	Lookup table for units of measure code (to units text) Select _HART_EUTable_Generic

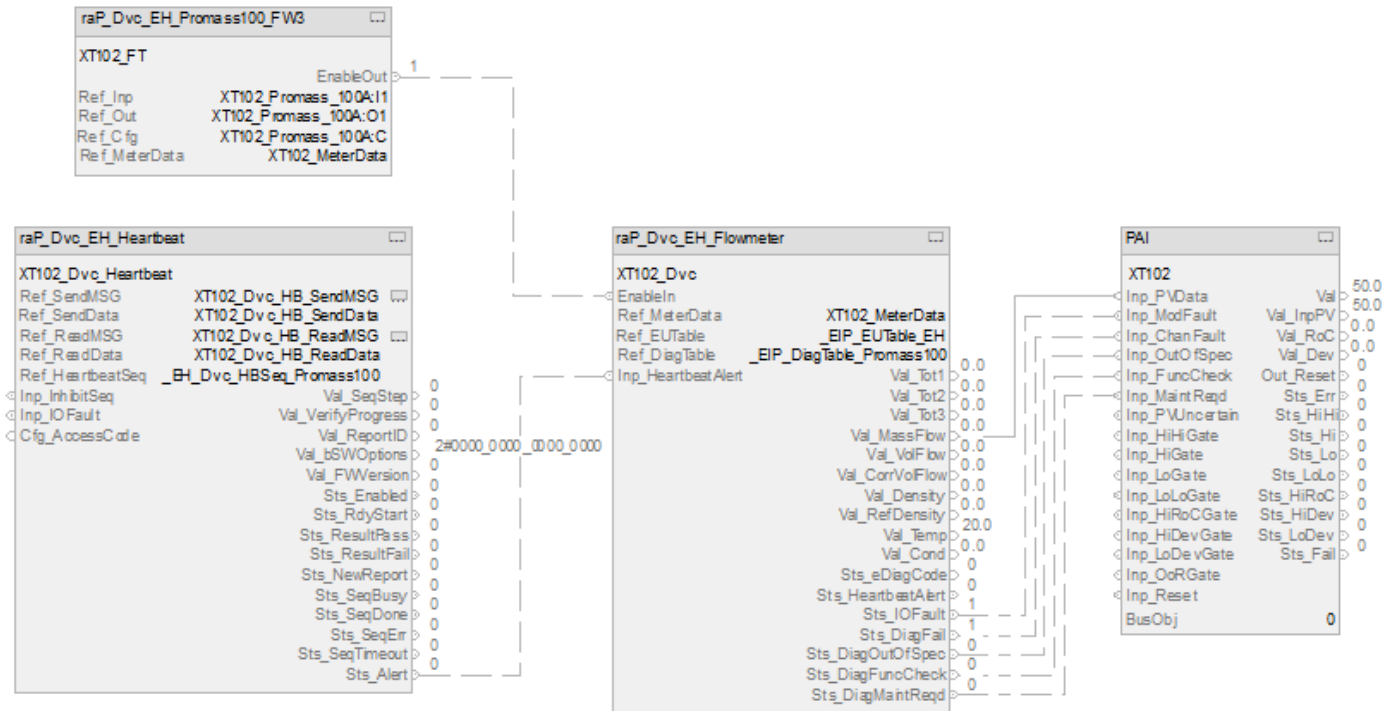
### Available Diagnostic Tables

Diagnostic tables are available for these HART devices. The HART number indicates the version of the table.

Option	Description
_HART5_DiagTable_FMG_60	HART Cmd48 Diagnostic Lookup Table: E+H Gammapilot FMG 60
_HART5_DiagTable_Generic	HART Cmd48 Diagnostic Lookup Table: Generic HART5 device
_HART5_DiagTable_LevelflexM	HART Cmd48 Diagnostic Lookup Table: E+H Levelflex M
_HART5_DiagTable_MicropilotM	HART Cmd48 Diagnostic Lookup Table: E+H Micropilot M
_HART5_DiagTable_ProsonicM	HART Cmd48 Diagnostic Lookup Table: E+H Prosonic M
_HART5_DiagTable_ProsonicS	HART Cmd48 Diagnostic Lookup Table: E+H Prosonic S
_HART5_DiagTable_Prowirl73	HART Cmd48 Diagnostic Lookup Table: E+H Prowirl 73
_HART5_DiagTable_TMass65l	HART Cmd48 Diagnostic Lookup Table: E+H TMass 65l
_HART5_DiagTable_TMT162	HART Cmd48 Diagnostic Lookup Table: E+H TMT 162
_HART5_DiagTable_TMT182	HART Cmd48 Diagnostic Lookup Table: E+H TMT182 Temperature
_HART6_DiagTable_Pressure_M	HART Cmd48 Diagnostic Lookup Table: E+H Cerabar, Deltabar, Deltapilot M
_HART7_DiagTable_GammapilotFMG5x_rev1	HART Cmd48 Diagnostic Lookup Table: E+H Gammapilot FMG5x rev 1.x
_HART7_DiagTable_Generic	HART Cmd48 Diagnostic Lookup Table: Generic HART7 device
_HART7_DiagTable_LevelflexFMP5x	HART Cmd48 Diagnostic Lookup Table: E+H Levelflex FMP5x
_HART7_DiagTable_LiquilineCM44x	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline CM442 / 444 / 448
_HART7_DiagTable_LiquilineCM82_rev1	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline CM82 FW Rev. 1
_HART7_DiagTable_LiquilineM_Cond	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M Conductivity
_HART7_DiagTable_LiquilineM_Cond_rev4	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M Conductivity rev 4.x
_HART7_DiagTable_LiquilineM_Oxy_rev4	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M Oxygen rev 4.x
_HART7_DiagTable_LiquilineM_Oxygen	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M Oxygen
_HART7_DiagTable_LiquilineM_pH_rev4	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M pH/ORP rev 4.x
_HART7_DiagTable_LiquilineM_pH_ORP	HART Cmd48 Diagnostic Lookup Table: E+H Liquiline M pH / ORP
_HART7_DiagTable_LiquistationCSFxx	HART Cmd48 Diagnostic Lookup Table: E+H Liquistation CSFxx
_HART7_DiagTable_Metso_ND7x_ND9x	HART Cmd48 Diagnostic Lookup Table: Metso ND7xxx and ND9xxx Positioners
_HART7_DiagTable_MicropilotFMR5x	HART Cmd48 Diagnostic Lookup Table: E+H Micropilot FMR5x
_HART7_DiagTable_MicropilotFMR6x	HART Cmd48 Diagnostic Lookup Table: E+H Micropilot FMR6x
_HART7_DiagTable_MicropilotFMR20	HART Cmd48 Diagnostic Lookup Table: E+H Micropilot FMR 20
_HART7_DiagTable_Pressure_S	HART Cmd48 Diagnostic Lookup Table: E+H Cerabar, Deltabar, Deltapilot S
_HART7_DiagTable_Promag53	HART Cmd48 Diagnostic Lookup Table: E+H Promag 53
_HART7_DiagTable_Promag100	HART Cmd48 Diagnostic Lookup Table: E+H Promag 100
_HART7_DiagTable_Promag200	HART Cmd48 Diagnostic Lookup Table: E+H Promag 200
_HART7_DiagTable_Promag300_500	HART Cmd48 Diagnostic Lookup Table: E+H Promag 300 and Promg 500
_HART7_DiagTable_Promag400	HART Cmd48 Diagnostic Lookup Table: E+H Promag 400
_HART7_DiagTable_Promag400_rev6	HART Cmd48 Diagnostic Lookup Table: E+H Promag 400 rev 6
_HART7_DiagTable_Promag400_rev9	HART Cmd48 Diagnostic Lookup Table: E+H Promag 400 rev 9
_HART7_DiagTable_Promass83	HART Cmd48 Diagnostic Lookup Table: E+H Promass 83
_HART7_DiagTable_Promass100	HART Cmd48 Diagnostic Lookup Table: E+H Promass 100
_HART7_DiagTable_Promass200	HART Cmd48 Diagnostic Lookup Table: E+H Promass 200
_HART7_DiagTable_Promass300_500	HART Cmd48 Diagnostic Lookup Table: E+H Promass 300 and Promass 500
_HART7_DiagTable_ProsonicFlow_100_rev1	HART Cmd48 Diagnostic Lookup Table: E+H Prosonic Flow 100 rev 1.x
_HART7_DiagTable_ProsonicFlow300_500rev1	HART Cmd48 Diagnostic Lookup Table: E+H Prosonic Flow 300 or 500 rev 1.x
_HART7_DiagTable_ProsonicFlowB200	HART Cmd48 Diagnostic Lookup Table: E+H Prosonic Flow B200
_HART7_DiagTable_Prowirl200	HART Cmd48 Diagnostic Lookup Table: E+H Prowirl
_HART7_DiagTable_TMT72_rev1	HART Cmd48 Diagnostic Lookup Table: E+H TMT72 rev 1
_HART7_DiagTable_TMT82	HART Cmd48 Diagnostic Lookup Table: E+H TMT82
_HART7_DiagTable_TMT162_rev4	HART Cmd48 Diagnostic Lookup Table: E+H TMT162 rev 4.x
_HART7_DiagTable_TrustSensTM37x_rev1	HART Cmd48 Diagnostic Lookup Table: E+H TrustSens TM37x rev 1.x

For more information, see [HART Integration on page 61](#).

## CS\_PAIEtherNetIP Sheet



The CS\_PAIEtherNetIP control strategy operates the same as the CS\_PAI control strategy but relies on EtherNet/IP™ data. Substitute XT102 for XT101.

This control strategy uses the raP\_Dvc\_EH\_Flowmeter instruction to integrate an Endress+Hauser flowmeter. The raP\_Dvc\_EH\_Heartbeat heartbeat instruction provides the ability to initiate Heartbeat Verification from the operator faceplate. For more information, see EtherNet/IP Integration.

For more information on integrating E+H devices, see Endress+Hauser EtherNet/IP Instrumentation for PlantPAx DCS Reference Manual, [PROCES-RM212A](#).

### raP\_Dvc\_EH\_Flowmeter Outputs to PAI Inputs

Output	Description
EnableOut	Enable Output - System Defined Parameter
Val_MassFlow	Mass flow rate value (Mass flow EU)

### raP\_Dvc\_EH\_Flowmeter Status Outputs

Output	Description
Sts_IOFault	I/O communication with the flowmeter is faulted
Sts_DiagFail	Diagnostic indicates device failure
Sts_DiagOutOfSpec	Diagnostic indicates device operating outside of specified accuracy
Sts_DiagFuncCheck	Diagnostic indicates device operating with substitute value
Sts_DiagMaintReqd	Diagnostic indicates device requires maintenance

## raP\_Dvc\_EH\_Flowmeter Configuration Considerations

Operand	Type	Description
PlantPAx control	raP_Dvc_EH_Flowmeter	Instance of data structure (backing tag) required for proper operation of instruction
Ref_MeterData	raP_UDT_Dvc_EH_Flowmeter_Data	Flowmeter data from E+H meter
Ref_EUTable	RAC_CODE_DESCRIPTION[1]	Lookup table for engineering unit code to text
Ref_DiagTable	RAC_CODE_DESCRIPTION[1]	Lookup table for diagnostic code to text

### Available Diagnostic Tables

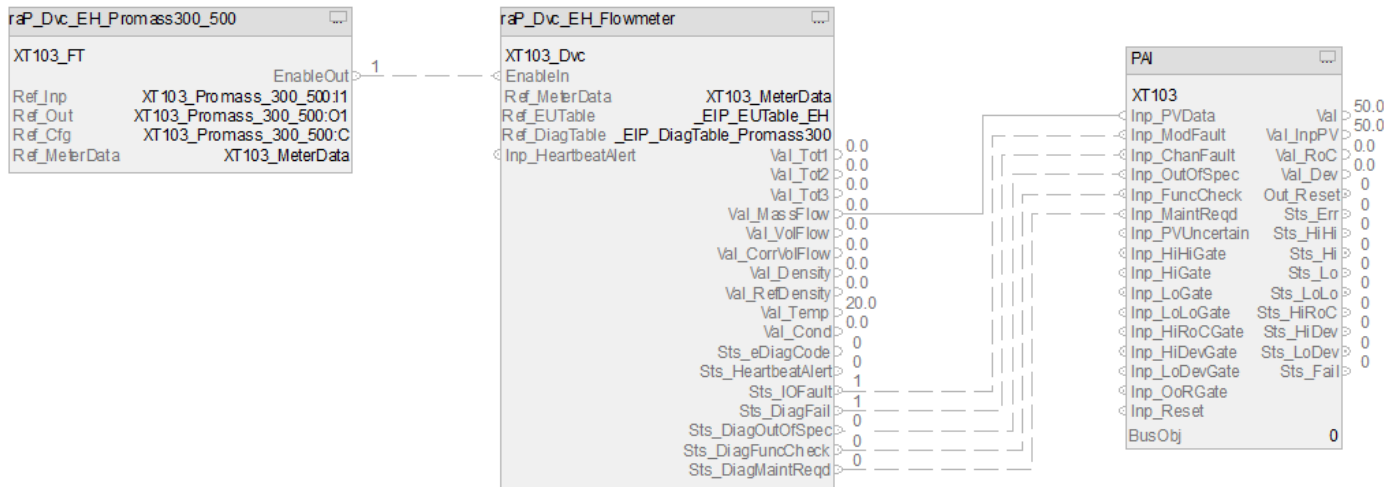
*Diagnostic tables are available for these EtherNet/IP devices. The EtherNet/IP number indicates the version of the table.*

Option	Description
_EIP_EUTable_EH	TagDescript - EH EtherNet/IP devices engineering units lookup table.
_EIP_DiagTable_Promass300	TagDescript - PV - EH EtherNet/IP Promass300_500 diagnostics lookup table
_EIP_DiagTable_Promass100	TagDescript - EH EtherNet/IP Promag 100/Promag 400/Promass 100 diagnostics lookup table
_EIP_DiagTable_LiquilineCM44x	TagDescript - PV - EH EtherNet/IP Liquiline CM44x diagnostics lookup table

For more information, see [EtherNet/IP Integration on page 85](#).



## CS\_PAIEtherNetIP\_NoHB Sheet



The CS\_PAIEtherNetIP\_NoHB control strategy operates the same as the CS\_PAIEtherNet control strategy but relies on EtherNet/IP NoHB data. Substitute XT103 for XT101.

This control strategy uses the raP\_Dvc\_EH\_Flowmeter instruction to integrate an Endress+Hauser flowmeter, but without heartbeat input. For more information, see [EtherNet/IP Integration on page 85](#).

For more information on integrating E+H devices, see Endress+Hauser EtherNet/IP Instrumentation for PlantPAx DCS Reference Manual, [PROCES-RM212A](#).

### raP\_Dvc\_EH\_Heartbeat Outputs to raP\_Dvc\_EH\_Flowmeter Inputs

Output	Description
Sts.Alert	Notify the raP_Dvc_EH_Flowmeter block that this block requires operator attention.

### raP\_Dvc\_EH\_Heartbeat Configuration Considerations

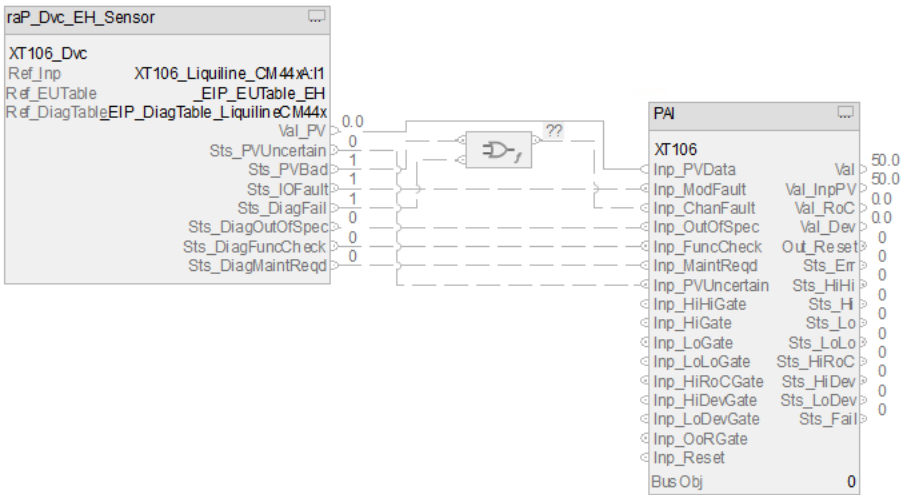
Operand	Type	Description
PlantPAx control	raP_Dvc_EH_Heartbeat	Instance of data structure (backing tag) required for proper operation of instruction
Ref_MeterData	raP_UDT_Dvc_EH_Flowmeter_Data	Flowmeter data from E+H meter on EtherNet/IP
Ref_EUTable	RAC.CODE_DESCRIPTION[1]	Lookup table for engineering unit code to text
Ref_DiagTable	RAC.CODE_DESCRIPTION[1]	Lookup table for diagnostic code to text

#### Available Diagnostic Tables

Diagnostic tables are available for these EtherNet/IP devices. The EtherNet/IP number indicates the version of the table.

Option	Description
_EIP_EUTable_EH	TagDescription - EH EtherNet/IP devices engineering units lookup table.
_EIP_DiagTable_Promass300	TagDescription - PV - EH EtherNet/IP Promass300.500 diagnostics lookup table
_EIP_DiagTable_Promag100	TagDescription - EH EtherNet/IP Promag 100 / Promag 400 / Promag 100 diagnostics lookup table
_EIP_DiagTable_LiquilineCM44x	TagDescription - PV - EH EtherNet/IP Liquiline CM44x diagnostics lookup table

# CS\_PAIEtherNetIP\_Sensor Sheet



The CS\_PAIEtherNetIP\_Sensor control strategy operates the same as the CS\_PAI control strategy but relies on EtherNet/IP\_Sensor data. Substitute XT106 for XT101.

This control strategy uses the raP\_Dvc\_EH\_Sensor instruction to integrate an Endress+Hauser analyzer device. For more information, see [EtherNet/IP Integration on page 85](#).

For more information on integrating E+H devices, see Endress+Hauser EtherNet/IP Instrumentation for PlantPAx DCS Reference Manual, [PROCES-RM212A](#).

## raP\_Dvc\_EH\_Sensor Outputs to PAI Input

Output	Description
Val_PV	Sensor primary variable (PV) value (PVEU)
Sts_PVUncertain	Device reports PV value is uncertain (quality)
Sts_IOFault	I/O communication with the flowmeter is faulted
Sts_DiagFail	Diagnostic indicates device failure
Sts_DiagOutOfSpec	Diagnostic indicates device operating outside of specified accuracy
Sts_DiagMainReqd	Diagnostic indicates device requires maintenance

## raP\_Dvc\_EH\_Sensor Configuration Considerations

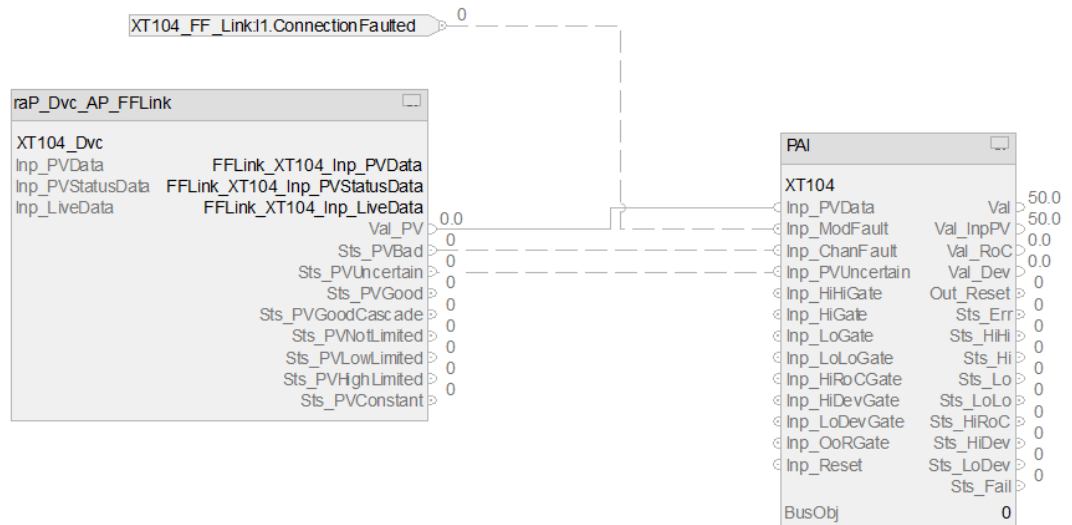
Operand	Type	Description
PlantPAx control	raP_Dvc_EH_Sensor	Instance of data structure (backing tag) required for proper operation of instruction
Ref_Inp	EH:CM44:I1:0	Input assembly: data from E+H Liquiline CM44x on EtherNet/IP
Ref_EUTable	RAC.CODE.DESCRPTION[2]	Lookup table for engineering units code to text
Ref_DiagTable	RAC.CODE.DESCRPTION[2]	Lookup table for diagnostic code to text

### Available Diagnostic Tables

Diagnostic tables are available for these EtherNet/IP devices.

Option	Description
_EIP_EUTable_EH	TagDescript - EH EtherNet/IP devices engineering units lookup table.
_EIP_DiagTable_Promass300	TagDescript - PV - EH EtherNet/IP Promass300..500 diagnostics lookup table
_EIP_DiagTable_Promass100	TagDescript - EH EtherNet/IP Promag 100 / Promag 400 / Promass 100 diagnostics lookup table
_EIP_DiagTable_LiquilineCM44x	TagDescript - PV - EH EtherNet/IP Liquiline CM44x diagnostics lookup table

## CS\_PAIF\_F Sheet



The CS\_PAIF\_F control strategy operates the same as the CS\_PAIF control strategy but relies on FOUNDATION Fieldbus data. Substitute XT104 for XT101.

This control strategy uses the raP\_Dvc\_AP\_FFLink instruction to integrate a FOUNDATION Fieldbus device. For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

For more information on integrating FOUNDATION Fieldbus devices, see Profibus PA and FOUNDATION Fieldbus Linking Devices in a PlantPax DCS Reference Manual, publication [PROCES-RM213](#).

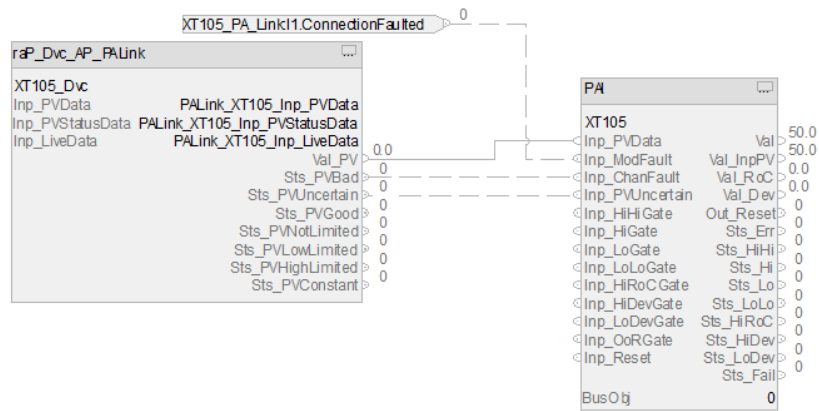
### raP\_Dvc\_AP\_FFLink Outputs to PAI Inputs

Output	Description
Val_PV	Process variable (PV engineering units). Map this output to Inp_PVData on PAI
Sts_PVBad	PV Status (quality) is bad. See detail bits for additional information. Map this output to Inp_ChanFault on PAI
Sts_PVUncertain	PV Status (quality) is uncertain. See detail bits for additional information. Map this output to Inp_PVUncertain on PAI

### raP\_Dvc\_AP\_FFLink Configuration Considerations

Operand	Type	Description
PlantPax control	raP_Dvc_AP_FFLink	Instance of data structure (backing tag) required for proper operation of instruction
Inp_PVData	REAL	Process variable data
Inp_PVStatusData	SINT	Process variable status data (byte)
Inp_LiveData	BOOL	Device is communicating and updating live PV and status

# CS\_PA\_PA Sheet



The CS\_PA\_PA control strategy operates the same as the CS\_PA control strategy but relies on Profibus PA data. Substitute XT105 for XT101.

This control strategy uses the raP\_Dvc\_AP\_PALink instruction to integrate a Profibus PA device. For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

For more information on integrating Profibus PA devices, see Profibus PA and FOUNDATION Fieldbus Linking Devices in a PlantPax DCS Reference Manual, publication [PROCES-RM213](#).

## raP\_Dvc\_AP\_PALink Outputs to PAI Inputs

Output	Description
Val_PV	Process variable value
Sts_PVBad	PV Status (quality) is bad. See detail bits for additional information. Map this output to Inp_ChanFault on PAI
Sts_PVUncertain	PV Status (quality) is uncertain. See detail bits for additional information. Map this output to Inp_PVUncertain on PAI

## raP\_Dvc\_AP\_PALink Configuration Considerations

Operand	Type	Description
PlantPax control	raP_Dvc_AP_PALink	Instance of data structure (backing tag) required for proper operation of instruction
Inp_PVData	REAL	Process variable data
Inp_PVStatusData	SINT	Process variable status data (byte)
Inp_LiveData	BOOL	Device is communicating and updating live PV and status

## ACM Considerations for PAI

- Configure these parameters first because they affect the visibility of the remaining parameters in the PAI object.
- If you use a specific I/O signal type, select the type for the IO.Signal\_Type parameter and select None for the ACM\_UsedIn parameter
  - If the PAI is used in another control strategy, select None for the IO.Signal\_Type parameter and select the type of control strategy for the ACM\_UsedIn parameter

## ACM-Based Parameters for a PAI Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
ACM_UsedIn	always	PAI can be standalone (None) or used as part of another PAID, PAIM, PPID, or PDBC control strategy
IO_Signal_Type	ACM_UsedIn=None	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter) ACM_UsedIn=None	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>02 - Device Configuration</b>		
Cfg_HasRoC	always	Set to monitor the PV rate of change
Cfg_HasDev	always	Set to monitor the PV deviation
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Inp_PV	ACM_UsedIn=None IO_SignalType=None IO_SignalType=HART	Link to the PV input reference
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	ACM_UsedIn=None IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table.
Ref_HartDevice	ACM_UsedIn=None IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	ACM_UsedIn=None IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	ACM_UsedIn=None IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	ACM_UsedIn=None IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>04 - Alarm Configuration</b>		
Cfg_HasFailAlm	always	If Cfg_HasFailAlm =True, ACM displays section 4.01 - Input Fail Alarm with additional parameters
Cfg_HasHiHiAlm	always	If Cfg_HasHiHiAlm=True, ACM displays section 4.02 - Hi Hi Alarm with additional parameters
Cfg_HasHiAlm	always	If Cfg_HasHiAlm=True, ACM displays section 4.03 - Hi Hi Alarm with additional parameters
Cfg_HasLoAlm	always	If Cfg_HasLoAlm=True, ACM displays section 4.04 - Lo Alarm with additional parameters
Cfg_HasLoLoAlm	always	If Cfg_HasLoLoAlm=True, ACM displays section 4.05 - Lo Alarm with additional parameters
Cfg_HasHiRoCAlm	Cfg_HasRoC=True	If Cfg_HasHiRoCAlm=True, ACM displays section 4.06 - Hi RoC Alarm with additional parameters
Cfg_HasHiDevAlm	Cfg_HasDev=True	If Cfg_HasHiDevAlm=True, ACM displays section 4.08 - Hi Dev Alarm with additional parameters

Parameter	Visible When	Details
Cfg_HasLoDevAlm	Cfg_HasDev=True	If Cfg_HasLoDevAlm =True, ACM displays section 4.09 - Lo Dev Alarm with additional parameters
<b>04.01 - Input Fail Alarm</b>		
Ref_OoRGate	Cfg_HasFailAlm =True ACM_UsedIn=None	Link to the gate reference
<b>04.02 - Hi Hi Alarm</b>		
Ref_HiHiGate	Cfg_HasHiHiAlm =True ACM_UsedIn=None	Link to the gate reference
<b>04.03 - Hi Alarm</b>		
Ref_HiGate	Cfg_HasHiAlm =True ACM_UsedIn=None	Link to the gate reference
<b>04.04 - Lo Alarm</b>		
Ref_LoGate	Cfg_HasLoAlm =True ACM_UsedIn=None	Link to the gate reference
<b>04.05 - Lo Lo Alarm</b>		
Ref_LoLoGate	Cfg_HasLoLoAlm =True ACM_UsedIn=None	Link to the gate reference
<b>04.06 - Hi RoC Alarm</b>		
Ref_HiRoCGate	Cfg_HasHiRoCAlm=True ACM_UsedIn=None	Link to the gate reference
<b>04.08 - Hi Dev Alarm</b>		
Ref_HiDevGate	Cfg_HasHiDevAlm=True ACM_UsedIn=None	Link to the gate reference
<b>04.09 - Lo Dev Alarm</b>		
Ref_LoDevGate	Cfg_HasLoDevAlm=True ACM_UsedIn=None	Link to the gate reference

## Additional Sub-Object for a PAI Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy. See <a href="#">Event Logging on page 49</a>

## Process Analog Dual Sensor Input (PAID) Control Strategies

Use a PAID control strategy to provide one analog Process Variable (PV) by using two analog input signals, from sources such as dual sensors, dual transmitters, and dual-input channels. The PAID instruction monitors the conditions of the channels and reports configured PV quality. The PAID instruction has functions for input selection, averaging, and failure detection. Should one of the two upstream PAI signals have bad quality, the PAID continues to provide an output using the remaining good quality signal. If both upstream signals are flagged as bad, the PAID PV is also flagged as bad. Additional functions, such as filtering and alarming, are done by a downstream PAI control strategy.

The following PAID control strategies are available as routines in the process library:

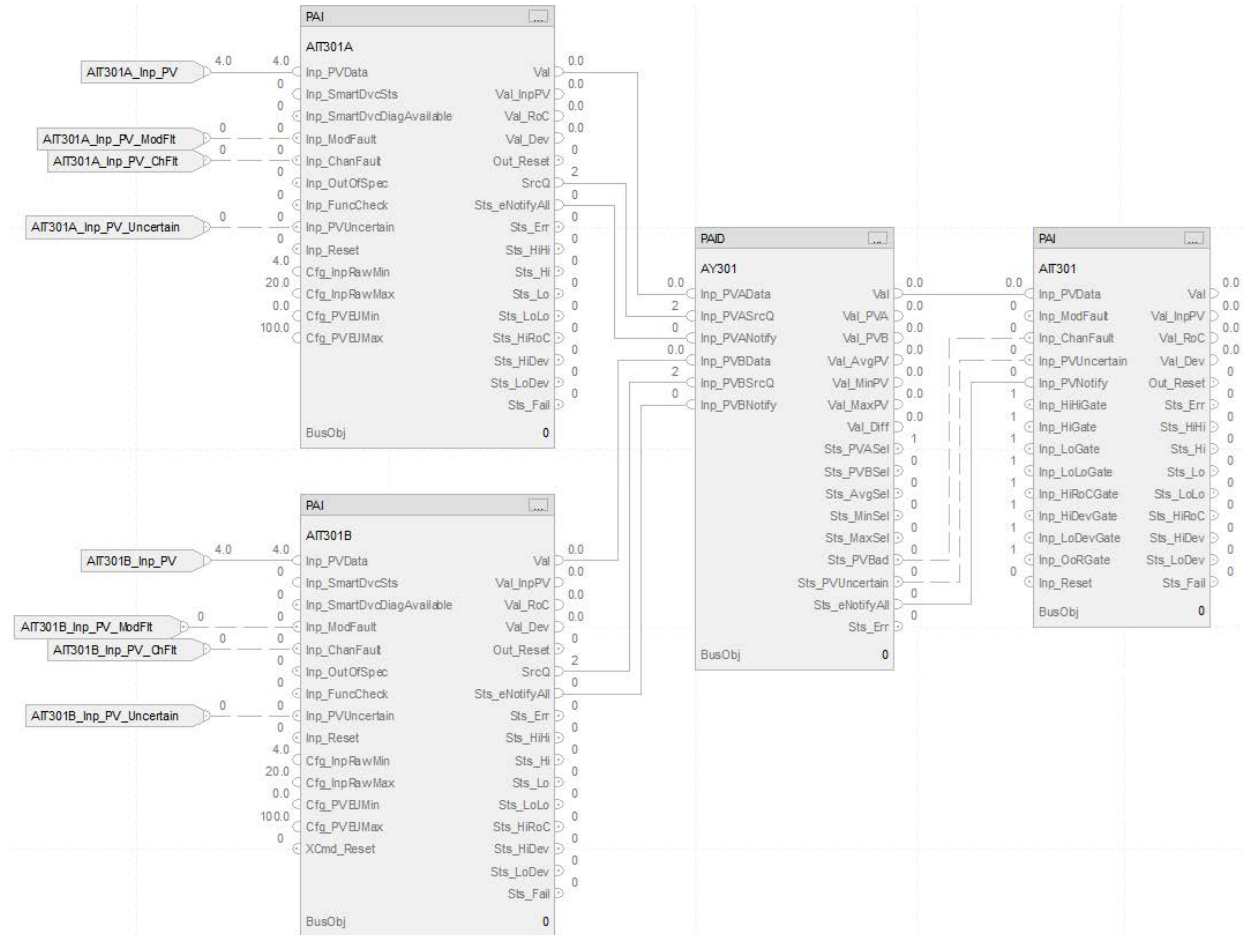
- CS\_PAID
- CS\_PAID\_HART
- CS\_PAID\_EtherNetIP
- CS\_PAID\_EtherNetIP\_NoHB
- CS\_PAID\_FF
- CS\_PAID\_PA

Import the appropriate control strategy as a **routine** in your controller project.

Each PAID control strategy contains one Function Block sheet:

Sheet	Description
CS_PAID	Process Analog Dual Sensor Input instruction
CS_PAID_HART	Process Analog Dual Sensor Input instruction with HART input
CS_PAID_EtherNetIP	Process Analog Dual Sensor Input instruction with EtherNetIP input
CS_PAID_EtherNetIP_NoHB	Process Analog Dual Sensor Input instruction with NoHB EtherNetIP input
CS_PAID_FF	Process Analog Dual Sensor Input instruction with FOUNDATION Fieldbus input
CS_PAID_PA	Process Analog Dual Sensor Input instruction with Profibus PA input

## CS\_PAID Sheet



### PAI Input References

See [CS\\_PAID Sheet on page 148](#) for details.

- Substitute AIT301A for the first instance of XT101
- Substitute AIT301B for the second instance of XT101



## PAI Outputs to PAID Inputs

Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)
SrcQ	Value for Inp_PVSrcQ parameter Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

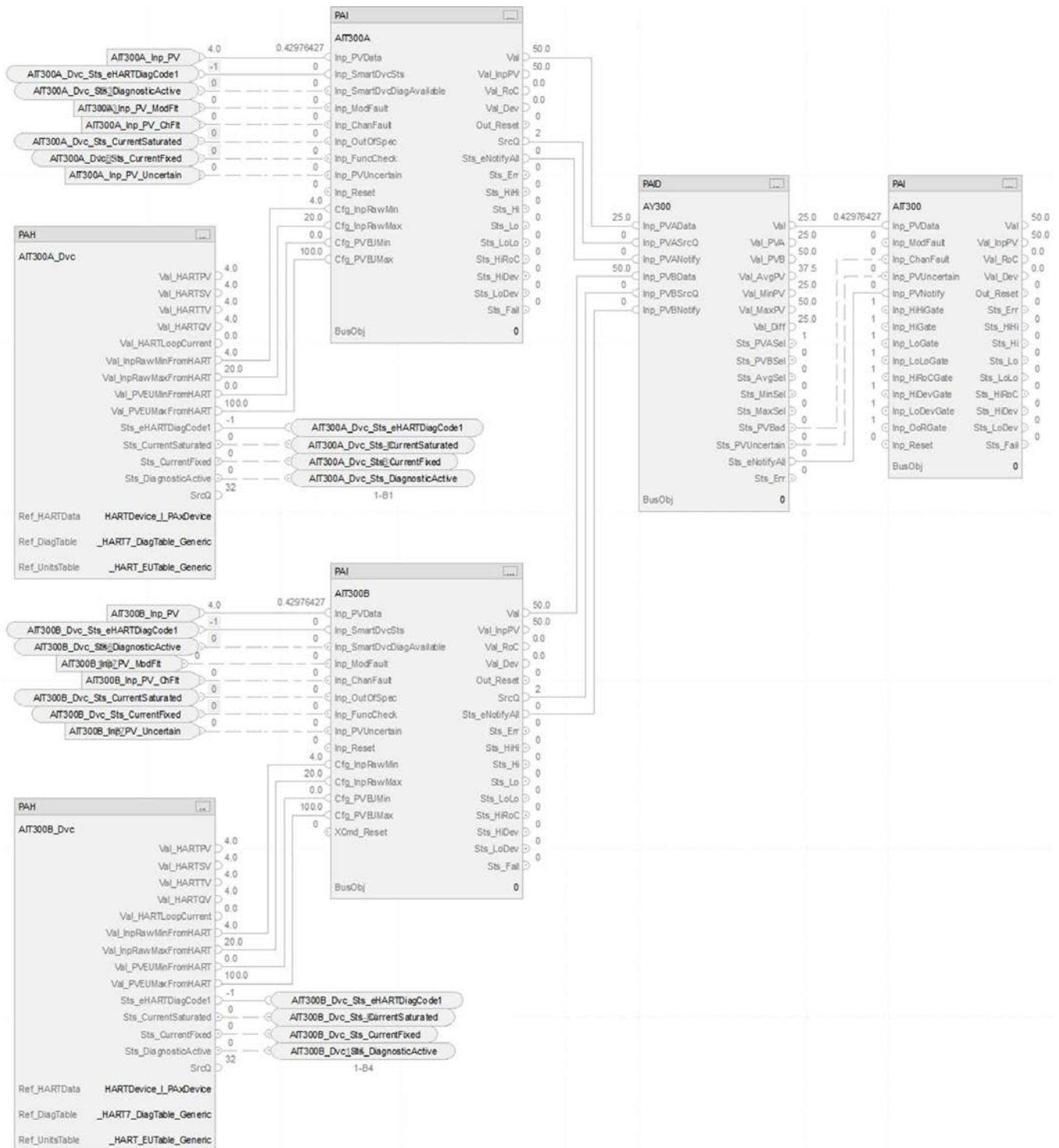
## PAID Outputs to PAI Inputs

Parameter	Description
Val	Analog PV, including substitute PV, if used (PV units)
Sts_PVBad	1 = PV bad quality or out of range
Sts_PVUncertain	1 = PV value is uncertain (quality)
Sts_eNotifyAll	Alarm status

## PAID Configuration Considerations

Operand	Type	Description
PlantPax® control	P_ANALOG_INPUT_DUAL	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PAID\_HART Sheet



The CS\_PAID\_HART control strategy operates the same as the CS\_PAID control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute AIT300A for the first instance of XT100
- Substitute AIT300B for the second instance of XT100

For more information, see [HART Integration on page 61](#).

## CS\_PAID\_EtherNetIP Sheet

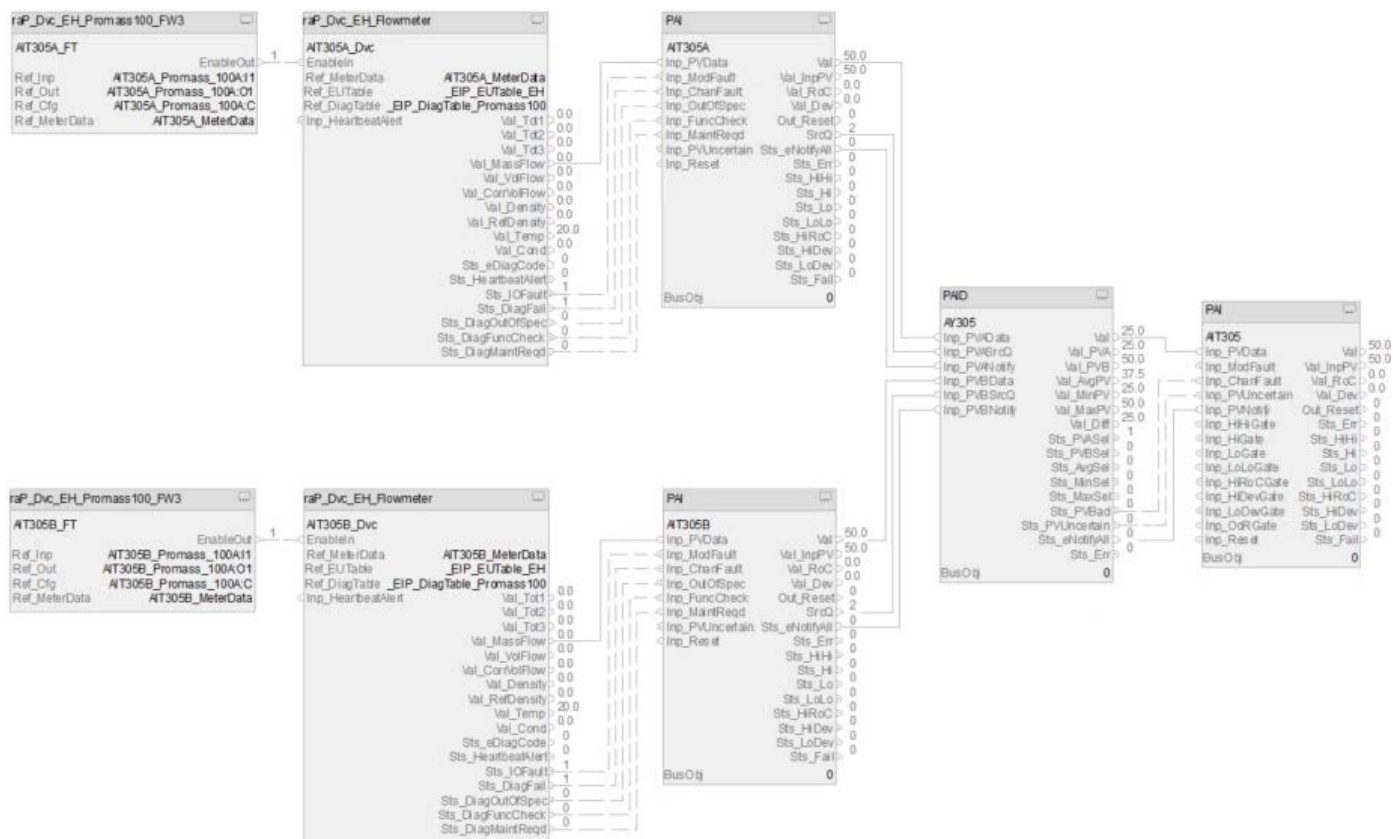


The CS\_PAID\_EtherNetIP control strategy operates the same as the CS\_PAID control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAID\\_EtherNetIP Sheet on page 151](#).
- Substitute AIT302A for the first instance of XT100
- Substitute AIT302B for the second instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PAID\_EtherNetIP\_NoHB Sheet



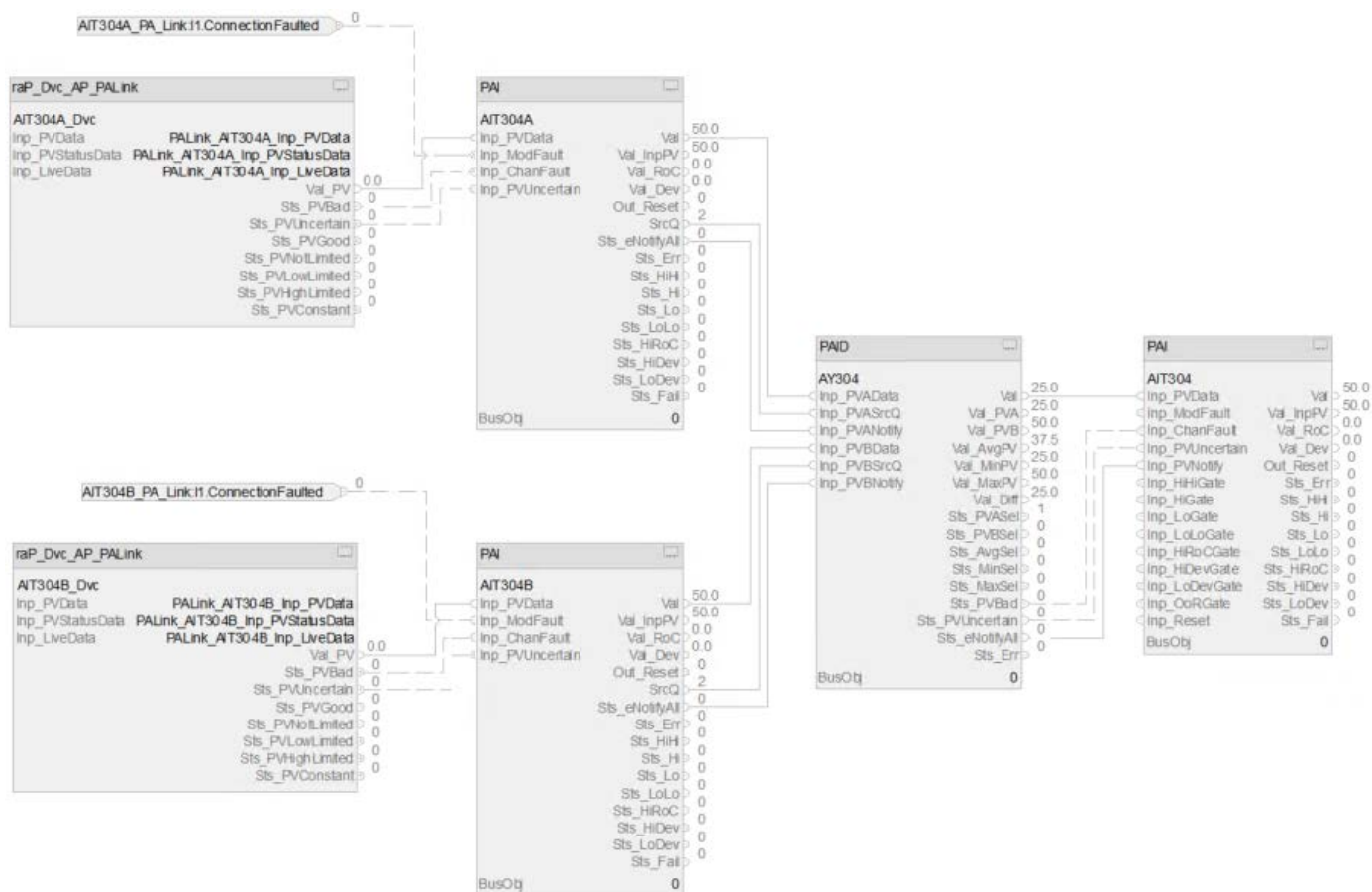
The CS\_PAID\_EtherNetIP\_NoHB control strategy operates the same as the CS\_PAID control strategy but relies on Ethernet/IP NoHB input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute AIT305A for the first instance of XT100
- Substitute AIT305B for the second instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).



## CS\_PAID\_PA Sheet



The CS\_PAID\_PA control strategy operates the same as the CS\_PAID control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PA\\_I PA Sheet on page 156](#).
- Substitute AIT304A for the first instance of XT100
- Substitute AIT304B for the second instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## ACM Considerations for PAID

Configure the **IO.SignalType** parameter first because it affects the visibility of the remaining parameters in the PAID object.



## ACM-Based Parameters for a PAID Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_RefTag	always	Link to analog input reference for PAID Outputs
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_HasPVA	always	Set if Inp_PVA is connected in logic
Cfg_HasPVB	always	Set if Inp_PVB is connected in logic
Cfg_HasHARTA	IO_Signal_Type=HART CfgHasPVA=True	Set if analog input A is connected to a HART device.
Cfg_HasHARTB	IO_Signal_Type=HART CfgHasPVB=True	Set if analog input B is connected to HART device.
Cfg_HasEHA	IO_Signal_Type=EH_EthernetIP CfgHasPVA=True	Set if analog input A is connected to a E+H EtherNet/IP device.
Cfg_HasEHB	IO_Signal_Type=EH_EthernetIP CfgHasPVB=True	Set if analog input B is connected to E+H EtherNet/IP device
Cfg_HasFFA	IO_Signal_Type=FF CfgHasPVA=True	Set if analog input A is connected to a FOUNDATION Fieldbus device
Cfg_HasFFB	IO_Signal_Type=FF CfgHasPVB=True	Set if analog input B is connected to FOUNDATION Fieldbus device
Cfg_HasPAA	IO_Signal_Type=PA CfgHasPVA=True	Set if analog input A is connected to a Profibus PA device
Cfg_HasPAB	IO_Signal_Type=PA CfgHasPVB=True	Set if analog input B is connected to Profibus PA device
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>

### 03.00 - IO Configuration

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table

### 03.01 - IO Configuration

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

PAIA_RefTag	Cfg_HasPVA=True	Link to PAI reference
Inp_PVA	Cfg_HasPVA=True IO_SignalType=None IO_SignalType=HART	Link to the analog input A PV reference
Ref_HartDeviceA	IO_Signal_Type=HART Cfg_HasPVA=True Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModuleA	IO_Signal_Type=EH_EthernetIP Cfg_HasPVA=True Cfg_HasEHA=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_ModuleA	IO_Signal_Type=FF Cfg_HasPVA=True Cfg_HasFFA=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_ModuleA	IO_Signal_Type=PA Cfg_HasPVA=True Cfg_HasPAA=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

### 03.02 - IO Configuration

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

PAIB_RefTag	Cfg_HasPVB=True	Link to PAI reference
-------------	-----------------	-----------------------

Parameter	Visible When	Details
Inp_PVB	Cfg_HasPVB=True IO_SignalType=None IO_SignalType=HART	Link to the analog input B PV reference
Ref_HartDeviceB	IO_SignalType=HART CfgHasHARTB=True CfgHasPVB=True Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModuleB	IO_SignalType=EH_EthernetIP Cfg_HasPVB=True Cfg_HasEHB=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_ModuleB	IO_SignalType=FF Cfg_HasPVB=True Cfg_HasFFB=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_ModuleB	IO_SignalType=PA Cfg_HasPVB=True Cfg_HasPAB=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.03 - Ref PAI Alarm Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Ref_HiHiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasFailAlm	always	If Cfg_HasFailAlm =True, ACM displays section 4.01 - Input Fail Alarm with additional parameters
Cfg_HasDiffAlm	always	If Cfg_HasDiffAlm=True, ACM displays section 4.07 - Signal Difference Alarm with additional parameters
Cfg_HasNoneGoodAlm	always	If Cfg_HasNoneGoodAlm=True, ACM displays section 4.10 - No Good Alarm with additional parameters
Cfg_HasOneGoodAlm	always	If Cfg_HasOneGoodAlm=True, ACM displays section 4.11 - One Good Alarm with additional parameters

**Additional Sub-Object for a PAID Instance**

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>



## Process Analog Input Multi Sensor (PAIM) Control Strategies

Use a PAIM control strategy to provide one analog Process Variable (PV) by using as many as eight analog input signals from sources such as sensors, transmitters, and input channels. The PAIM instruction monitors the conditions of the channels and reports configured PV quality. The PAIM instruction has functions for input selection, averaging, and failure detection. In addition, there is configuration selection for the minimum number of good, unrejected input signals required to have a good PV value, and an alarm if the required number of good inputs is not met. Configure which PV to use if there are only two unrejected signals remaining: the lesser, the greater, or the average of the two. Additional functions, such as for filtering and alarming, are done by a downstream PAI block.

The following PAIM control strategies are available as routines in the process library:

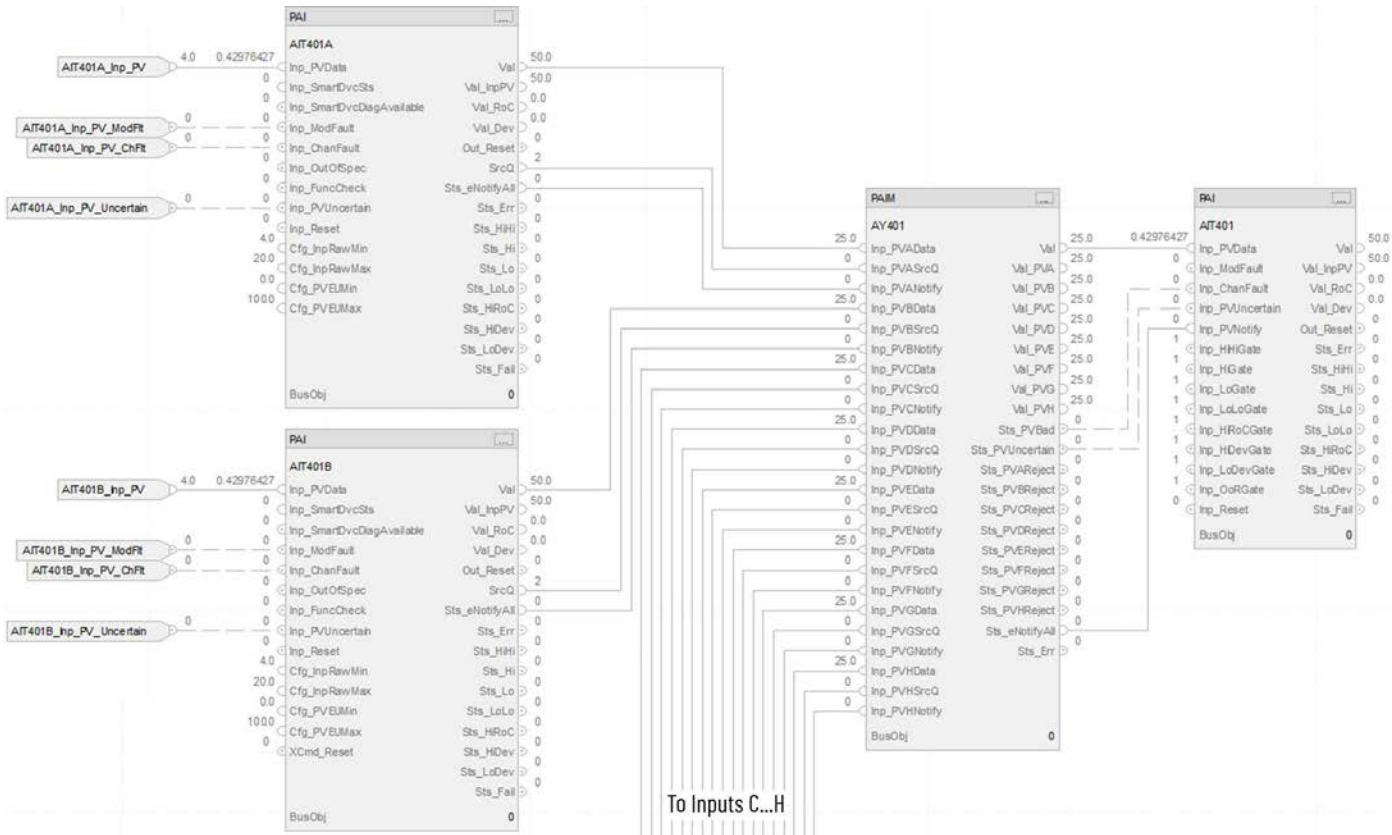
- CS\_PAIM
- CS\_PAIM\_HART
- CS\_PAIM\_EtherNetIP
- CS\_PAIM\_EtherNetIP\_NoHB
- CS\_PAIM\_FF
- CS\_PAIM\_PA

Import the appropriate control strategy as a **routine** in your controller project.

Each PAIM control strategy contains one Function Block sheet:

Sheet	Description
CS_PAIM	Process Analog Input Multi Sensor Input instruction
CS_PAIM_HART	Process Analog Input Multi Sensor Input instruction with HART input
CS_PAIM_HART_EtherNetIP	Process Analog Input Multi Sensor Input instruction with EtherNetIP input
CS_PAIM_EtherNetIP_NoHB	Process Analog Input Multi Sensor Input instruction with NoHB EtherNetIP input
CS_PAIM_FF	Process Analog Input Multi Sensor Input instruction with FOUNDATION Fieldbus input
CS_PAIM_PA	Process Analog Input Multi Sensor Input instruction with Profibus PA input

CS\_PAIM Sheet



PAI Input References

- See [CS\\_PA1 Sheet on page 148](#) for details.
- Substitute AIT401x for each instance of XT101

## PAI Outputs to PAIM Inputs

Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)
SrcQ	Value for Inp_PVSrcQ parameter Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

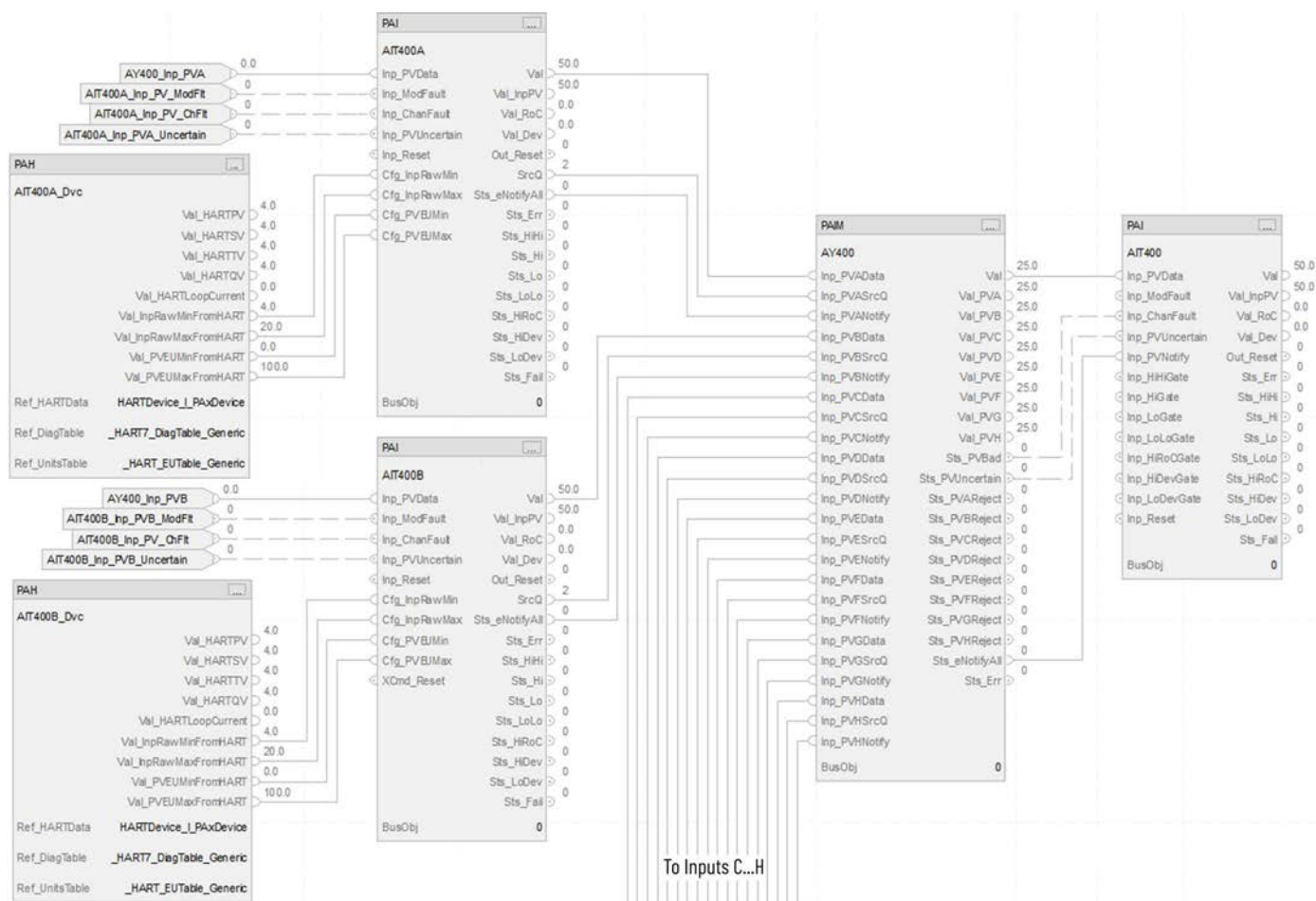
## PAIM Outputs to PAI Inputs

Parameter	Description
Val	Analog PV, including substitute PV, if used (PV units)
Sts_PVBad	1 = PV bad quality or out of range
Sts_PVUncertain	1 = PV value is uncertain (quality)
Sts_eNotifyAll	Alarm status

## PAIM Configuration Considerations

Operand	Type	Description
PlantPax® control	P_ANALOG_INPUT_MULTI	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PAIM\_HART Sheet



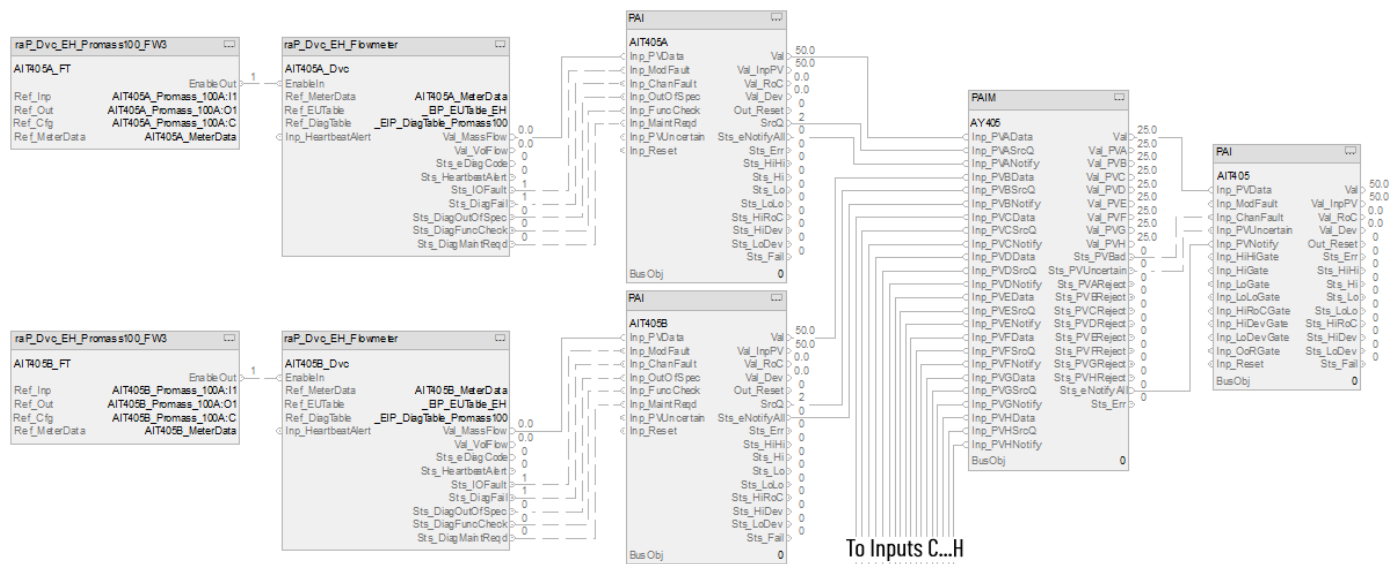
The CS\_PAIM\_HART control strategy operates the same as the CS\_PAIM control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute AIT400x for each instance of XT100

For more information, see [HART Integration on page 61](#).

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PAIM\_EtherNetIP\_NoHB Sheet

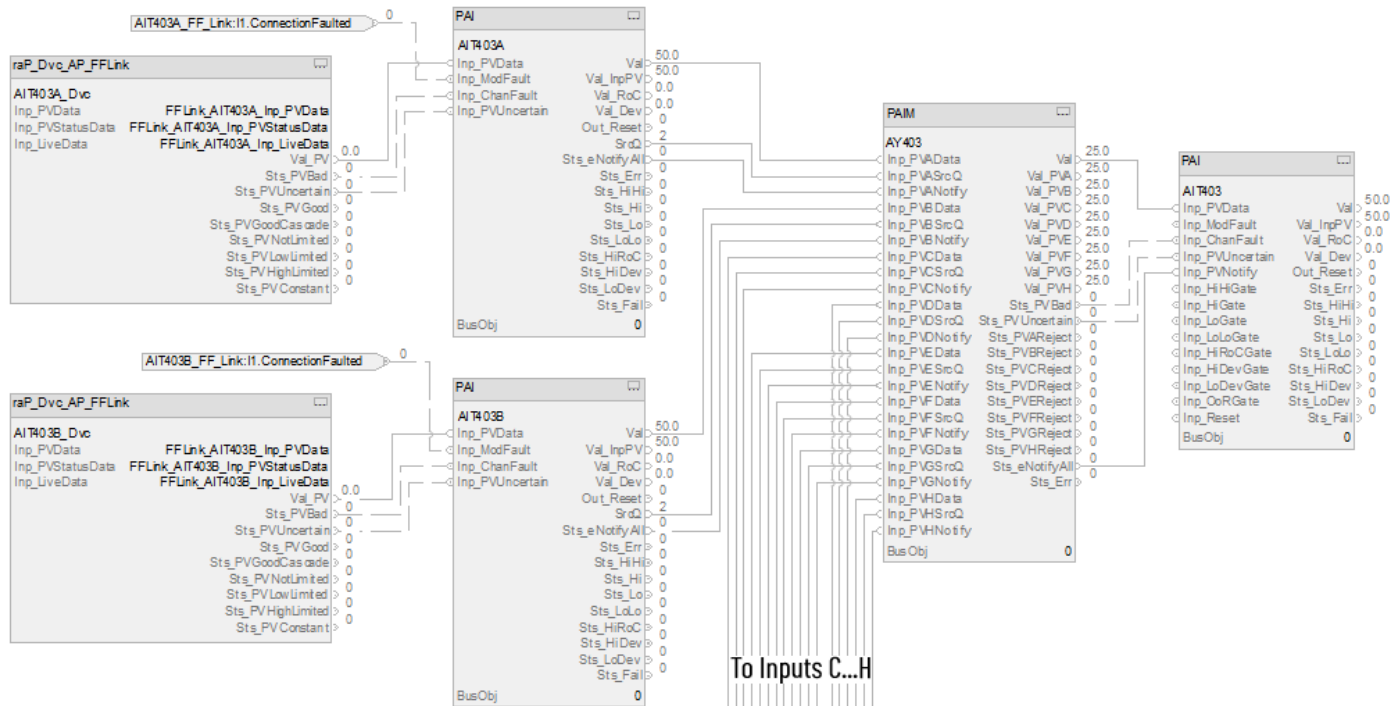


The CS\_PAIM\_EtherNetIP\_NoHB control strategy operates the same as the CS\_PAIM control strategy but relies on EtherNet/IP NoHB input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PA1\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute AIT405x for each instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PAIM\_FF Sheet



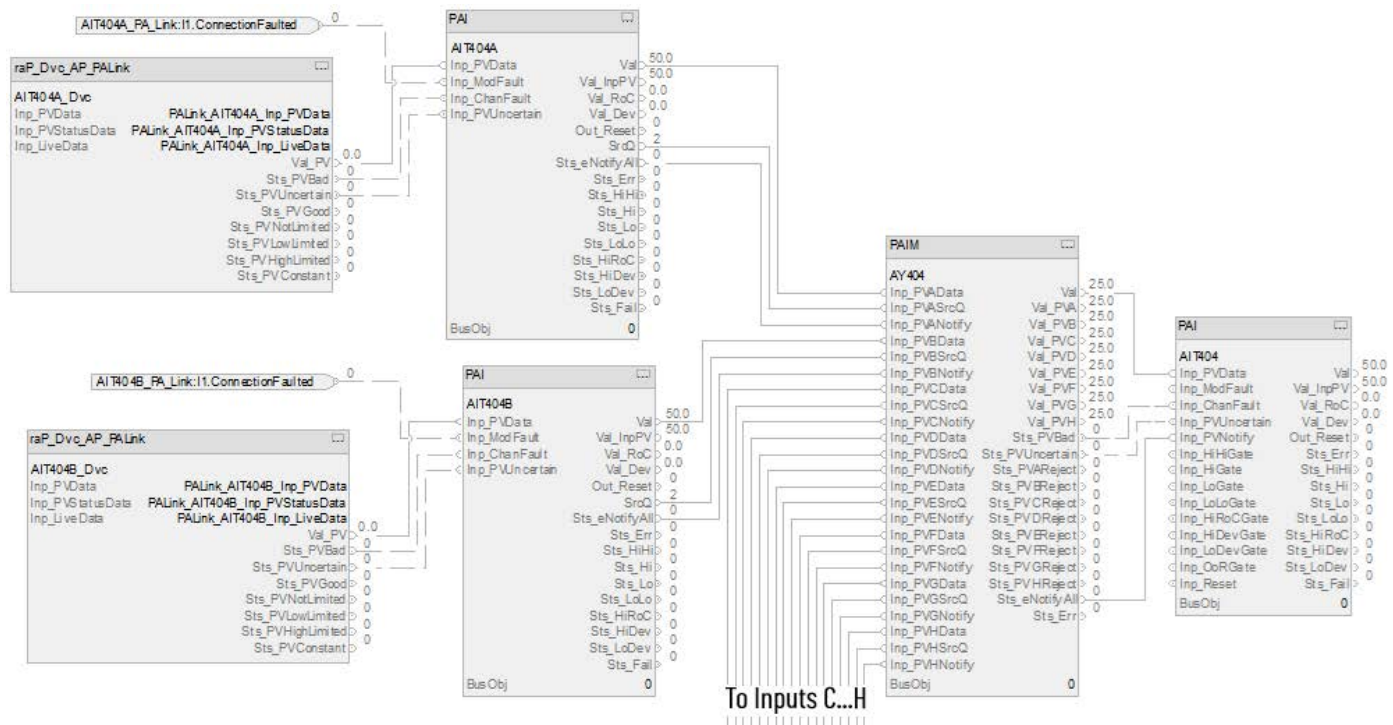
The CS\_PAIM\_FF control strategy operates the same as the CS\_PAIM control strategy but relies on FOUNDATION Fieldbus input data.

- For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAIM\\_FF Sheet on page 155](#).
- Substitute AIT403x for each instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).



CS\_PAIM\_PA Sheet



The CS\_PAIM\_PA control strategy operates the same as the CS\_PAIM control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute AIT404x for each instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

ACM Considerations for PAIM

Configure the IO\_SignalType parameter first because it affects the visibility of the remaining parameters in the PAIM object.

ACM-Based Parameters for a PAIM Instance

Where x = A, B, C, D, E, F, G, H

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_RefTag	always	Link to analog input reference for PAIM outputs
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_HasPVx	always	Set if the associated Inp_PV is connected in logic
Cfg_HasHARTx	IO_Signal_Type=HART CfgHasPVx=True	Set if analog input x is connected to a HART device
Cfg_HasEHx	IO_Signal_Type=EH_EthernetIP CfgHasPVx=True	Set if analog input x is connected to a E+H EtherNet/IP device
Cfg_HasFFx	IO_Signal_Type=FF CfgHasPVx=True	Set if analog input x is connected to a FOUNDATION Fieldbus device
Cfg_HasPAx	IO_Signal_Type=PA CfgHasPVx=True	Set if analog input x is connected to a Profibus PA device



Parameter	Visible When	Details
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>

**03.00 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table

**03.0# - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Where x: 1=A, 2=B, 3=C, 4=D, 5=E, 6=F, 7=G, 8=H

PAIx_RefTag	Cfg_HasPVx=True	Link to PAI reference
Inp_PVx	Cfg_HasPVx=True IO_SignalType=None IO_SignalType=HART	Link to the analog input x PV reference
Ref_HartDevicex	IO_Signal_Type=HART Cfg_HasPVA=True Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModulex	IO_Signal_Type=EH_EthernetIP Cfg_HasPVA=True Cfg_HasEHA=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Modulex	IO_Signal_Type=FF Cfg_HasPVA=True Cfg_HasFFA=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Modulex	IO_Signal_Type=PA Cfg_HasPVA=True Cfg_HasPAA=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.09 - Ref PAI Alarm Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Ref_HiHiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasFailAlm	always	If Cfg_HasFailAlm =True, ACM displays section 4.01 - Input Fail Alarm with additional parameters
Cfg_HasMinGoodAlm	always	If Cfg_HasMinGoodAlm=True, ACM displays section 4.11 - Min Good Alarm with additional parameters
Cfg_HasAnyRejectAlm	always	If Cfg_HasAnyRejectAlm=True, ACM displays section 4.13 - Any Reject Alarm with additional parameters

## Additional Sub-Object for a PAIM Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Process Analog Output (PAO) Control Strategies

Use a PAO control strategy to drive an analog field device to a reference value. The reference value can be entered by operator entry at the HMI or by a program input. The control strategies use a single output to an analog output channel to drive the field device. Optional opened and closed limit indications can be configured requiring additional digital input(s).

Alternatively, use the PAO control strategies to position the field device by using two digital output pulses (one to pulse open and another to pulse close). Pulsed outputs to position the field device require two digital output channels to position the device, as well as an additional analog channel to represent the current field device position. Digital positioning also requires additional configuration in the PAO instruction for the pulse timing.

In addition to positioning a field device based on program or HMI entries, these control strategies provide the ability to position (shed) the device based on IO fault status and interlock conditions.

The following PAO control strategies are available as routines in the process library:

- CS\_PAO (with interlocks)
- CS\_PAO\_noIntlk (without interlocks)
- CS\_PAO\_HART (HART input with interlocks)
- CS\_PAO\_HART\_noIntlk (HART input with no interlocks)

Import the appropriate control strategy as a **routine** in your controller project.

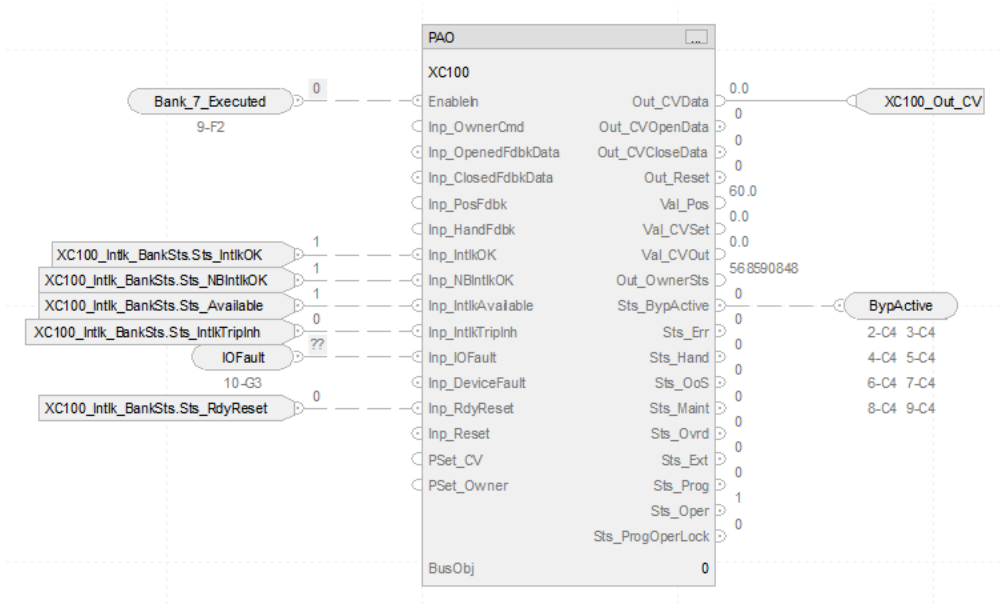
The PAO control strategies contain these Function Block sheets:

Sheet	Description
CS_PAO	Process Analog Output instruction
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	Only in CS_PAO. The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors one analog output channel for I/O fault input and raises alarm on an I/O fault.

The PAO HART control strategies contain these Function Block sheets:

Sheet	Description
CS_PAO_HART	Process Analog Output instruction with HART input
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	Only in CS_PAO. The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors one analog output channel for I/O fault input and raises alarm on an I/O fault.

CS\_PAO Sheet



PAO Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XC100_Inp_PosFdbk	Tieback input. Feedback from actual device position PV (CV engineering units). Valid any float. Default is 0.0.
XC100_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XC100_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XC100_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XC100_Intlk_BankSts.Sts_IntkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
IOFault	Input connection from IO Faults sheet
XC100_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

PAO Output References

Parameter	Description
XC100_Out_CVData	Control Variable output CV output in raw (I/O Card) units. Extended properties of this member: Engineering Unit - Raw units (text) used for the analog output Destination: Analog output channel or downstream REAL tag
BypActive	Output connection to interlock bank sheet

For a HART analog output, include these outputs:

HART Parameter	Description
XC101_Val_CVOut	Value of CV Output after optional rate limiting, in engineering units. Extended Properties of this member: Engineering Unit - Engineering units (text) used for the analog output Destination: Analog output channel or downstream REAL tag
XC101_Sts_Available	1 = Analog output available for control by automation (Program)

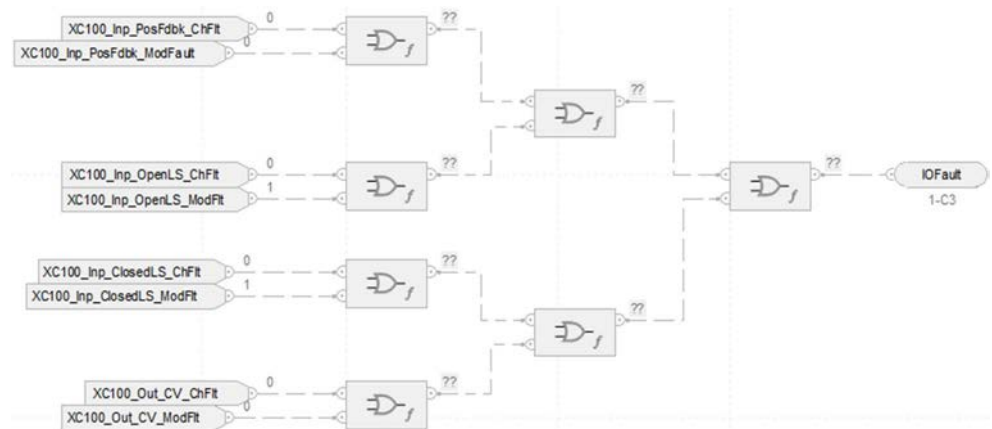
## PAO Configuration Considerations

Operand	Type	Description
PlantPax® control	P_ANALOG_OUTPUT	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

If you use digital output pulses, configure these PAO **Parameter**:

Parameter	Description
Cfg_HasPulseOut	Enables the pulse1 = Device provides pulse output (open, close).
Cfg_HasOpenedFdbk	1 = Use device opened feedback for failure checking.
Cfg_HasCloseFdbk	1 = Device provides closed feedback signal.

## IO Faults Sheet



## Fault Input References

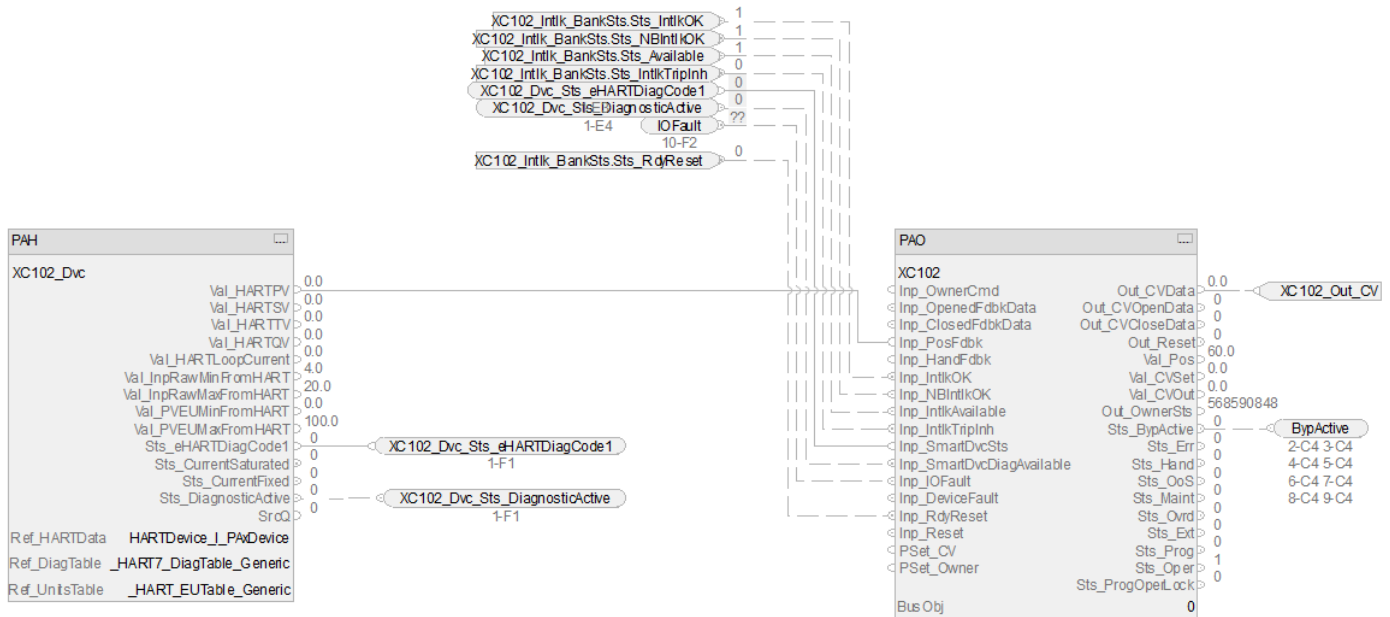
Parameter	Description
XC100_Inp_PosFdbk_ChanFault	Tieback input channel fault
XC100_Inp_PosFdbk_ModFault	Tieback input module fault
XC100_Inp_OpenLS_ChFit	Open limit switch channel fault
XC100_Inp_OpenLS_ModFit	Open limit switch module fault
XC100_Inp_ClosedLS_ChFit	Closed limit switch channel fault
XC100_Inp_ClosedLS_ModFit	Closed limit switch module fault
XC100_Out_CVData_ChanFault	Control Variable output channel fault
XC100_Out_CVData_ModFault	Control Variable output data module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PAO sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## CS\_PAO\_HART Sheet



The CS\_PAO\_HART control strategy, including the Interlock and IO Fault sheets, operates the same as the CS\_PAO control strategy but relies on HART input data. Substitute XC101 for XC100.

For more information, see [HART Integration on page 61](#).

## ACM Considerations for PAO

Configure these parameters first because they affect the visibility of the remaining parameters in the PAO object.

- If you use HART data, set `IO_Signal_Type=HART` and `Cfg_UsedInOther=False`
- If the PAO is used in another control strategy, set `IO_Signal_Type=None` and `Cfg_UsedInOther=True`

## ACM-Based Parameters for a PAO Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_UsedInOther	always	PAO can be standalone (False) or used as part of another PPID_PAO, CC, IMC, MMC, or RMPS control strategy (True)
OTHER_RefTag	Cfg_UsedInOther=True	Link to the other control strategy that uses this PAO instance.
IO_Signal_Type	Cfg_UsedInOther=False	Select the signal type: None or HART.
Use_OOAP	Has_OOAP=True (controller parameter) Cfg_UsedInOther=False	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>00.01.00 - Data - HART</b>		
Cfg_UseHARTText	IO_Signal_Type=HART	Set to use the text configuration from the HART device
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True Cfg_UsedInOther=False	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	Cfg_UsedInOther=False	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
<b>02 - Device Configuration</b>		
Cfg_HasPulseOut	Always	Set if the device has pulse output (open or close)
<b>02.01 - Device Configuration</b>		
Cfg_HasOpenedFdbk	always	Set if the device has an open feedback input
Cfg_HasClosedFdbk	always	Set if the device has a closed feedback input
Cfg_HasPosFdbk	IO_Signal_Type=None	Set if the device has a position feedback input
Cfg_HasCombinedFdbk	always	Set if the device has combined feedback providing open, closed, and position.
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Ref_PSet_CV	always	Link the Pset CV reference tag. This will be connected to the PAO.Pset_CV input parameter.
Inp_OpenLS	Cfg_UsedInOther=False Cfg_HasOpenedFdbk=True Cfg_UsedInOther=False Cfg_HasCombinedFdbk=True	Link the open limit switch feedback input reference
Inp_ClosedLS	Cfg_UsedInOther=False Cfg_HasClosedFdbk=True Cfg_UsedInOther=False Cfg_HasCombinedFdbk=True	Link the closed limit switch feedback input reference
Out_CV	Cfg_UsedInOther=False Cfg_HasPulseOut=False	Link the CV output reference
Out_CVOpen	Cfg_HasPulseOut=True	Link the CV open output reference
Out_CVClose	Cfg_HasPulseOut=True	Link the CV close output reference
<b>03.00 - IO Configuration</b>		
Cfg_UseHARTDigitalData	Cfg_UsedInOther=False IO_Signal_Type=HART	Set the references for the HART PV, SV, TV, and values
Hart_Type	Cfg_UsedInOther=False IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table

Parameter	Visible When	Details
Ref_HartDevice	Cfg_UsedInOther=False IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
<b>04 - Alarm Configuration</b>		
Cfg_HasDevAlm	always	If Cfg_HasDevAlm=True, ACM displays section 4.01 - Dev Alarm with additional parameters
Cfg_DeviceFaultAlm	always	If Cfg_DeviceFaultAlm=True, ACM displays section 4.02 - Device Fault Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.04 - Interlock Trip Alarm with additional parameters

## Additional Sub-Objects for a PAO Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>



## Notes:



## Process Boolean Logic (PBL) Control Strategy

Use the PBL control strategy to process as many as four digital inputs by applying as many as eight gates of configurable logic. Gate types available include AND, OR, XOR (Exclusive-OR), Set/Reset, Select, and Majority. A benefit of the PBL control strategy is that assembly of the logical gates is done from the HMI, which helps to make sure that the HMI representation is accurate with respect to the underlying logic.

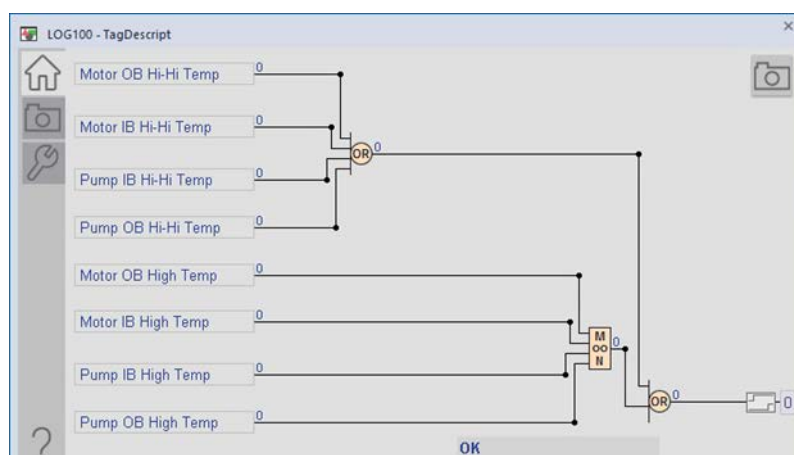
The PBL instruction can record its current state (via snapshot of current graphical representation):

- After a change in output state
- On Operator or Program command
- Based on a logic loopback input

Use the PBL instruction in these situations:

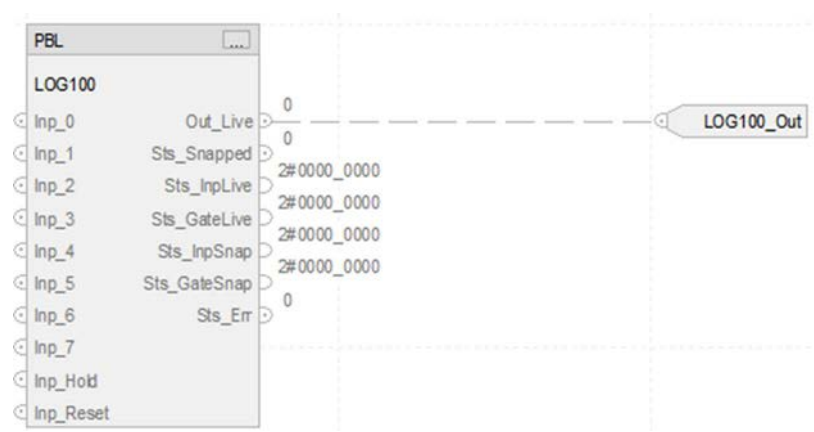
- A project requires an Interlock or Permissive condition that is more complicated than the simple OR-ing or AND-ing provided by the PINTLK (Interlocks) or PPERM (Permissives) Add-On Instructions.
- A project requires some Boolean (combination) logic that can be reconfigured from the HMI online, or which requires the snapshot capability for saving a copy of the logic state with a time stamp.
- A project contains more than the 16 interlock conditions or permissive conditions provided by the PINTLK and PPERM Add-On Instructions, but some of the conditions can be grouped together under one identification. For example, all bearing over-temperature signals for a pump and motor (Pump Inboard Bearing, Pump Outboard Bearing, Motor Inboard Bearing, and Motor Outboard Bearing) can be ORed together in a PBL instruction and the result presented to a PINTLK instruction as a single Bearing Overtemp condition.

The PBL logic is typically configured from an HMI display.



The CS\_PBL control strategy is available as a routine in the process library. Import the control strategy as a **routine** in your controller project. The PBL control strategy contains one CS\_PBL Function Block sheet.

CS\_PBL Sheet



PBL Input References

Parameter	Description
Inp_0	Logic inputs
Inp_1	
Inp_2	
Inp_3	
Inp_4	
Inp_5	
Inp_6	
Inp_7	

PBL Output Reference

Parameter	Description
LOG100_Out_Live	Condition logic output (result) after delay.

PBL Configuration Considerations

Operand	Type	Description
PBL tag	P_BOOLEAN_LOGIC	Instance of data structure (backing tag) required for proper operation of instruction

ACM Considerations for a PBL Instance

Configure the parameters to process as many as four digital inputs by applying as many as eight gates of configurable logic.

## ACM-Based Parameters for a PBL Instance

Parameter	Visible When	Details
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
RefTag_Inp0	always	Link to logic input 0
RefTag_Inp1	always	Link to logic input 1
RefTag_Inp2	always	Link to logic input 2
RefTag_Inp3	always	Link to logic input 3
RefTag_Inp4	always	Link to logic input 4
RefTag_Inp5	always	Link to logic input 5
RefTag_Inp6	always	Link to logic input 6
RefTag_Inp7	always	Link to logic input 7
Out_Live	always	Link to the output reference

## Additional Sub-Object for a PBL Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**

## Process Discrete 2-, 3-, or 4-State Device (PD4SD) Control Strategy

The Process Discrete 2-, 3-, 4-state Device Add-On Instruction controls and monitors feedback (using up to four discrete outputs and up to four discrete feedbacks) from a discrete 2-state, 3-state, or 4-state device in various modes while monitoring for fault conditions. These devices include multiple-speed motors or multiple-position valves.

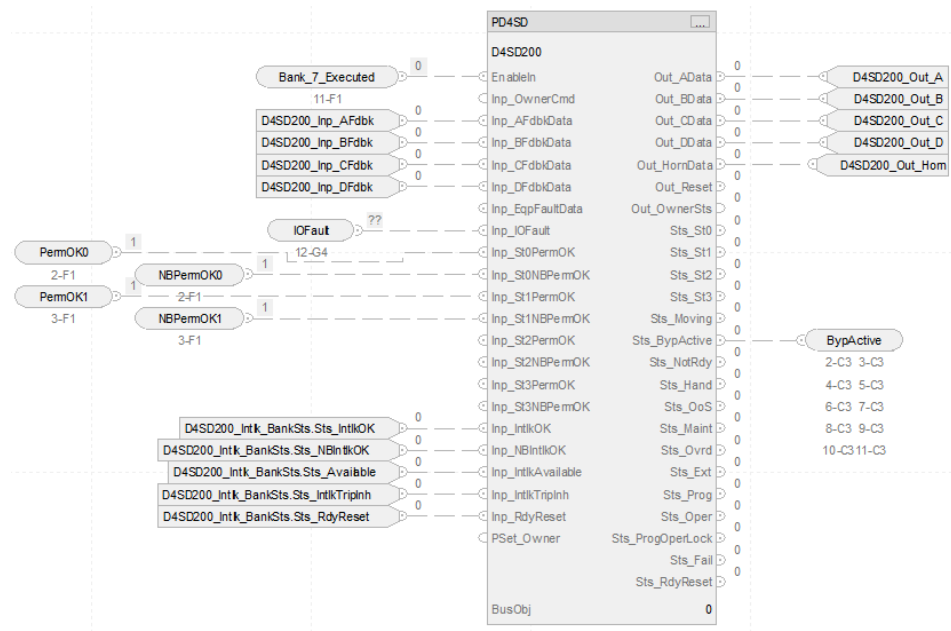
The CS\_PD4SD control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PD4SD control strategy contains these Function Block sheets:

Sheet	Description
CS_PD4SD	Discrete State Device Add-On Instruction
Permissives 0 Permissives 1 Permissives 2 Permissives 3	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Fault	The logic monitors as many as four discrete input channels and as many as five discrete output channels for I/O fault input and raises an alarm on an I/O fault.

## CS\_PD4SD Sheet



### PD4SD Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
D4SD200_Inp_FdbkA	Input Signal: Feedback A from device.
D4SD200_Inp_FdbkB	Input Signal: Feedback B from device.
D4SD200_Inp_FdbkC	Input Signal: Feedback C from device.
D4SD200_Inp_FdbkD	Input Signal: Feedback D from device.
IOFault	Input connection from IO Faults sheet
PermOK0	Input connection from Permissives sheet 0 (State 1) 1 = On permissives OK, device can turn On
NBPermOK0	Input connection from Permissives sheet 0 (State 1) 1 = Non-bypassable On permissives OK, device can turn On
PermOK1	Input connection from Permissives sheet 1 (State 2) 1 = On permissives OK, device can turn On
NBPermOK1	Input connection from Permissives sheet 1 (State 2) 1 = Non-bypassable On permissives OK, device can turn On
PermOK2	Input connection from Permissives sheet 2 (State 3) 1 = On permissives OK, device can turn On
NBPermOK2	Input connection from Permissives sheet 2 (State 3) 1 = Non-bypassable On permissives OK, device can turn On
PermOK3	Input connection from Permissives sheet 3 (State 4) 1 = On permissives OK, device can turn On
NBPermOK3	Input connection from Permissives sheet 3 (State 4) 1 = Non-bypassable On permissives OK, device can turn On
D4SD200_Intlk_BankSts_Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
D4SD200_Intlk_BankSts_Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
D4SD200_Intlk_BankSts_Sts_Available	Interlock bank status 1 = Available
D4SD200_Intlk_BankSts_Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
D4SD200_Intlk_BankSts_Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset



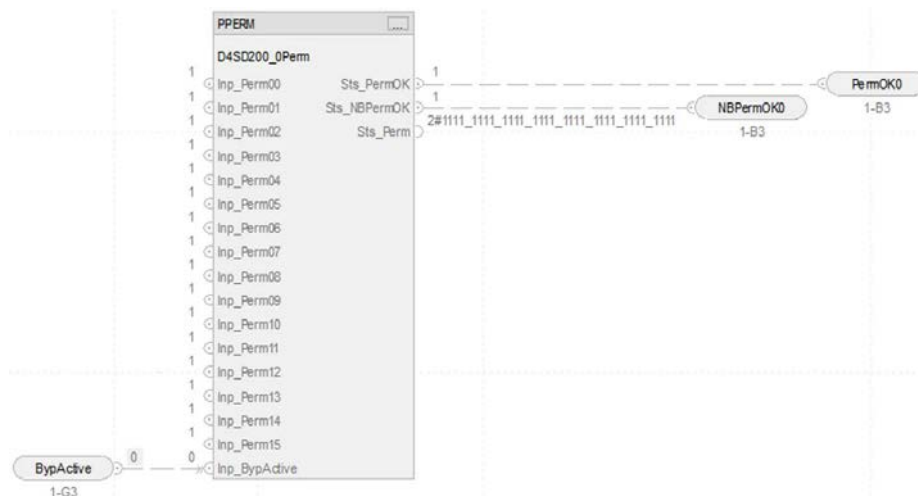
## PD4SD Output References

Parameter	Description
D4SD200_Out_A	Output A to device
D4SD200_Out_B	Output B to device
D4SD200_Out_C	Output C to device
D4SD200_Out_D	Output D to device
D4SD200_Out_Horn	1 = Sound audible before commanded state change
D4SD200_Out_Reset	1 = Reset command has been received and accepted
BypActive	Output connection to permissives and interlock bank sheets

## PD4SD Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_DISCRETE_4STATE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## Permissive Sheet



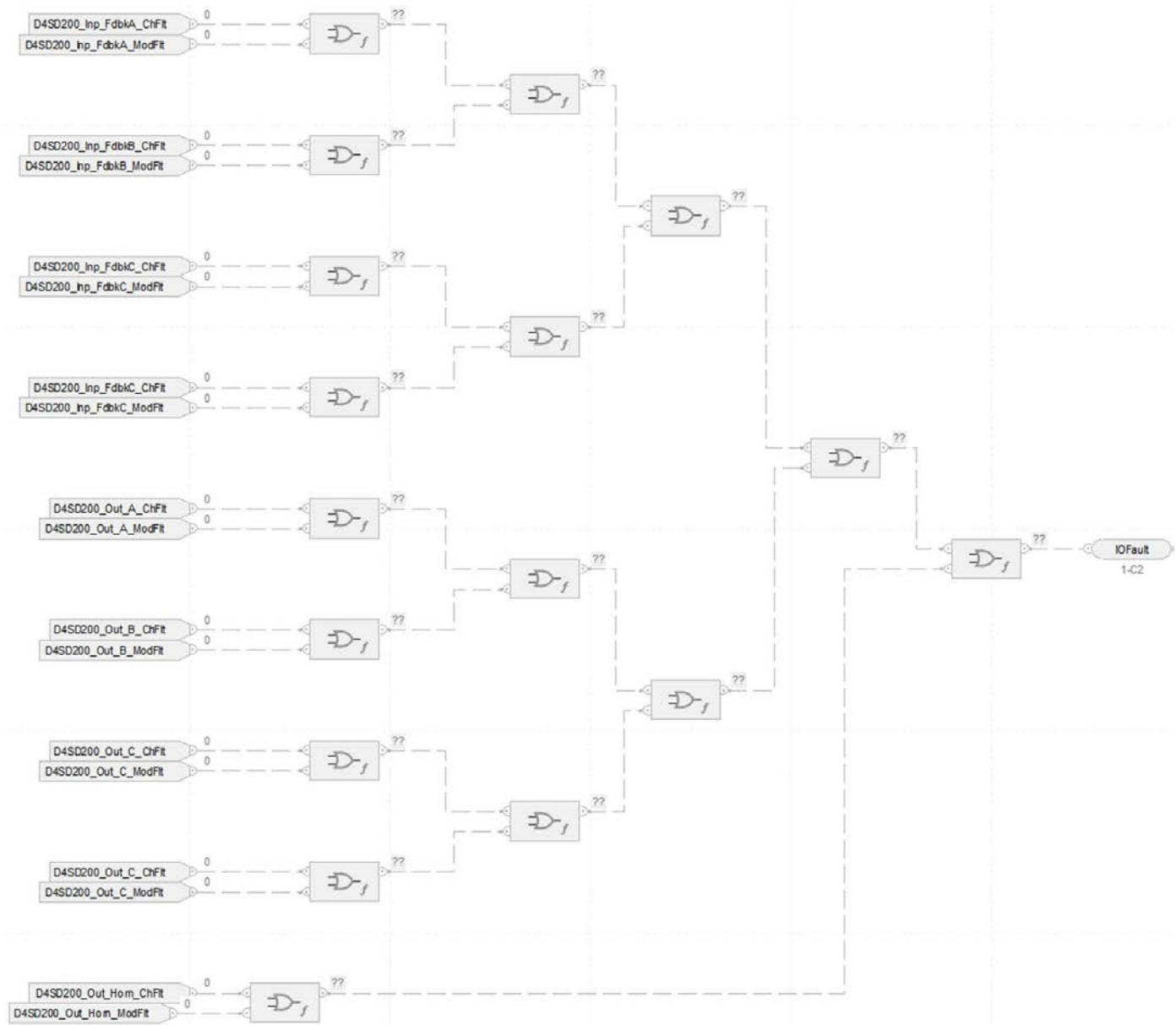
## PPERM Input References

Parameter	Description
BypActive	Input connection from the CS_PD4SD sheet

## PPERM Output References

Parameter	Description
PermOK0 PermOK1 PermOK2 PermOK3	Overall permissive status (1 = OK to energize)
NBPermOK0 NBPermOK1 NBPermOK2 NBPermOK3	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

IO Fault Sheet



## Fault Input References

Parameter	Description
D4SD200_Inp_FdkA_ChFlt D4SD200_Inp_FdkB_ChFlt D4SD200_Inp_FdkC_ChFlt D4SD200_Inp_FdkD_ChFlt	Tieback input channel faults
D4SD200_Inp_FdkA_ModFault D4SD200_Inp_FdkB_ModFault D4SD200_Inp_FdkC_ModFault D4SD200_Inp_FdkD_ModFault	Tieback input module faults
D4SD200_Out_A_ChFlt D4SD200_Out_B_ChFlt D4SD200_Out_C_ChFlt D4SD200_Out_D_ChFlt	Output channel faults
D4SD200_Out_A_ModFlt D4SD200_Out_B_ModFlt D4SD200_Out_C_ModFlt D4SD200_Out_D_ModFlt	Output module faults
D4SD200_Out_Horn_ChFlt	Sound audible for output channel fault
D4SD200_Out_Horn_ModFlt	Sound audible for output module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PD4SD sheet

For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

## ACM Considerations for a PD4SD Instance

Configure the Cfg\_NumState parameter to define the number of device states.

## ACM-Based Parameters for a PD4SD Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_NumStates	always	Enter the number of device states (2, 3, or 4) The default is 2.
Cfg_HasPerm0Obj	Cfg_NumStates >= 1	Set to create an instance of the PPERM instruction to allow a state 0 command
Cfg_HasPerm1Obj	Cfg_NumStates >= 2	Set to create an instance of the PPERM instruction to allow a state 1 command
Cfg_HasPerm2Obj	Cfg_NumStates >= 3	Set to create an instance of the PPERM instruction to allow a state 2 command
Cfg_HasPerm3Obj	Cfg_NumStates >= 4	Set to create an instance of the PPERM instruction to allow a state 3 command
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Inp_FdbkA	always	Link to the Feedback A reference
Inp_FdbkB	always	Link to the Feedback B reference
Inp_FdbkC	always	Link to the Feedback C reference
Inp_FdbkD	always	Link to the Feedback D reference
Out_A	always	Link to the Output A reference
Out_B	always	Link to the Output B reference
Out_C	always	Link to the Output C reference
Out_D	always	Link to the Output D reference
Out_Horn	always	Link to the horn output reference
<b>04 - Alarm Configuration</b>		
Cfg_HasEqpFaultAlm	always	If Cfg_HasEqpFaultAlm=True, ACM displays section 4.07 - Equipment Fault Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.09 - Fail Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.02 - I/O Fault Alarm with additional parameters

## Additional Sub-Objects for a PD4SD Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_0 Permissive_1 Permissive_2 Permissive_3	Configure permissives to allow state commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>



## Process Deadband Controller (PDBC) Control Strategies

Use the PDBC control strategy to maintain a PV within a deadband of the SP by triggering one or two digital outputs (a raise output and a lower output).

The following PDBC control strategies are available as routines in the process library:

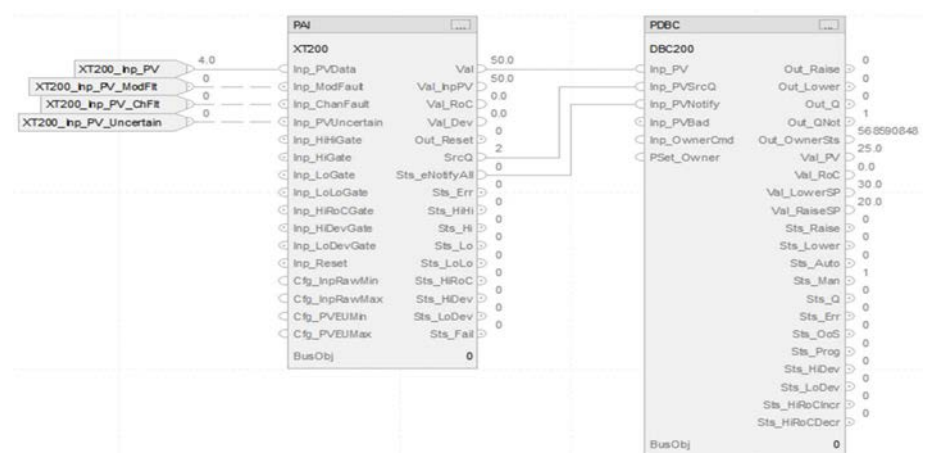
- CS\_PDBC
- CS\_PDBC\_HART
- CS\_PDBC\_EtherNetIP
- CS\_PDBC\_FF
- CS\_PDBC\_PA

Import the appropriate control strategy as a **routine** in your controller project.

The PDBC control strategy contains one Function Block sheet:

Sheet	Description
CS_PDBC	Process Deadband Controller instruction
CS_PDBC_HART	Process Deadband Controller with HART input
CS_PDBC_EtherNetIP	Process Deadband Controller with EtherNetIP
CS_PDBC_FF	Process Deadband Controller with FOUNDATION Fieldbus input
CS_PDBC_PA	Process Deadband Controller with PA input

### CS\_PDBC Sheet



### PAI Input References

See [CS\\_PA Sheet on page 148](#) for details.

- Substitute XT200 the instances of XT101

## PAI Outputs to PDBC Inputs

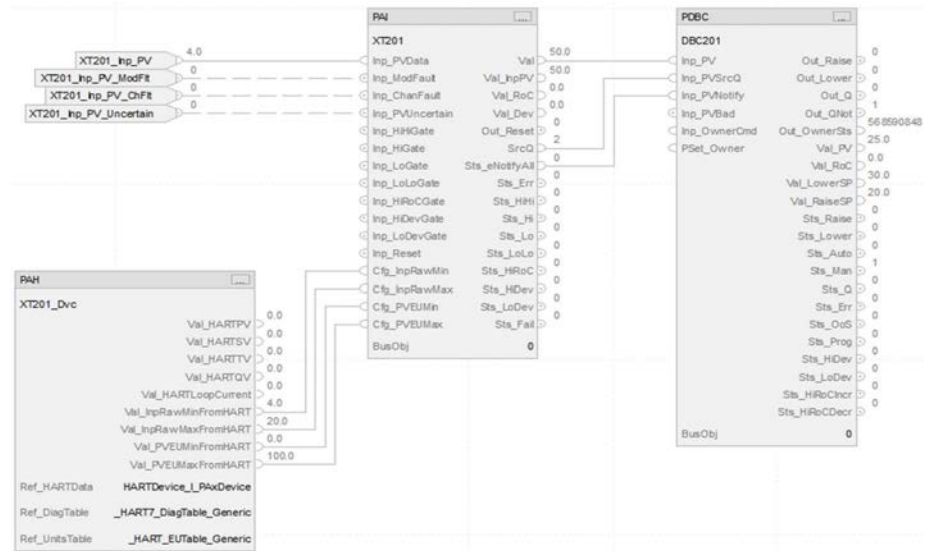
Parameter	Description
Val	Value for PV parameter Process Variable (PVEU)
SrcQ	Value for PDBC Inp_PVSrcQ parameter  Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PDBC Inp_PVNotify parameter  Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

## PDBC Configuration Considerations

Operand	Type	Description
PlantPax® control	P_DEADBAND	Instance of data structure (backing tag) required for proper operation of instruction.
BusObj	BUS_OBJ	Bus component for organization control 0 if not using organization Bus[x].Obj when using organization. See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .



## CS\_PDBC\_HART Sheet

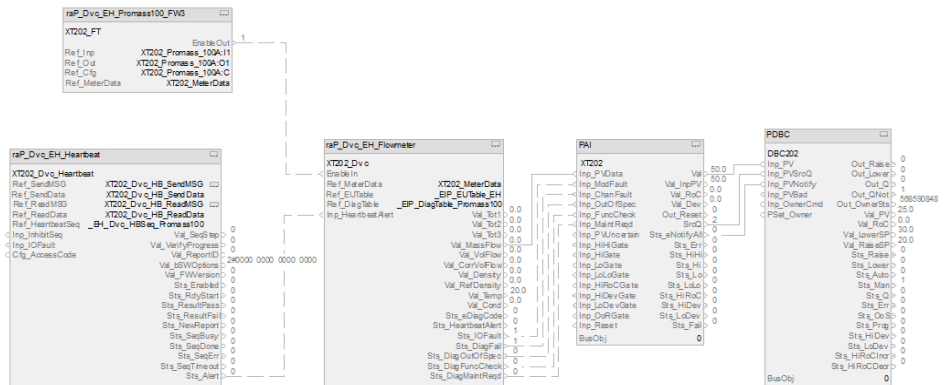


The CS\_PDBC\_HART control strategy operates the same as the CS\_PDBC control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute DBC201 for the PV data instance of XT100
- Substitute XT201 for the remaining instances of XT100

For more information, see [HART Integration on page 61](#).

## CS\_PDBC\_EtherNetIP Sheet

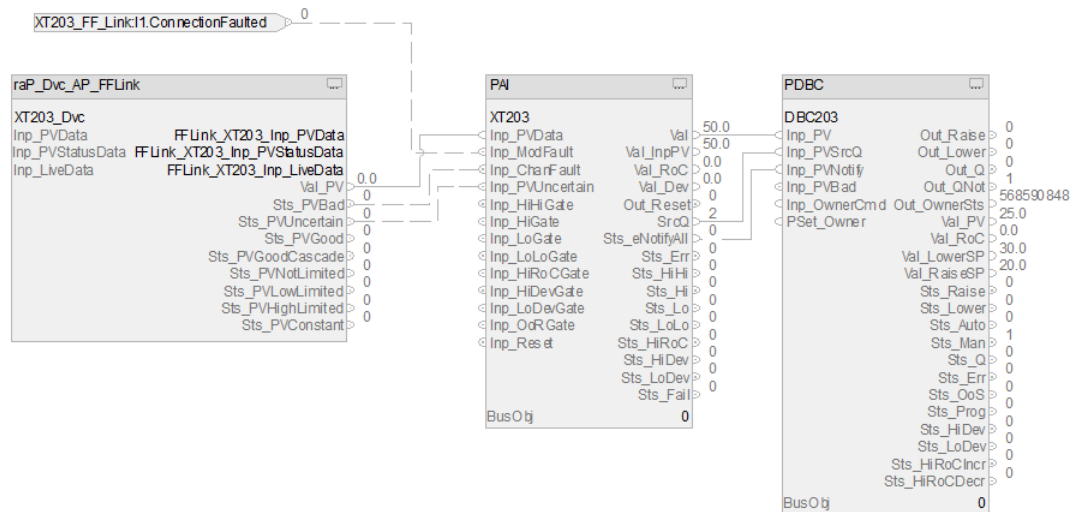


The CS\_PDBC\_EtherNetIP control strategy operates the same as the CS\_PDBC control strategy but relies on EtherNet/IP™ input data.

- For information on the EtherNet/IP outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute DBC202 for the PV data instance of XT100
- Substitute XT202 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PDBC\_FF Sheet

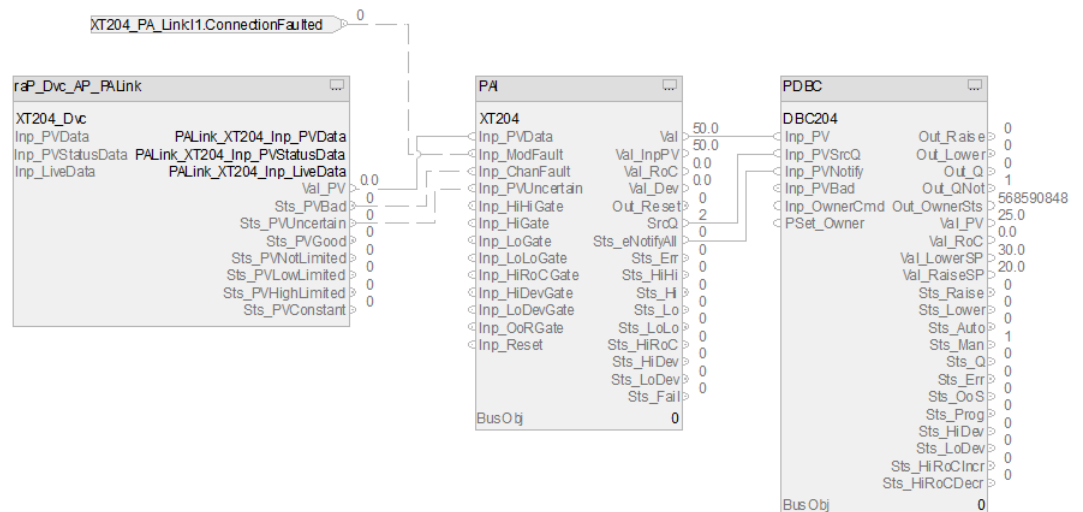


The CS\_PDBC\_FF control strategy operates the same as the CS\_PDBC control strategy but relies on FOUNDATION Fieldbus input data.

- For information on the FOUNDATION Fieldbus outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute DBC203 for the PV data instance of XT
- Substitute XT202 for the remaining instances of XT

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PDBC\_PA Sheet



The CS\_PDBC\_PA control strategy operates the same as the CS\_PDBC control strategy but relies on PROFIBUS PA input data.

- For information on the Profibus PA outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute DBC204 for the PV data instance of XT
- Substitute XT204 for the remaining instances of XT

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## ACM Considerations for PDBC

Configure these parameters first because they affect the visibility of the remaining parameters in the PDBC object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for a PDBC Instance

Parameter	Visible When	Details
00 - Selection		
PAI_Type	always	Important: Select this parameter first as the option affects the remaining parameters. Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
01 - Options		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
03.00 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PAI_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV input reference
	IO_SignalType=HART	
03.00.10 - Ref PAI Alarm Configuration		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference

Parameter	Visible When	Details
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference

#### 04 - Alarm Configuration

Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.01 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm =True, ACM displays section 4.02 - Lo Dev Alarm with additional parameters
Cfg_HasHiRoCIncr	always	If Cfg_HasHiRoCIncr=True, ACM displays section 4.03 - Hi RoCIncr Alarm with additional parameters
Cfg_HasHiRoCDec	always	If Cfg_HasHiRoCDec=True, ACM displays section 4.04 - Hi RoCDec Alarm with additional parameters

## Additional Sub-Object for a PDBC Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

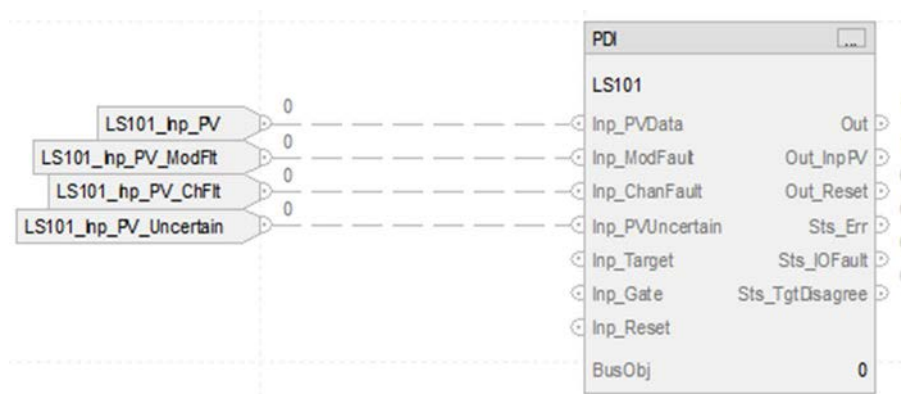
**Notes:**

## Process Discrete Input (PDI) Control Strategy

Use the PDI control strategy to monitor a discrete (true or false) input and check for alarm conditions. The PDI instruction processes a signal from a channel of a discrete input module. Use the PDI instruction with any discrete (BOOL) signal.

The CS\_PDI control strategy is available as a routine in the process library. Import the control strategy as a **routine** in your controller project. The PDI control strategy contains one CS\_PDI Function Block sheet.

### CS\_PDI Sheet



### PDI Input References

Parameter	Description
LS101.Inp_PVData	Process variable input Source: sensor or input
LS101.Inp_PV_ModFit	Process variable input module fault 1 = I/O module failure or module communication status bad 0 = OK
LS101.Inp_PV_ChFit	Process variable input channel fault 1 = I/O channel fault or failure 0 = OK
LS101.Inp_PV_Uncertain	Process variable input uncertain Indicates the channel data accuracy is undetermined 1 = The channel data is uncertain This input sets Sts_PVUncertain if not in Virtual

For examples on how to map device input tags to the Inp\_PVData, Inp\_ModFault, Inp\_ChanFault, and Inp\_PVUncertain references, see [PlantPax Control Strategies on page 21](#).

### PDI Configuration Considerations

Operand	Type	Description
PDI tag	P-DISCRETE-INPUT	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS-OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## ACM Considerations for a PDI Instance

Configure the PDI parameters to monitor a discrete input.

### ACM-Based Parameters for a PDI Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
<b>03 - IO Configuration</b> Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Inp_PV	always	Link to the PV reference
<b>04 - Alarm Configuration</b>		
Cfg_TgtDisagreeAlm	always	If Cfg_TgtDisagreeAlm=True, ACM displays section 4.01 - Target Disagree Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.02 - I/O Fault Alarm with additional parameters
<b>04.01 - Tag Disagree Alarm Configuration</b>		
Ref_Gate	Cfg_TgtDisagreeAlm=True	Link to the gate condition for the Tag Disagree alarm. Any BOOL tag can be used.

### Additional Sub-Object for a PDI Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>



## Process Discrete Output (PDO) Control Strategies

Use a PDO control strategy to drive a discrete (true/false) output, monitor discrete inputs serving as feedbacks from a device driven by the discrete output, and check for alarm conditions. Use the PDO instruction for a channel of a discrete output module. Use the PDO instruction with any discrete (BOOL) signal.

The PDO instruction operates in a variety of modes, and can provide steady, single pulsed, or continually pulsed output.

The following PDO control strategies are available as routines in the process library:

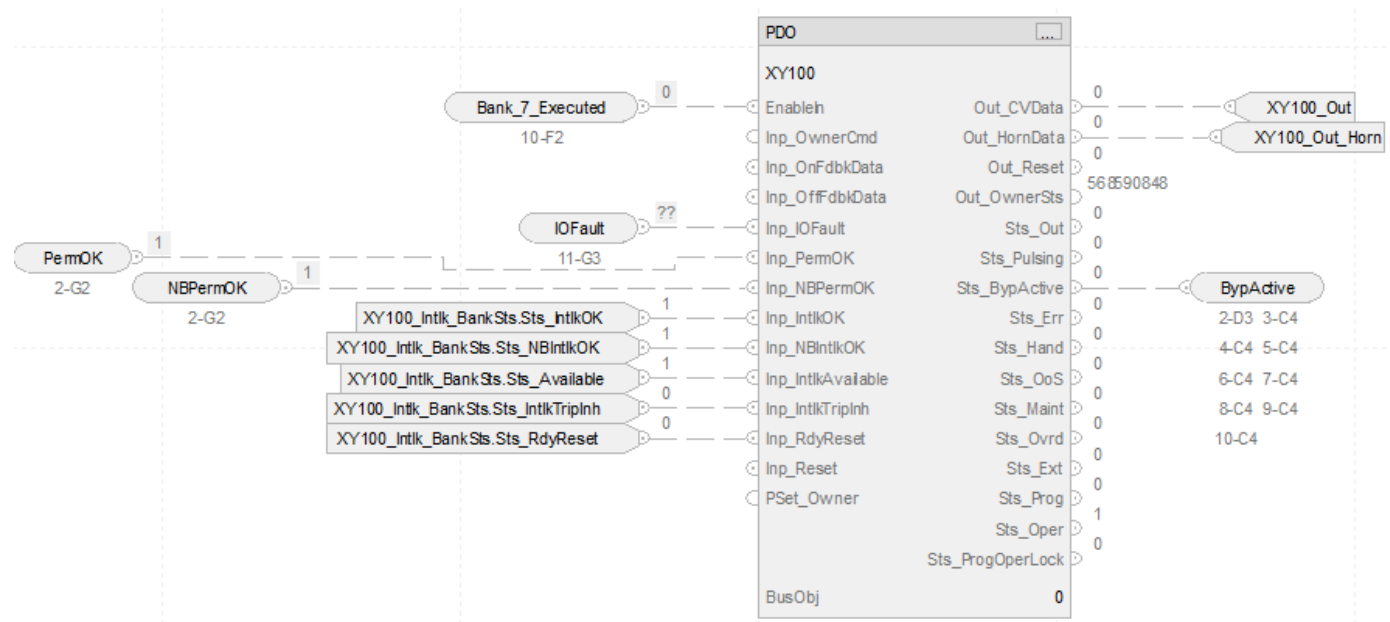
- CS\_PDO (with interlocks)
- CS\_PDO\_noIntlk (without interlocks)

Import the appropriate control strategy as a **routine** in your controller project.

The PDO control strategies contain these Function Block sheets:

Sheet	Description
CS_PDO	Process Discrete Output instruction
Permissive	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	Only in CS_PDO The PDO instruction monitors bypassable and non-bypassable Interlocks that force the output to the configured safe state. There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors the input and output modules and channels used to interface with the device for fault conditions and raises an alarm on an I/O fault.

## CS\_PDO Sheet



### PDO Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
IOFault	Input connection from the IO Faults sheet
PermOK	Input connection from Permissive sheet 1 = On permissives OK, device can turn On
NBPermOK	Input connection from Permissive sheet 1 = Non-bypassable On permissives OK, device can turn On
XY100_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XY100_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XY100_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XY100_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
XY100_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

### PDO Output References

Parameter	Description
XY100_Out_CVData	Control Variable output CV output in raw (I/O Card) units. Extended properties of this member: Engineering Unit - Raw units (text) used for the analog output
XY100_Out_Horn	1 = Sound audible prior to commanded state change
BypActive	Output connection to permissives and interlock bank sheets

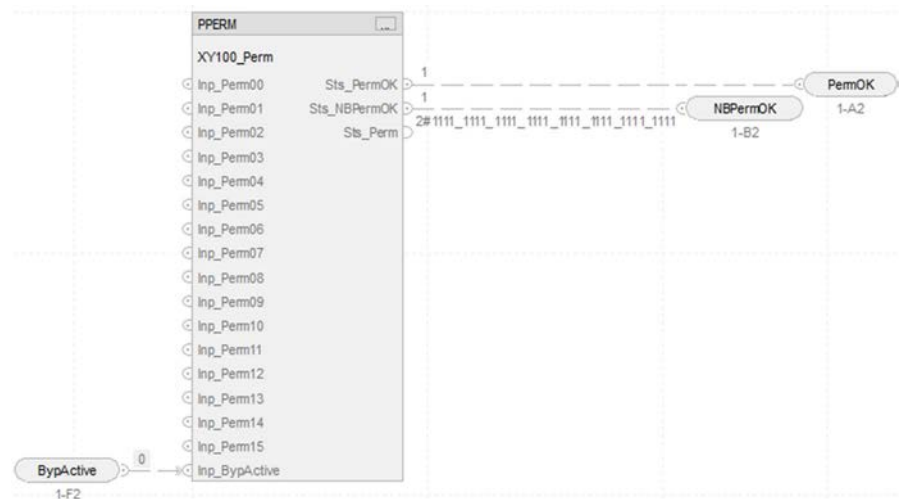
## PDO Configuration Considerations

Operand	Type	Description
PlantPax® control	P_DISCRETE_OUTPUT	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

If you use digital input pulses, configure these PDO operands:

Parameter	Description
Cfg_HasPulseOut	1 = Enable pulsing functions
Cfg_HasOnFdbk	1 = Device provides an On feedback signal
Cfg_HasOffFdbk	1 = Device provides an Off feedback signal

## Permissive Sheet



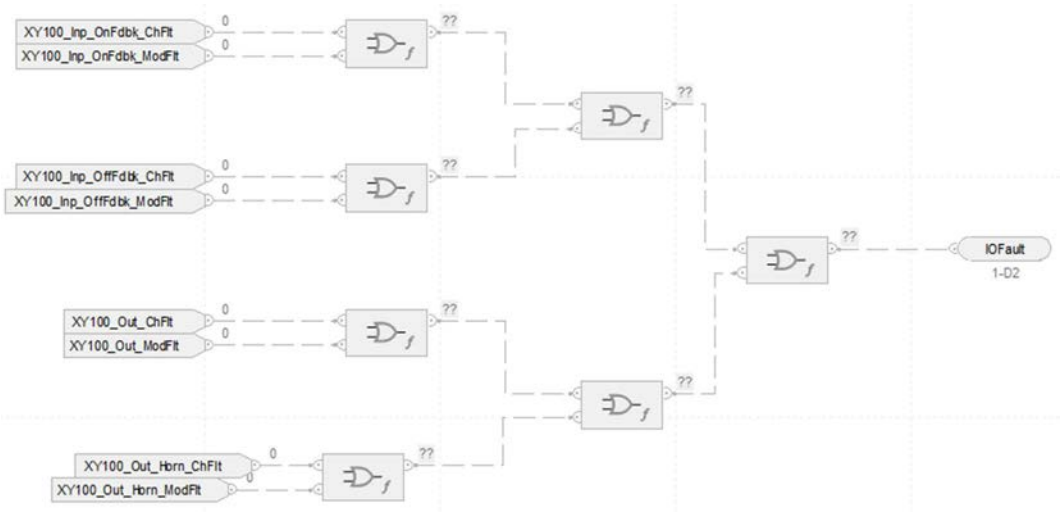
## PPERM Input References

Parameter	Description
BypActive	Input connection from the interlock bank sheet

## PPERM Output References

Parameter	Description
PermOK	Overall permissive status (1 = OK to energize)
NBPermOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

# IO Faults Sheet



## Fault Input References

Parameter	Description
XY100_Inp_OnFdbk_ChFlt	On feedback channel fault
XY100_Inp_OnFdbk_ModFlt	On feedback module fault
XY100_Inp_OffFdbk_ChFlt	Off feedback channel fault
XY100_Inp_OffFdbk_ModFlt	Off feedback module fault
XY100_Out_ChFlt	Output channel fault
XY100_Out_ModFlt	Output module fault
XY100_Out_Horn_ChFlt	Output horn channel fault
XY100_Out_Horn_ModFlt	Output horn module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PDO sheet

For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

## ACM Considerations for a PDO Instance

Configure the PDO parameters to drive a discrete output.

## ACM-Based Parameters for a PDO Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device. See <a href="#">Process Controller on page 36</a>
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow an output command.
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
<b>02.01- Device Configuration Feedback</b>		
Cfg_HasOnFdbk	always	Set if device has on feedback
Cfg_HasOffFdbk	always	Set if device has off feedback
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Inp_OnFdbk	Cfg_HasOnFdbk=True	Link to the on feedback reference
Inp_OffFdbk	Cfg_HasOffFdbk=True	Link to the off feedback reference
Out	always	Link to the output reference
Out_Horn	always	Link to the horn output reference
<b>04 - Alarm Configuration</b>		
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.04 - Interlock Trip Alarm with additional parameters
Cfg_HasOffFailAlm	always	If Cfg_HasOffFailAlm=True, ACM displays section 4.01 - Off Fail Alarm with additional parameters
Cfg_HasOnFailAlm	always	If Cfg_HasOnFailAlm=True, ACM displays section 4.02 - On Fail Alarm with additional parameters

## Additional Sub-Objects for a PDO Instance

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive	Configure permissives to allow an output command See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>



## Process Dosing Flow Meter (PDOSEFM) Control Strategies

Use the PDOSEFM control strategy to control an ingredient addition that uses a flow meter to measure the quantity of ingredient added. The flow meter can be an analog flow meter (signal proportional to flow), a pulse generating flow meter (pulse count proportional to quantity delivered), or a digital flow meter providing flow rate or quantity (totalized flow) information.

The following PDOSEFM control strategies are available as routines in the process library:

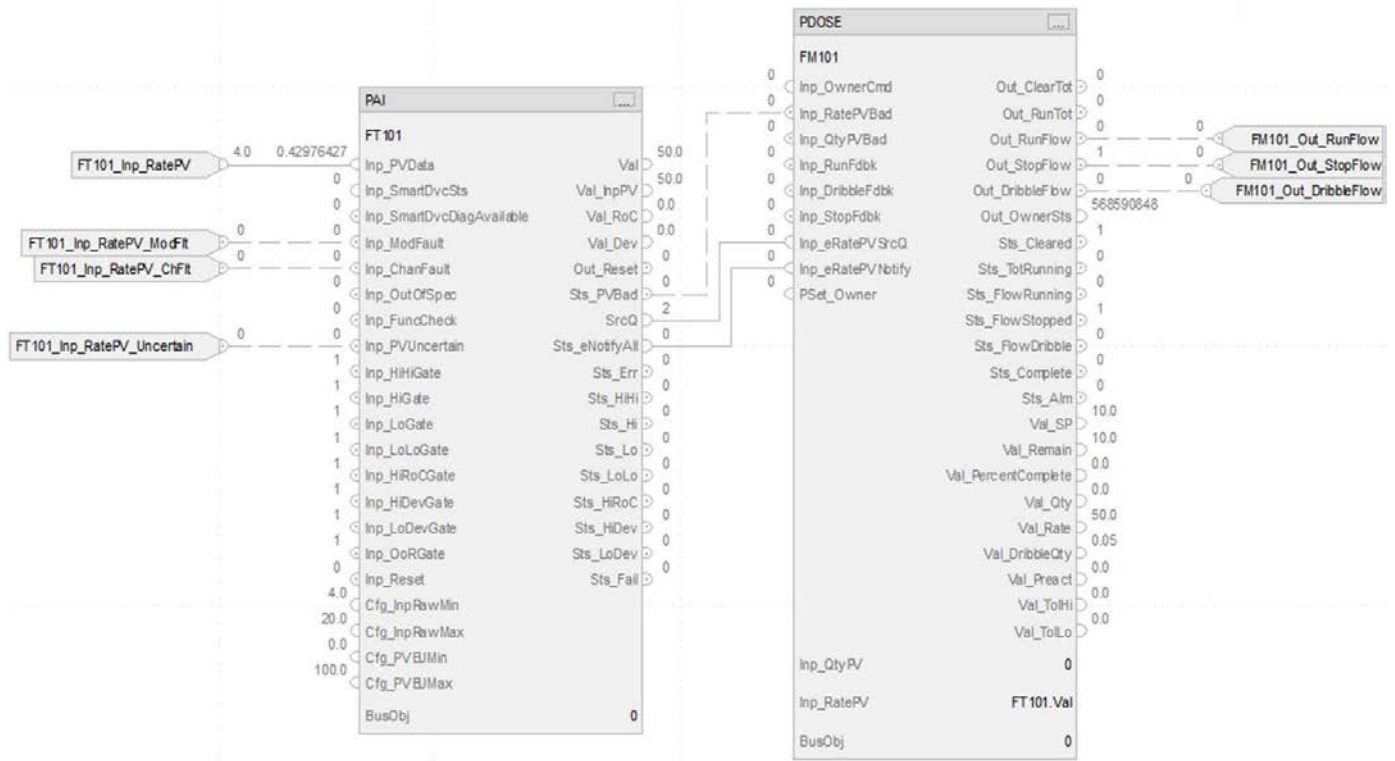
- CS\_PDOSEFM
- CS\_PDOSEFM\_HART
- CS\_PDOSEFM\_EtherNetIP
- CS\_PDOSEFM\_EtherNetIP\_NoHB
- CS\_PDOSEFM\_FF
- CS\_PDOSEFM\_PA

Import the appropriate control strategy as a **routine** in your controller project.

The PDOSEFM control strategy contains one Function Block sheet:

Sheet	Description
CS_PDOSEFM	Process Dosing Flow Meter instruction
CS_PDOSEFM_HART	Process Dosing Flow Meter instruction with HART input
CS_PDOSEFM_EtherNetIP	Process Dosing Flow Meter instruction with EtherNetIP input
CS_PDOSEFM_EtherNetIP_NoHB	Process Dosing Flow Meter instruction with No HB EtherNetIP input
CS_PDOSEFM_FF	Process Dosing Flow Meter instruction with Foundation Fieldbus input
CS_PDOSEFM_PA	Process Dosing Flow Meter instruction with PA input

CS\_PDOSEFM Sheet



PAI Input References

- See [CS\\_PAI Sheet on page 148](#) for details.
- Substitute FM101 for the PV data instance of XT101
  - Substitute FT101 for the remaining instances XT101



## PAI Outputs to PDOSE Inputs

Parameter	Description
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad
SrcQ	Source and quality of primary value or status: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	All alarm status enumerated values including related objects: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged, or unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

## PDOSE Output References

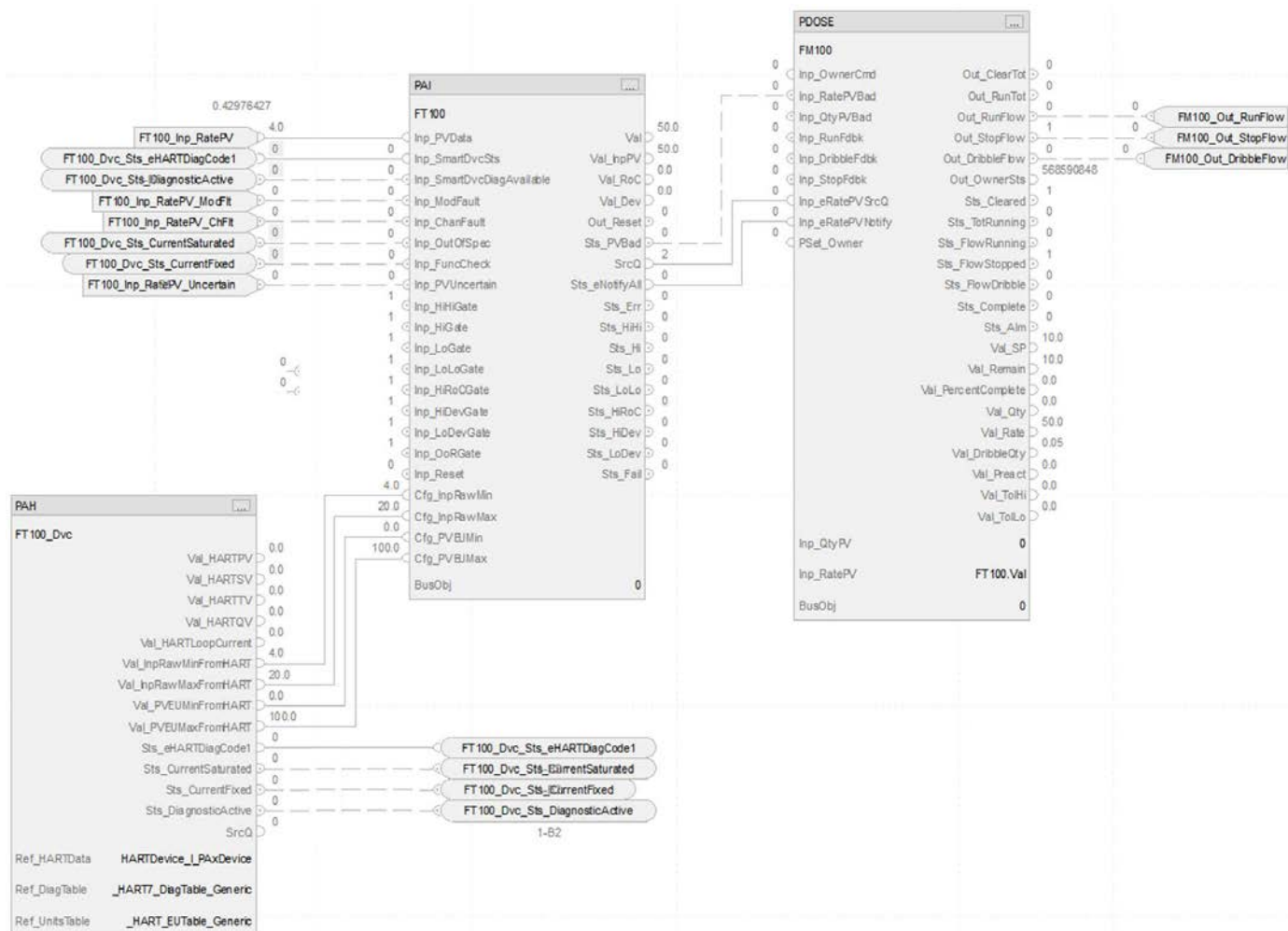
Parameter	Description
FM101_Out_RunFlow	1 = Deliver at full (fast) flow
FM101_Out_StopFlow	1 = Stop delivery equipment
FM101_Out_DribbleFlow	1 = Deliver at dribble (slow) flow

## PDOSE Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_DOSING	Instance of data structure (backing tag) required for proper operation of instruction
Inp_QtyPV	REAL	Quantity from flowmeter (EU or pulse count). Input is disabled if Sts_CalcQty is either of the following: <ul style="list-style-type: none"> <li>• 1=integrate Inp_RatePV to get quantity</li> <li>• 0=use Inp_QtyPV</li> </ul>
Inp_RatePV	REAL	Flow rate from flowmeter (EU/Time, see Cfg_RateTime). Input is disabled if Sts_CalcRate is either of the following: <ul style="list-style-type: none"> <li>• 1=differentiate Inp_QtyPV to get rate</li> <li>• 0=use Inp_RatePV</li> </ul>
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

For a flowmeter, you usually use the rate input. If the flowmeter provides a rate and a totalized quantity, use both the rate and quantity parameters. When both parameters are connected, the instruction uses the meter's quantity and does not need to calculate a quantity from the rate. Connect the clear totalizer output back to the meter to reset the totalizer as needed.

## CS\_PDSEFM\_HART Sheet

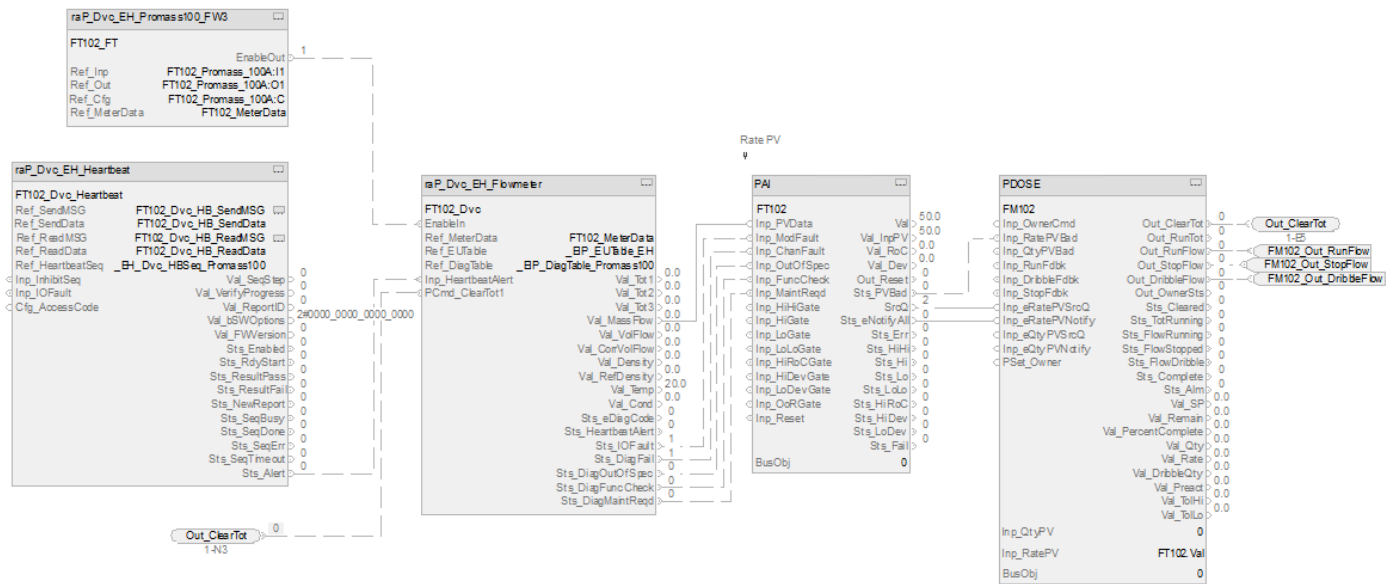


The CS\_PDOSEFM\_HART control strategy operates the same as the CS\_PDOESFM control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute FM100 for the PV data instance of XT100
- Substitute FT100 for the remaining instances of XT100

For more information, see [HART Integration on page 61](#).

## CS\_PDSEFM\_EtherNetIP Sheet

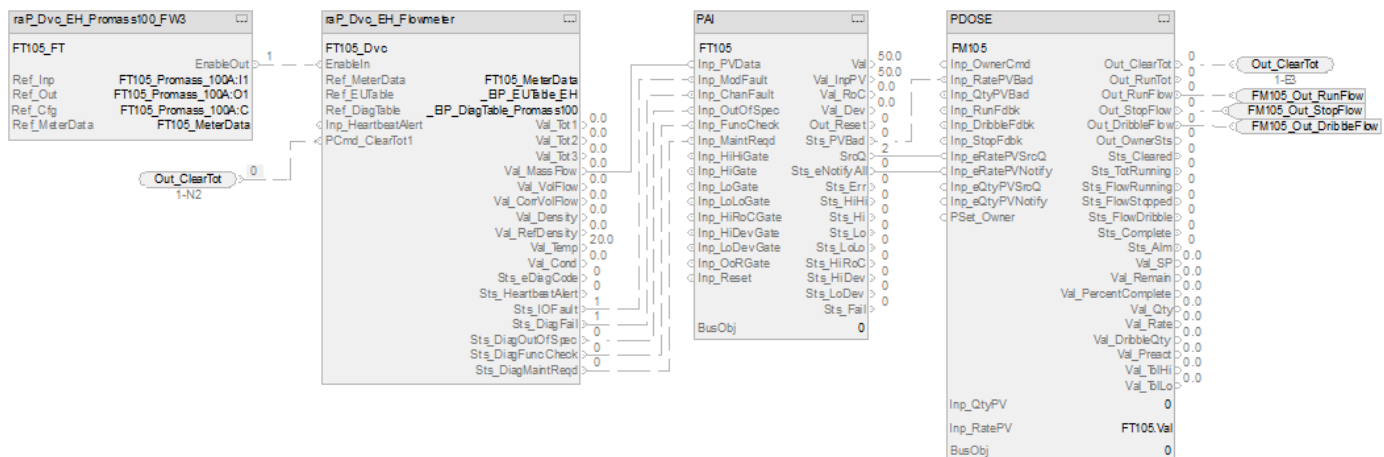


The CS\_PDSEFM\_EtherNetIP control strategy operates the same as the CS\_PDSEFM control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute FM102 for the PV data instance of XT100
- Substitute FT102 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PDOSEFM\_EtherNetIP\_NoHB Sheet

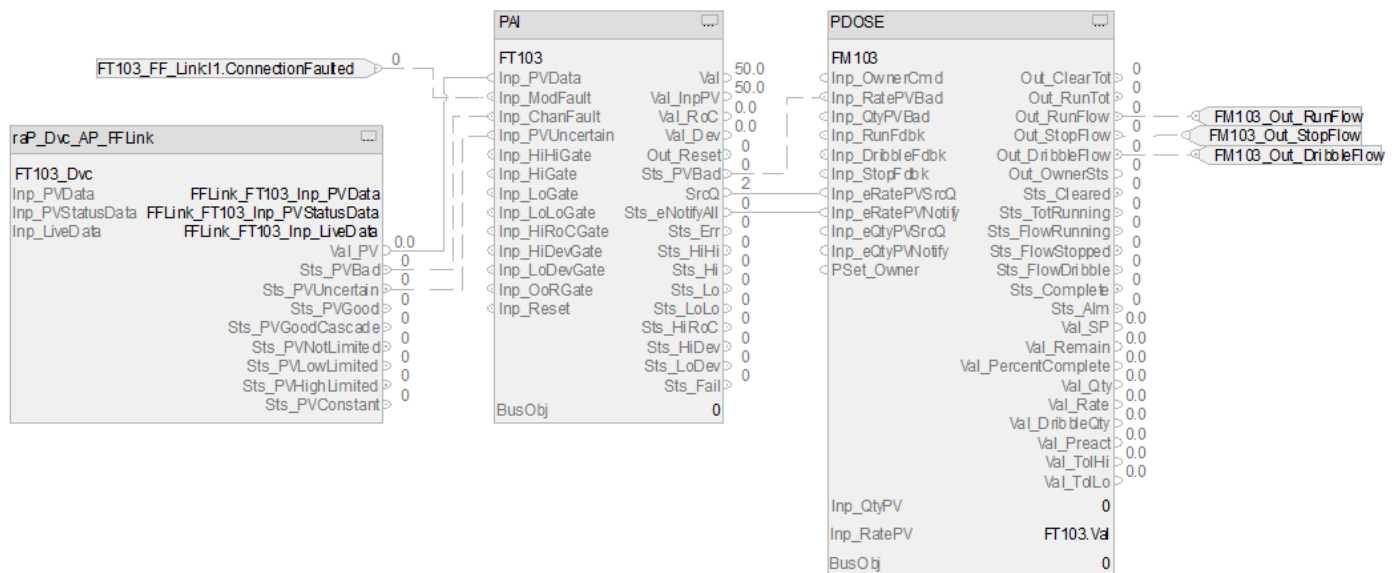


The CS\_PDOSEFM\_EtherNetIP\_NoHB control strategy operates the same as the CS\_PDOSEFM control strategy but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute FM105 for the PV data instance of XT100
- Substitute FT105 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PDOSEFM\_FF Sheet



The CS\_PDOSEFM\_FF control strategy operates the same as the CS\_PDOSEFM control strategy but relies on FOUNDATION Fieldbus input data.

- For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute FM103 for the PV data instance of XT100
- Substitute FT103 for the remaining instances of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).



## ACM Considerations for PDOSE

Configure these parameters first because they affect the visibility of the remaining parameters in the PAI object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

### ACM-Based Parameters for a PDOSE Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_Type	always	Important: Select this parameter first as the option affects the remaining parameters. Define the type: flow meter (FM) or weigh scale (WS)
Cfg_HasQtyPV	Cfg_Type=FM	Set to require a quantity PV signal.
Cfg_HasRatePV	Cfg_Type=WS	Set to require a rate PV signal.
PAI_Type	always	Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI (Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_Has_OutClearTot	IO_Signal_Type=EH_EtherNetIP	Set if OutClearTot is the input to the PCmdClearTot1 parameter
Use_00AP	Has_00AP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_00AP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Bus</b>		
Bus_Instance	Has_00AP=True (controller parameter)	Link to a bus array instance. This should be unique for each module
<b>01.01 - Options Rate</b>		
Cfg_UseHARTDigitalData_Rate	Cfg_Type=FM IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling_Rate	Cfg_Type=FM IO_Signal_Type=HART	Set to connect HART scaling from PAH object
Hart_Type_Rate	Cfg_Type=FM IO_Signal_Type=HART	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_Hart_Device_Rate	Cfg_Type=FM IO_Signal_Type=HART Cfg_UseHARTDigitalData_Qty=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
<b>01.10 - Options Qty</b>		
Cfg_UseHARTDigitalData_Qty	Cfg_Type=FM Cfg_HasQtyPV=True IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling_Qty	Cfg_Type=FM Cfg_HasQtyPV=True IO_Signal_Type=HART	Set to connect HART scaling from PAH object
Hart_Type_Qty	Cfg_Type=FM Cfg_HasQtyPV=True IO_Signal_Type=HART	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_Hart_Device_Qty	Cfg_Type=FM Cfg_HasQtyPV=True Cfg_UseHARTDigitalData_Qty=False IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
<b>02.01 - Device Configuration Feedback</b>		
Cfg_HasEqpFdbk	always	Set if the device provides run (dribble if used) and stop feedback

Parameter	Visible When	Details
<b>03 - IO Configuration</b> Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Inp_QtyPV	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel) IO_Signal_Type=None	Link to the PV quantity input reference
	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel) IO_Signal_Type=HART	
Inp_RatePV	Cfg_Type=FM PAI_Type=PAI(Single_channel) IO_Signal_Type=None	Link to the PV rate input reference
	Cfg_Type=FM PAI_Type=PAI(Single_channel) IO_Signal_Type=HART	
Inp_RunFdbk	Cfg_HasEqpFdbk=True	Link to the device running input reference
Inp_StopFdbk	Cfg_HasEqpFdbk=True	Link to the device stopped input reference
Inp_DribbleFdbk	Cfg_HasEqpFdbk=True	Link to the device dribble input reference
Out_RunFlow	always	Link to the run flow output reference
Out_StopFlow	always	Link to the stop flow output reference
Out_DribbleFlow	always	Link to the dribble flow output reference
Ref_EtherNetIPModule	IO_Signal_Type=EH_EtherNetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects

**03.00 - IO Configuration Rate**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

PAI_Rate_RefTag	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_Type=FM PAI_Type=External PAI(Single_channel)	
PAID_Rate_RefTag	Cfg_Type=FM PAI_Type=PAID(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Rate_RefTag	Cfg_Type=FM PAI_Type=PAIM(Multi-channel)	Link to the analog input (multi-channel) reference
Ref_FF_Module_Rate	Cfg_Type=FM IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module_Rate	Cfg_Type=FM IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.00.20 - Ref PAI Rate Alarm Configuration**

Ref_HiHiGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference



Parameter	Visible When	Details
Ref_LoDevGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate_Rate	Cfg_Type=FM PAI_Type=PAI(Single_channel)	Link to the gate reference
<b>03.01 - IO Configuration Qty</b>		
PAI_Qty_RefTag	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=External PAI(Single_channel)	
PAID_Qty_RefTag	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAID(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Qty_RefTag	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi-channel) reference
Ref_FF_Module	Cfg_Type=FM Cfg_HasQtyPV=True IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	Cfg_Type=FM Cfg_HasQtyPV=True IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03.01.20 - Ref PAI Qty Alarm Configuration</b>		
Ref_OoRGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiHiGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	Cfg_Type=FM Cfg_HasQtyPV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
<b>04 - Alarm Configuration</b>		
Cfg_HasOverTolAlm	always	If Cfg_HasOverTolAlm=True, ACM displays section 4.04 - Over Tolerance Alarm with additional parameters
Cfg_HasUnderTolAlm	always	If Cfg_HasUnderTolAlm=True, ACM displays section 4.05 - Under Tolerance Alarm with additional parameters
Cfg_HasZeroFaultAlm	always	If Cfg_HasZeroFaultAlm=True, ACM displays section 4.06 - Zero Fault Alarm with additional parameters



Parameter	Visible When	Details
Cfg_HasEqpFaultAlm	always	If Cfg_HasEqpFaultAlm=True, ACM displays section 4.07 - Equipment Fault Alarm with additional parameters
Cfg_HasHiFlowRateAlm	always	If Cfg_HasHiFlowRateAlm=True, ACM displays section 4.08 - Hi Flow Rate Alarm with additional parameters
Cfg_HasLoFlowRateAlm	always	If Cfg_HasLoFlowRateAlm=True, ACM displays section 4.09 - Lo Flow Rate Alarm with additional parameters

## Additional Sub-Object for a PDOSE Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**

## Process Dosing Weigh Scale (PDOSEWS) Control Strategy

Use the PDOSEWS control strategy to control an ingredient addition that uses a weigh scale to measure the quantity of ingredient added. The weigh scale can be on the receiving vessel, indicating a gain in weight, or on the sourcing vessel, indicating a loss in weight. The weigh scale can be connected using an analog input, device network, or other connection.

The following PDOSEWS control strategies are available as routines in the process library:

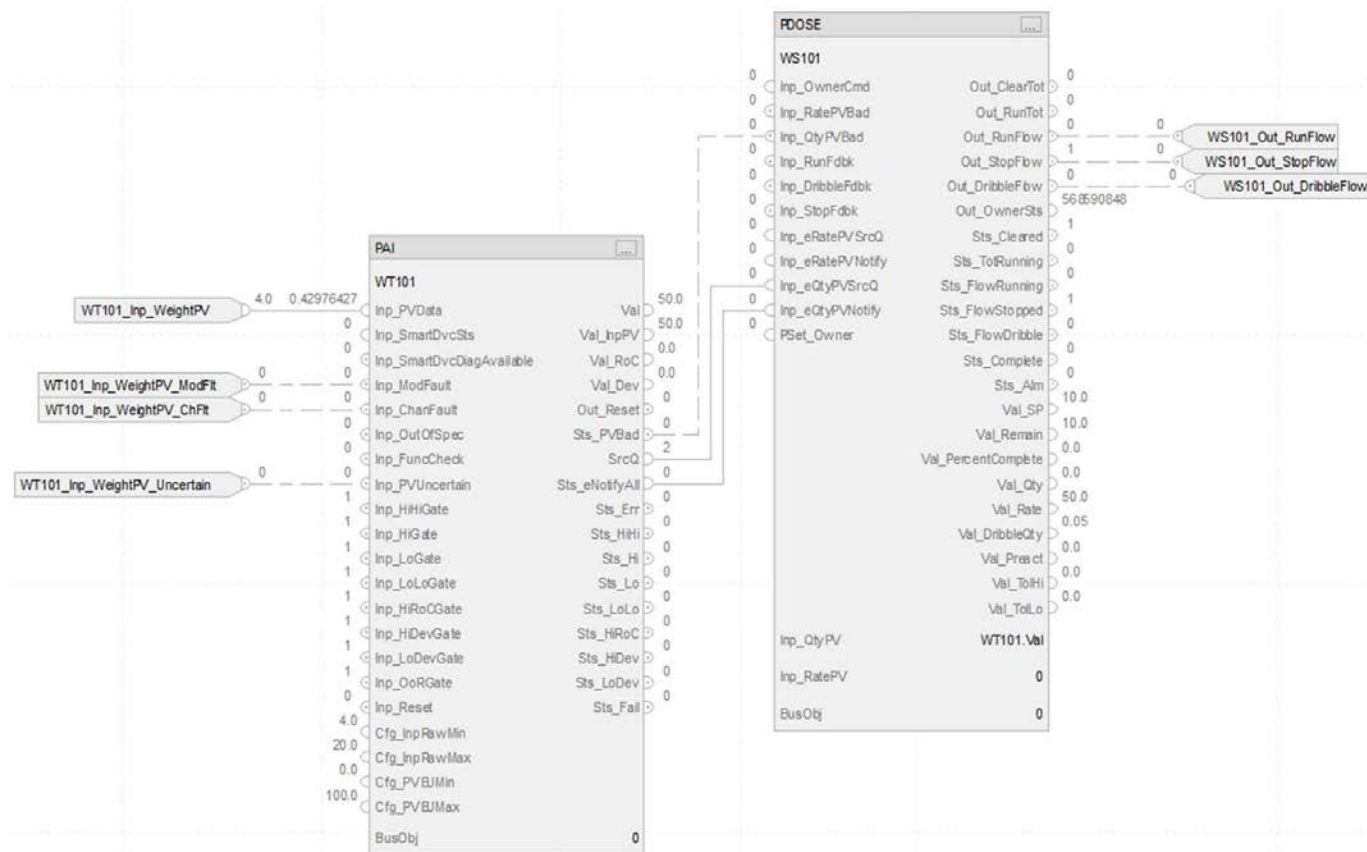
- CS\_PDOSEWS
- CS\_PDOSEWS\_HART
- CS\_PDOSEWS\_EtherNetIP
- CS\_PDOSEWS\_EtherNetIP\_NoHB
- CS\_PDOSEWS\_FF
- CS\_PDOSEWS\_PA

Import the appropriate control strategy as a **routine** in your controller project.

The PDOSEWS control strategy contains one Function Block sheet:

Sheet	Description
CS_PDOSEWS	Process Dosing Weigh Scale instruction
CS_PDOSEWS_HART	Process Dosing Weigh Scale instruction with HART input
CS_PDOSEWS_EtherNetIP	Process Dosing Weigh Scale instruction with EtherNetIP input
CS_PDOSEWS_EtherNetIP_NoHB	Process Dosing Weigh Scale instruction with No HB EtherNetIP input
CS_PDOSEWS_FF	Process Dosing Weigh Scale instruction with FOUNDATION Fieldbus input
CS_PDOSEWS_PA	Process Dosing Weigh Scale instruction with PA input

## CS\_PDOSEWS Sheet



### PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

- Substitute WS101 for the PV data instance of XT101
- Substitute WT101 for the remaining instances of XT101

## PAI Outputs to PDOSE Inputs

Parameter	Description
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad
SrcQ	Source and quality of primary value or status: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	All alarm status enumerated values including related objects: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged, or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

## PDOSE Output References

Parameter	Description
WS101_Out_RunFlow	1 = Deliver at full (fast) flow
WS101_Out_StopFlow	1 = Stop delivery equipment
WS101_Out_DribbleFlow	1 = Deliver at dribble (slow) flow

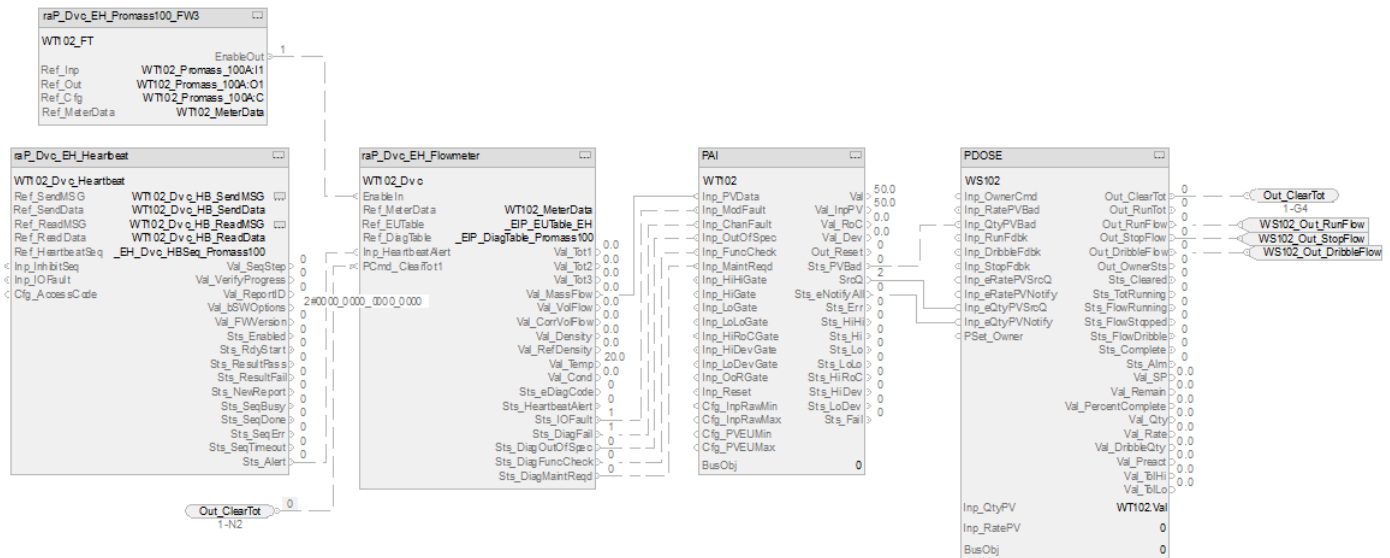
## PDOSE Configuration Considerations

Operand	Type	Description
PlantPax® control	P_DOSING	Instance of data structure (backing tag) required for proper operation of instruction
Inp_QtyPV	REAL	Quantity per time from weigh scale (EU or pulse count). Input is disabled if Sts_CalcQty is either of the following: • 1=integrate Inp_RatePV to get quantity • 0=use Inp_QtyPV
Inp_RatePV	REAL	Rate of material that is added or removed from weigh scale (EU/Time, see Cfg_RateTime). Input is disabled if Sts_CalcRate is either of the following: • 1=differentiate Inp_QtyPV to get rate • 0=use Inp_RatePV
BusObj	BUS_OBJ	Bus component for organization control • 0 if not using organization • Bus[x].Obj when using organization See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

A weigh scale usually only provides a quantity signal (how much material is in the device). Connect the PDOSE instruction to the quantity parameter; the instruction calculates the rate by measuring how much the quantity changes (differentiate with respect to time).



## CS\_PDOSEWS\_EtherNetIP Sheet

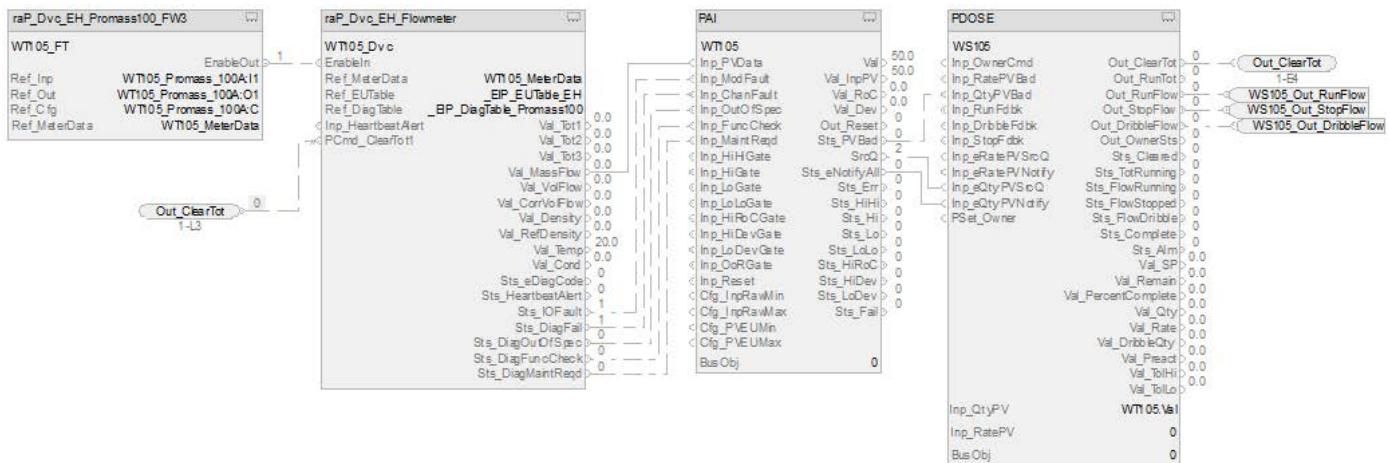


The CS\_PDOSEWS\_EtherNetIP control strategy operates the same as the CS\_PDOSEWS control strategy but relies on EtherNet/IP™ input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute for WT102 for XT100.

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PDOSEWS\_EtherNetIP\_NoHB Sheet

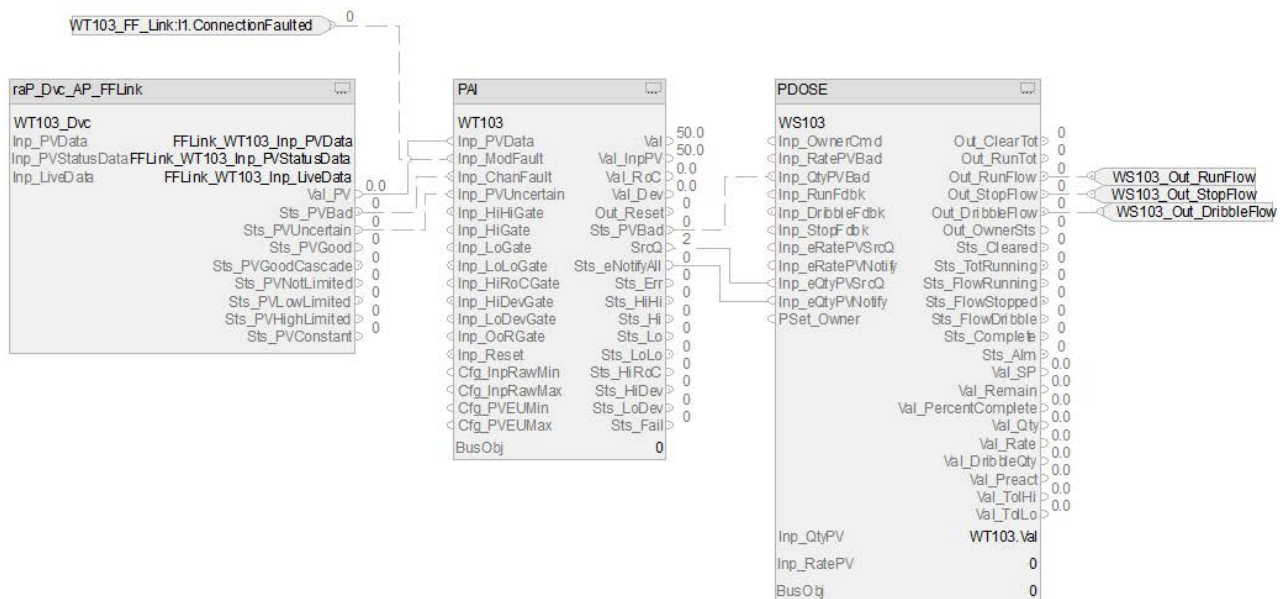


The CS\_PDOSEWS\_EtherNetIP\_NoHB control strategy operates the same as the CS\_PDOSEWS control strategy but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute for WT105 for XT100.

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PDOSEWS\_FF Sheet

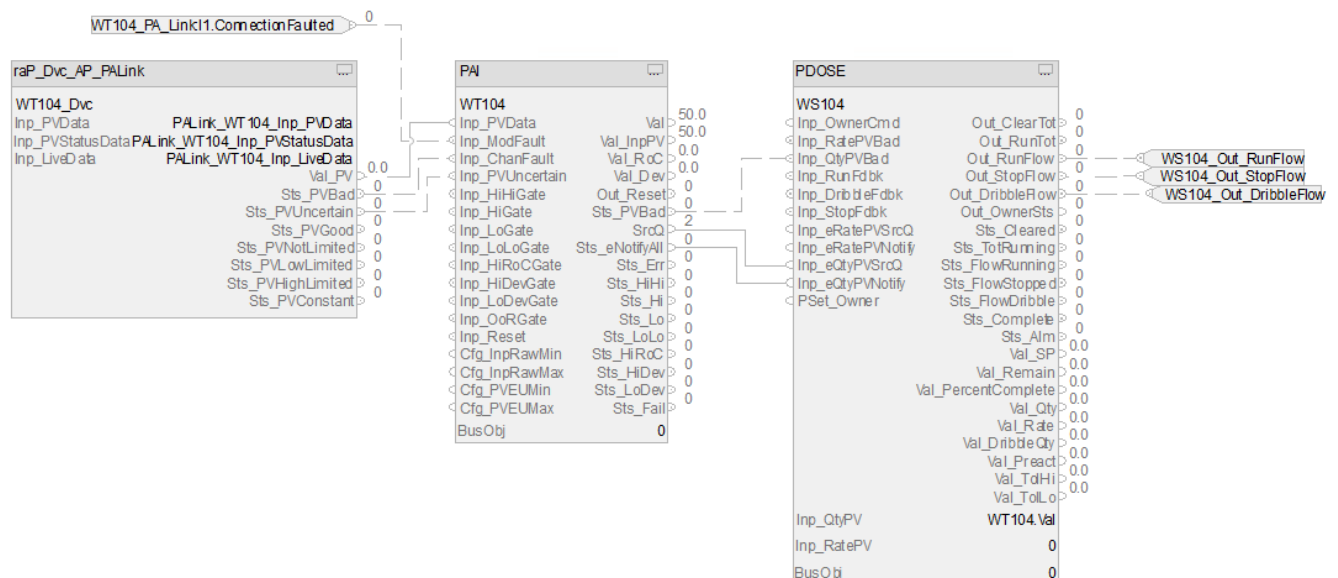


The CS\_PDOSEWS\_FF control strategy operates the same as the CS\_PDOSEWS control strategy but relies on FOUNDATION Fieldbus input data.

- For information on FOUNDATION Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute for WT103 for XT100.

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PDOSEWS\_PA Sheet



The CS\_PDOSEWS\_PA control strategy operates the same as the CS\_PDOSEWS control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute for WT104 for XT100.

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).



## ACM Considerations for PDOSE

Configure these parameters first because they affect the visibility of the remaining parameters in the PAI object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

### ACM-Based Parameters for a PDOSE Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_Type	always	Important: Select this parameter first as the option affects the remaining parameters. Define the type: flow meter (FM) or weigh scale (WS)
Cfg_HasQtyPV	Cfg_Type=FM	Set to require a quantity PV signal.
Cfg_HasRatePV	Cfg_Type=WS	Set to require a rate PV signal.
PAI_Type	always	Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI (Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_Has_OutClearTot	IO_Signal_Type=EH_EtherNetIP	Set if OutClearTot is the input to the PCmdClearTot1 parameter
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Bus</b>		
Bus_Instance	Has_OOAP=True (controller parameter)	Link to a bus array instance. This should be unique for each module
<b>01.01 - Options Rate</b>		
Cfg_UseHARTDigitalData_Rate	Cfg_Type=WS Cfg_HasRatePV=True IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling_Rate	Cfg_Type=WS Cfg_HasRatePV=True IO_Signal_Type=HART	Set to connect HART scaling from PAH object
Hart_Type_Rate	Cfg_Type=WS Cfg_HasRatePV=True IO_Signal_Type=HART	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_Hart_Device_Rate	Cfg_Type=WS Cfg_HasRatePV=True Cfg_UseHARTDigitalData_Qty=False IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
<b>01.10 - Options Qty</b>		
Cfg_UseHARTDigitalData_Qty	Cfg_Type=WS IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling_Qty	Cfg_Type=WS IO_Signal_Type=HART	Set to connect HART scaling from PAH object
Hart_Type_Qty	Cfg_Type=WS IO_Signal_Type=HART	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_Hart_Device_Qty	Cfg_Type=WS IO_Signal_Type=HART Cfg_UseHARTDigitalData_Qty=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
<b>02.01 - Device Configuration Feedback</b>		
Cfg_HasEqpFdbk	always	Set if the device provides run (dribble if used) and stop feedback

Parameter	Visible When	Details
<b>03 - IO Configuration</b> Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Inp_QtyPV	Cfg_Type=WS PAI_Type=PAI(Single_channel) IO_Signal_Type=None	Link to the PV quantity input reference
	Cfg_Type=WS PAI_Type=PAI(Single_channel) IO_Signal_Type=HART	
Inp_RatePV	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel) IO_Signal_Type=None	Link to the PV rate input reference
	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel) IO_Signal_Type=HART	
Inp_RunFdbk	Cfg_HasEqpFdbk=True	Link to the device running input reference
Inp_StopFdbk	Cfg_HasEqpFdbk=True	Link to the device stopped input reference
Inp_DribbleFdbk	Cfg_HasEqpFdbk=True	Link to the device dribble input reference
Out_RunFlow	always	Link to the run flow output reference
Out_StopFlow	always	Link to the stop flow output reference
Out_DribbleFlow	always	Link to the dribble flow output reference
Ref_EtherNetIPModule	IO_Signal_Type=EH_EtherNetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects

**03.00 - IO Configuration Rate**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

PAI_Rate_RefTag	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=External PAI(Single_channel)	
PAID_Rate_RefTag	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAID(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Rate_RefTag	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi-channel) reference
Ref_FF_Module_Rate	Cfg_Type=WS Cfg_HasRatePV=True IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module_Rate	Cfg_Type=WS Cfg_HasRatePV=True IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.00.20 - Ref PAI Rate Alarm Configuration**

Ref_HiHiGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_LoLoGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate_Rate	Cfg_Type=WS Cfg_HasRatePV=True PAI_Type=PAI(Single_channel)	Link to the gate reference
<b>03.01 - IO Configuration Qty</b>		
PAI_Qty_RefTag	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_Type=WS PAI_Type=External PAI(Single_channel)	
PAID_Qty_RefTag	Cfg_Type=WS PAI_Type=PAID(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Qty_RefTag	Cfg_Type=WS PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi-channel) reference
Ref_FF_Module	Cfg_Type=WS IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	Cfg_Type=WS IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03.01.20 - Ref PAI Qty Alarm Configuration</b>		
Ref_OoRGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiHiGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	Cfg_Type=WS PAI_Type=PAI(Single_channel)	Link to the gate reference
<b>04 - Alarm Configuration</b>		
Cfg_HasOverTolAlm	always	If Cfg_HasOverTolAlm=True, ACM displays section 4.04 - Over Tolerance Alarm with additional parameters
Cfg_HasUnderTolAlm	always	If Cfg_HasUnderTolAlm=True, ACM displays section 4.05 - Under Tolerance Alarm with additional parameters
Cfg_HasZeroFaultAlm	always	If Cfg_HasZeroFaultAlm=True, ACM displays section 4.06 - Zero Fault Alarm with additional parameters

Parameter	Visible When	Details
Cfg_HasEqpFaultAlm	always	If Cfg_HasEqpFaultAlm=True, ACM displays section 4.07 - Equipment Fault Alarm with additional parameters
Cfg_HasHiFlowRateAlm	always	If Cfg_HasHiFlowRateAlm=True, ACM displays section 4.08 - Hi Flow Rate Alarm with additional parameters
Cfg_HasLoFlowRateAlm	always	If Cfg_HasLoFlowRateAlm=True, ACM displays section 4.09 - Lo Flow Rate Alarm with additional parameters

**Additional Sub-Object for a PDOSE Instance**

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Process Analog Fanout (PFO) Control Strategies

Use a PFO control strategy to send (fanout) one primary analog output signal to up to 8 secondary users or devices. Each secondary output has configurable gain, offset, and clamping limits.

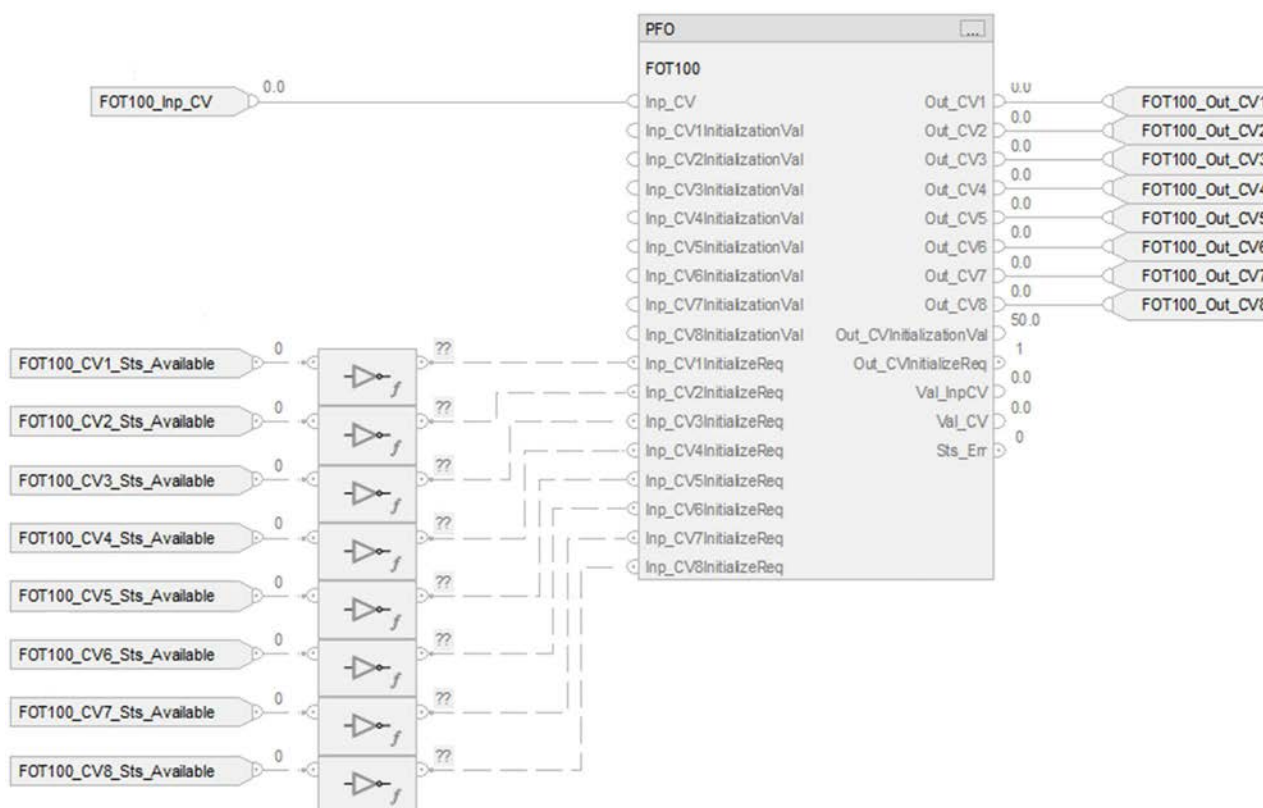
The PFO instruction receives an input CV (controlled variable) from a primary PID loop or analog output and applies rate-of-change limiting to the input signal. This control strategy is a base component of a PPID Split Range control strategy.

The following CS\_PFO control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PFO control strategy contains the CS\_PFO Function Block sheet.

### CS\_PFO Sheet



## PFO Input References

Parameter	Description
FOT100_Inp_CV	Input CV from upstream block's output (engineering units)
FOT100_CV1_Sts_Available	Initialize request from downstream block #1 = set Out_CV1 to Inp_CV1InitializationVal.
FOT100_CV2_Sts_Available	Initialize request from downstream block #2 = set Out_CV2 to Inp_CV2InitializationVal.
FOT100_CV3_Sts_Available	Initialize request from downstream block #3 = set Out_CV3 to Inp_CV3InitializationVal.
FOT100_CV4_Sts_Available	Initialize request from downstream block #4 = set Out_CV4 to Inp_CV4InitializationVal.
FOT100_CV5_Sts_Available	Initialize request from downstream block #5 = set Out_CV5 to Inp_CV5InitializationVal.
FOT100_CV6_Sts_Available	Initialize request from downstream block #6 = set Out_CV6 to Inp_CV6InitializationVal.
FOT100_CV7_Sts_Available	Initialize request from downstream block #7 = set Out_CV7 to Inp_CV7InitializationVal.
FOT100_CV8_Sts_Available	Initialize request from downstream block #8 = set Out_CV8 to Inp_CV8InitializationVal.

## PFO Output References

Parameter	Description
FOT100_Out_CV1	Output to downstream block #1 (out 1 engineering unit).
FOT100_Out_CV2	Output to downstream block #2 (out 2 engineering units).
FOT100_Out_CV3	Output to downstream block #3 (out 3 engineering units).
FOT100_Out_CV4	Output to downstream block #4 (out 4 engineering units).
FOT100_Out_CV5	Output to downstream block #5 (out 5 engineering units).
FOT100_Out_CV6	Output to downstream block #6 (out 6 engineering units).
FOT100_Out_CV7	Output to downstream block #7 (out 7 engineering units).
FOT100_Out_CV8	Output to downstream block #8 (out 8 engineering units).

## PFO Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_ANALOG_FANOUT	Instance of data structure (backing tag) required for proper operation of instruction

## ACM Considerations for PFO

Configure the parameters to send (fanout) one primary analog output signal to as many as 8 secondary users or devices.

## ACM-Based Parameters for a PFO Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
CV_Type	always	Select the type of control variable device CV_Type=AOUT or VSD
Cfg_UsedInPIDE	always	PAI can be standalone (False) or used as part of another PIDE control strategy (True)
PPID_RefTag	Cfg_UsedInPIDE=True	
<b>00.0x - Data - General - CVx Where x=2...8</b>		
Cfg_HasCVx	Number of PAOs in referenced PPID(Splitrange) >= x	Set if the output CV is connected
	Cfg_UsedInPIDE=False	
Cfg_HasNavCVx	Cfg_HasCVx=True	Set to enable the navigation buttons
<b>02.01 - Device Configuration</b>		
CVInitVal_RefTag	CV_Type=AOUT Cfg_UsedInPIDE=False	Link to output CV reference At least one output reference is required when CV_Type=AOUT
<b>02.0x - Device Configuration Where x=2...8</b>		
CVxInitVal_RefTag	Cfg_HasCVx=True CV_Type=AOUT Cfg_UsedInPIDE=False	Link to output CV reference
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
<b>Where x=2...8</b>		
Ref_InpCV	Cfg_UsedInPIDE=False	Link to the input CV reference
Aout1_RefTag	CV_Type=AOUT Cfg_UsedInPIDE=False	Link to the analog input reference At least one analog input reference is required when CV_Type=AOUT
Aoutx_RefTag	CV_HasCVx=AOUT CV_Type=AOUT Cfg_UsedInPIDE=False	Link to the analog input reference
VSD1_RefTag	CV_Type=VSD Cfg_UsedInPIDE=False	Link to the PVSD input reference At least one PVSD input reference is required when CV_Type=VSD
VSDx_RefTag	CV_HasCVx=VSD CV_Type=VSD Cfg_UsedInPIDE=False	Link to the PVSD input reference

## Additional Sub-Object for a PFO Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**



## Process High or Low Selector (PHLS) Control Strategies

The PHLS control strategy is a base component of the PPID Override control strategy. Use a PHLS control strategy to select the highest or the lowest of as many as six incoming controlled variables (CVs). The instruction sends the selected CV as its output and the output(s) of the 'unselected' PPID controller(s) are kept within  $K_p \cdot \text{Error}$  of the active PPID controller output to help ensure a quick response when another PPID's output becomes the limiting output.

For example, three PID controls feed a PHLS instruction that is configured to select the lowest of the three PID outputs as the speed reference for a drive. In normal operation, the discharge pressure PID has control, and the other PIDs track the output of the discharge pressure loop. When motor current exceeds its setpoint, or if suction pressure falls below its setpoint, the limit constrained PPID takes control to help prevent motor overcurrent or pump cavitation.

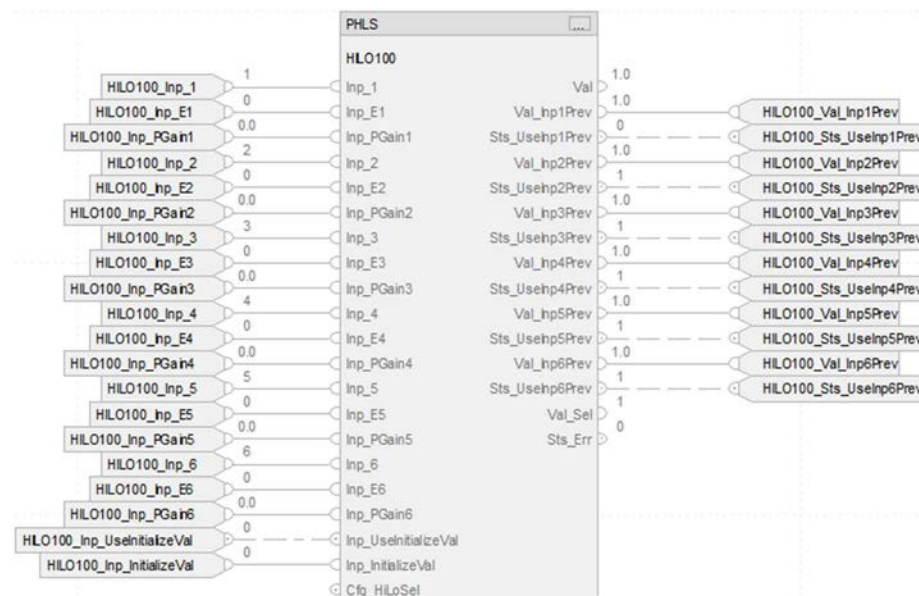
Scaling of the output of this block to CVEU can be done by a downstream PAO block. This block also supports initialization from a downstream block; the initialization is forwarded to upstream blocks.

The CS\_PHS control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PHLS control strategy contains the CS\_PHLs Function Block sheet.

# CS\_PHLS Sheet



## PHLS Input References

Parameter	Description
HIL0100_Inp_1 HIL0100_Inp_2 HIL0100_Inp_3 HIL0100_Inp_4 HIL0100_Inp_5 HIL0100_Inp_6	Each input# is a CV value.
HIL0100_Inp_E1 HIL0100_Inp_E2 HIL0100_Inp_E3 HIL0100_Inp_E4 HIL0100_Inp_E5 HIL0100_Inp_E6	Loop error from primary input# (optional, used for offset calculation).
HIL0100_Inp_PGain1 HIL0100_Inp_PGain2 HIL0100_Inp_PGain3 HIL0100_Inp_PGain4 HIL0100_Inp_PGain5 HIL0100_Inp_PGain6	Proportional gain from primary input# (optional, used for offset calculation).
HIL0100_Inp_UselInitializeVal	Use an initialization value from a downstream block.
HIL0100_Inp_InitializeVal	Initialization value from a downstream block.

## PHLS Output References

Parameter	Description
HIL0100_Val_Inp1Prev HIL0100_Val_Inp2Prev HIL0100_Val_Inp3Prev HIL0100_Val_Inp4Prev HIL0100_Val_Inp5Prev HIL0100_Val_Inp6Prev	Previous (Feedback) input value for primary input#
HIL0100_Sts_UselInp1Prev HIL0100_Sts_UselInp2Prev HIL0100_Sts_UselInp3Prev HIL0100_Sts_UselInp4Prev HIL0100_Sts_UselInp5Prev HIL0100_Sts_UselInp6Prev	Request for primary input# to use feedback Val_Inp#Prev

## PHLS Configuration Considerations

Operand	Type	Description
PHLS tag	P_HIGH_LOW_SELECT	Instance of data structure (backing tag) required for proper operation of instruction

## ACM Considerations for PHLS

Configure the parameters to select the highest or the lowest of as many as six incoming controlled variables (CVs).

## ACM-Based Parameters for a PHLS Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_UsedInPID	always	Set if this object is used as part of another PID control strategy.
<b>00.02.0x - Data - General - PPIDx</b> Where x=1...6		
Cfg_HasInpx	always	Set if input x is connected
PPID_RefTagx	Cfg_HasInpx=True Cfg_UsedInPID=False	Link to PID output reference
Input_RefTagx	Design_View (project parameter) > 7 Cfg_HasInpx=True Cfg_UsedInPID=True	Link to input reference for PID navigation
<b>03 - IO Configuration</b>		
Inp_CVInitReq	Cfg_UsedInPID=False	Use this request when reinitializing. Tag will be connected to the Inp_UselInitializeValue parameter
Inp_CVInitVal	Cfg_UsedInPID=False	Initialization value from downstream block. Valid = Any float. Tag will be connected to the Inp_InitializeValue parameter

## Additional Sub-Object for a PHLS Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**

## Process Lead Lag Standby (PLLS) Control Strategy

Use a PLLS control strategy to control of a parallel group of motors or drives. The number of devices to control depends on the demand on the system. The group can be configured to consist of as few as two or as many as 30 devices. The minimum demand can be set as low as 0, so that all devices are stopped at minimum demand. The maximum demand can be set as high as the number of devices in the group.

The PLLS control strategy has two options in the process library, with these routines:

Option	Routine	Description
<b>CS_PLLS_PMTR</b> CS_PLLS_PMTR Parameters and Local Tags MainRoutine B01_GRPMT100_Status_Mapping B02_GRPMT100 B03_GRPMT100_Command_Mapping Interlocks MT400 MT401 MT402 Permissive	B01_GRPMT100_Status_Mapping	Map motor status into the inputs of PLLS routine.
	B02_GRPMT100	Function Block control strategy routine for motors
	B03_GRPMT100_Command_Mapping	Map the commands of the PLLS out to the commands of the motor
	MT400	PMTR function block for a motor
	MT401	PMTR function block for a second motor
	MT402	PMTR function block for a third motor
<b>CS_PLLS_PVSD</b> CS_PLLS_PVSD Parameters and Local Tags MainRoutine B01_GRPVSD100_Status_Mapping B02_GRPVSD100 B03_GRPVSD100_Command_Mapping Interlocks MT800 MT801 MT802 Permissive	B01_GRPVSD100_Status_Mapping	Map drive status into the inputs of PLLS routine
	B02_GRPVSD100	Function Block control strategy routine for motors
	B03_GRPVSD100_Command_Mapping	Map the commands of the PLLS out to the commands of the drive
	MT800	Function block for a drive
	MT801	Function block for a second drive
	MT802	Function block for a third drive

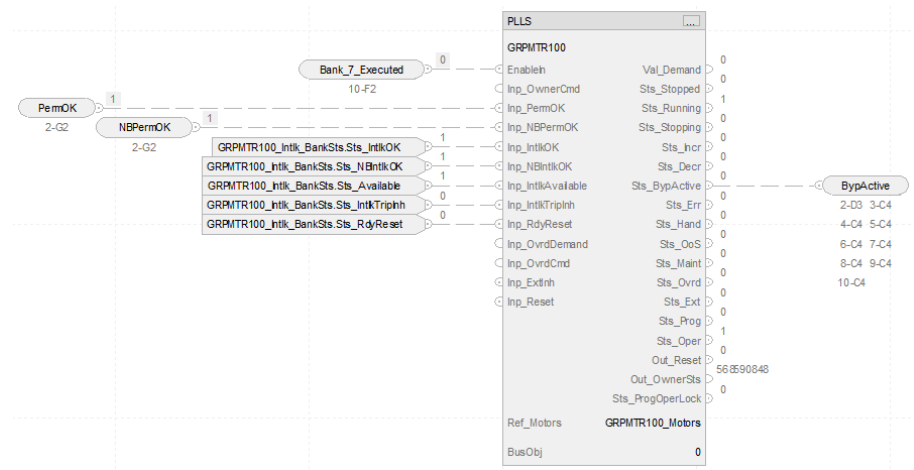
Import the PLLS as **program** in your controller project. The execution order of the routines is important for the proper operation of this control strategy.

- 
- IMPORTANT** For proper operation of this control strategy:
- In order for PLLS to align with the states of the motors on first scan, the motor / drive logic must be executed before this control strategy.
  - The routines in this strategy must be executed in the correct order (see the MainRoutine for your PLLS option):
1. GRPMTR100\_Status\_Mapping
  2. GRPMTR100
  3. GRPMTR100\_CommandMapping.
-

## CS\_PLLS\_PMTR Sheet

The B02GRPMTR100 routine contains these Function Block sheets:

Sheet	Description
CS_PLLS	Process Lead Lag Standby instruction for group drive control
Permissive	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PLLS instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.



## PLLS Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
PermOK	Input connection from permissive sheet 1 = On permissives OK, group can start
NBPermOK	Input connection from permissive sheet 1 = Non-bypassable On permissives OK, group can start
GRPMTR100_Intlk_BankSts_Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
GRPMTR100_Intlk_BankSts_Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
GRPMTR100_Intlk_BankSts_Sts_Available	Interlock bank status, 1 = Available
GRPMTR100_Intlk_BankSts_Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops group but does not raise trip alarm
GRPMTR100_Intlk_BankSts_Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

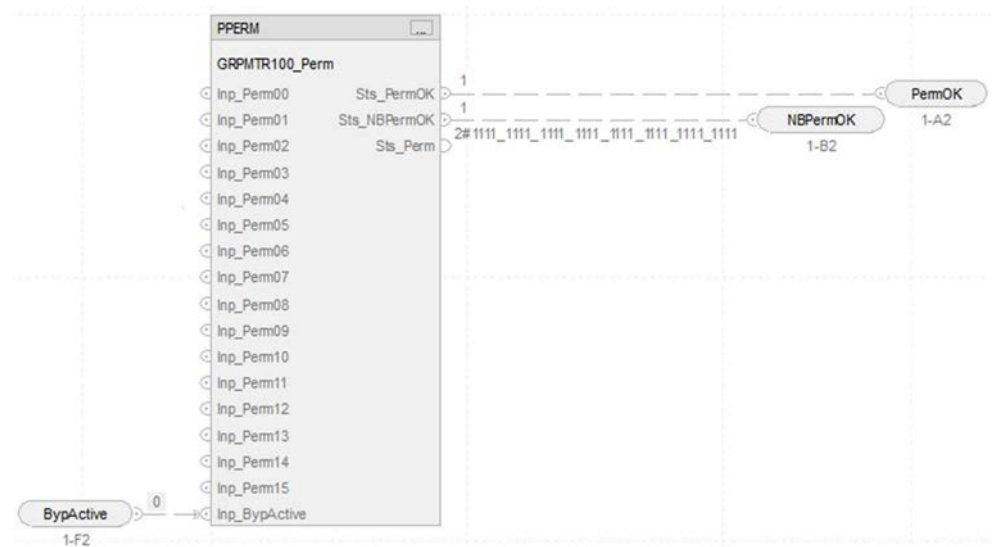
## PLLS Output References

Parameter	Description
ByActive	Output connection to permissive and interlock bank sheets

## PLLS Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_LEAD_LAG_STANDBY	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .
Ref_Motors	P_LEAD_LAG_STANDBY_MOTOR array	Reference to GRPMTR100_Motors array, used for motor status and motor command mapping.

## Permissive Sheet



## PPERM Input References

Parameter	Description
ByActive	Input connection from CS_PLLS_PMTR sheet

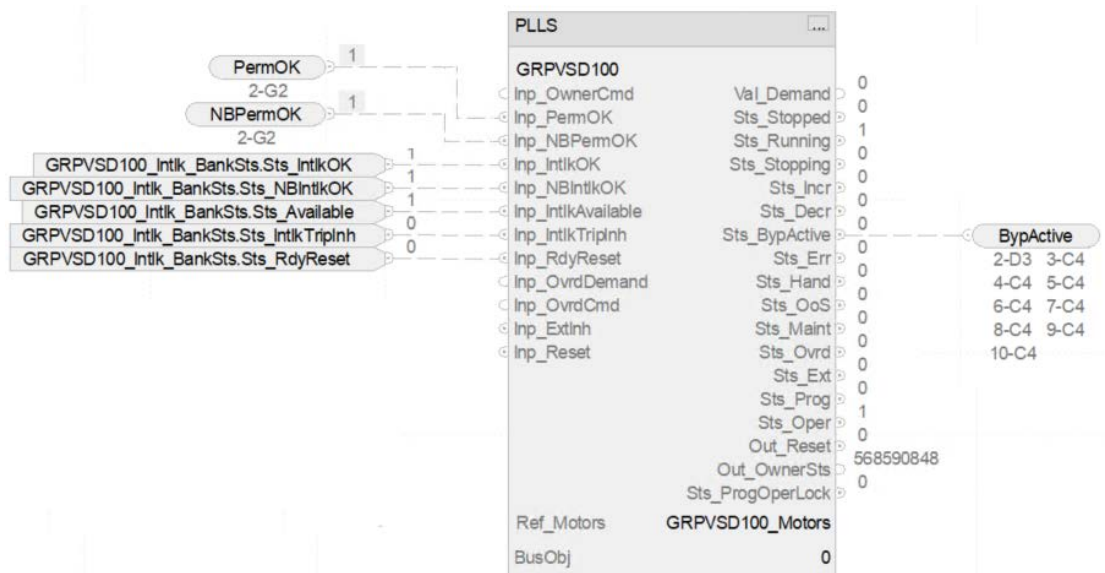
## PPERM Output References

Parameter	Description
PpermOK	Overall permissive status (1 = OK to start group)
NBPermOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to start group)

## CS\_PLLS\_PVSD Sheet

The B02GRPVSVD100 routine contains these Function Block sheets:

Sheet	Description
CS_PLLS	Process Lead Lag Standby instruction for group drive control
Permissive	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PLLS instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.



### PLLS Input References

Parameter	Description
PermOK	Input connection from permissive sheet 1 = On permissives OK, group can start
NBPermOK	Input connection from permissive sheet 1 = Non-bypassable On permissives OK, group can start
GRPVSVD100_Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
GRPVSVD100_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
GRPVSVD100_Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
GRPVSVD100_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops group but does not raise trip alarm
GRPVSVD100_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

### PLLS Output References

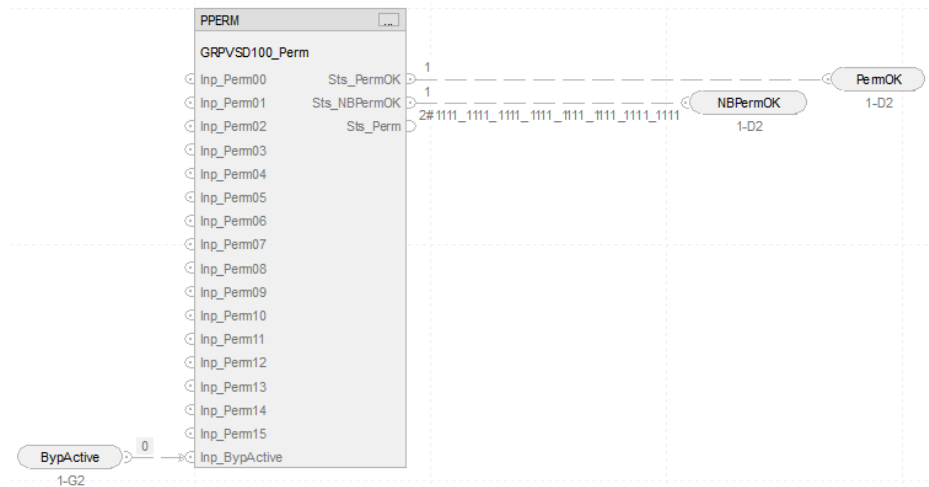
Parameter	Description
BypActive	Output connection to permissive and interlock bank sheets



## PLLS Configuration Considerations

Operand	Type	Description
PlantPax® control	P_LEAD_LAG_STANDBY	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .
Ref_Motors	P_LEAD_LAG_STANDBY_MOTOR array	Reference to GRPVSD100_Motors array, used for motor status and motor command mapping.

## Permissive Sheet



## PPERM Input References

Parameter	Description
BypActive	Input connection from CS_PLLS_PVSD sheet

## PPERM Output References

Parameter	Description
PermOK	Overall permissive status (1 = OK to start group)
NBPermOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to start group)

## ACM Considerations for PLLS

Configure the parameters to control a parallel group of motors or drives. Specify the Cfg\_RefDvcType parameter first as it affects the visibility of the remaining parameters.

## ACM-Based Parameters for a PLLS Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow a start command.
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Cfg_RefDvcType	always	Important: Select this parameter first as the option affects the remaining parameters.  Define the PLLS type: Motors or Drives
<b>04 - Alarm Configuration</b>		
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasCantStartAlm	always	If Cfg_HasCantStartAlm=True, ACM displays section 4.02 - Cant Start Alarm with additional parameters
Cfg_HasCantStopAlm	always	If Cfg_HasCantStopAlm=True, ACM displays section 4.03 - Cant Stop Alarm with additional parameters

## Additional Sub-Objects for a PLLS Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissives	Configure permissives to allow an output command See <a href="#">Permissives on page 50</a>
RefDevice_Mapping	Configure the command direction See <a href="#">RefDevice Mapping on page 251</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## RefDevice Mapping

On the RefDevice\_Mapping tab, add a sub-object to configure the command direction. This example shows Cfg\_RefDvcType=Motors.

Object Configuration Wizard

Name: GRPMTR100

Description: Description

Catalog Number: PLLS (4.2) - Published

Solution: (RA-LIB) Process 5 Task: Program:

Parameters Interlocks Permissive RefDevice\_Mapping Events

Name	Index	Reference_Motor	CommandDirection_Motor	AllowNav	SubObject Description
Devic...	0		Forward	<input type="checkbox"/>	

Parameter	Visible When	Details
Reference_Motor	always	Link to the motor for each sub-object
CommandDirection_Motor	Cfg_RefDvcType=Motors	Configure the direction of the linked motor in the Reference_Motor parameter on this tab
Reference_VSD	always	Configure the direction of the linked motor in the Reference_VSD parameter on this tab
CommandDirection_VSD	Cfg_RefDvcType=Drives	Link to the drive for each sub-object

## Notes:

## Process Motor (PMTR) Control Strategies

Use a PMTR control strategy to monitor and control a fixed single-speed, two-speed, or reversing motor using a full-voltage contactor or intelligent motor controller (soft starter). The motor can be run or jogged, including jogging reverse or jogging fast. The control strategy uses a Device Object to interface with the hardware motor controller.

Additional features provided in this control strategy, include a Runtime and Start Counter (to record the total run time and number of drive starts) and a Restart Inhibit (to limit the number of starts within a specified time period to protect the motor windings from overheating).

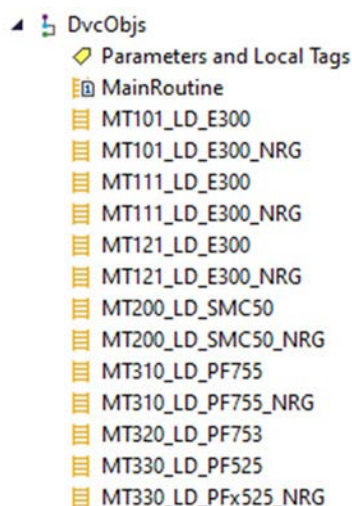
The following PMTR control strategies are available as routines in the process library:

Motor Controller Type	Control Strategy
E300™ Electronic Overload Relay	CS_PMTR_1S_E300 CS_PMTR_1S_E300_Energy CS_PMTR_2S_E300 CS_PMTR_2S_E300_Energy CS_PMTR_REV_E300 CS_PMTR_REV_E300_Energy
SMC™-50 Motor Controller	CS_PMTR_1S_SMC50 CS_PMTR_1S_SMC50_Energy
Basic	CS_PMTR_1S CS_PMTR_1S_Hand CS_PMTR_2S CS_PMTR_2S_Hand CS_PMTR_REV CS_PMTR_REV_Hand

Import the appropriate control strategy as a **routine** in your controller project.

Also, import the appropriate device object as a routine in your controller project. These objects are from the Power Device Library and must be downloaded separately from the PlantPax® Process Library.

Each ‘\_NRG’ object uses the Energy object to group energy parameters for the device. Use this object with the corresponding, energy-related control strategy.

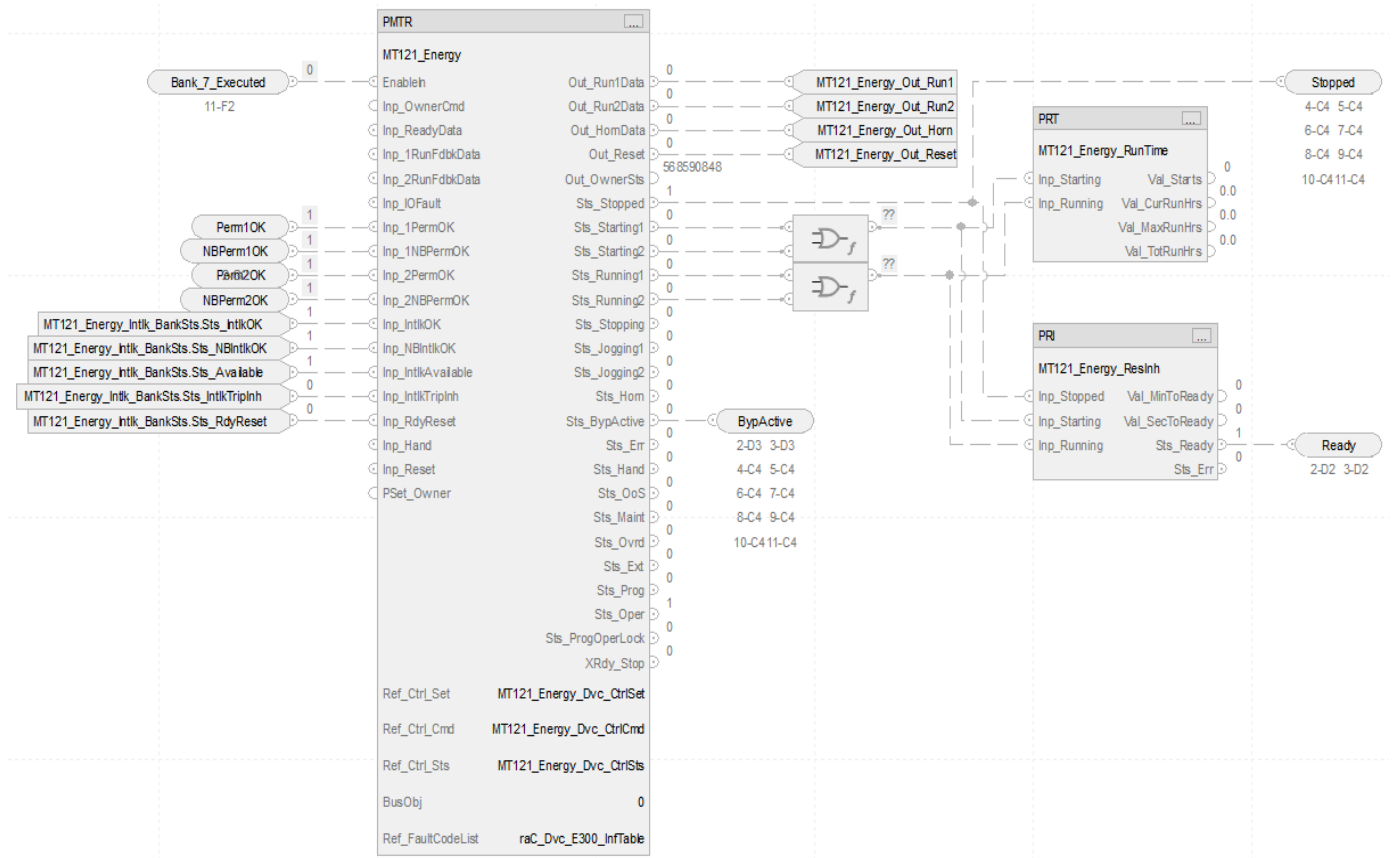


The PMTR control strategies contain these Function Block sheets:

Sheet	Description
CS_PMTR	Process Motor instruction
Permissives 1 Permissives 2	<p>Process Permissives instruction</p> <p>The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.</p> <p>The Permissives 2 sheet is only in the control strategies for two-speed and reversing motor controllers</p>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	<p>The PMTR instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable).</p> <p>There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default.</p> <p>Use the sheets and interlocks that you need and delete the remainder.</p>

In the input and output reference descriptions on each sheet, [device] = PMTR instance tag.

# CS\_PMTR Sheet



## PMTR Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
Perm1OK	Input connection from Permissives 1 sheet (single speed) 1 = On permissives OK, device can turn On
NBPerm1OK	Input connection from Permissives 1 sheet (single speed) 1 = Non-bypassable On permissives OK, device can turn On
Perm2OK	Input connection from Permissives 2 sheet (second speed or reverse) 1 = On permissives OK, device can turn On
NBPerm2OK	Input connection from Permissives 2 sheet (second speed or reverse) 1 = Non-bypassable On permissives OK, device can turn On
[device]_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
[device]_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
[device]_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
[device]_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
[device]_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

## PMTR Output References

Parameter	Description
[device]_Out_Run1	Single speed 1=Start/Run Motor Reverse or Fast 0=Stop Motor (for held starter type)
[device]_Out_Run2	Second speed or reverse 1=Start/Run Motor Reverse or Fast 0=Stop Motor (for held starter type)
[device]_Out_Horn	1 = Notification before commanded state change
[device]_Out_Reset	1 = Reset command has been received and accepted
ByActive	Output connection to permissives and interlock bank sheet
Ready	Output connection to the permissive sheet
Stopped	Output connection to interlock bank sheet

The Boolean OR performs a bitwise OR based on these PMTR outputs:

- Sts\_Stopped
- Sts\_Starting1
- Sts\_Starting2
- Sts\_Running1
- Sts\_Running2

The result feeds these instructions:

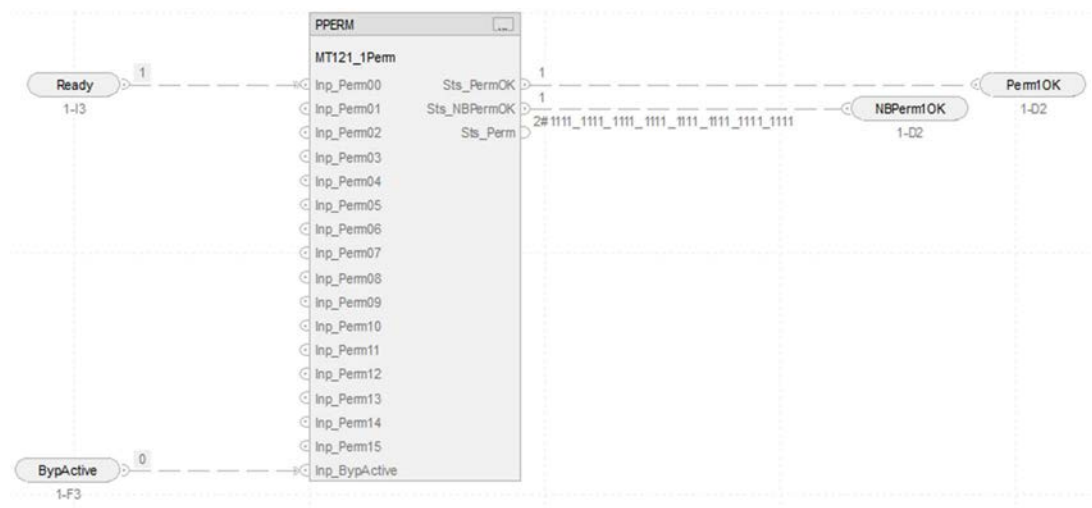
Instruction	Description
Process Run Time and Start Counter (PRT)	The PRT instruction records the total run time and number of instances the drive starts.
Process Restart Inhibit (PRI)	The PRI instruction helps prevent the drive from starting repeatedly within specified time periods. Continual starts or start attempts in a short period overheat the motor windings and damage the motor.

## PMTR Configuration Considerations

Operand	Type	Description
PlantPAx control	P_MOTOR_DISCRETE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .
Ref_Ctrl_Set	RAC_ITF_DVC_PWRDISCRETE_SET	Power Discrete Automation Device Object Settings Interface Preconfigured in the device object ladder routine
Ref_Ctrl_Cmd	RAC_ITF_DVC_PWRDISCRETE_CMD	Power Discrete Automation Device Object Command Interface Preconfigured in the device object ladder routine
Ref_Ctrl_Sts	RAC_ITF_DVC_PWRDISCRETE_STS	Power Discrete Automation Device Object Status Interface Preconfigured in the device object ladder routine
Ref_FaultCodeList	RAC_CODEDESCRIPTION[400]	Fault Code to Fault Description lookup table for the motor controller Preconfigured in the device object ladder routine



Permissive Sheet



PPERM Input References

Parameter	Description
Ready	Input connection from the CS_PMTR sheet
BypActive	Input connection from the CS_PMTR sheet

PPERM Output References

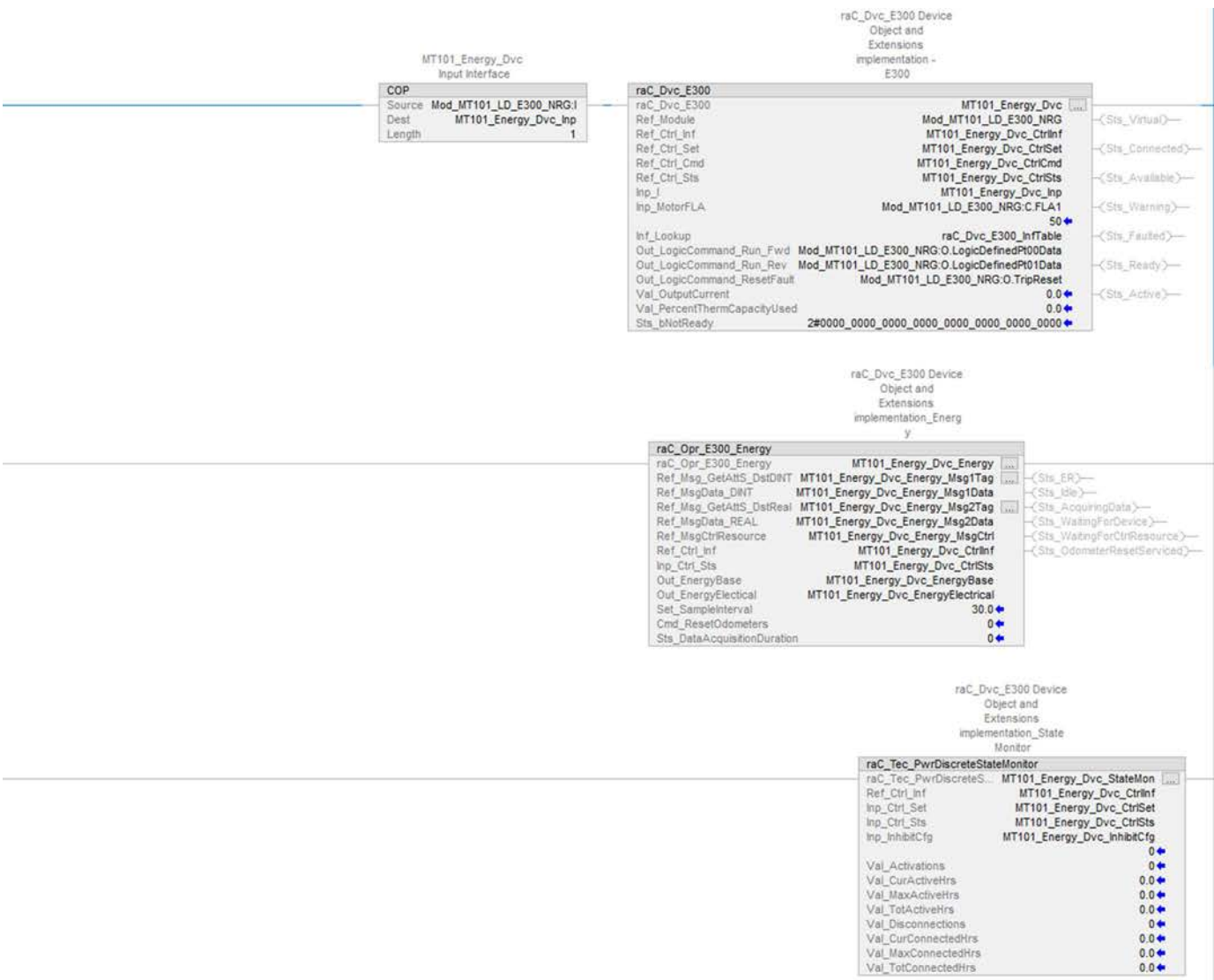
Parameter	Description
Perm1OK Perm2OK	Overall permissive status (1 = OK to energize)
NBPerm1OK NBPerm2OK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

# Motor Controller Device Objects

## Single Speed



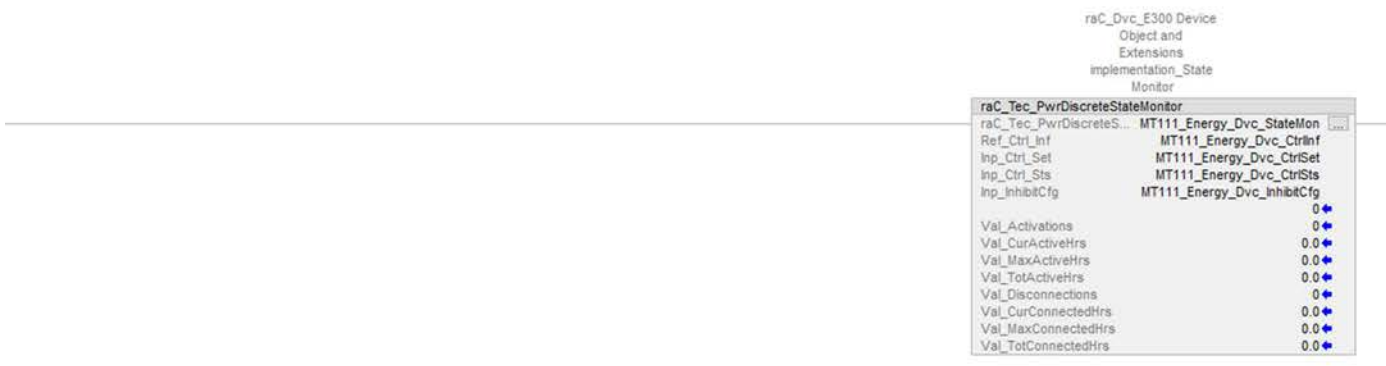
## Single Speed with Energy Parameters



## Two Speed



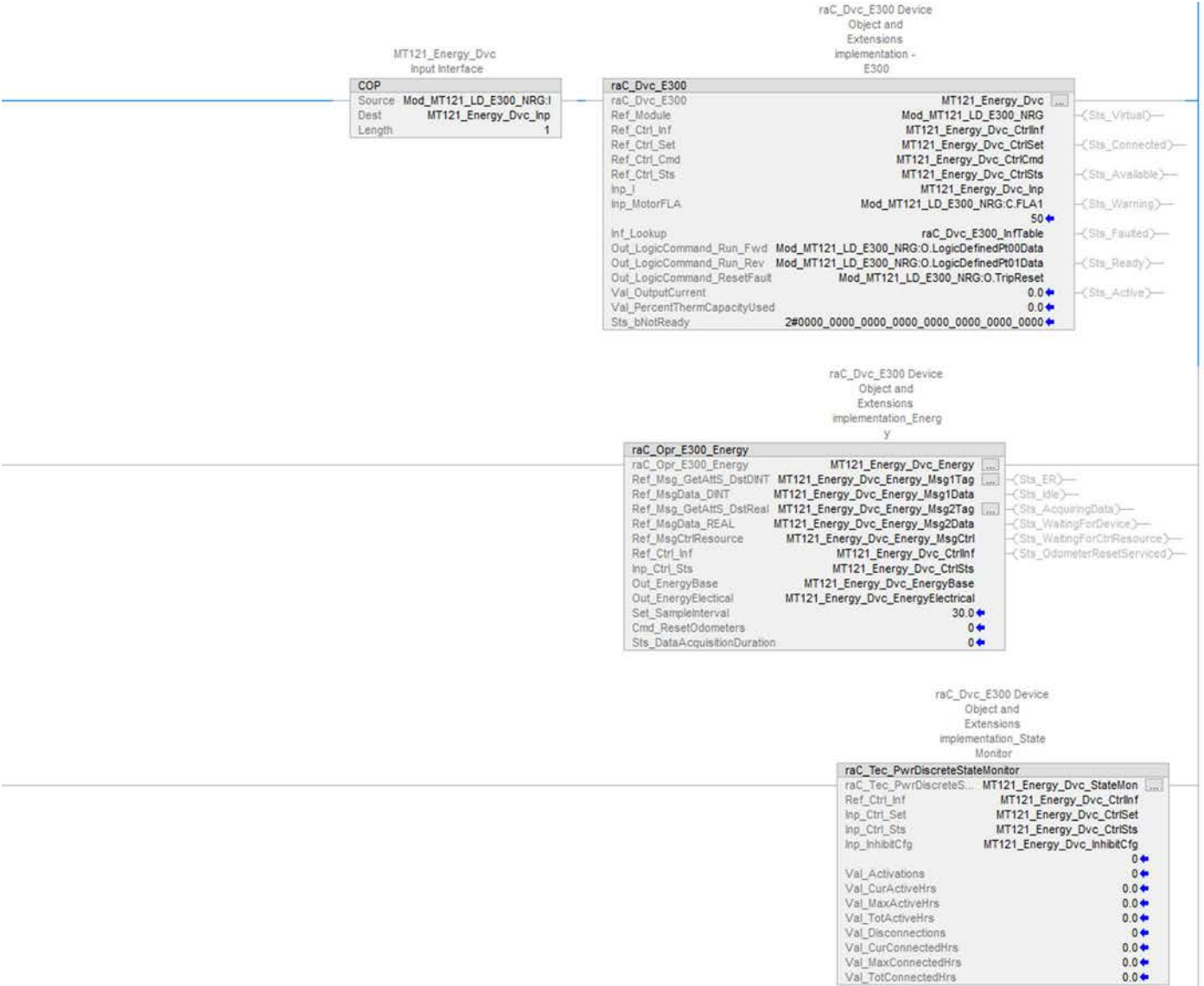
## Two Speed with Energy Parameters



Reversing



Reversing with Energy Parameters



## ACM Considerations for a PMTR Instance

Configure the motor type via the `Cfg_DvcType` parameter:

- Single speed non reversing
- Two speed (Fast/Slow)
- Reversing (Forward/Reverse)

## ACM-Based Parameters for a PMTR Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_DvcType	always	Important: Select this parameter first as the option affects the remaining parameters. Define the PMTR device type
Cfg_HasDvcObj	always	Set if device has connection to a device object (Ex. E300)
Cfg_DvcObj_Tag	Cfg_HasDvcObj=True	Link to the device object. See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_Has1PermObj	always	Set to create an instance of the PPERM instruction to allow an Out_Run1 command.
Cfg_Has2PermObj	Cfg_DvcType=Reversing	Set to create an instance of the PPERM instruction to allow an Out_Run2 command.
	Cfg_DvcType=TwoSpeed	
Cfg_HasResInhObj	always	Set to create an instance of the restart inhibit (PRI) instruction See <a href="#">Statistics Objects on page 57</a>
Cfg_HasRunTimeObj	always	Set to create an instance of a runtime (PRT) instruction See <a href="#">Statistics Objects on page 57</a>
Cfg_HasHand	always	Set to enable a hand switch input (Inp_Hand)
<b>02 - Device Configuration</b>		
Cfg_HasRunFdbk	always	Set if Motor has run feedback
Cfg_HasStart1	always	Set to enable Motor Forward/Slow output (Out_Run1)
Cfg_HasStart2	always	Set to enable Motor Reverse/Fast output (Out_Run2)
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Input_1RunFdbk	always	Link to the Motor Running Forward/Slow input reference
Input_2RunFdbk	Cfg_HasRunFdbk=True Cfg_DvcType=Reversing	Link to the Motor Running Reverse/Fast input reference
	Cfg_HasRunFdbk=True Cfg_DvcType=TwoSpeed	
Inp_Hand	Cfg_HasHand	Link to the Motor Hand Switch input reference
Out_Run1	always	Link to the Motor Run Forward/Slow output reference
Out_Run2	Cfg_HasRunFdbk=True Cfg_DvcType=Reversing	Link to the Motor Run Reverse/Fast output reference
	Cfg_HasRunFdbk=True Cfg_DvcType=TwoSpeed	
Out_Horn	always	Link to the Motor Horn output reference
Out_Reset	always	Link to the Motor Reset output reference
<b>04 - Alarm Configuration</b>		

Parameter	Visible When	Details
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.02 - I/O Fault Alarm with additional parameters
Cfg_HasFailToStartAlm	Cfg_HasRunFdbk=True	If Cfg_HasFailToStartAlm=True, ACM displays section 4.03 - Fail to Start Alarm with additional parameters
Cfg_HasFailToStopAlm	Cfg_HasRunFdbk=True	If Cfg_HasFailToStopAlm=True, ACM displays section 4.04 - Fail to Stop Alarm with additional parameters
Cfg_HasMotorFault	always	If Cfg_HasMotorFault=True, ACM displays section 4.05 - Motor Fault Alarm with additional parameters

## Additional Sub-Objects for a PMTR Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object?	Description?
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_1 Permissive_2	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Linked Libraries	Configure device libraries needed for your project See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>





## Process n-Position (PNPOS) Control Strategy

The Process n-Position Device (PNPOS) instruction controls a circular or linear discrete device. The device can have between 2 and 30 positions. The instruction lets you select each position with associated outputs and feedbacks.

For linear devices, the PNPOS instruction can be configured to:

- Return to Position 1 on every move, approaching the target position from the 'same side' on each move to improve position repeatability.
- Move directly to the new position.

For circular devices, the PNPOS instruction can be configured to:

- Move only "clockwise" to increasing positions. For example, with an 8-position device, a move from position 1 to position 6 could be clockwise only (from position 1 through positions 2, 3, 4, and 5 to position 6).
- Move in whichever direction provides the shortest move. For example, with an 8-position device, it could use the shortest path (from position 1 through positions 8 and 7 to position 6).

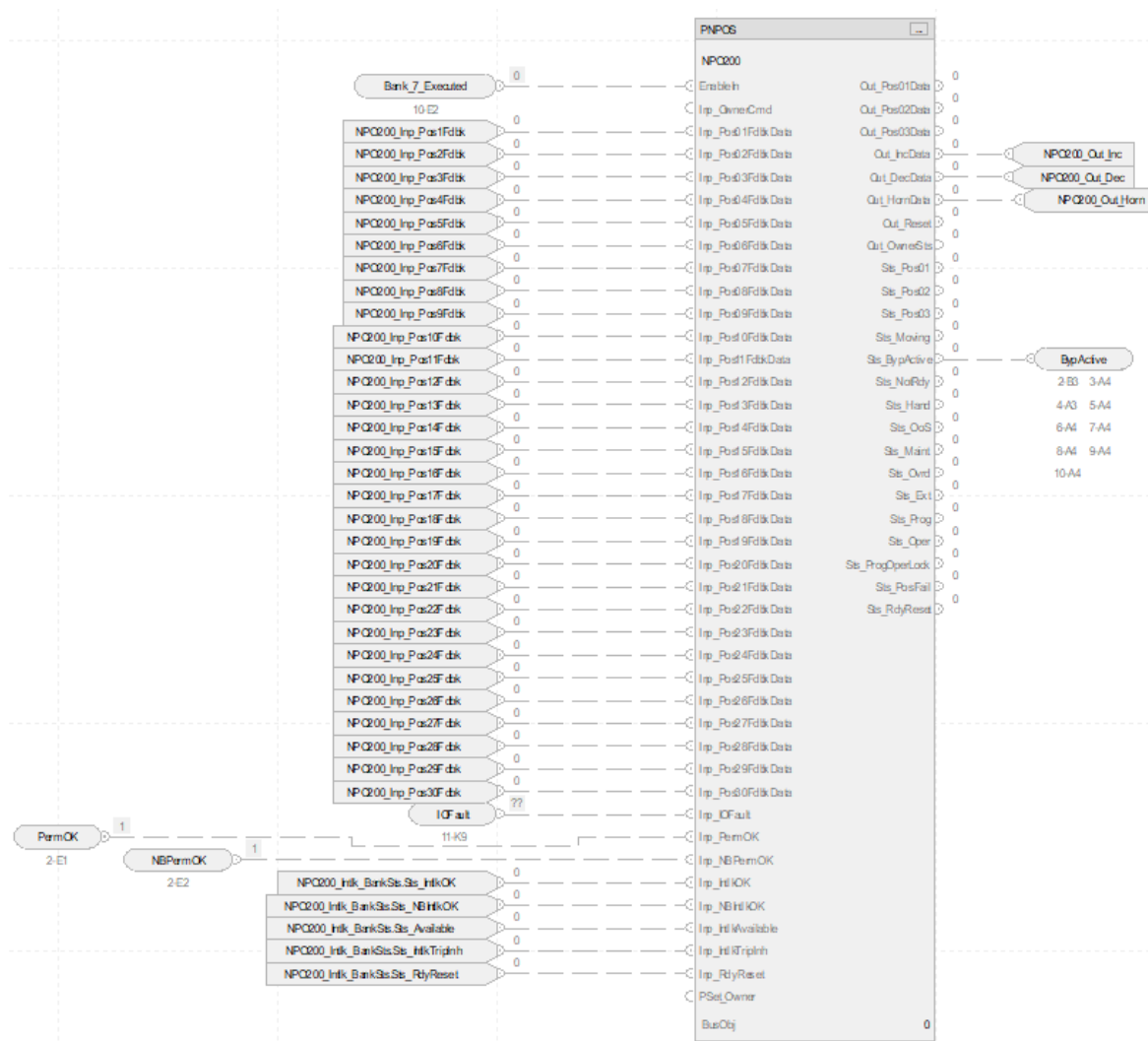
The CS\_PNPOS control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The NPOS control strategy contains these Function Block sheets:

Sheet	Description
CS_PNPOS	n-Position Device Add-On Instruction
Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Fault	The logic monitors as many as 30 discrete input channels and as many as three discrete output channels for I/O fault input and raises an alarm on an I/O fault.

## CS\_PNPOS Sheet



## PNPOS Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
NP0200_Inp_PosxFdbk Where x=1-30	Position x feedback, 1 = Device is confirmed at Position x.
IOFault	Input connection from IO Faults sheet
PermOK	Input connection from Permissives sheet 1 = On permissives OK, device can turn On
NBPermOK	Input connection from Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
NP0200_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
NP0200_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run

Parameter	Description
NP0200_Intlk_BankSts.Sts.Available	Interlock bank status 1 = Available
NP0200_Intlk_BankSts.Sts.IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
NP0200_Intlk_BankSts.Sts.RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

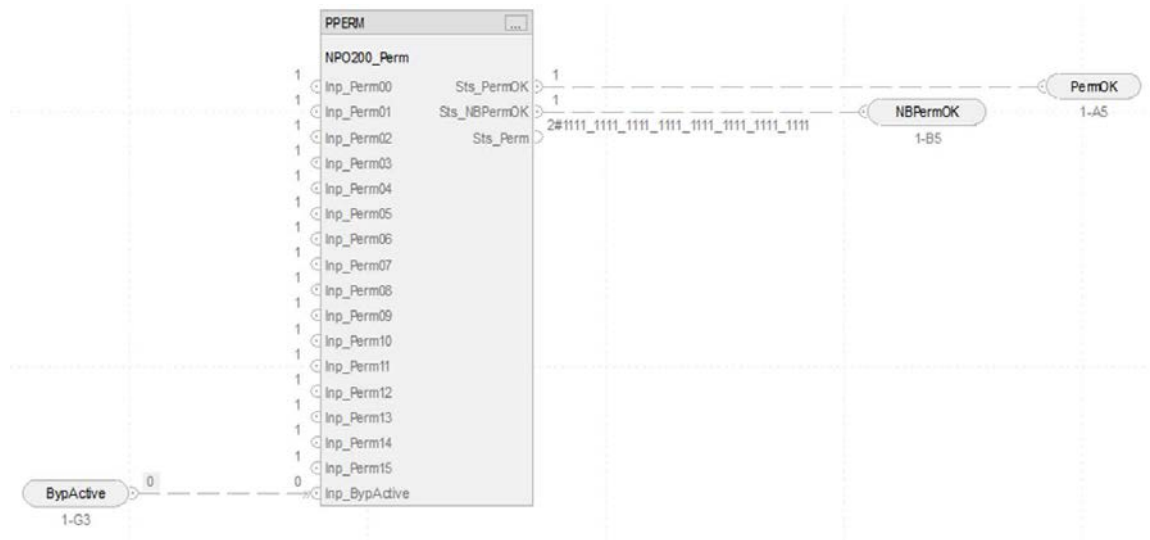
## PNPOS Output References

Parameter	Description
NP0200_Out_Horn	1 = Sound audible before commanded state change
NP0200_Out_Inc	Increment output
NP0200_Out_Dec	Decrement output
BypActive	Output connection to permissives and interlock bank sheets

## PNPOS Configuration Considerations

Operand	Type	Description
PlantPax® control	P_DISCRETE_N_POSITION	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## Permissive Sheet



## PPERM Input References

Parameter	Description
BypActive	Input connection from the CS_PNPOS sheet

## PPERM Output References

Parameter	Description
PermOK	Overall permissive status, 1 = OK to energize
NBPermOK	Non-bypassable permissive status, 1 = all non-bypassable permissives OK to energize



Fault Input References

Parameter	Description
NP0200_Inp_PosxFdbk_ChFlt Where x=1-30	Tieback input x channel fault
NP0200_Out_Dec_ChFlt	Decrease output channel fault
NP0200_Out_Dec_ModFlt	Decrease output module fault
NP0200_Out_Inc_ChFlt	Increase output channel fault
NP0200_Out_Inc_ModFlt	Increase output module fault
D4SD100_Out_Horn_ChFlt	Sound audible for output channel fault
D4SD100_Out_Horn_ModFlt	Sound audible for output module fault

Fault Output References

Parameter	Description
IOFault	Output connection to CS_PNPOS sheet

For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

ACM Considerations for a  
PNPOS Instance

Configure the Cfg\_NumPos parameter to define the number of device positions (2...30).

## ACM-Based Parameters for a PNPOS Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device. See <a href="#">Process Controller on page 36</a>
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow a position command.
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
<b>02.01 - Device Configuration</b>		
Cfg_HasPosFdbk	always	Set if the device has position feedback
Cfg_HasLockFdbk	always	Set if the device has locked feedback
Cfg_HasCylFdbk	always	Set if the device has cylinder feedback
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on configuration of the controller object I/O parameters. See <a href="#">I/O Mapping on page 38</a> .		
Out_Inc	always	Link to the output that drives the increasing position (CW) reference
Out_Dec	always	Link to the output that drives the decreasing position (CCW) reference
Out_Horn	always	Link to the horn output reference
<b>03.xx - IO Configuration</b>		
Assign a compatible I/O point for the associated feedback, where xx=1...30		
Inp_PosxxFdbk	always	Link to the position feedback input reference, where xx=1...30
<b>04 - Alarm Configuration</b>		
Cfg_HasPosFailAlm	always	If Cfg_HasPosFailAlm=True, ACM displays section 4.03 - PosFail Alarm with additional parameters
Cfg_HasLockFailAlm	always	If Cfg_HasLockFailAlm=True, ACM displays section 4.04 - LockFail Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.02 - I/O Fault Alarm with additional parameters

## Additional Sub-Objects for a PNPOS Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive	Configure permissives to allow an output command See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**

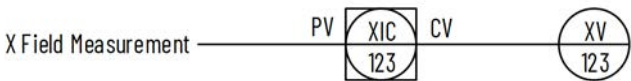


# Process Proportional + Integral + Derivative (PPID) Basic Control Strategies

Use this basic PPID control strategy to manipulate one Control Variable (CV) in response to an error (the difference between the Process Variable (PV) readings and the Setpoint (SP, the target PV) settings.

To scale the CV to align with the associated IO module channel range or to accommodate a fail-open (FO) valve (or air to close) use either of the following options:

- Use a basic PPID with Analog Output control strategy
- Insert a scalar Instruction between the PPID CV and the analog output channel reference



The PPID control strategies are pre-configured to enable selectable controller actions (CV Action, SP Action, and Loop Mode Action) based on various shed conditions (Interlock trip, CV fail, PV fail, and SP fail).

The following PPID control strategies are available as routines in the process library:

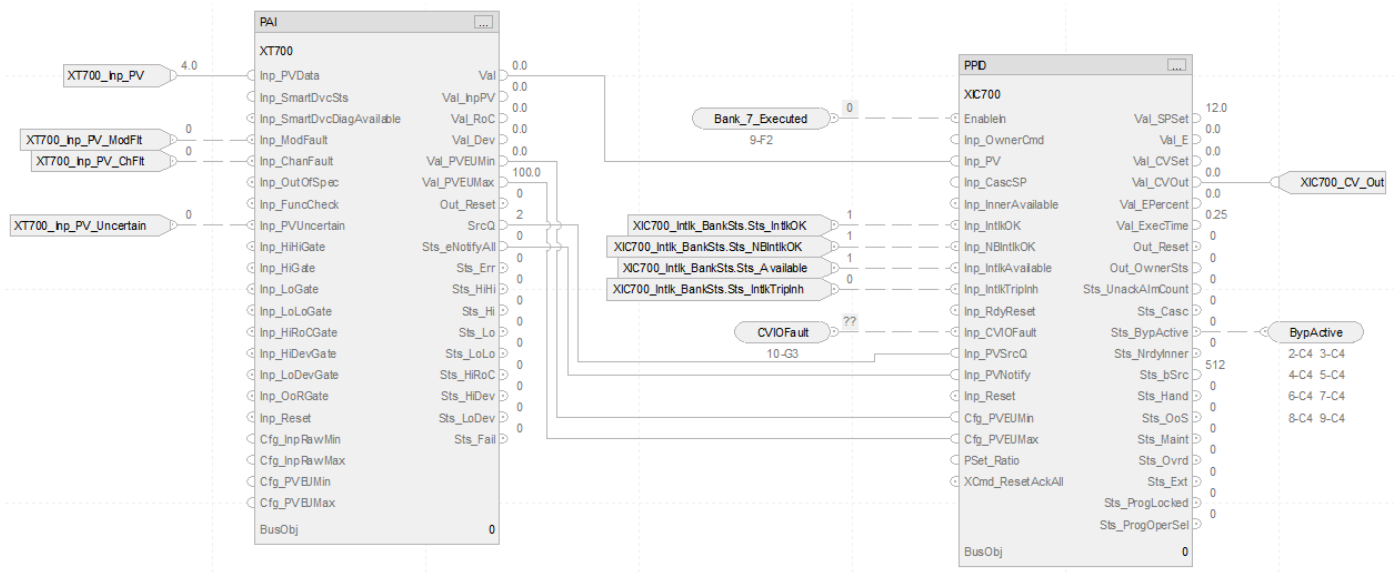
- CS\_PPID
- CS\_PPID\_HART
- CS\_PPID\_EtherNetIP
- CS\_PPID\_EtherNetIP\_NoHB
- CS\_PPID\_FF1
- CS\_PPID\_PA

Import the appropriate control strategy as a **routine** in your controller project.

Each PPID control strategy contains these sheets.

Sheet	Description
CS_PPID	PPID instruction
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PPID instruction monitors bypassablebypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The PPID instruction monitors Control Variable faults.

CS\_PPID Sheet



PAI Input References

See [CS\\_PA1 Sheet on page 148](#) for details. Substitute XT700 for XT101

PAI Outputs to PPID Inputs

Parameter	Description
Val	Value for PPID In_PV parameter Process Variable (PVEU)
SrcQ	Value for PPID In_PVSrcQ parameter In_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID In_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

## PPID Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XIC700_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XIC700_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XIC700_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XIC700_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet

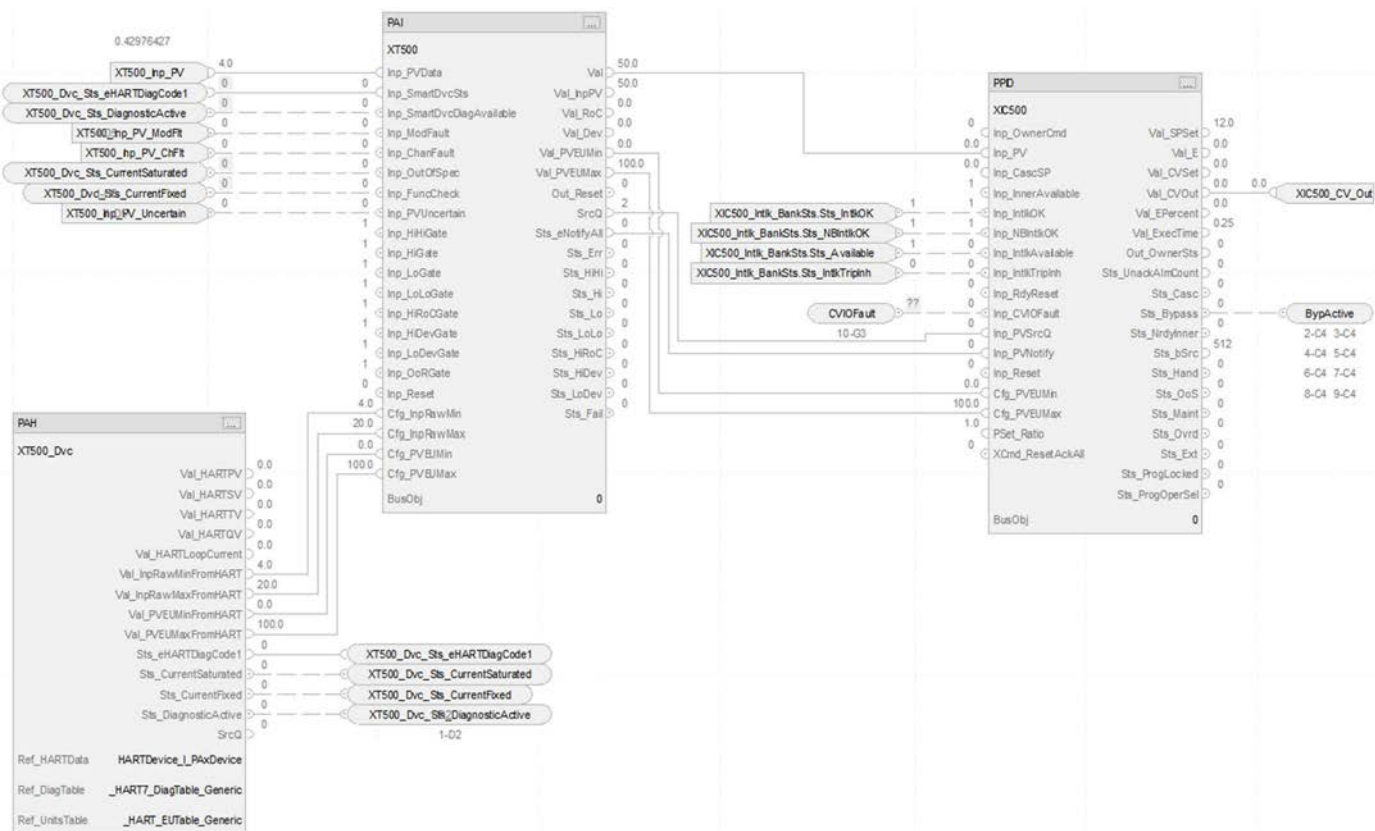
## PPID Output References

Parameter	Description
XIC700_Out_CV	Control Variable output Loop CV after clamping and ramping (CVEU)
BypActive	Output connection to interlock bank sheet

## PPID Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID HART Sheet

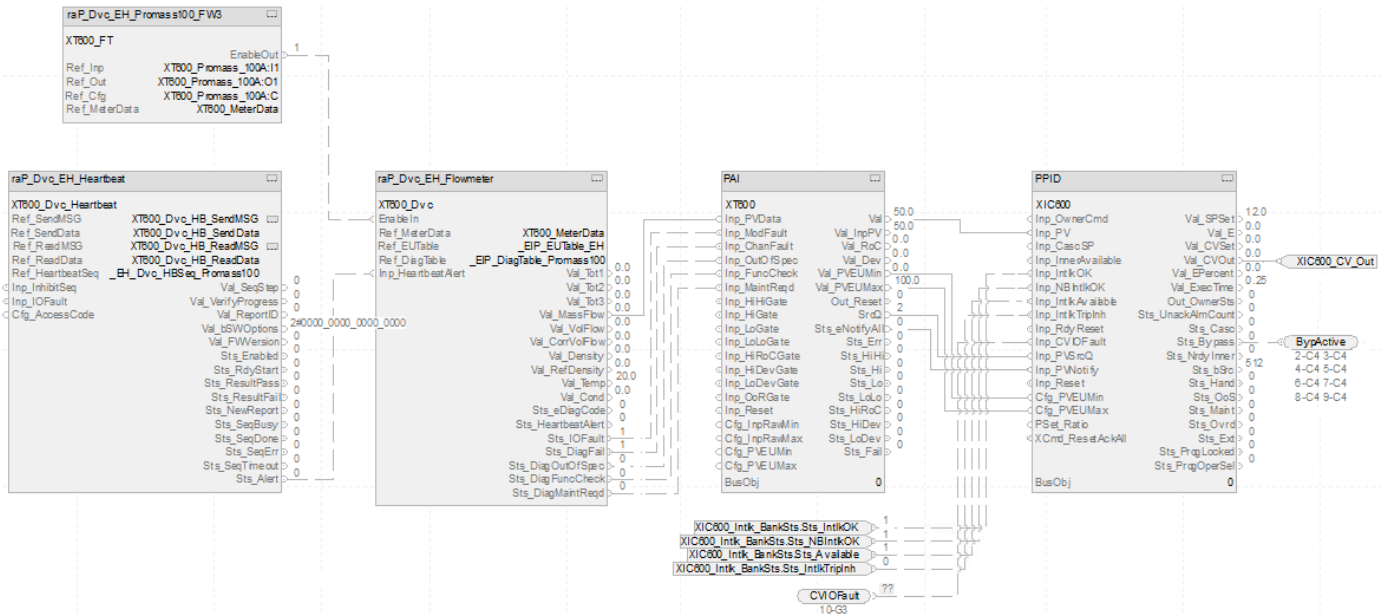


The CS\_PPID HART sheet operates the same as the CS\_PPID sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute for XT500 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_EtherNetIP Sheet

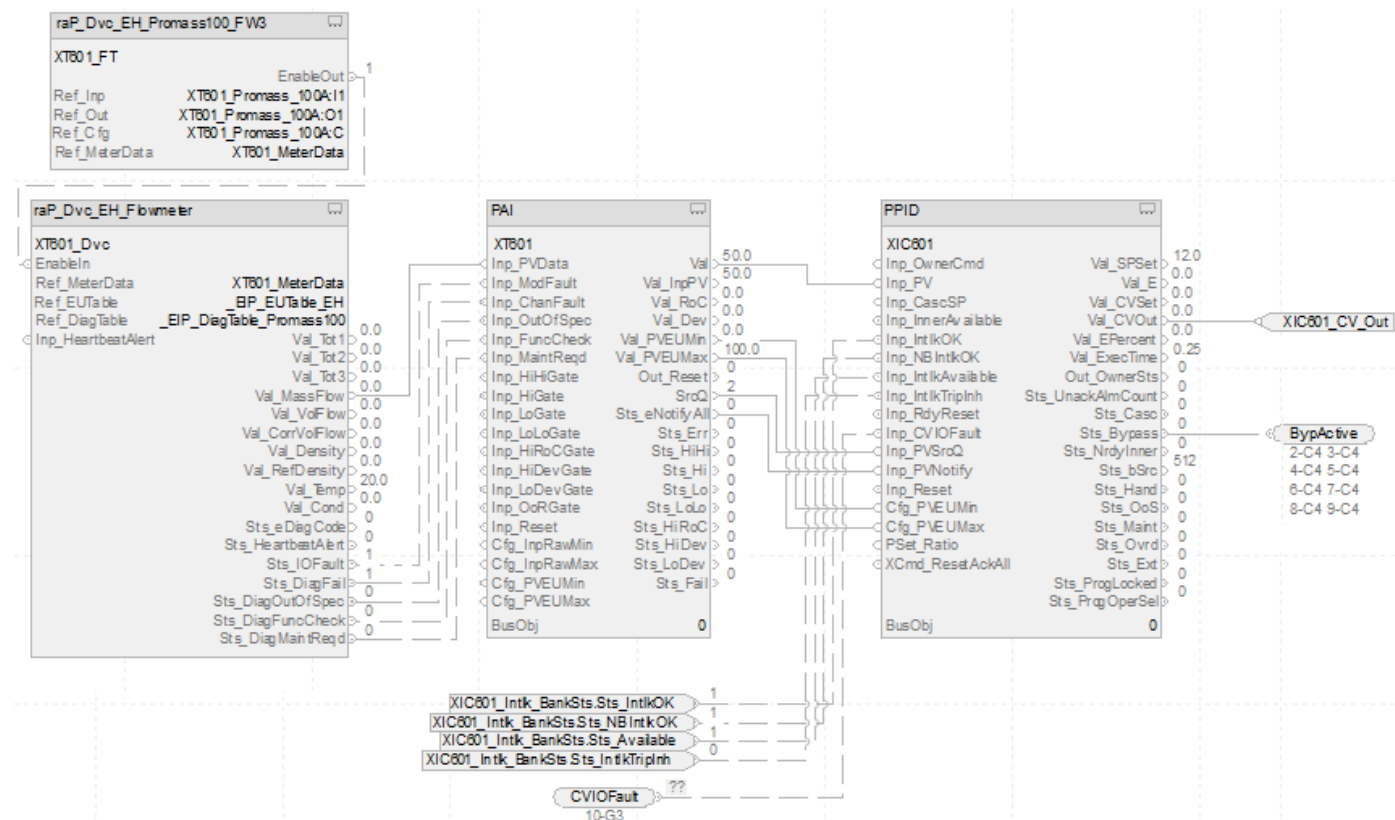


The CS\_PPID EtherNet/IP™ sheet operates the same as the CS\_PPID sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute for XT600 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_EtherNetIP\_NoHB Sheet

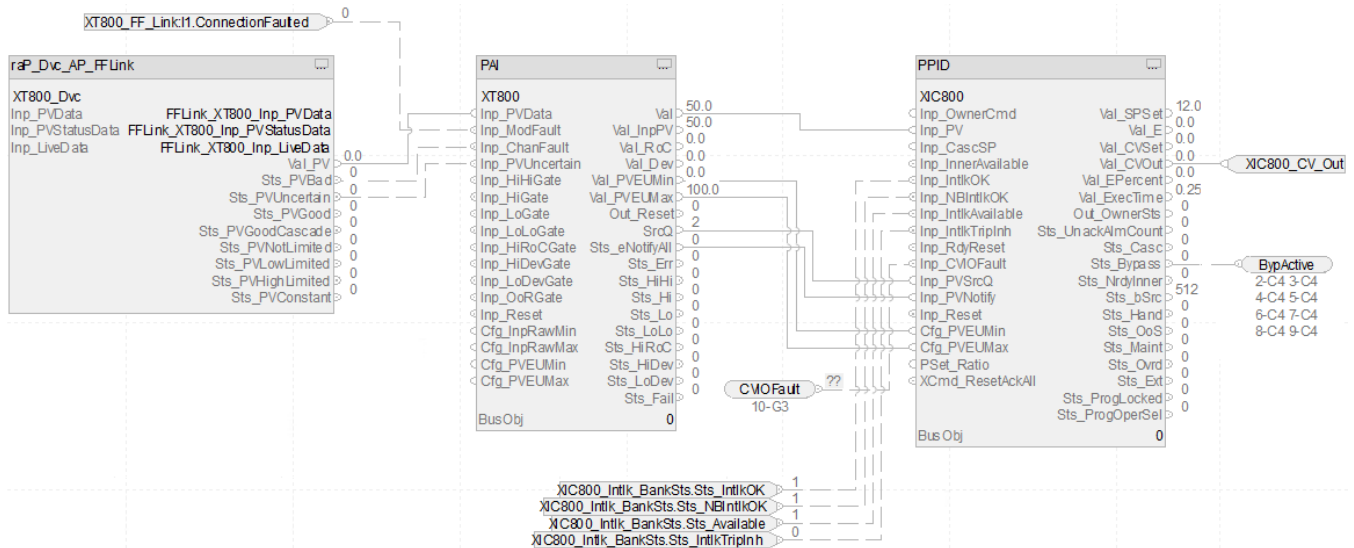


The CS\_PPID EtherNet/IP NoHB sheet operates the same as the CS\_PPID sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute for XT601 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_FF1 Sheet

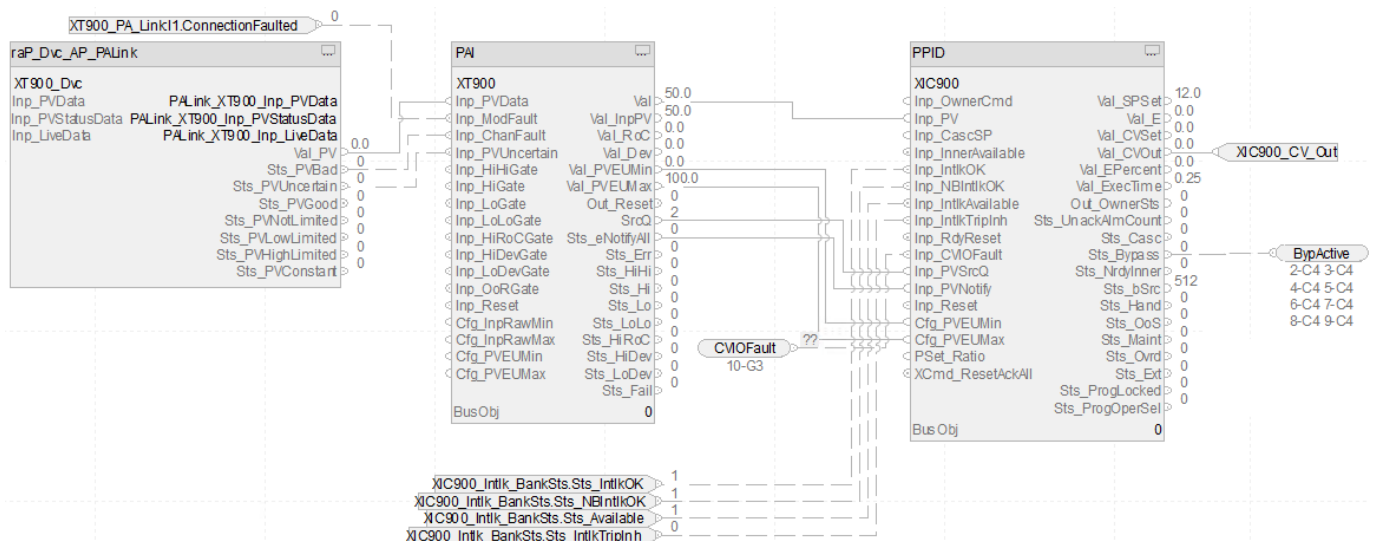


The CS\_PPID\_FF1 sheet operates the same as the CS\_PPID sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PA\\_FF Sheet on page 155](#).
- Substitute for XT500 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_PA Sheet



The CS\_PPID PA sheet operates the same as the CS\_PPID sheet but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PA\\_PA Sheet on page 156](#).
- Substitute for XT900 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## I/O Faults Sheet



### Fault Input References

Parameter	Description
XICxxx_Out_CV_ChFlt	Channel fault 1 = I/O channel fault or failure 0 = OK
XICxxx_Out_CV_ModFlt	Module fault 1 = I/O module failure or module communication status bad 0 = OK

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the type of analog input via the PAI\_Type or PPID\_Type parameter, depending on whether this PPID object is used in another PIDE object
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

### ACM-Based Parameters for a PPID Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_UsedInPIDE	always	Set if this object is used as part of another PPID control strategy.
PPID_Type	Cfg_UsedInPIDE=True	Define the PPID type: Cascade Override
PAI_Type	Cfg_UsedInPIDE=False	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	Cfg_UsedInPIDE=False	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter) Cfg_UsedInPIDE=False	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	Cfg_UsedInPIDE=False IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	Cfg_UsedInPIDE=False IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table



Parameter	Visible When	Details
Ref_HartDevice	Cfg_UsedInPIDE=False IO_Signal_Type=HART Cfg_UsedHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Cfg_HasIntlkObj	Cfg_UsedInPIDE=False	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance	Has_00AP=True (controller parameter) Use_00AP=True Cfg_UsedInPIDE=False	Link to a bus array instance. This should be unique for each device

**03 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

PAI_RefTag	Cfg_UsedInPIDE=False PAI_Type=PAI(Single_channel) Cfg_UsedInPIDE=False PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference Cfg_UsedInPIDE=False PAI_Type=PAI(Single_channel)	Link to the PV input reference
CV_Out	Cfg_UsedInPIDE=False	Link to the CV output reference

**03.00 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.01 - Ref PAI Alarm Configuration**

Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Parameter	Visible When	Details
Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

## Additional Sub-Objects for a PPID Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

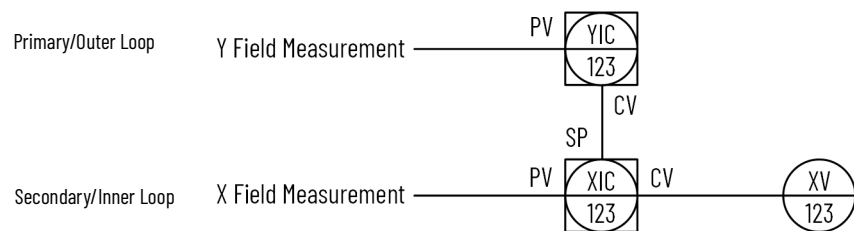
Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Notes:



## Process Proportional + Integral + Derivative (PPID) with Cascade Control Strategies

Cascade control is defined as when an outer (primary) control loop's output (CV) is used as a setpoint (SP) to an inner (secondary) control loop. The PPID Cascade control strategy is useful when external disturbances to the inner loop process variable are frequent, which can eventually cause disturbances to the process variable of the outer control loop. Controlling the disturbance at the faster acting inner loop compensates for the resulting disturbance to the outer control loop. Also, non-linearities in the final control element can also be controlled at the faster acting Inner loop reducing potential disturbances to the outer loop.



The provided control strategies are pre-configured to provide the following features:

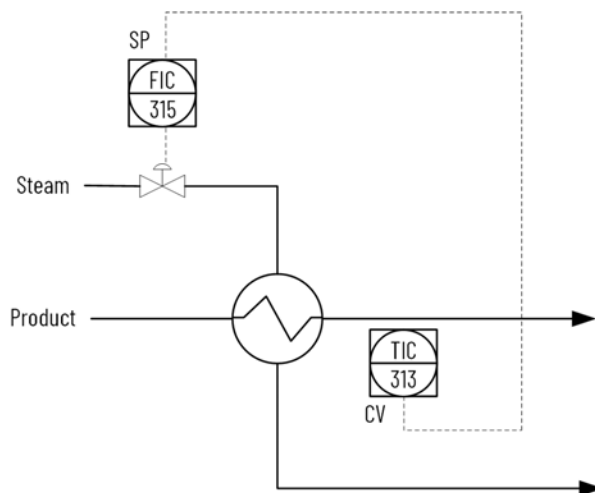
- Bumpless transfer: when the inner control loop is not available, the outer loop output (CV) tracks the inner loop setpoint.
- Anti-windup: when the inner loop hits a CV limit, the outer control loop output is prevented from winding up (increasing or decreasing) beyond that limit.

Another feature pre-configured in the control strategies is the visibility of the whole cascade control strategy status at both the outer and inner control loops. This lets you place the inner and outer PPIDs on different operator displays, while each PPID indicates the status of the whole strategy.

## PPID with Cascade Control Example

In this example, the temperature of product flows through a heat exchanger where the exiting product temperature is ultimately maintained at setpoint by modulating the flow of steam to the heat exchanger.





































A cascade loop provides better control by opening the steam valve when the steam flow drops before the product temperature drops. To implement a cascade loop, use a PPID instruction to control the steam valve opening based on a process variable signal from a steam flow transmitter. This is the inner loop of the cascaded pair. A second PPID instruction (outer loop) uses the product temperature as a process variable and sends its CV output into the setpoint of the inner loop. In this manner, the outer temperature loop modulates the steam flow setpoint of the inner loop. The steam flow loop is then responsible for providing the amount of steam that is requested by the temperature loop to maintain a constant product temperature.



An external disturbance to the outer loop (such as an increase in product flow) results in a reduction in temperature. In this scenario, the outer loop increases its output to increase the steam flow setpoint to bring the product temperature back to setpoint.

If an upstream disturbance reduces the steam pressure, the steam flow controller (inner loop) reacts by opening the steam valve to maintain steam flow that mitigates any resulting disturbance to the product temperature (outer loop).

The following PPID Cascade control strategies (consisting of multiple routines) are available in the process library:

Control Strategy	Routines
CS_PPID_CASC	 CS_PPID_CASC <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> XIC760</li> <li> XIC770</li> </ul>
CS_PPID_CASC_HART	 CS_PPID_CASC_HART <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> XIC560</li> <li> XIC570</li> </ul>
CS_PPID_CASC_EtherNetIP	 CS_PPID_CASC_EtherNetIP <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> Interlocks</li> <li> XIC660</li> <li> XIC670</li> </ul>
CS_PPID_CASC_EtherNetIP_NoHB	 CS_PPID_CASC_EtherNetIP_NoHB <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> Interlocks</li> <li> XIC661</li> <li> XIC671</li> </ul>
CS_PPID_CASC_FF	 CS_PPID_CASC_FF <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> FFLinkMap</li> <li> Interlocks</li> <li> XIC860</li> <li> XIC870</li> </ul>
CS_PPID_CASC_PA	 CS_PPID_CASC_PA <ul style="list-style-type: none"> <li> Parameters and Local Tags</li> <li> MainRoutine</li> <li> Interlocks</li> <li> PALinkMap</li> <li> XIC960</li> <li> XIC970</li> </ul>

Import the **routines** for the appropriate control strategy in your controller project. Each control strategy contains multiple routines; each routine contains multiple Function Block sheets.

Each PPID Cascade control strategy contains these routines:

- Inner Loop
- Outer Loop





*PAI Outputs to PPID Inputs*

Parameter	Description
Val_InpPV	Analog input value in engineering units (actual, before Substitute PV selection)
Val_PVEUmin	Value for PPID Cfg_PVEUMin parameter PV minimum value for scaling from engineering units to %, PV at 0% (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value for scaling from engineering units to %, PV at 100% (PVEU). Valid any float greater than Cfg_PVEUMin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: <div style="display: flex; justify-content: space-between;"> <div> 0 = Good, live, confirmed good  1 = Good, live, assumed good  2 = Good, no feedback, assumed good  8 = Test, simulated  9 = Test, loopback  10 = Test, manually entered  16 = Uncertain, live, off-spec  17 = Uncertain, substituted at device </div> <div> 18 = Uncertain, substituted at instruction  19 = Uncertain, using last known good  20 = Uncertain, using replacement value  32 = Bad, signal failure  33 = Bad, channel fault  34 = Bad, module/communications fault  35 = Bad, invalid configuration </div> </div>

*PPID Input References*

Parameter	Description
XIC770.Val_CVout	Outer Loop CV after clamping and ramping (CVEU).
XIC760.Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XIC760.Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XIC760.Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XIC760.Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection to IO Faults sheet
XIC770.Sts_eNotify	Alarm status from outer loop: 0 = Not in alarm, acknowledged, 1 = Not in alarm, unacknowledged or reset required, 2 = Low severity alarm, acknowledged, 3 = Low severity alarm, unacknowledged, 4 = Medium severity alarm, acknowledged, 5 = Medium severity alarm, unacknowledged, 6 = High severity alarm, acknowledged, 7 = High severity alarm, unacknowledged, 8 = Urgent severity alarm, acknowledged, 9 = Urgent severity alarm, unacknowledged.

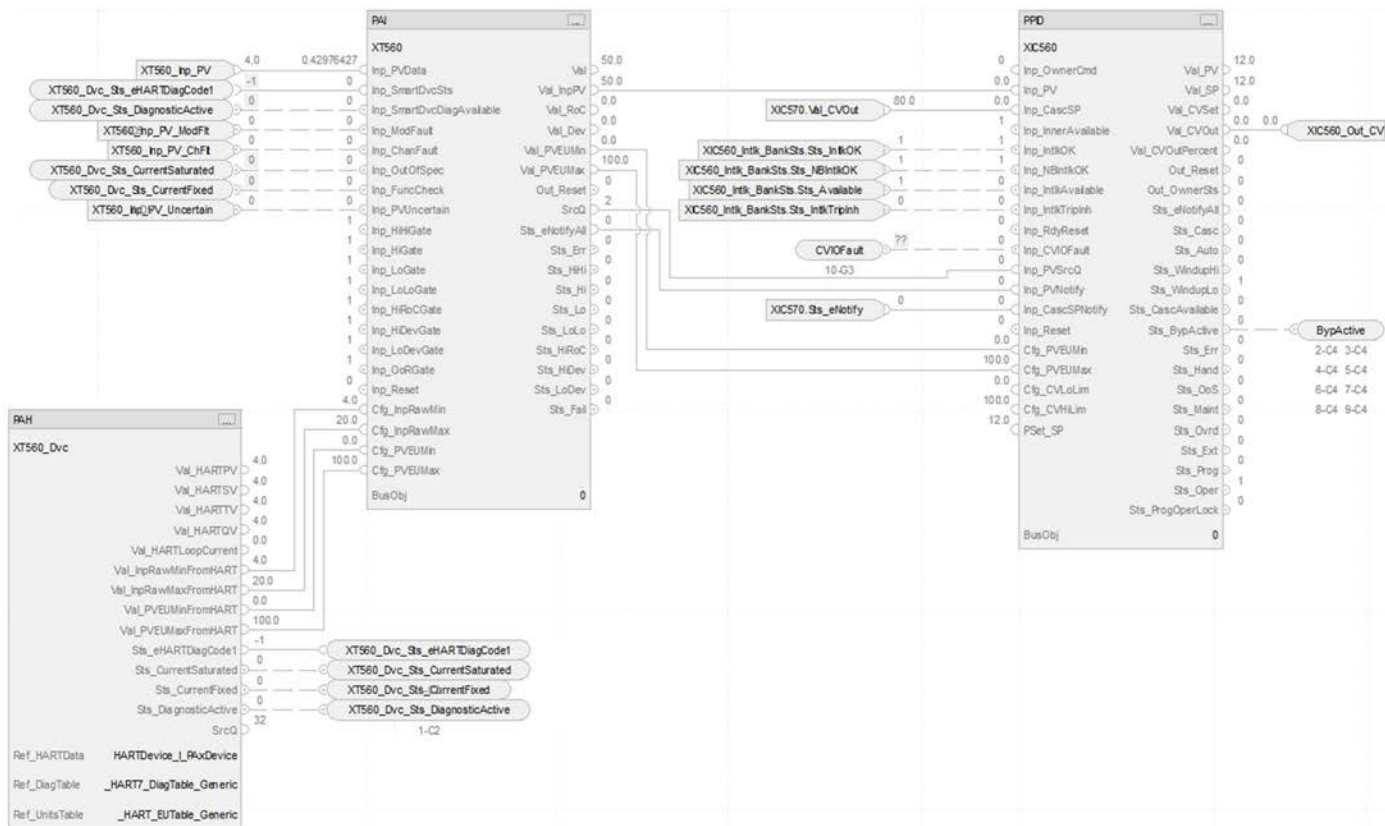
*PPID Output References*

Parameter	Description
XIC760.Out_CV	Control Variable output Loop CV after clamping and ramping (CVEU)
BypActive	Output connection to interlock bank sheet

*PPID Configuration Considerations*

Operand	Type	Description
PlantPax® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>0 if not using organization</li> <li>Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID\_CASC - Inner Loop HART Sheet (XIC560)

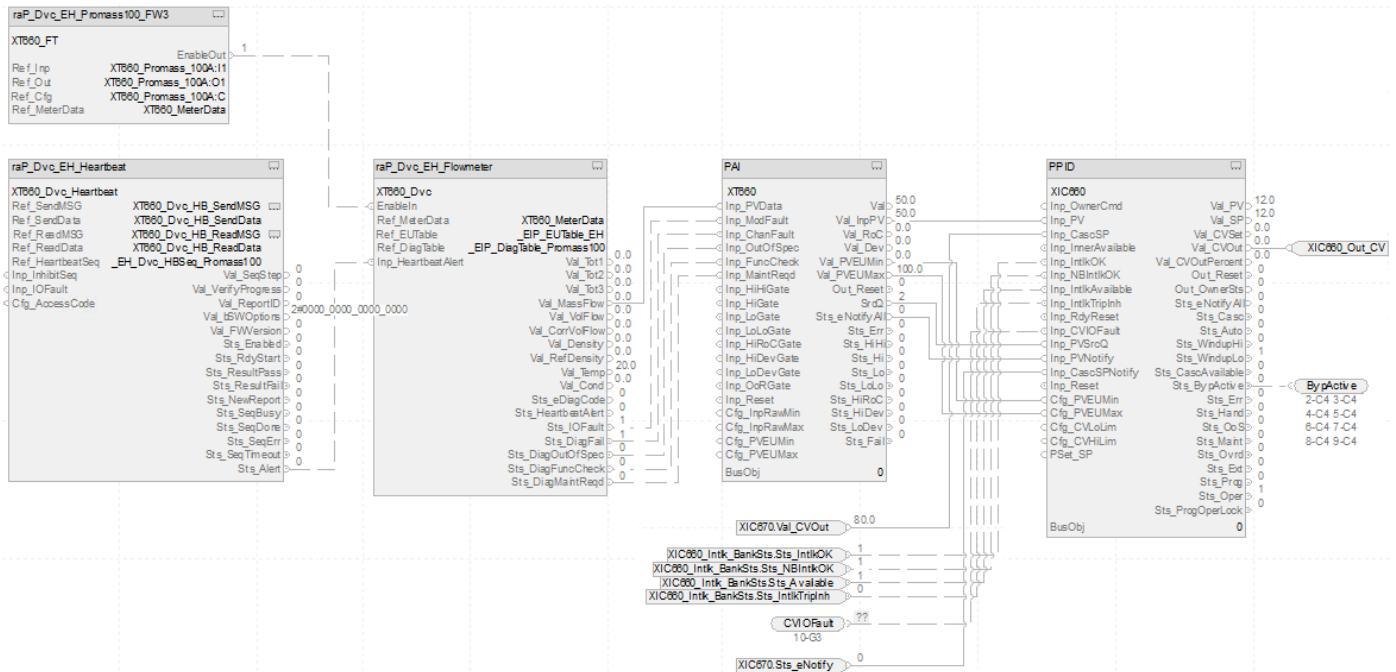


The CS\_PPID\_CASC -Inner Loop HART sheet operates the same as the CS\_PPID\_CASC - Inner Loop but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XT560 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_CASC - Inner Loop EtherNet/IP Sheet (XIC660)

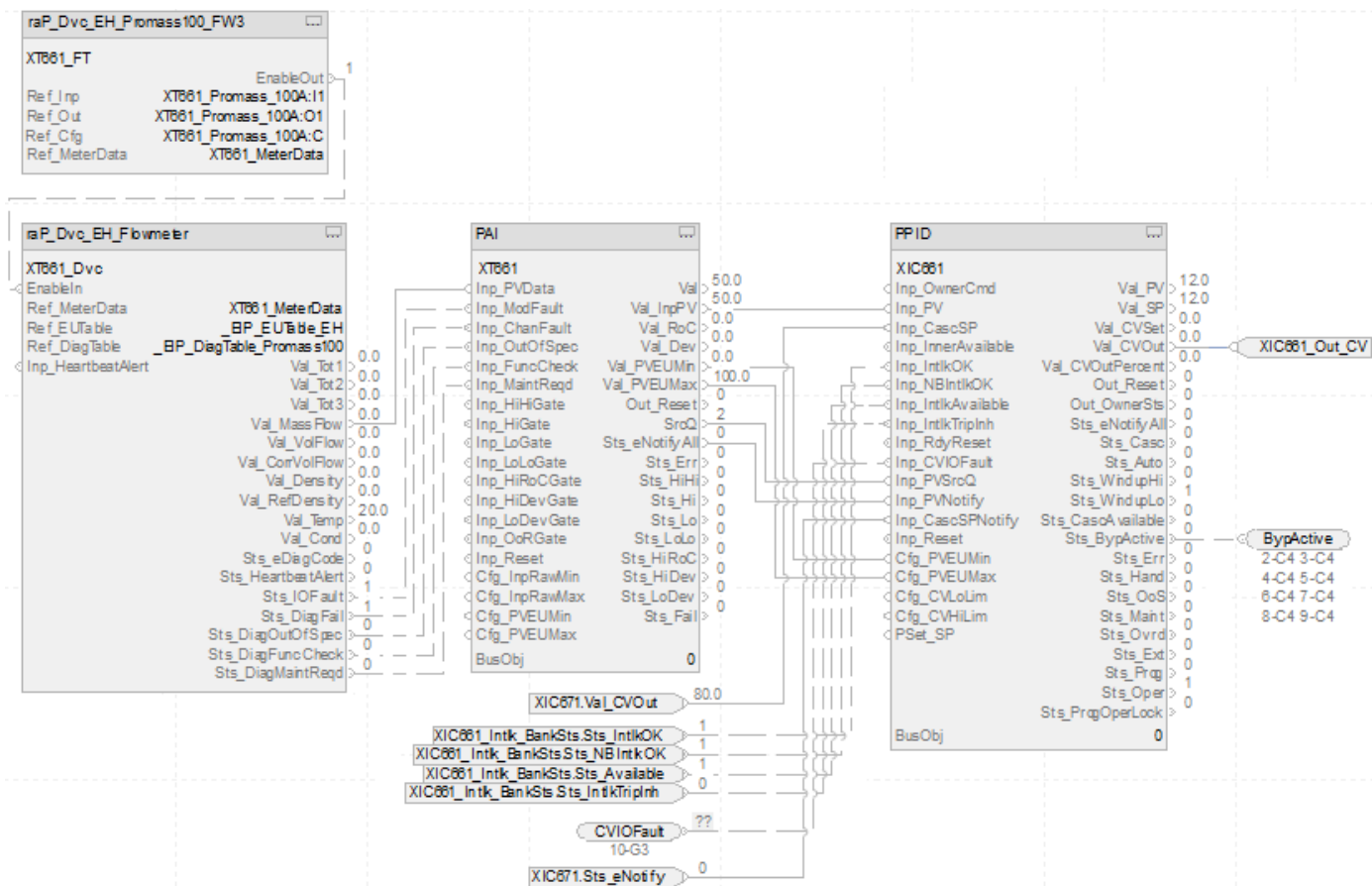


The CS\_PPID\_CASC -Inner Loop EtherNet/IP sheet operates the same as the CS\_PPID\_CASC - Inner Loop but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PA1\\_EtherNet/IP Sheet on page 151](#).
- Substitute XT660 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_CASC - Inner Loop EtherNet/IP NoHB Sheet (XIC661)

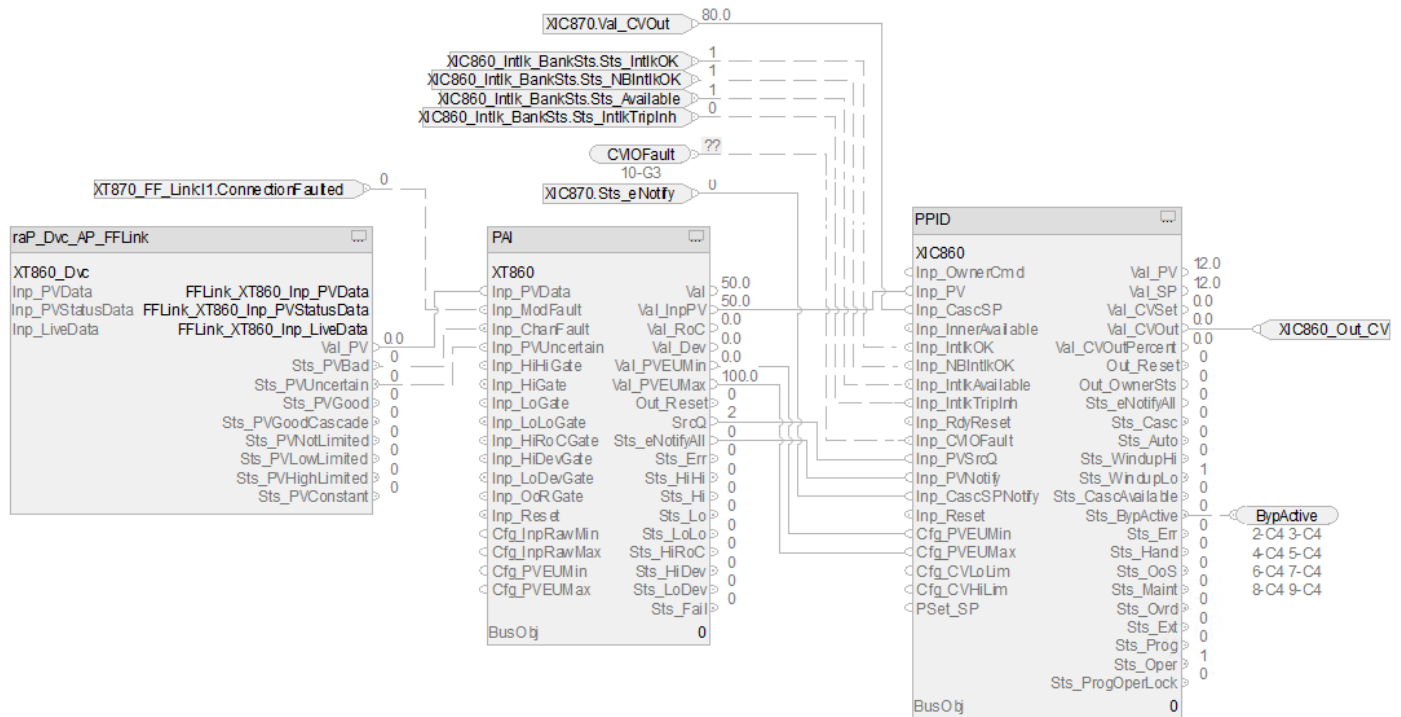


The CS\_PPID\_CASC - Inner Loop EtherNet/IP NoHB sheet operates the same as the CS\_PPID\_CASC - Inner Loop but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XT661 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_CASC - Inner Loop FF Sheet (XIC860)

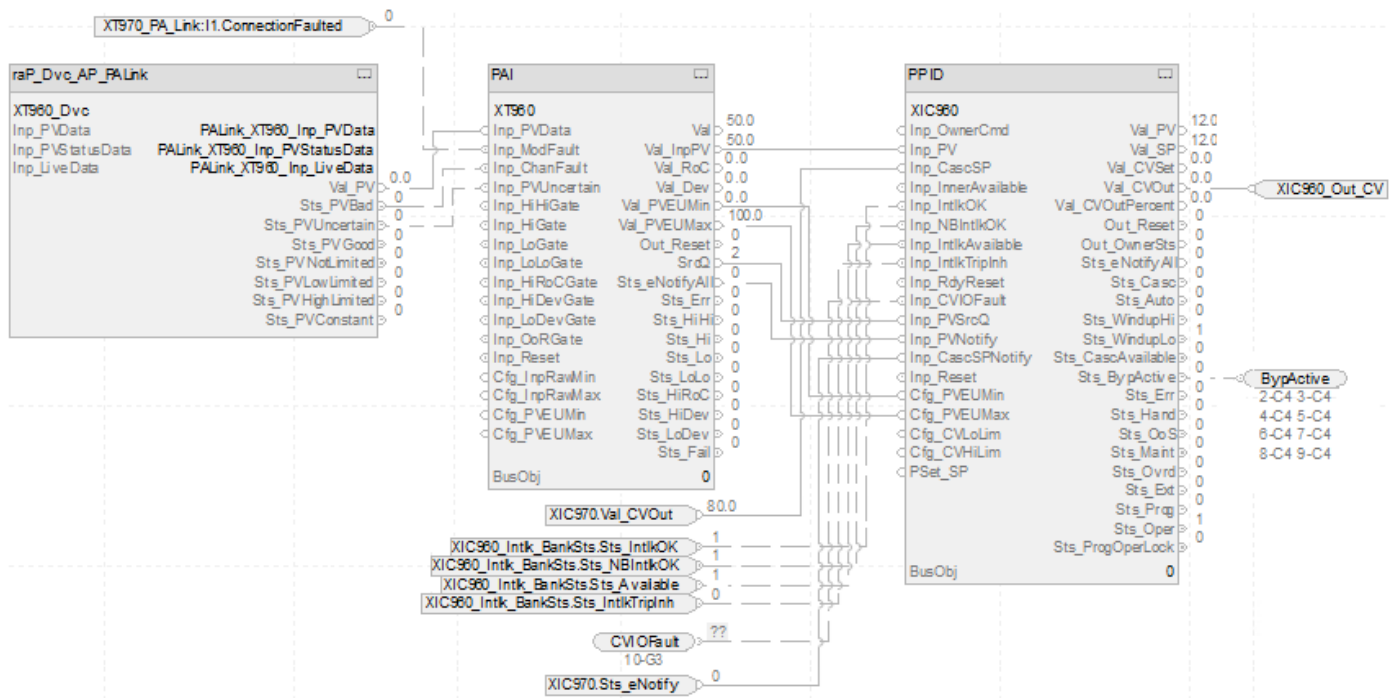


The CS\_PPID\_CASC - Inner Loop FF sheet operates the same as the CS\_PPID\_CASC - Inner Loop but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XT860 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_CASC - Inner Loop PA Sheet (XIC960)



The CS\_PPID\_CASC -Inner Loop PA sheet operates the same as the CS\_PPID\_CASC - Inner Loop but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XT960 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## Interlock Bank Inner Loop Sheet



### PINTLK Input Reference

Parameter	Description
BypActive	Input connection from CS_PPID_CASC - Inner Loop sheet

### PINTLK Configuration Considerations

Operand	Type	Description
PlantPAx control	P_INTERLOCK	Instance of data structure (backing tag) required for proper operation of instruction
Ref_IntlkBankSts	P_INTERLOCK_BANK_STATUS	Reference interlock bank status

For more information about interlocks and how to configure multiple banks, see [Interlock Options on page 29](#).

## IO Faults Inner Loop Sheet



### Faults Input References

Parameter	Description
XIC760_Out_CV_ChFit	Channel fault, 1 = I/O channel fault or failure, 0 = OK
XIC760_Out_CV_ModFit	Module fault, 1 = I/O module failure or module communication status bad, 0 = OK

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_CASC Inner Loop sheet

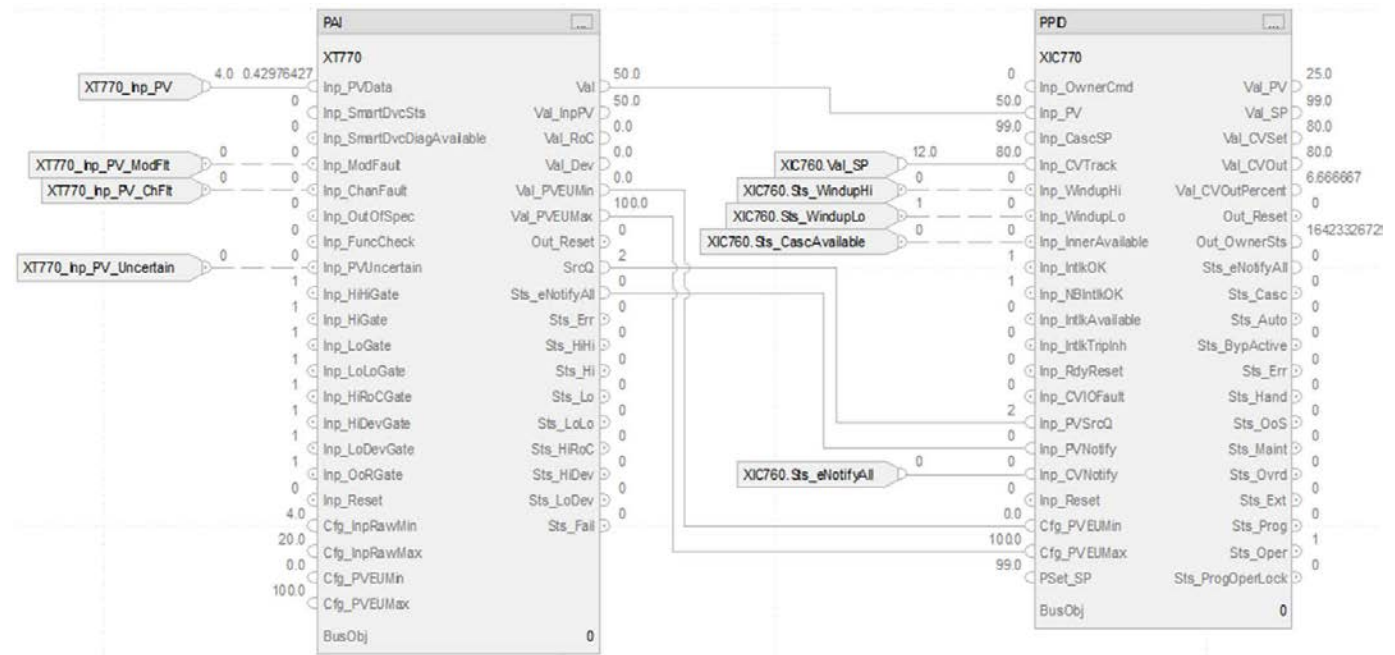
For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

ROUTINE Outer Loop

Each routine contains one sheet

Sheet	Description
CS_PPID_CASC - Outer Loop	PPID outer loop <ul style="list-style-type: none"><li>• XIC770 analog</li><li>• XIC570 HART</li><li>• XC670 EtherNet/IP</li><li>• XC671 EtherNet/IP with no heartbeat</li><li>• XC870 FOUNDATION Fieldbus</li><li>• XC970 Profibus PA</li></ul>

CS\_PPID\_CASC-Outer Loop Sheet (XIC770)



PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details. Substitute XT770 for every instance of XT101



*PAI Outputs to PPID Inputs*

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUmin parameter PV minimum value for scaling from engineering units to %, PV at 0% (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value for scaling from engineering units to %, PV at 100% (PVEU). Valid any float greater than Cfg_PVEUmin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: <div style="display: flex; justify-content: space-between;"> <div> 0 = Good, live, confirmed good  1 = Good, live, assumed good  2 = Good, no feedback, assumed good  8 = Test, simulated  9 = Test, loopback  10 = Test, manually entered  16 = Uncertain, live, off-spec  17 = Uncertain, substituted at device </div> <div> 18 = Uncertain, substituted at instruction  19 = Uncertain, using last known good  20 = Uncertain, using replacement value  32 = Bad, signal failure  33 = Bad, channel fault  34 = Bad, module/communications fault  35 = Bad, invalid configuration </div> </div>
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: <div style="display: flex; justify-content: space-between;"> <div> 0 = Not in alarm, acknowledged  1 = Not in alarm, unacknowledged or reset required  2 = Low severity alarm, acknowledged  3 = Low severity alarm, unacknowledged  4 = Medium severity alarm, acknowledged </div> <div> 5 = Medium severity alarm, unacknowledged  6 = High severity alarm, acknowledged  7 = High severity alarm, unacknowledged  8 = Urgent severity alarm, acknowledged  9 = Urgent severity alarm, unacknowledged </div> </div>

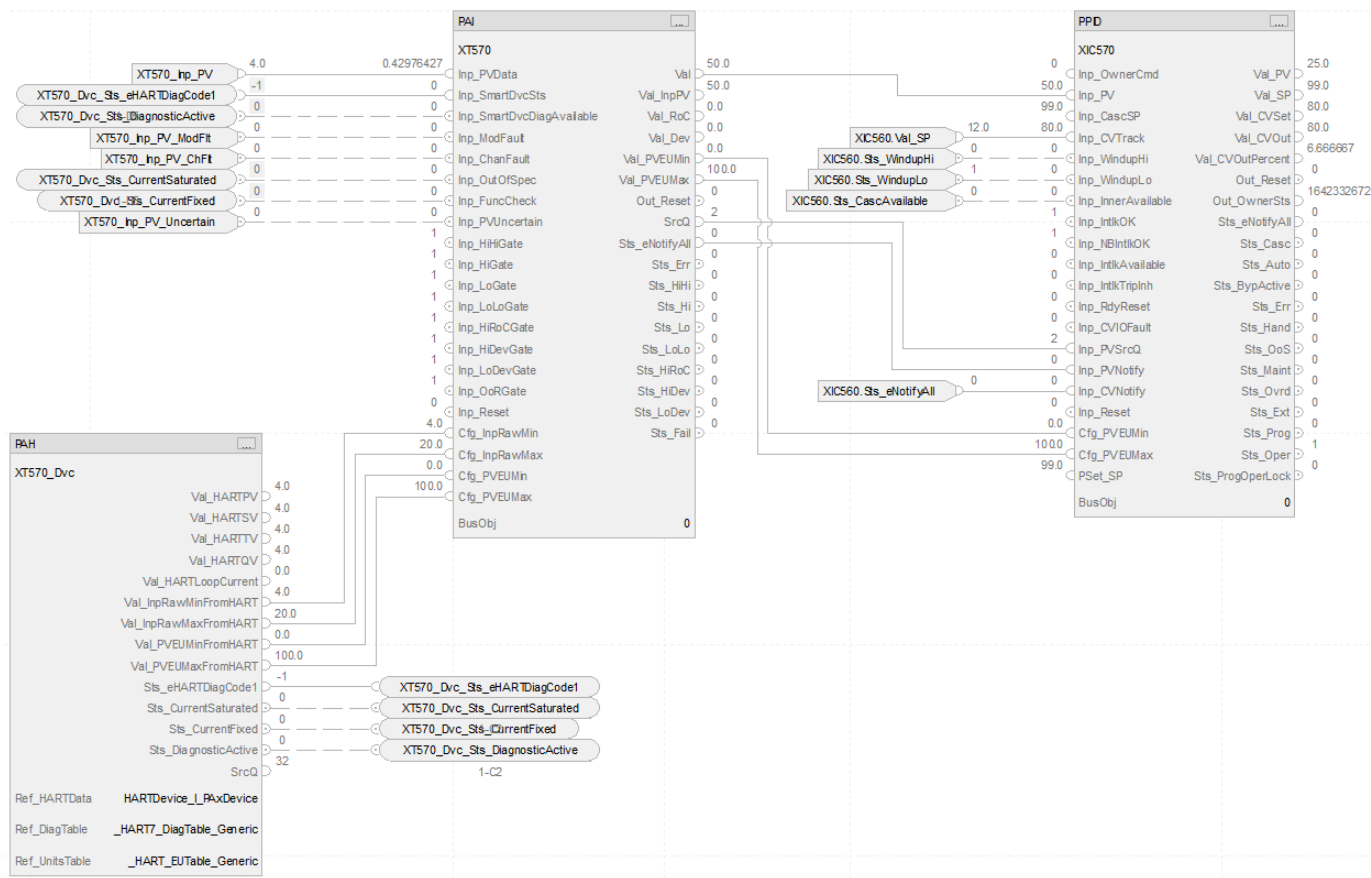
*PPID Input References*

Parameter	Description
XIC760.Inp_CVTrack	Inner loop CV to track if Cfg_UseCVTrack = 1 or if Inp_InnerAvailable = 0 (CVEU)
XIC760.Sts_WindupHi	1 = The inner loop winding up High, usually connects to Inp_WindupHi of outer loop
XIC760.Sts_WindupLo	1 = The inner loop winding up Low, usually connects to Inp_WindupLo of outer loop
XIC760.Sts_CascAvailable	1 = Inner loop is available. 0 = Inner loop is not available, PPID tracks Inp_CVTrack, typically inner loop SP or actuator position.
XIC760.Sts_eNotifyAll	Alarm status from inner loop: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

*PPID Configuration Considerations*

Operand	Type	Description
PlantPAx control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>0 if not using organization</li> <li>Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID\_CASC - Outer Loop HART Sheet (XIC570)

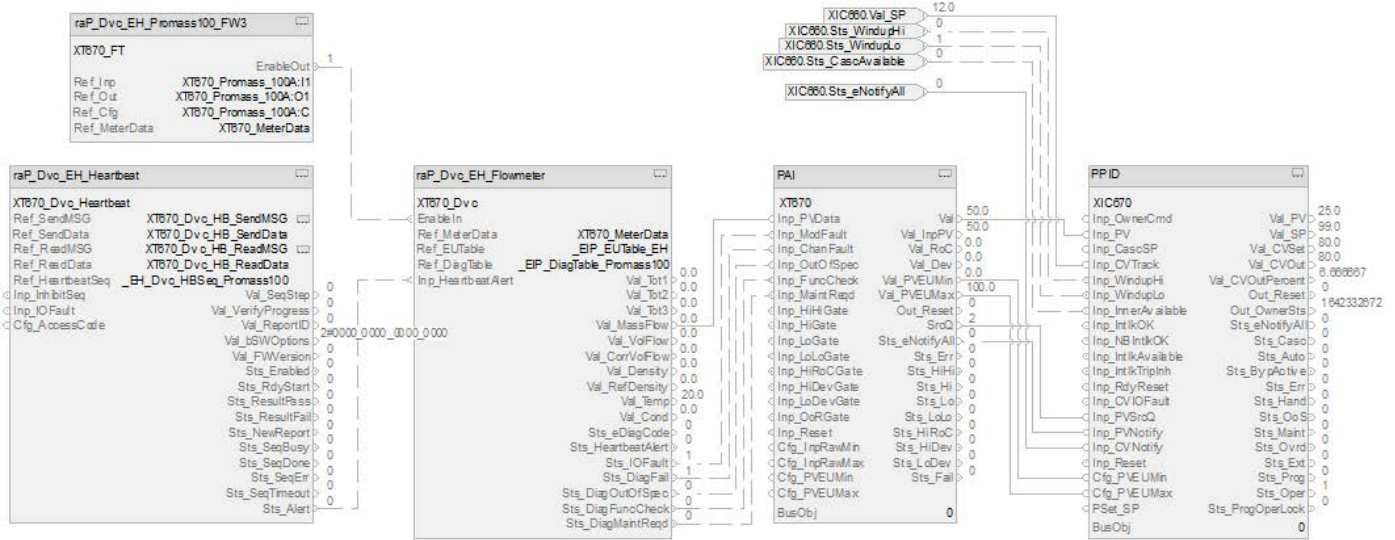


The CS\_PPID\_CASC - Outer Loop HART sheet operates the same as the CS\_PPID\_CASC - Outer Loop sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute XT570 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_CASC - Outer Loop EtherNet/IP Sheet (XIC670)

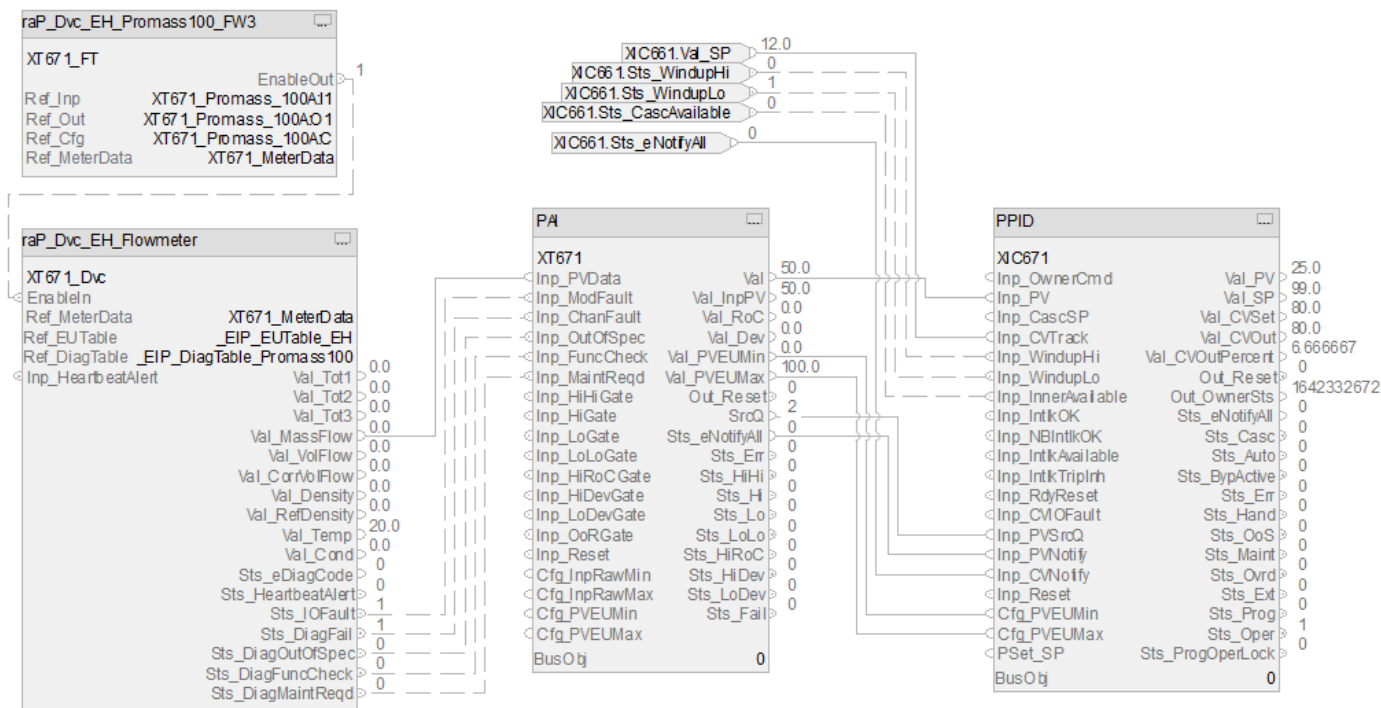


The CS\_PPID\_CASC - Outer Loop EtherNet/IP sheet operates the same as the CS\_PPID\_CASC - Outer Loop sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XT670 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_CASC - Outer Loop EtherNet/IP NoHB Sheet (XIC671)

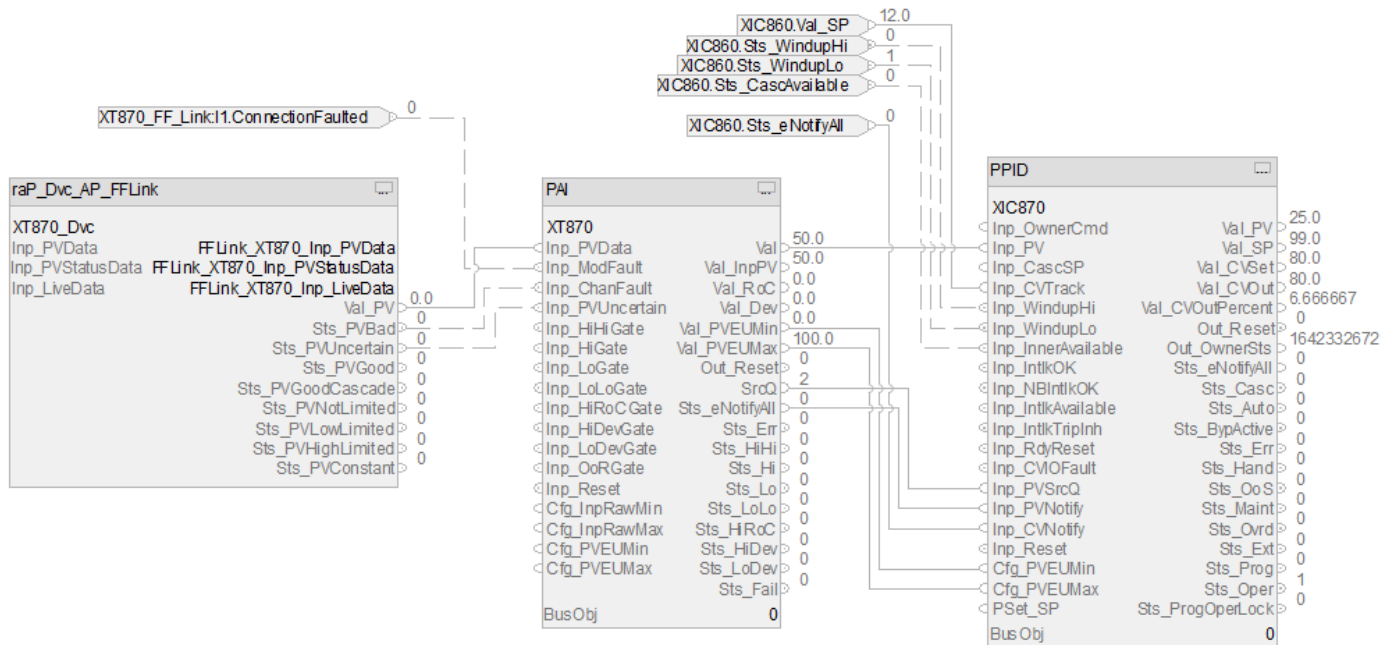


The CS\_PPID\_CASC - Outer Loop EtherNet/IP NoHB sheet operates the same as the CS\_PPID\_CASC - Outer Loop sheet but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XT671 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_CASC - Outer Loop FF Sheet (XIC870)

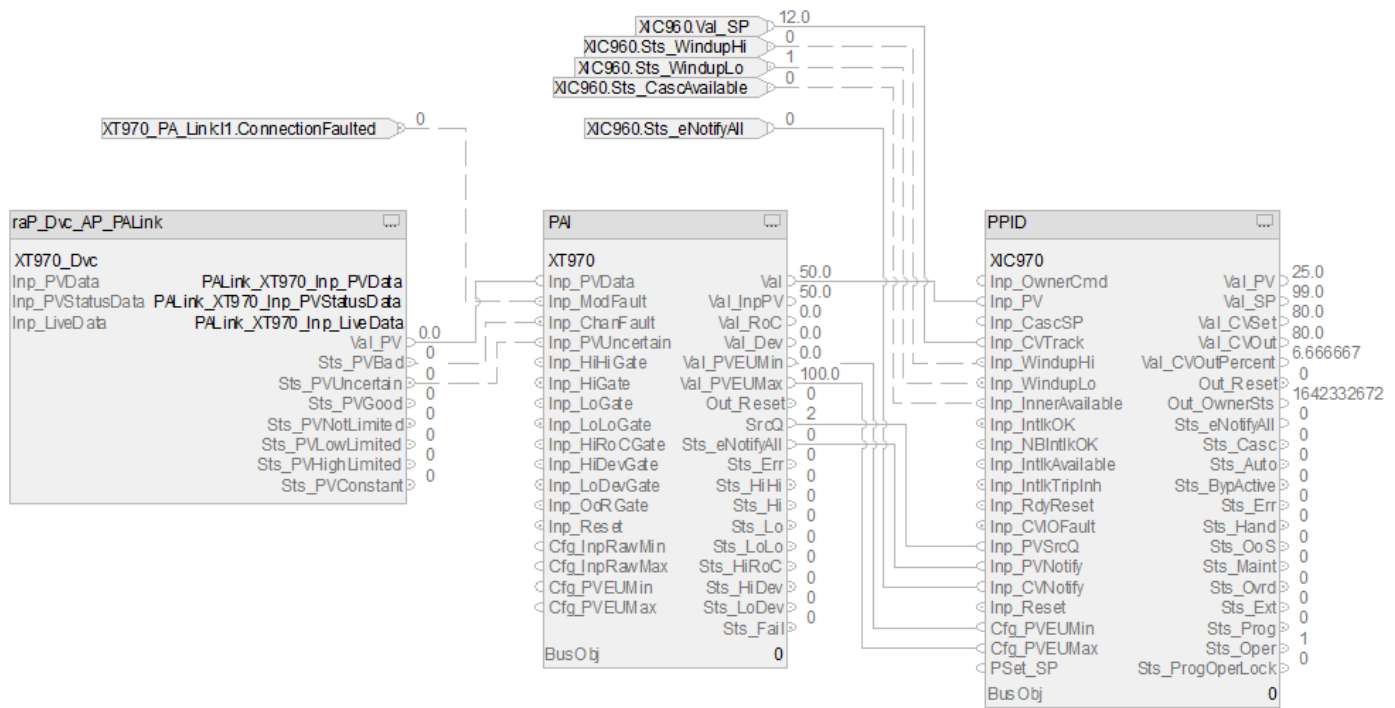


The CS\_PPID\_CASC - Outer Loop FF sheet operates the same as the CS\_PPID\_CASC - Outer Loop sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAIF Sheet on page 155](#).
- Substitute XT870 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

CS\_PPID\_CASC - Outer Loop PA Sheet (XIC970)



The CS\_PPID\_CASC - Outer Loop PA sheet operates the same as the CS\_PPID\_CASC - Outer Loop sheet but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XT970 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

IO Faults Outer Loop Sheet



Faults Input References

Parameter	Description
XICxxx_Out_CV_ChFlt	Channel fault, 1 = I/O channel fault or failure, 0 = OK
XICxxx_Out_CV_ModFlt	Module fault, 1 = I/O module failure or module communication status bad, 0 = OK

Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_CASC Outer Loop sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID with Cascade Control

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the type of analog input via the PAI\_Type parameter for the outer loop and PAI1\_Type parameter for the inner loop.
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type and PV1\_IO\_Signal\_Type parameters

Select the IO\_Signal\_Type and PV1\_IO\_Signal\_Type parameters first because they affect the visibility of the remaining parameters in the PPID object.

## ACM-Based Parameters for a PPID Instance with Cascade Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
IO_Signal_Type	always	Select the signal type for the outer loop PAI: None, HART, EH_EthernetIP, FF, or PA.
PV1_IO_Signal_Type	always	Select the signal type for the inner loop PAI: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>00.03 - Outer Loop Selection</b>		
PAI_Type	always	Select the PAI type for the outer loop: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAI_RefTag	PAI_Type=PAI(Single_channel) PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
<b>00.04 - Inner Loop Selection</b>		
PPID_Ref_Inner_Loop	always	Link to the analog input reference for the inner loop
PAI1_Type	PPID_Ref_Inner_Loop is linked to an analog input reference	Select the PAI type for the inner loop: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAI1_RefTag1	PPID_Ref_Inner_Loop is linked to an analog input reference PAI_Type=PAI(Single_channel) PPID_Ref_Inner_Loop is linked to an analog input reference PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID1_RefTag1	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM1_RefTag1	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
<b>00.05 - Options Outer Loop</b>		
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
OuterLoop_Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
<b>00.06 - Options Inner Loop</b>		
Ref_PV1_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_PV1_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects



Parameter	Visible When	Details
Ref_PV1_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PV1_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
InnerLoop_Bus_Instance	Has_OOAP=True(controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device

**01 - Options**

Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART PV1_IO_Signal_Type= HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock

**03 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the outer loop PV input reference
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**03.01 - Ref PAI Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Inp_PV1	PPID_Ref_Inner_Loop is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the inner loop PV input reference
CV_Out	PPID_Ref_Inner_Loop is linked to an analog input reference	Link to the inner loop CV output reference
Ref_HiHiGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_LoGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate_PV1	PAI_RefTag1 is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

**Additional Sub-Objects for a PPID with Cascade Control Instance**

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

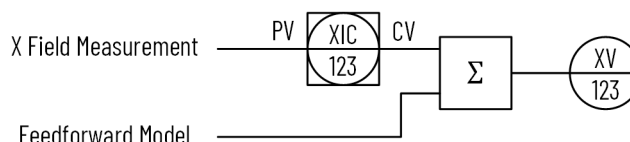
## Process Proportional + Integral + Derivative (PPID) Feedforward Control Strategies

Use the PPID Feedforward control strategy when feedback control (PPID control strategy) alone is not adequate to maintain the process variable at the setpoint. Rather than rely on feedback to make corrective changes to a process only after some load change has driven the process variable away from setpoint, control schemes with feedforward monitor the relevant load(s) and use that information to preemptively make stabilizing changes to the final control element such that the process variable will not be affected.

To scale the CV to align with the associated IO module channel range or to accommodate a fail-open (FO) valve (or air to close) use either of the following options:

- Use a basic PPID with Analog Output control strategy
- Insert a scalar instruction between the PPID CV and the analog output channel reference

The PPID Feedforward control strategies are pre-configured to enable controller actions (CV Action, SP Action and Loop Mode Action) based on various shed conditions (Interlock trip, CV fail, PV fail, and SP fail).

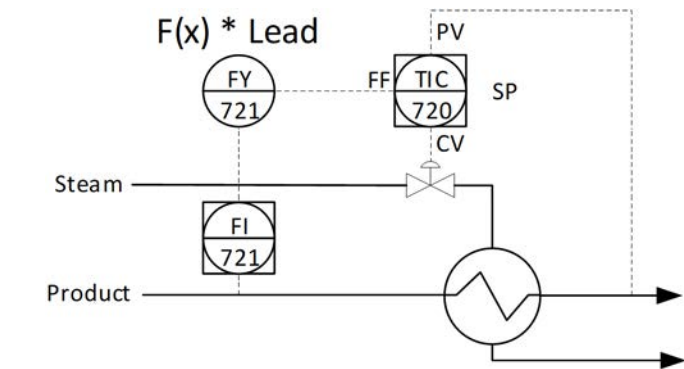


### PPID Feedforward Example

Consider a control system that manipulates steam flow to a heat exchanger to maintain the discharge temperature of the product at a constant setpoint value. The outlet temperature suffers temporary deviations from setpoint if load conditions change. The feedback control system can eventually bring the exiting product's temperature back to setpoint, but it cannot begin corrective action until after a load has driven the product temperature away from the setpoint. To improve control, build both feedforward action and feedback action into the design. The feedforward action lets the control system take corrective action in response to load changes before the process variable is affected.

In this example, the dominant load in the system is product flow rate. Adapting this control system to include feedforward requires installing a product flow transmitter that is characterized to provide feedforward action to the PID controller maintaining temperature. With feedforward control action in place, the steam flow rate immediately changes with product flow rate, preemptively compensating for the increased or decreased heat demand of the product.

The feedforward component of the strategy directly affects the steam valve position in response to product flow. However, the temperature response to the manipulation of the PPID output generally includes a process lag. To overcome the process lag, the feedforward model typically includes a lead function.



When the product flow rate to this heat exchanger suddenly increases, the lead function adds a surge to the feedforward signal, quickly opening the steam valve sending a surge of steam to the exchanger to help overcome the process response lag. The feedforward action is not perfect with this lead function added, but it is substantially improved.

The following PPID control strategies are available as routines in the process library:

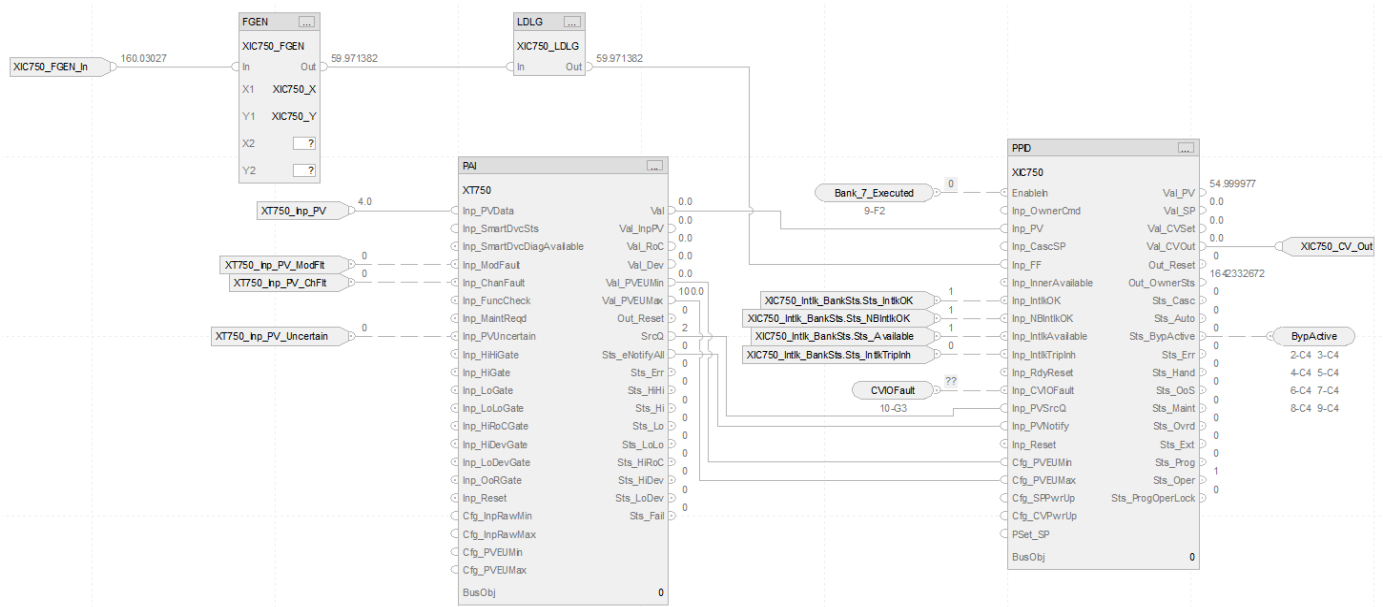
- CS\_PPID\_FF
- CS\_PPID\_FF\_HART
- CS\_PPID\_FF\_EtherNetIP
- CS\_PPID\_FF\_EtherNetIP\_NoHB
- CS\_PPID\_FF\_FF
- CS\_PPID\_FF\_PA

Import the appropriate control strategy as a **routine** in your controller project.

Each PPID Feedforward control strategy contains these sheets:

Sheet	Description
CS_PPID_FF	PPID instruction
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PPID instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors Control Variable faults.

## CS\_PPID\_FF Sheet



### PAI Input References

See [CS\\_PA1 Sheet on page 148](#) for details.

- Substitute XIC750 for the PV data instance of XT101
- Substitute XT750 for the remaining instances of XT101

## PAI Outputs to PPID Inputs

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUmin parameter PV minimum value for scaling from engineering units to %, PV at 0% (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value for scaling from engineering units to %, PV at 100% (PVEU). Valid any float greater than Cfg_PVEUmin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: <div> <div> 0 = Good, live, confirmed good  1 = Good, live, assumed good  2 = Good, no feedback, assumed good  8 = Test, simulated  9 = Test, loopback  10 = Test, manually entered  16 = Uncertain, live, off-spec  17 = Uncertain, substituted at device </div> <div> 18 = Uncertain, substituted at instruction  19 = Uncertain, using last known good  20 = Uncertain, using replacement value  32 = Bad, signal failure  33 = Bad, channel fault  34 = Bad, module/communications fault  35 = Bad, invalid configuration </div> </div>
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status <div> <div> 0 = Not in alarm, acknowledged  1 = Not in alarm, unacknowledged or reset required  2 = Low severity alarm, acknowledged  3 = Low severity alarm, unacknowledged  4 = Medium severity alarm, acknowledged </div> <div> 5 = Medium severity alarm, unacknowledged  6 = High severity alarm, acknowledged  7 = High severity alarm, unacknowledged  8 = Urgent severity alarm, acknowledged  9 = Urgent severity alarm, unacknowledged </div> </div>

## PPID Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XIC750_FGEN_In	FeedForward term (CVEU). Valid any float between -(Cfg_CVEUMax-Cfg_CVEUmin) and (Cfg_CVEUMax-Cfg_CVEUmin). Default is 0.0.
XIC750_Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
XIC750_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status, 1 = All non-bypassable interlocks OK to run
XIC750_Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
XIC750_Intlk_BankSts.Sts_IntlkTripInh	Interlock bank status, 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet
XIC750_SPPwrUp	Loop SP on power-up (PVEU) used when Cfg_PwrUpLM is not 0. The value is clamped to the SP range (Cfg_SPLoLim, Cfg_SPHiLim). Valid any float between Cfg_PVEUmin and Cfg_PVEUMax.
XIC750_CWPwrUp	Loop CV on power-up (CVEU) used when Cfg_PwrUpLM is not 0. Value can be clamped to the configured limits (Cfg_CVLoLim, Cfg_CHiLim) in cascade or auto, and in manual if so configured. Valid any float between Cfg_CVEUmin and Cfg_CVEUMax.
XIC750_PSet_SP	Program setting for SP, loop mode Auto (PVEU). Valid any float. Default is 0.0.

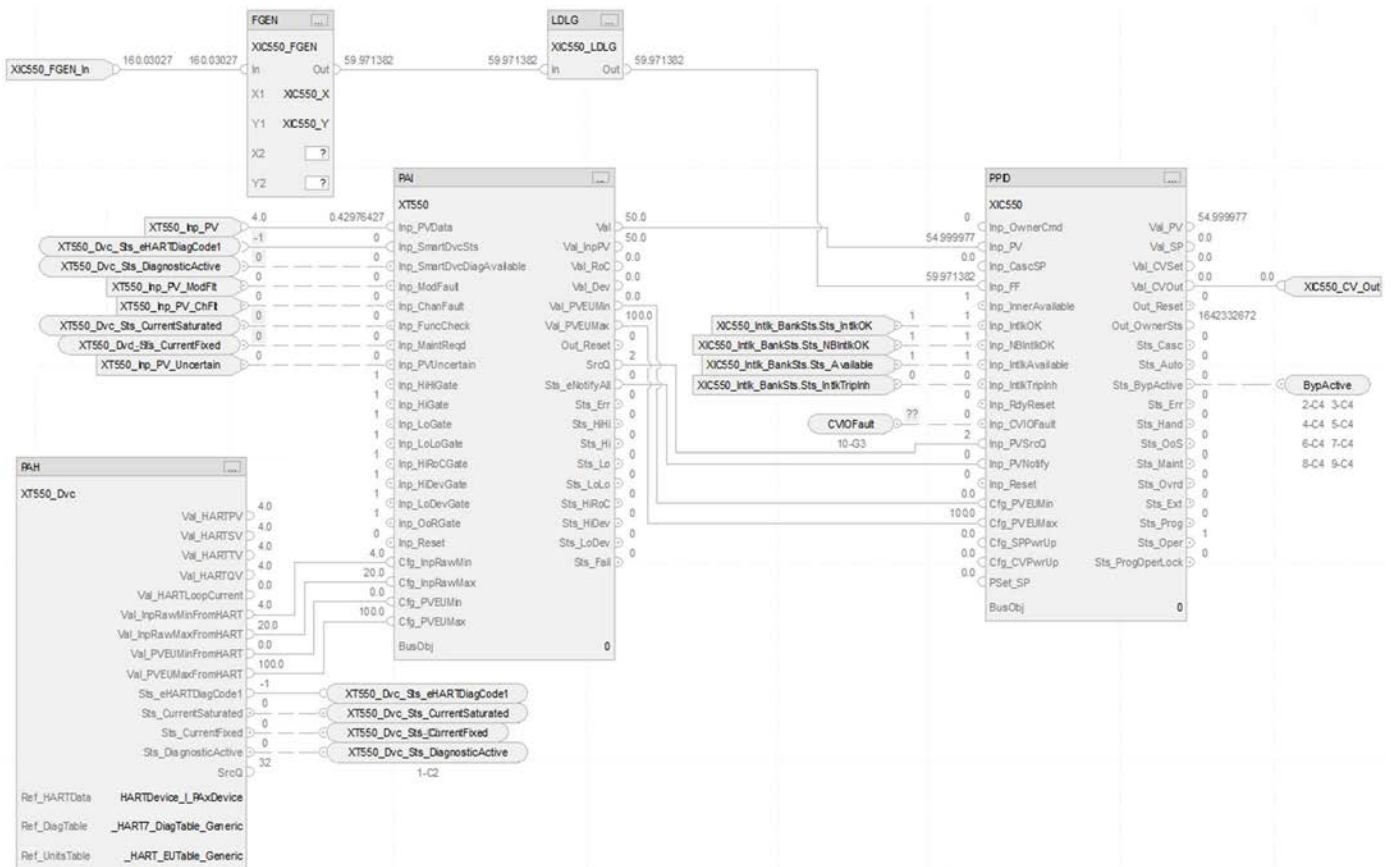
## PPID Output References

Parameter	Description
XIC750_CV_Out	Control Variable output Loop CV after clamping and ramping (CVEU)
BypActive	Output connection to interlock bank sheet

## PPID Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID\_FF HART Sheet

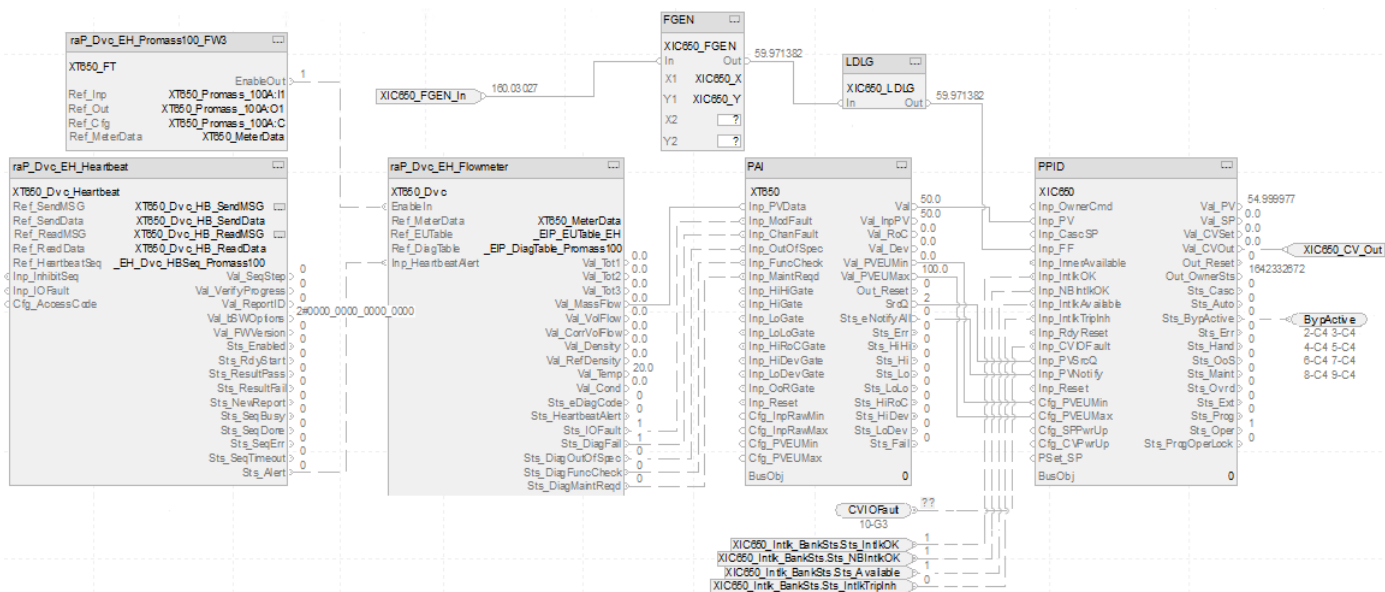


The CS\_PPID\_FF HART sheet operates the same as the CS\_PPID\_FF sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XIC550 for the PV data instance of XT101
- Substitute XT550 for the remaining instances of XT101

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_FF\_EtherNetIP Sheet



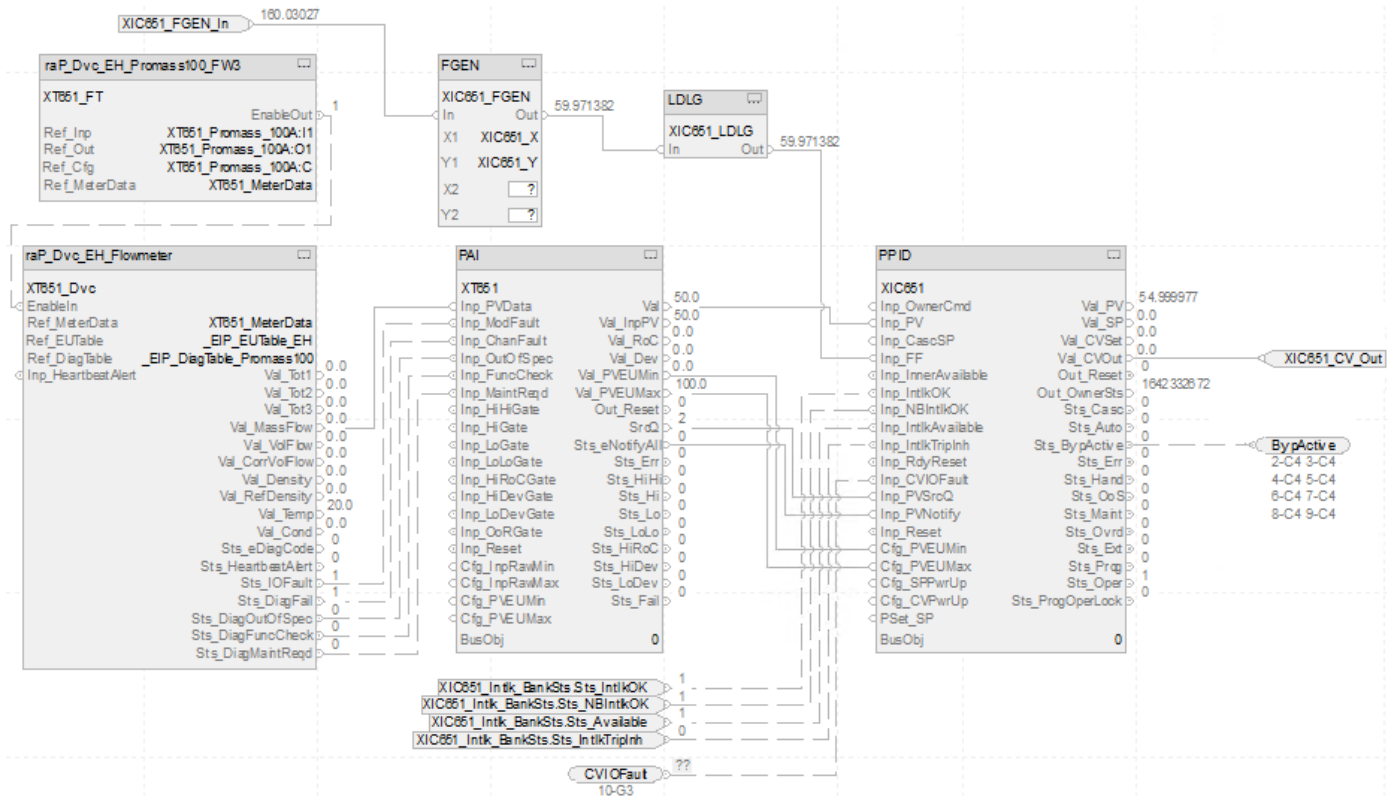
The CS\_PPID\_FF EtherNet/IP™ sheet operates the same as the CS\_PPID\_FF sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XIC650 for the PV data instance of XT101
- Substitute XT650 for the remaining instances of XT101

For more information, see [EtherNet/IP Integration on page 85](#).



## CS\_PPID\_FF\_EtherNetIP\_No HB Sheet

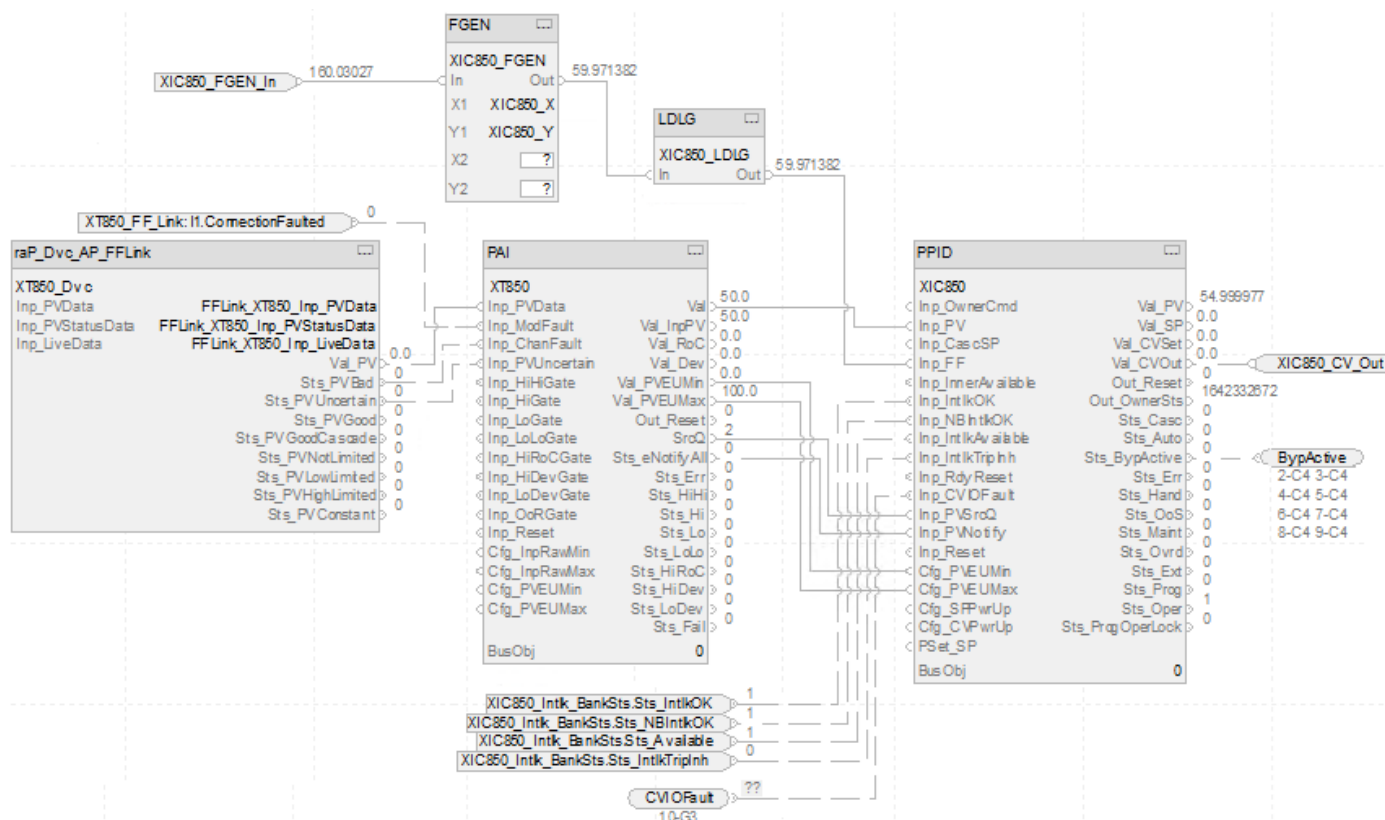


The CS\_PPID\_FF EtherNet/IP NoHB sheet operates the same as the CS\_PPID\_FF sheet but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XIC651 for the PV data instance of XT101
- Substitute XT651 for the remaining instances of XT101

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_FF\_FF

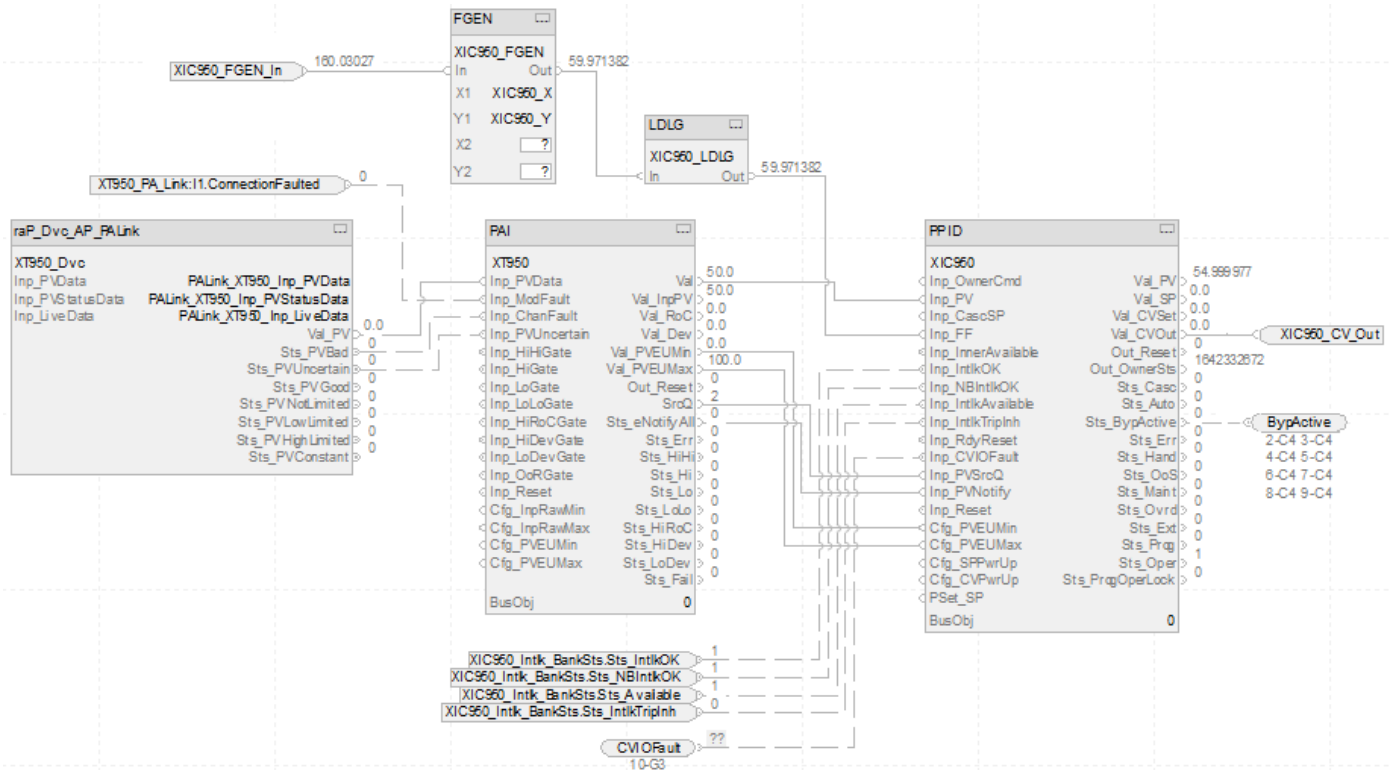


The CS\_PPID\_FF\_FF sheet operates the same as the CS\_PPID\_FF sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XIC850 for the PV data instance of XT101
- Substitute XT850 for the remaining instances of XT101

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_FF\_PA



The CS\_PPID\_FF\_PA sheet operates the same as the CS\_PPID\_FF sheet but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XIC950 for the PV data instance of XT101
- Substitute XT950 for the remaining instances of XT101

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## I/O Faults Sheet



### Fault Input References

Parameter	Description
XICxxx_Out_CV_ChFit	Channel fault, 1 = I/O channel fault or failure, 0 = OK Source: PAI instruction
XICxxx_Out_CV_ModFit	Module fault, 1 = I/O module failure or module communication status bad, 0 = OK Source: PAI instruction

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_FF sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID with Feedforward Control

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the type of analog input via the PAI\_Type parameter.
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter.

## ACM-Based Parameters for a PPID Instance with Feedforward Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PAI_RefTag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the PV input reference
CV_Out	always	Link to the CV output reference
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03.01 - Ref PAI Alarm Configuration</b>		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

**Additional Sub-Objects for a PPID with Feedforward Control Instance**

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Notes:



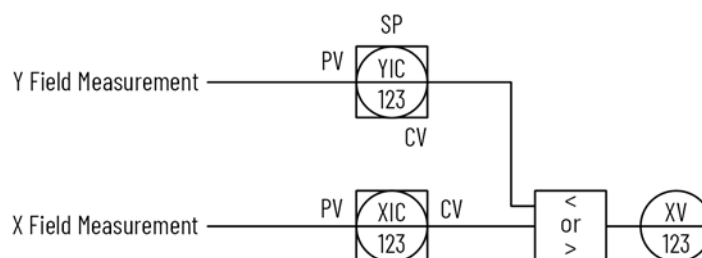


## Process Proportional + Integral + Derivative (PPID) Override Control Strategies

The PPID Override control strategy selectively chooses the output of up to six PPID controllers based on configuration (High Select or Low Select) to drive an analog output device. The output(s) of the 'unselected' PPID controller(s) are kept within  $K_p \cdot \text{Error}$  of the active PPID controller output to help ensure a quick response when another PPID's output becomes the limiting output.

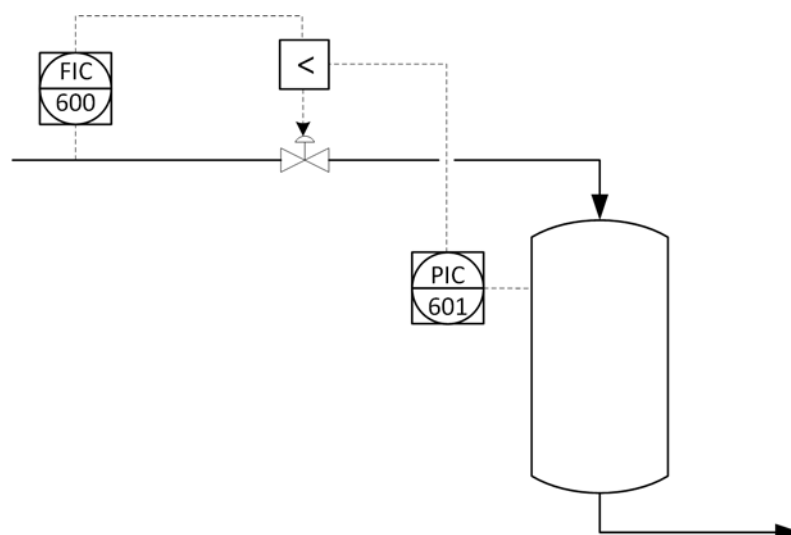
To scale the CV to align with the associated I/O module channel range or to accommodate a fail-open (FO) valve (or air to close) use either of the following options:

- Use a basic PPID with Analog Output control strategy
- Insert a scalar instruction between the PPID CV and the analog output channel reference



### PPID Override Example

In this example, the primary control maintains a desired flow of product (FIC-600) into the vessel. To maintain the vessel integrity, it is desired to keep the vessel pressure below a set value. This is accomplished by using the override control strategy where the vessel pressure controller (PIC-601) restricts the valve opening if the vessel pressure exceeds the set value. The lower of the two CVs is selected to drive the final control element (FV-600).



These PPID Override control strategies (consisting of multiple routines) are available in the process library:

Control Strategy	Routines
CS_PPID_OVERRIDE	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>HILO790</li> <li>XC790</li> <li>XIC790</li> <li>XIC800</li> </ul> </li> </ul>
CS_PPID_OVERRIDE_HART	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE_HART               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>HILO590</li> <li>XC590</li> <li>XIC590</li> <li>XIC600</li> </ul> </li> </ul>
CS_PPID_OVERRIDE_EtherNetIP	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE_EtherNetIP               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>HILO690</li> <li>Interlocks</li> <li>XC690</li> <li>XIC1690</li> <li>XIC690</li> </ul> </li> </ul>
CS_PPID_OVERRIDE_EtherNetIP_NoHB	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE_EtherNetIP_NoHB               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>HILO691</li> <li>Interlocks</li> <li>XC691</li> <li>XIC1691</li> <li>XIC691</li> </ul> </li> </ul>
CS_PPID_OVERRIDE_FF	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE_FF               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>FFLinkMap</li> <li>HILO890</li> <li>Interlocks</li> <li>XC890</li> <li>XIC1890</li> <li>XIC890</li> </ul> </li> </ul>
CS_PPID_OVERRIDE_PA	<ul style="list-style-type: none"> <li>CS_PPID_OVERRIDE_PA               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>HILO990</li> <li>Interlocks</li> <li>PALinkMap</li> <li>XC990</li> <li>XIC1990</li> <li>XIC990</li> </ul> </li> </ul>

Import the **routines** for the appropriate control strategy in your controller project. Each control strategy contains multiple routines; each routine contains multiple Function Block sheets.

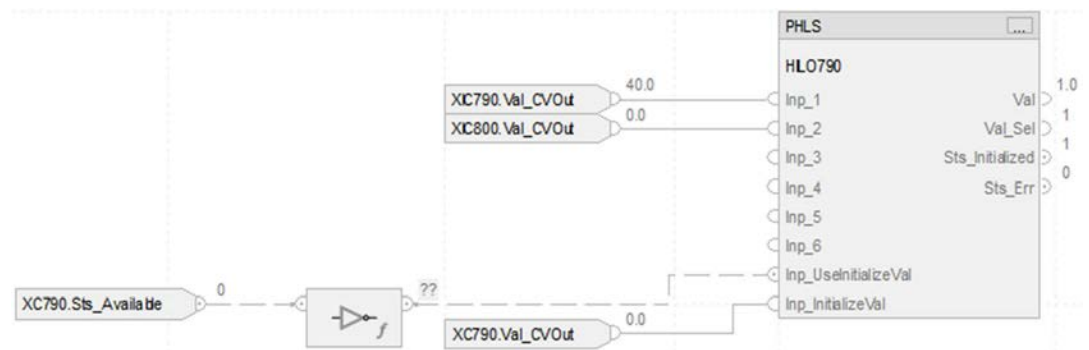
Each PPID Override control strategy contains these Routines:

- Override Low Select (the PHLS can be configured as High or Low Select)
- Process Analog Output
- Process Analog Input to Process PID (two instances)

# ROUTINE Override Low Select

Sheet	Description
CS_PHLS	Process High or Low Selector instruction <ul style="list-style-type: none"><li>• HILO790 analog</li><li>• HILO590 HART</li><li>• HILO690 EtherNet/IP™</li><li>• HILO691 EtherNet/IP with no heartbeat</li><li>• HILO890 FOUNDATION Fieldbus</li><li>• HILO990 Profibus PA</li></ul>

## CS\_PHLS Sheet



The control strategy, as supplied, uses only two PID control loops. The control strategy can support up to six PID control loops by exposing additional inputs of the PHLS instruction.

Option	Inp_1	Inp_2
Analog	XIC790	XIC800
HART	XIC590	XIC600
EtherNet/IP	XIC690	XIC1690
EtherNet/IP with no heartbeat	XIC691	XIC1691
FOUNDATION Fieldbus	XIC890	XIC1890
Profibus PA	XIC990	XIC1990

The control strategy uses a subset of the PHLS control strategy. See [Process High or Low Selector \(PHLS\) Control Strategies on page 241](#) for more details.

### PHLS Input References

Parameter	Description
XIC790_Val_CVOut	Control Variable output 1 Source: Val_CVOut from PID loop 1 (XIC790)
XIC800_Val_CVOut	Control Variable output 2 Source: Val_CVOut from PID loop 1 (XIC800)
XC790_Sts_Available	Status available of downstream PAO, 1= Instruction is initialized
XC790_Val_CVOut	Control Variable output as initialization value from downstream block.

### PHLS Output Reference

Parameter	Description
HILO790_Val	Control Variable output (selected minimum or maximum) for downstream block Destination: PAO input (PSet_CV)

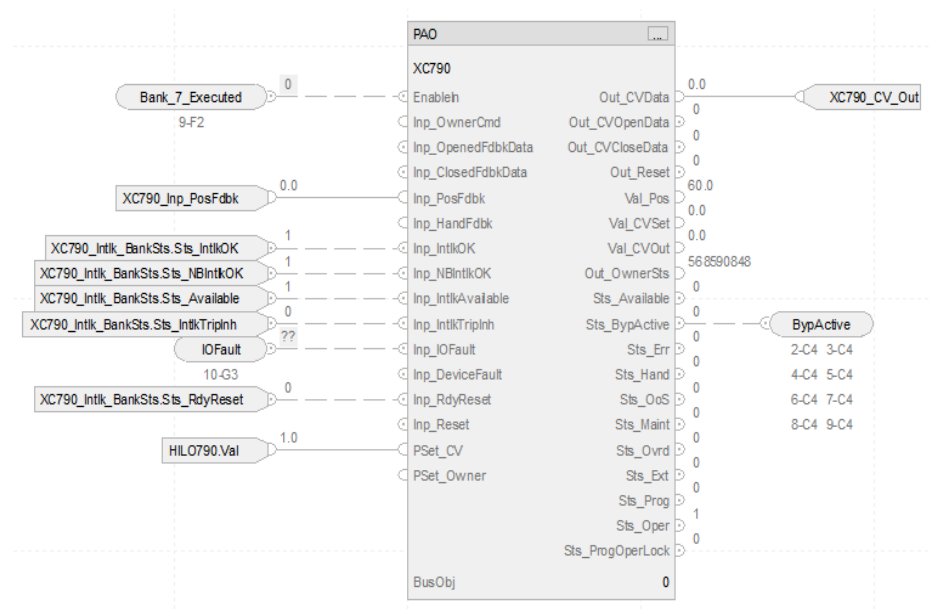
### PHLS Configuration Considerations

Operand	Type	Description
PlantPax® control	P_HIGH_LOW_SELECT	Data structure required for proper operation of instruction

## ROUTINE Process Analog Output

Sheet	Description
CS_PAO	Process High or Low Selector instruction <ul style="list-style-type: none"> <li>XC790 analog</li> <li>XC590 HART</li> <li>XC690 EtherNet/IP</li> <li>XC691EtherNet/IP with no heartbeat</li> <li>XC890 FOUNDATION Fieldbus</li> <li>XC990 Profibus PA</li> </ul>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors one analog output channel for I/O fault input and raises an alarm on an I/O fault.

### CS\_PAO Sheet



### Input References to PAO and PAO Output References

For details, see the [CS\\_PAO Sheet on page 180](#).

Substitute:

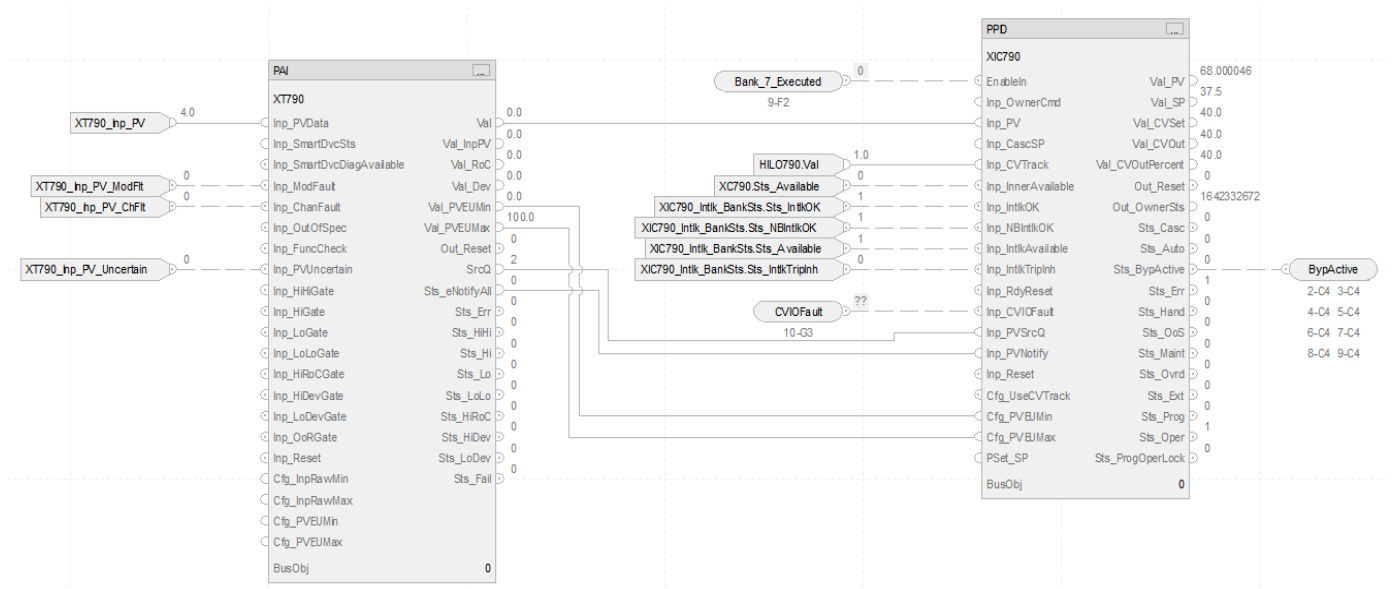
Input	XC101 =	XC100 =
Analog	XC790	HILO790
HART	XC590	HILO590
EtherNet/IP	XC690	HILO690
EtherNet/IP with no heartbeat	XC691	HILO691
FOUNDATION Fieldbus	XC890	HILO890
Profibus PA	XC990	HILO990

ROUTINE: Process Analog Input to Process PID

There are two routines; each routine contains these sheets.

Sheet	Description
CS_PPID	Process PID instruction with override <ul style="list-style-type: none"><li>• XIC790/XIC800 analog</li><li>• XIC590/XIC600 HART</li><li>• XIC690/XIC1690 EtherNet/IP</li><li>• XIC690/XIC1691 EtherNet/IP with no heartbeat</li><li>• XIC890/XIC1890 FOUNDATION Fieldbus</li><li>• XIC990/XIC1900 Profibus PA</li></ul>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PPID instruction monitors interlock conditions that cause output CV and SP to shed. CV shed can be configured to hold the last good CV value or to use the configured safe value. SP is shed to current PV. There are 8 interlock bank sheets; each bank exposes 16 interlocks but supports as many as 32 interlocks. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors Control Variable faults.

CS\_PPID Sheet



PAI Input References

For details, see [CS\\_PAI Sheet on page 148](#).

- Substitute XIC790 for the first instance of XT101
- Substitute XIC800 for the second instance of XT101

*PAI Outputs to PPID Inputs*

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUmin parameter PV minimum value in engineering units (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value in engineering units (PVEU). Valid any float greater than Cfg_PVEUmin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

*PPID Input References*

Parameter	Description
HIL0790_Val HIL0800_Val	Control Variable output CV to track if Cfg_UseCVTrack = 1 or if Inp_InnerAvailable = 0 (CVEU). Valid any float. Source: PIC control loop
XC790_Sts_Available	Status available
XC800_Sts_Available	1 = Inner loop is available 0 = Inner loop is not available
XIC790_Intlk_BankSts.Sts_IntlkOK XIC800_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XIC790_Intlk_BankSts.Sts_NBIntlkOK XIC800_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XIC790_Intlk_BankSts.Sts_Available XIC800_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XIC790_Intlk_BankSts.Sts_IntlkTriplnh XIC800_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet

*PPID Output References*

Parameter	Description
XIC790_Val_CVOut	Control Variable output Loop CV after clamping and ramping (CVEU) Destination: Analog output channel or downstream REAL tag
BypActive	Output connection to interlock bank sheet

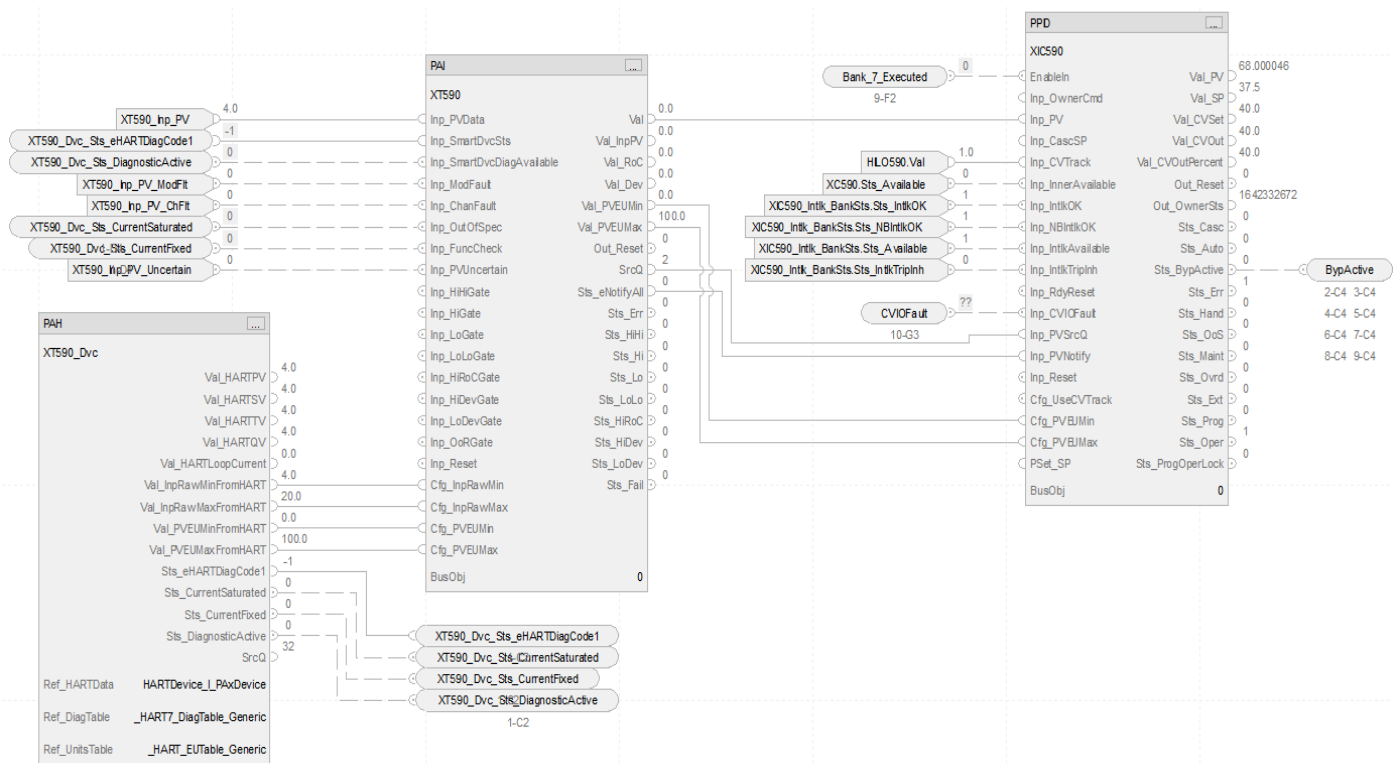
PPID Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"><li>0 if not using organization</li><li>Bus[x].Obj when using organization</li></ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

Override requires these additional configuration operands for each PPID:

Parameter	Description
Cfg_UseCVTrack	For each PID control loop, set Cfg_UseCVTrack=1 1= Use Inp_CVTrack reset feedback in tracking, for example, if PPID output is significantly faster than actuator or inner loop or in override select control
Cfg_CVTrackGain	For each PID control loop, set Cfg_CVTrackGain=Cfg_IGain (Tt=Ti) Tracking gain Kt (1/minute) for independent or tracking time constant Tt (minutes) for dependent gains for CV to track Inp_CVTrack if Cfg_UseCVTrack = 1.

CS\_PPID HART Sheet



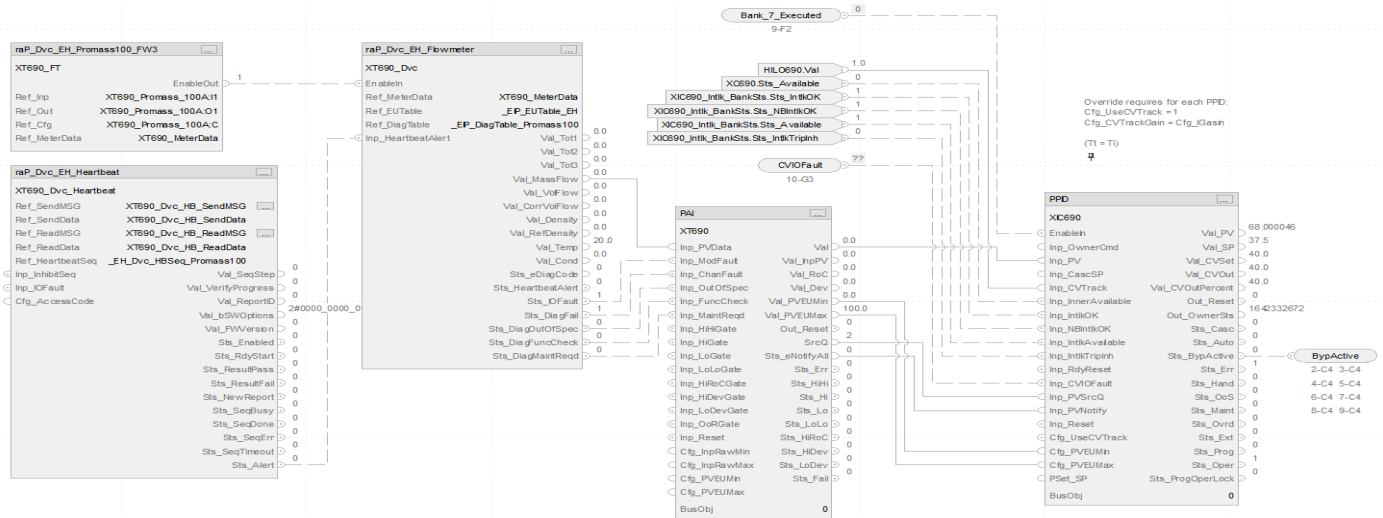
The CS\_PPID HART sheet operates the same as the CS\_PPID sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XIC590 for the first instance of XT100
- Substitute XIC600 for the second instance of XT100

For more information, see [HART Integration on page 61](#).



## CS\_PPID\_EtherNetIP Sheet

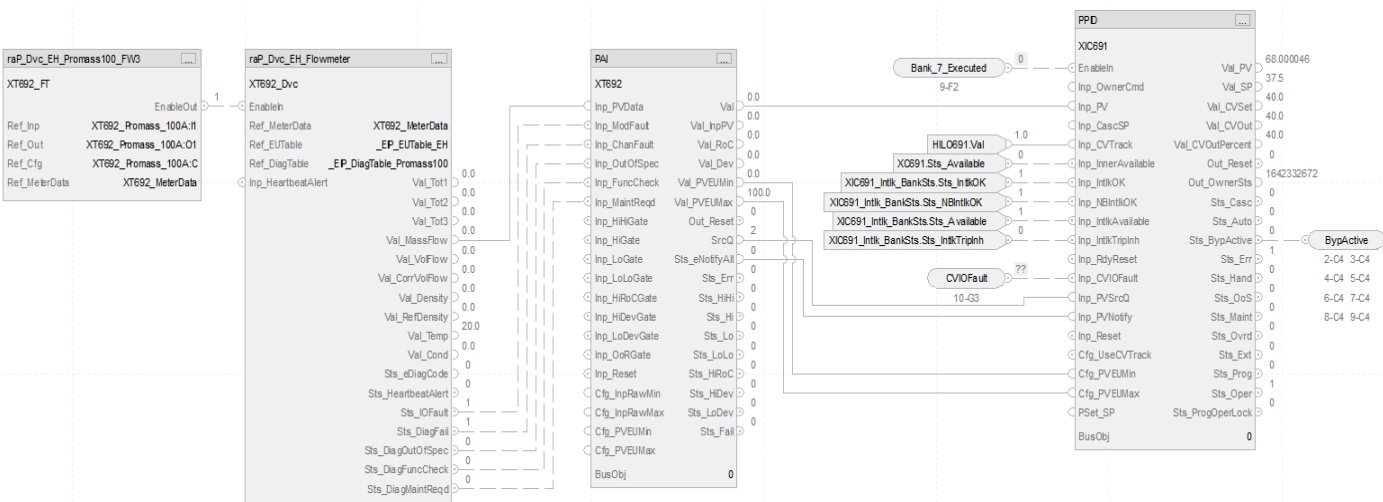


The CS\_PPID EtherNet/IP sheet operates the same as the CS\_PPID sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XIC690 for the first instance of XT100
- Substitute XIC1690 for the second instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_EtherNetIP\_NoHB Sheet

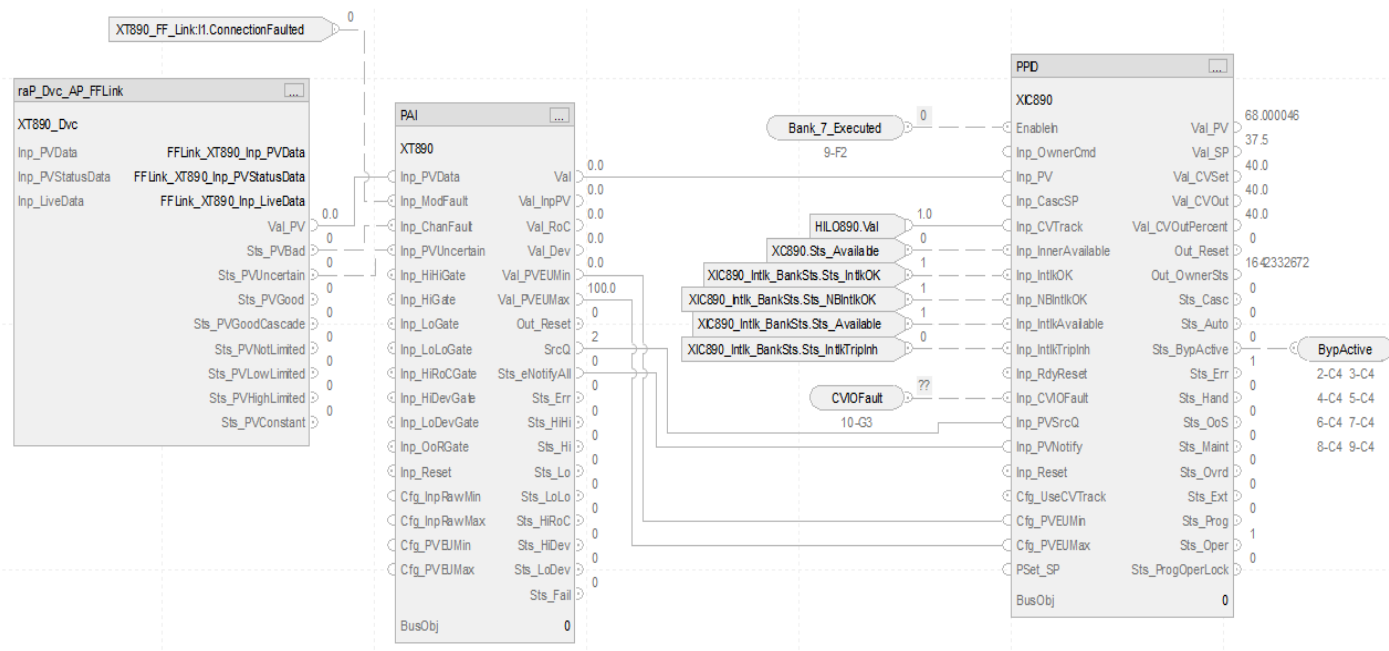


The CS\_PPID EtherNetIP\_NoHB sheet operates the same as the CS\_PPID sheet but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XIC691 for the first instance of XT100
- Substitute XIC1691 for the second instance of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_FF Sheet

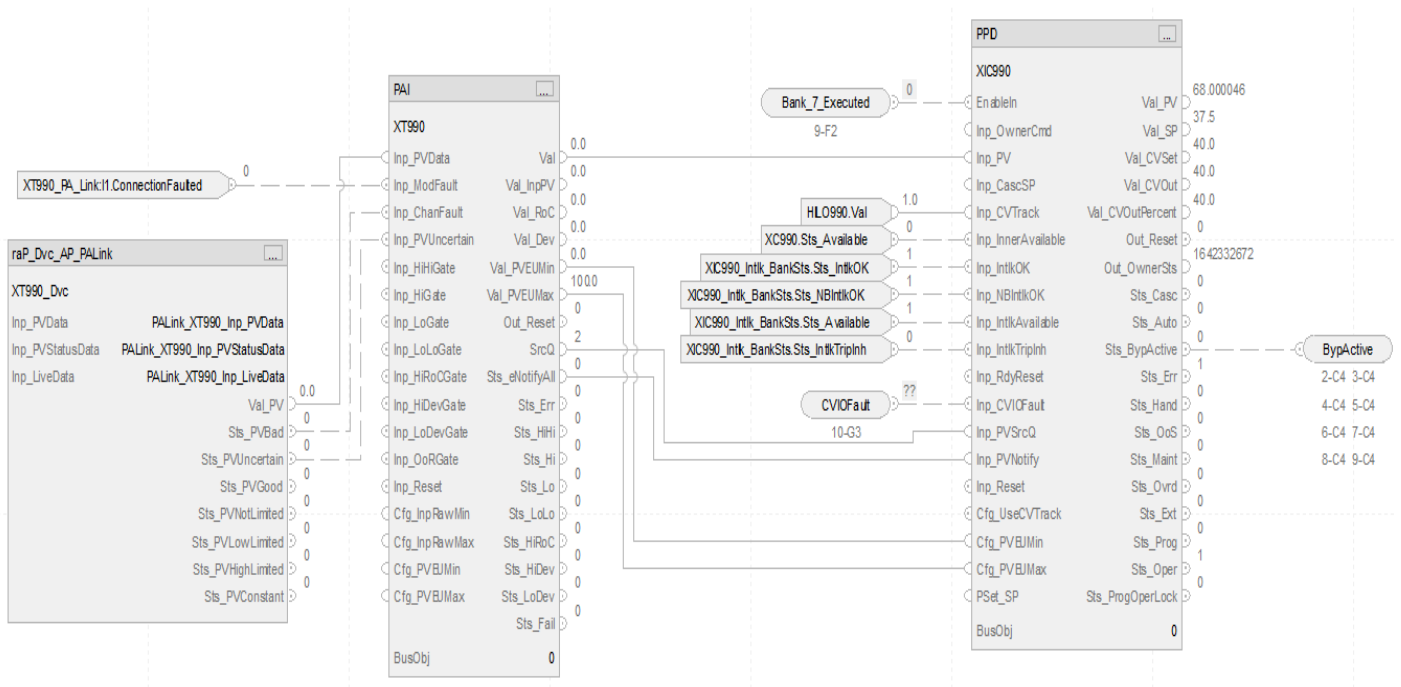


The CS\_PPID\_FF sheet operates the same as the CS\_PPID sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XIC890 for the first instance of XT100
- Substitute XIC1890 for the second instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_PA Sheet



The CS\_PPID\_PA sheet operates the same as the CS\_PPID sheet but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XIC990 for the first instance of XT100
- Substitute XIC1990 for the second instance of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

# 10 Faults Sheet



### Faults Input References

Parameter	Description
XICxxx_Out_CV_ChanFault	Control Variable output channel fault
XICxxx_Out_CV_ModFault	Control Variable output data module fault

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_OVERRIDE sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID with Override Control

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the number of PPID controllers via the Cfg\_PPIDSelecton parameter
- Specify the type of analog input via the PAI\_Type
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for a PPID Instance with Override Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_PPIDSelection	always	Specify the number of PPID controllers.
Use_00AP	Has_00AP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_00AP=True?	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36?</a>
<b>00.00.00 - Selection</b>		
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAI_RefTag	always	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
<b>00.00.0x - Selection</b> Where x = 1...5		
PAIx_Type	Cfg_PPIDSelection=(x+1)	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAIx_RefTag	Cfg_PPIDSelection=(x+1) PAI_Type=PAI(Single_channel)	Link to the analog input reference
PAIDx_RefTag	Cfg_PPIDSelection=(x+1) PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIMx_RefTag	Cfg_PPIDSelection=(x+1) PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
IO_Signal_Type_x	Cfg_PPIDSelection=(x+1) PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
<b>01.00 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance_PPID	Has_00AP=True (controller parameter) Use_00AP=True	Link to a bus array instance for the PPID. This should be unique for each device
Ref_EtherNetIPModule	IO_Signal_Type =EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

Parameter	Visible When	Details
<b>01.0x - IO Configuration</b>		
Where x = 1...5 Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Ref_HartDevicex	Cfg_PPIDSelection=(x+1) IO_Signal_Type_x=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModulex	Cfg_PPIDSelection=(x+1) IO_Signal_Type_x=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Modulex	Cfg_PPIDSelection=(x+1) IO_Signal_Type_x=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Modulex	Cfg_PPIDSelection=(x+1) IO_Signal_Type_x=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PHLS_RefTag	always	Link to the PHLS reference
PA0_RefTag	always	Link to the PA0 reference
Bus_Instance_PA0	Has_00AP=True (controller parameter) Use_00AP=True	Link to a bus array instance for the PA0. This should be unique for each device
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
CV_Out	always	Link to the CV output reference
Inp_PosFdk	always	Link to the input position feedback reference
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
Inp_PV	always	Link to the PV input reference
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference	Link to the gate reference
<b>03.0x - IO Configuration</b>		
Where x = 1...5 Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PPIDx_RefTag	Cfg_PPIDSelection=(x+1)	
Cfg_HasCVxIntlkObj	Cfg_PPIDSelection=(x+1)	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasCVxIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance_PPIDx	Cfg_PPIDSelection=(x+1) Has_00AP=True (controller parameter) Use_00AP=True	Link to a bus array instance for the PPID. This should be unique for each device

Parameter	Visible When	Details
Inp_PVx	Cfg_PPIDSelection=(x+1)	Link to the PV input reference
Ref_HiHiGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_HiGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_LoGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_LoLoGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_HiRoCGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_HiDevGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_LoDevGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference
Ref_0oRGate_PVx	PAI_RefTag is linked to an analog input reference Cfg_PPIDSelection=(x+1)	Link to the gate reference

#### 04 - Alarm Configuration

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

## Additional Sub-Objects for a PPID with Override Control Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
CVIntlk CV1Intlk CV2Intlk CV3Intlk	Configure an interlock for the CV instance See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CVEvents	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>
PPID2Events PPID3Events PPID4Events PPID5Events PPID6Events	Configure an event to monitor for the additional PPID instance See <a href="#">Event Logging on page 49</a>



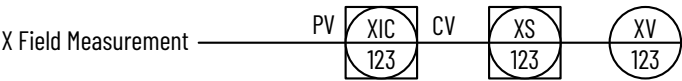


# Process Proportional + Integral + Derivative (PPID) Basic PPID with Process Analog Output (PAO) Control Strategies

This control strategy differs from the Basic PPID control strategy in that a PAO instruction is inserted between the output of the PPID and the reference signal to the analog output channel.

This strategy provides the capability to pulse outputs (pulse open and pulse close) to position a valve to the reference signal provided by the PPID CV.

You can also use this control strategy instead of the Basic PPID control strategy to scale the output to accommodate fail-open (FO) valves (or air to close). Alternatively, you can use a scalar instruction in place of the PAO to accommodate fail-open valves.



## PPID with PAO Control Strategies

Control Strategy	Routine
CS_PPID_PAO	<ul style="list-style-type: none"><li>CS_PPID_PAO<ul style="list-style-type: none"><li>Parameters and Local Tags</li><li>MainRoutine</li><li>XC730</li><li>XIC730</li></ul></li></ul>
CS_PPID_PAO_EtherNetIP	<ul style="list-style-type: none"><li>CS_PPID_PAO_EtherNetIP<ul style="list-style-type: none"><li>Parameters and Local Tags</li><li>MainRoutine</li><li>Interlocks</li><li>XC630</li><li>XIC630</li></ul></li></ul>
CS_PPID_PAO_EtherNetIP_NoHB	<ul style="list-style-type: none"><li>CS_PPID_PAO_EtherNetIP_NoHB<ul style="list-style-type: none"><li>Parameters and Local Tags</li><li>MainRoutine</li><li>Interlocks</li><li>XC631</li><li>XIC631</li></ul></li></ul>

## PPID with PAO Control Strategies

Control Strategy	Routine
CS_PPID_PAO_FF	<ul style="list-style-type: none"> <li>CS_PPID_PAO_FF               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>FFLinkMap</li> <li>Interlocks</li> <li>XC830</li> <li>XIC830</li> </ul> </li> </ul>
CS_PPID_PAO_PA	<ul style="list-style-type: none"> <li>CS_PPID_PAO_PA               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>PALinkMap</li> <li>XC930</li> <li>XIC930</li> </ul> </li> </ul>

## CS\_PPID HART Options

HART Option	Description	Routine
CS_PPID_PAO_HART	HART input to PAO (XC731)	<ul style="list-style-type: none"> <li>CS_PPID_PAO_HART               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>XC731</li> <li>XIC731</li> </ul> </li> </ul>
CS_PPID_HART_PAO	HART input to PAI (XIC530)	<ul style="list-style-type: none"> <li>CS_PPID_HART_PAO               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>XC530</li> <li>XIC530</li> </ul> </li> </ul>
CS_PPID_HART_PAO_HART	HART input to PAO (XC531) and HART input to PAI (XIC531)	<ul style="list-style-type: none"> <li>CS_PPID_HART_PAO_HART               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>XC531</li> <li>XIC531</li> </ul> </li> </ul>

Import the **routines** for the appropriate control strategy in your controller project. Each control strategy contains multiple routines; each routine contains multiple Function Block sheets.

Each PPID with PAO control strategy contains these routines:

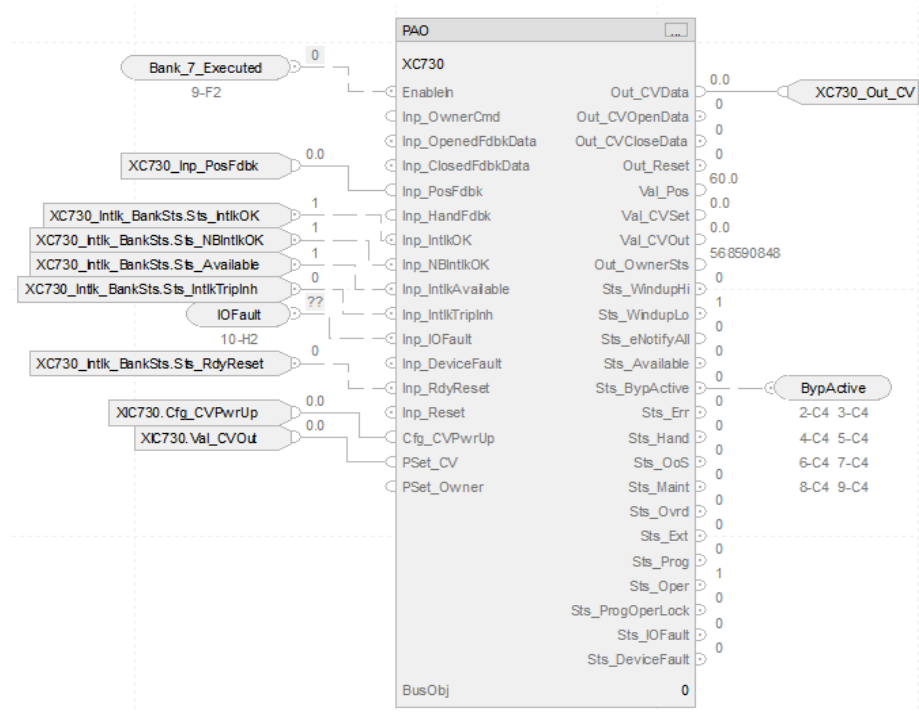
- PAO
- PPID

## ROUTINE PPID\_PAO

Each routine contains these sheets.

Sheet	Description
CS_PPID_PAO	Process analog output <ul style="list-style-type: none"> <li>• XC730 analog</li> <li>• XC531 and XC731 HART</li> <li>• XC630 EtherNet/IP™</li> <li>• XC631EtherNet/IP with no heartbeat</li> <li>• XC830 FOUNDATION Fieldbus</li> <li>• XC930 Profibus PA</li> </ul>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors Control Variable faults.

### CS\_PPID\_PAO Sheet (XC730)



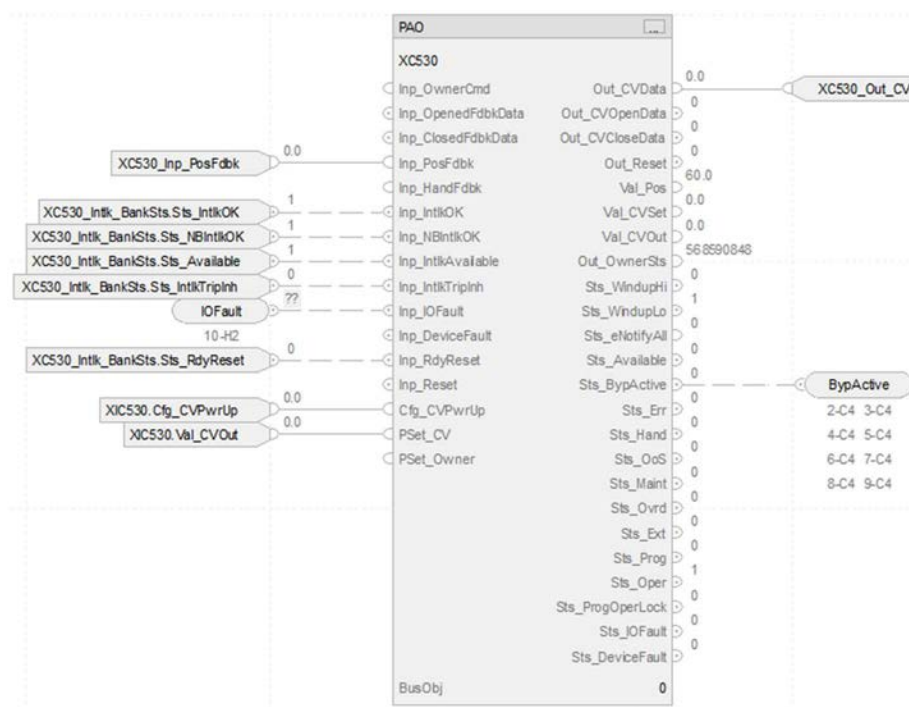
#### PAO Input References

For details, see the [CS\\_PAO Sheet on page 180](#). Substitute XC730 for XC100.

#### PAO Output References

Parameter	Description
XIC730_Out_CV	Control Variable output for PPID instructions CV output in raw (I/O Card) units. Extended properties of this member: Engineering Unit - Raw units (text) used for the analog output
BypActive	Output connection to interlock bank sheet

## CS\_PPID\_PAO HART Sheet (XC530)



The CS\_PPID\_PAO\_HART sheet operates the same as the CS\_PAO sheet but relies on HART input data.

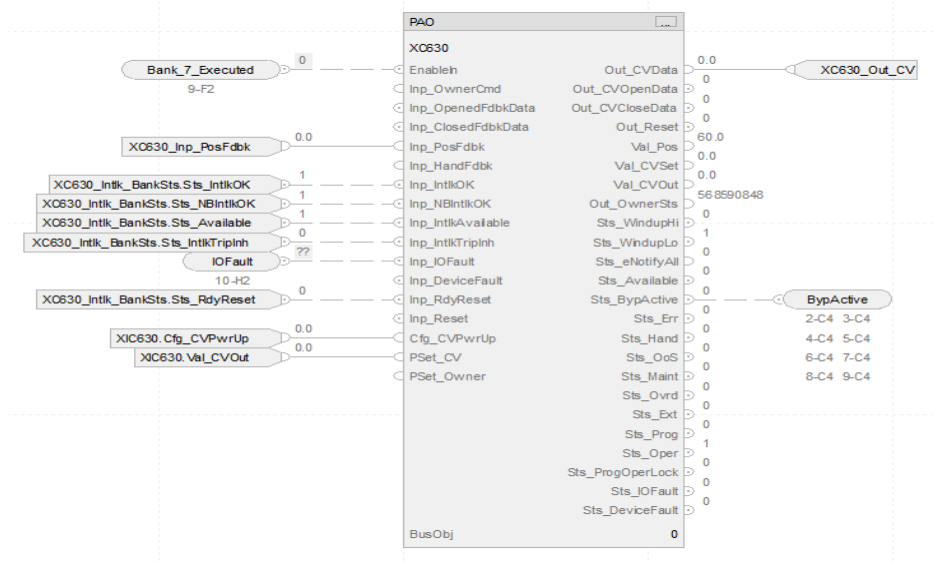
- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XC530 for XT100

For the CS\_HART\_PAO\_HART control strategy:

- For information on PAH outputs to PAO inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XC531 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_PAO EtherNet/IP Sheet (XC630)

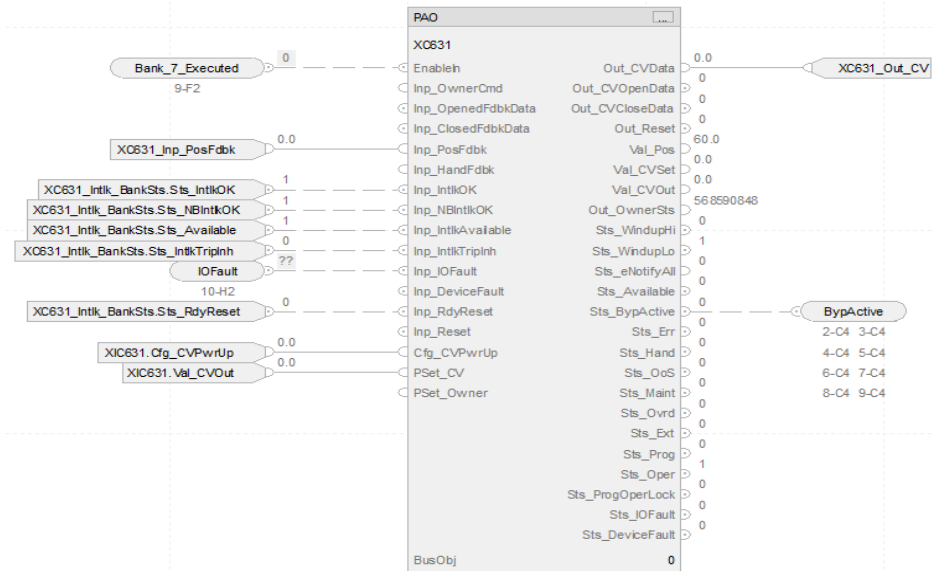


The CS\_PPID\_PAO EtherNet/IP sheet operates the same as the CS\_PAO sheet but relies on EtherNet/IP input data.

- For information on PAO inputs, see [CS\\_PAO Sheet on page 180](#).
- Substitute XC630 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_PAO EtherNet/IP NoHB Sheet (XC631)

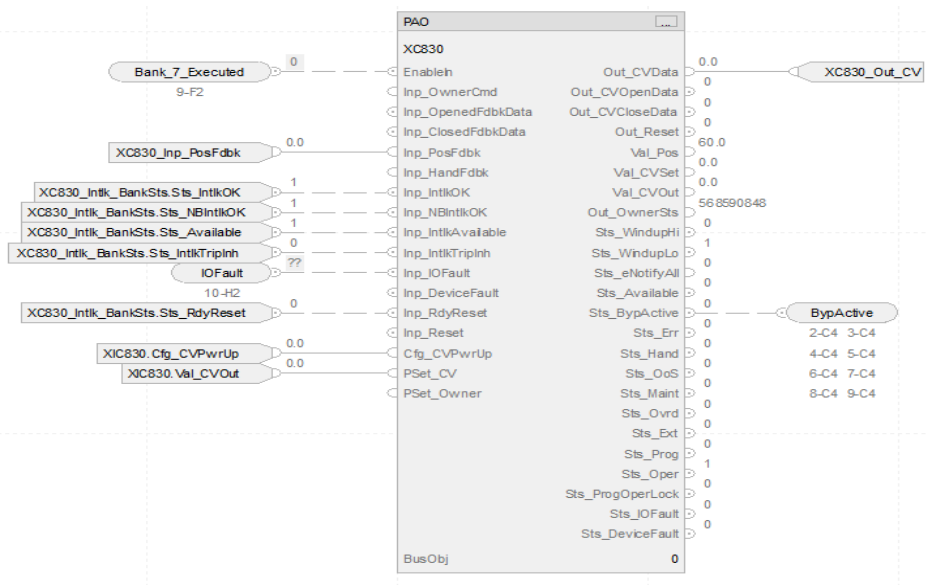


The CS\_PPID\_PAO EtherNet/IP NoHB sheet operates the same as the CS\_PAO sheet but relies on EtherNet/IP input data with no heartbeat.

- For information on PAO inputs, see [CS\\_PAO Sheet on page 180](#).
- Substitute XC631 for XC100

For more information, see [EtherNet/IP Integration on page 85](#).

CS\_PPID\_PAO FF Sheet (XC830)

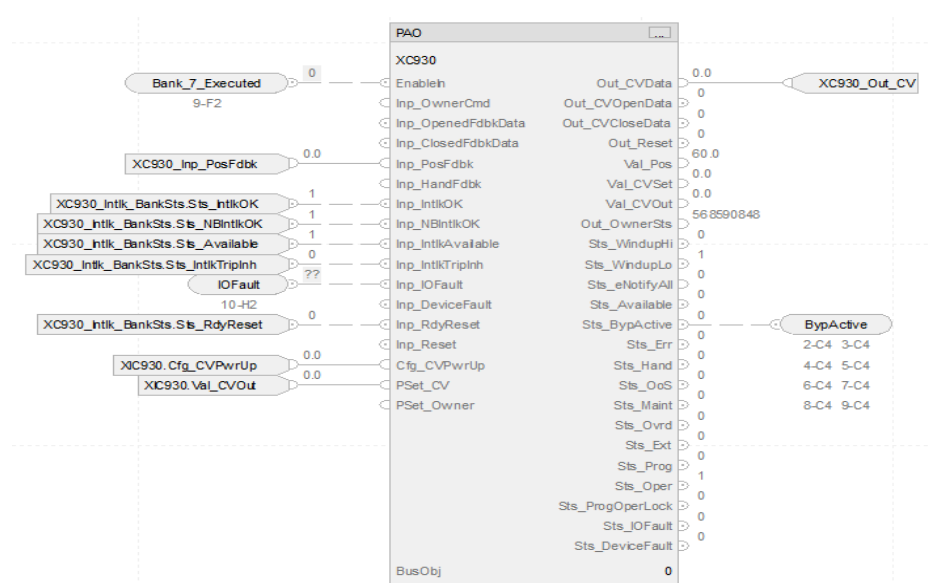


The CS\_PPID\_PAO FF sheet operates the same as the CS\_PAO sheet but relies on FOUNDATION Fieldbus input data.

- For information on PAO inputs, see [CS\\_PAO Sheet on page 180](#).
- Substitute XC830 for XC100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

CS\_PPID\_PAO PA Sheet (XC930)

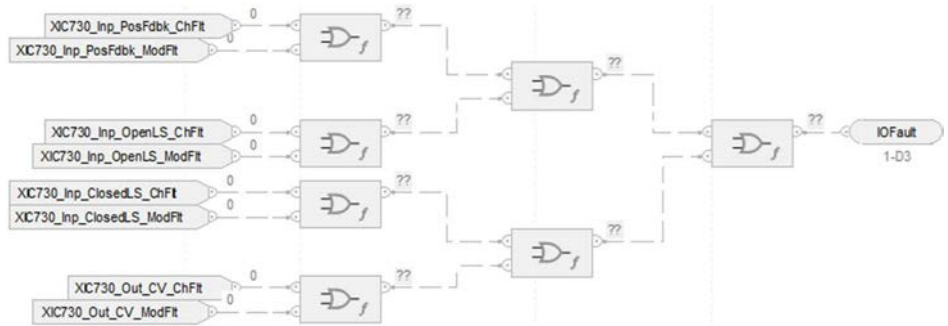


The CS\_PPID\_PAO PA sheet operates the same as the CS\_PAO sheet but relies on Profibus PA input data.

- For information on PAO inputs, see [CS\\_PAO Sheet on page 180](#).
- Substitute XC930 for XC100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## IO Faults Sheet



### Faults Input References

Parameter	Description
XICxxx_Inp_PosFdbk_ChFlt	Tieback Input Channel Fault Source: PAO instruction
XICxxx_Inp_PosFdbk_ModFlt	1 = This or parent I/O communication faulted (module not in Running state). Use this parameter to trigger device I/O Fault action. Source: PAO instruction
XICxxx_Inp_OpenLS_ChFlt	Opened Feedback Input Channel Fault Source: PAO instruction
XICxxx_Inp_OpenLS_ModFlt	1 = This or parent I/O communication faulted (module not in Running state). Use this parameter to trigger device I/O Fault action. Source: PAO instruction
XICxxx_Inp_ClosedLS_ChFlt	Closed Feedback Input Channel Fault Source: PAO instruction
XICxxx_Inp_ClosedLS_ModFlt	1 = This or parent I/O communication faulted (module not in Running state). Use this parameter to trigger device I/O Fault action. Source: PAO instruction
XICxxx_Out_CV_ChFlt	Control Variable Output Channel Fault 1 = I/O channel fault or failure 0 = OK Source: PAO instruction
XICxxx_Out_CV_ModFlt	1 = This or parent I/O communication faulted (module not in Running state). Use this parameter to trigger device I/O Fault action. Source: PAO instruction

### Fault Output Reference

Parameter	Description
IOFault	Output connection to CS_PAO sheet

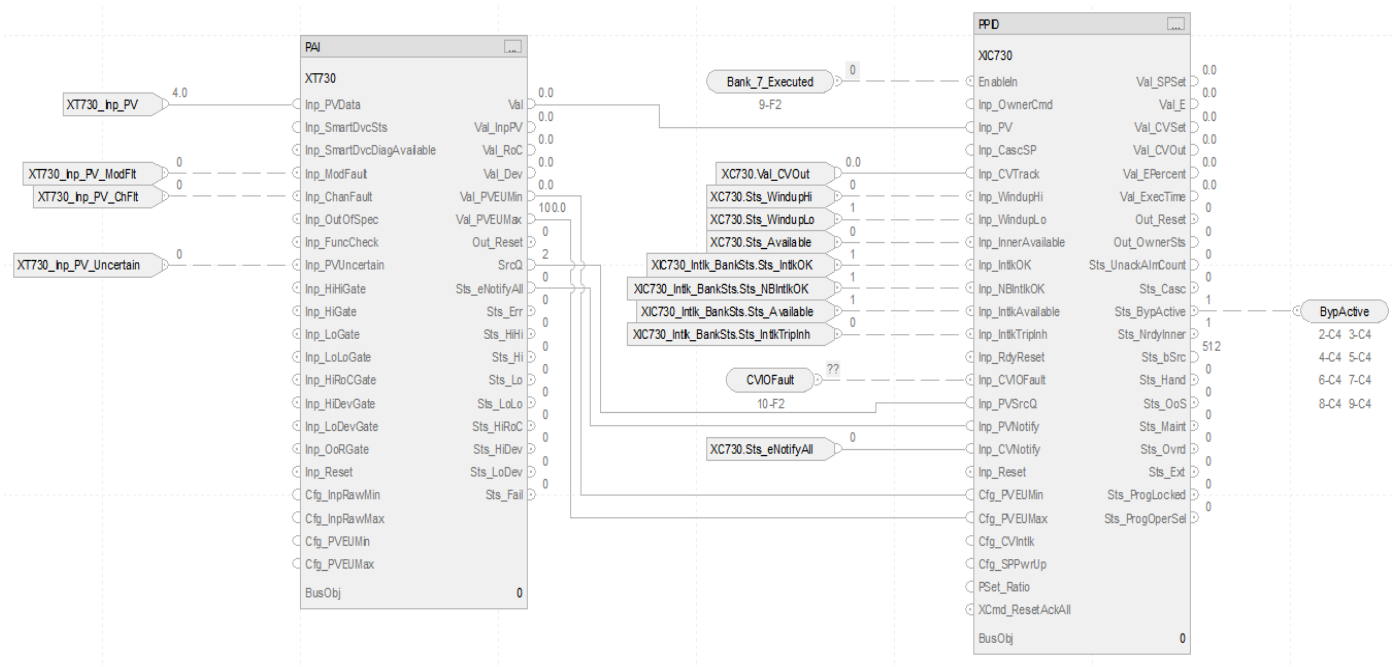
For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

ROUTINE PPID\_PAO\_PID

Each routine contains these sheets.

Sheet	Description
CS_PPID_PAO_PID	Process analog output <ul style="list-style-type: none"><li>• XIC730 analog</li><li>• XIC530 and XIC531 HART</li><li>• XIC630 EtherNet/IP</li><li>• XIC631EtherNet/IP with no heartbeat</li><li>• XIC830 FOUNDATION Fieldbus</li><li>• XIC930 Profibus PA</li></ul>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors Control Variable faults.

CS\_PPID\_PAO\_PID Sheet (XIC730)



PAI Input References

For details, see [CS\\_PAI Sheet on page 148](#). Substitute XT730 for XT101



*PAI Outputs to PPID Inputs*

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

*PPID Input References*

Parameter	Description
XC730.Val_CVout	Loop CV after clamping and ramping (CVEU). Source: outer loop
XC730.Sts_WindupHi	Windup high signal. When true, the CV cannot integrate in a positive direction. The signal is typically obtained from the Windup hi output from an inner loop. Default is false.
XC730.Sts_WindupLo	Windup low signal. When true, the CV cannot integrate in a negative direction. The signal is typically obtained from the Windup low output from an inner loop. Default is false.
XC730.Sts_Available	1 = Inner loop (slave object) is available. 0 = Inner loop is not available, PPID tracks Inp_CVTrack, typically inner loop SP or actuator position. Default is true.
XIC730.Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
XIC730.Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status, 1 = All non-bypassable interlocks OK to run
XIC730.Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
XIC730.Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet
XC730.Sts_eNotify	Alarm status from PAO: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

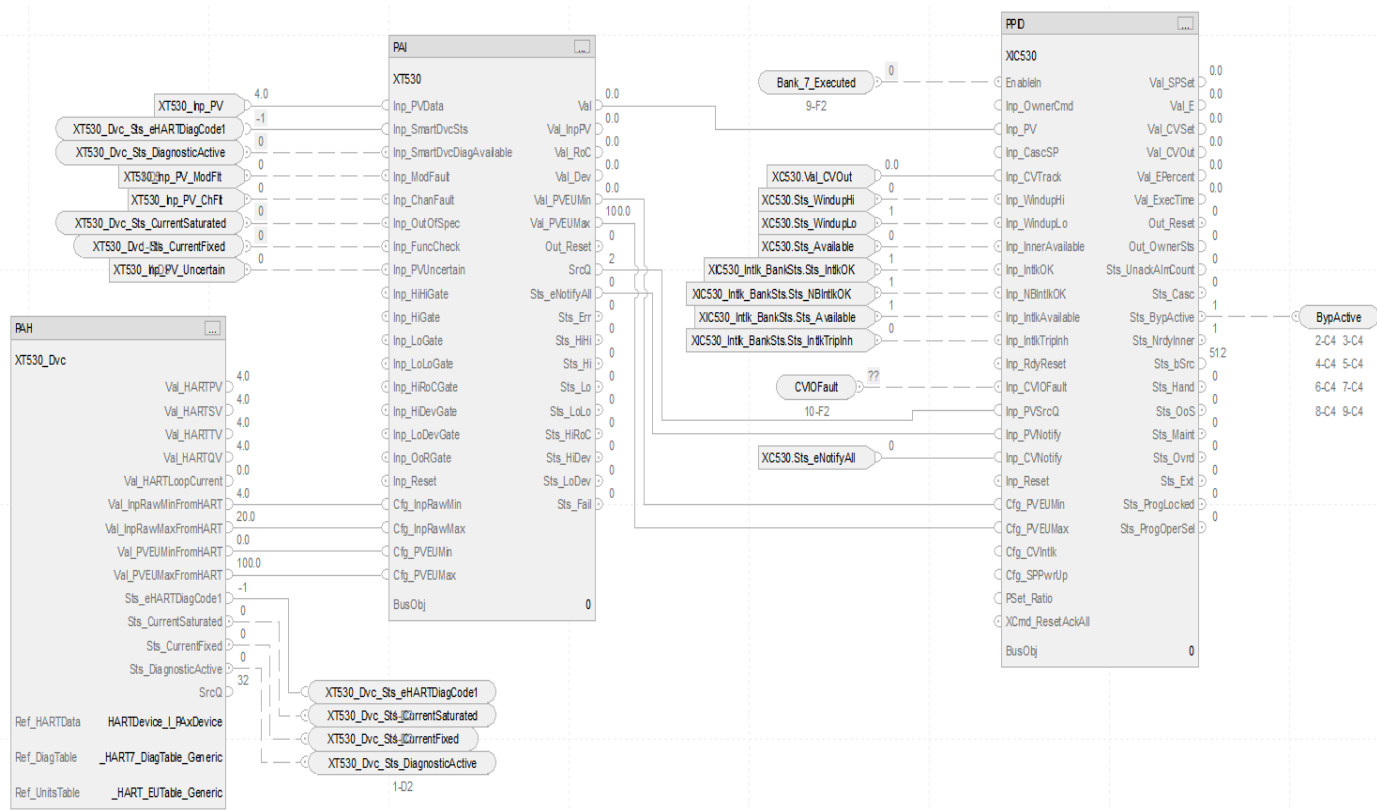
*PPID Output References*

Parameter	Description
BypActive	Output connection to interlock bank sheet

### PPID Configuration Considerations

Operand	Type	Description
PlantPax control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

### CS\_PPID\_PAO\_HART\_PID HART Sheet (XIC530)



The CS\_PPID\_PAO\_HART\_PID sheet operates the same as the CS\_PPID\_PAO\_PID sheet but relies on HART input data.

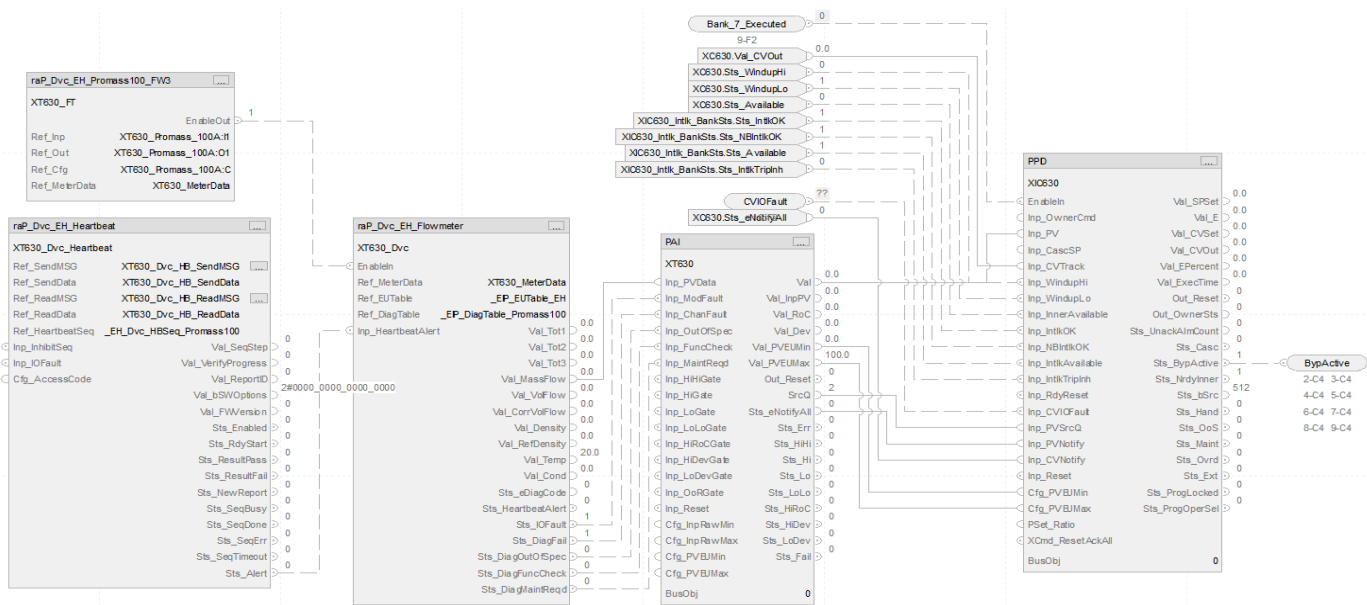
- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XT530 for XT100

If you are using the CS\_HART\_PAO\_HART control strategy:

- Substitute XT531 for XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_PAO\_PID EtherNet/IP Sheet (XIC630)

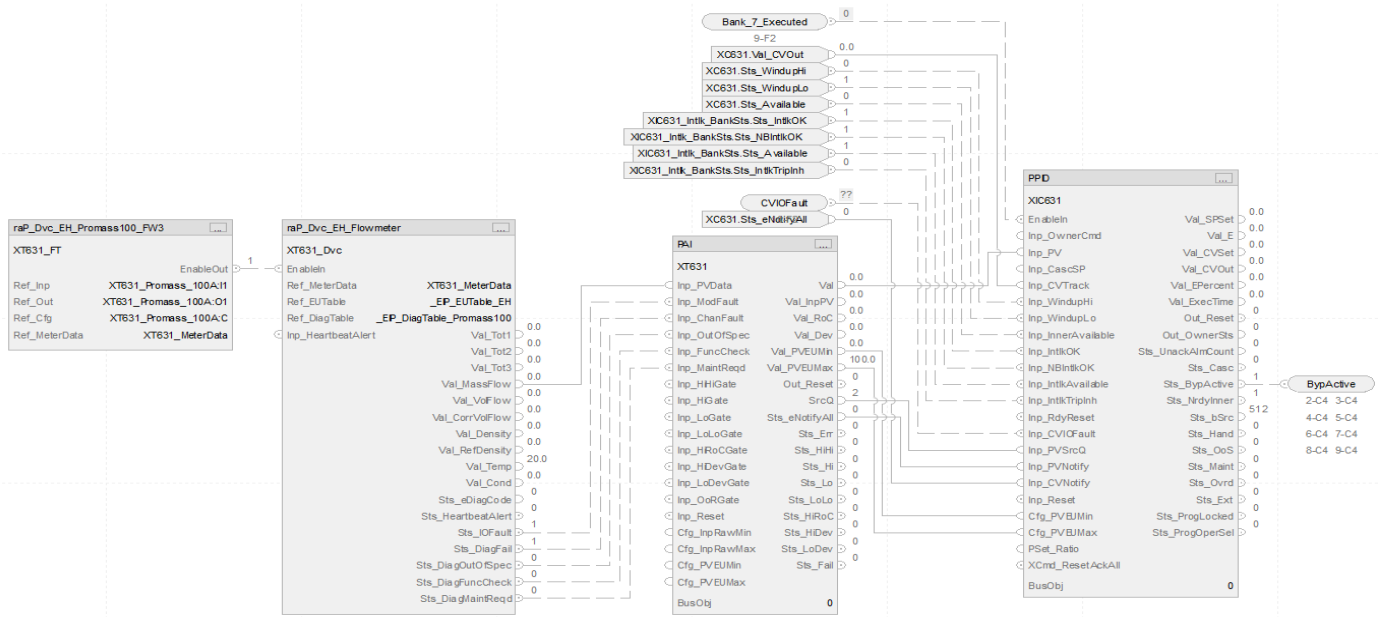


The CS\_PPID\_PAO\_HART\_PID EtherNet/IP sheet operates the same as the CS\_PPID\_PAO\_PID sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNet/IP Sheet on page 151](#).
- Substitute XT630 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_PAO\_PID EtherNet/IP NoHB (XIC631)

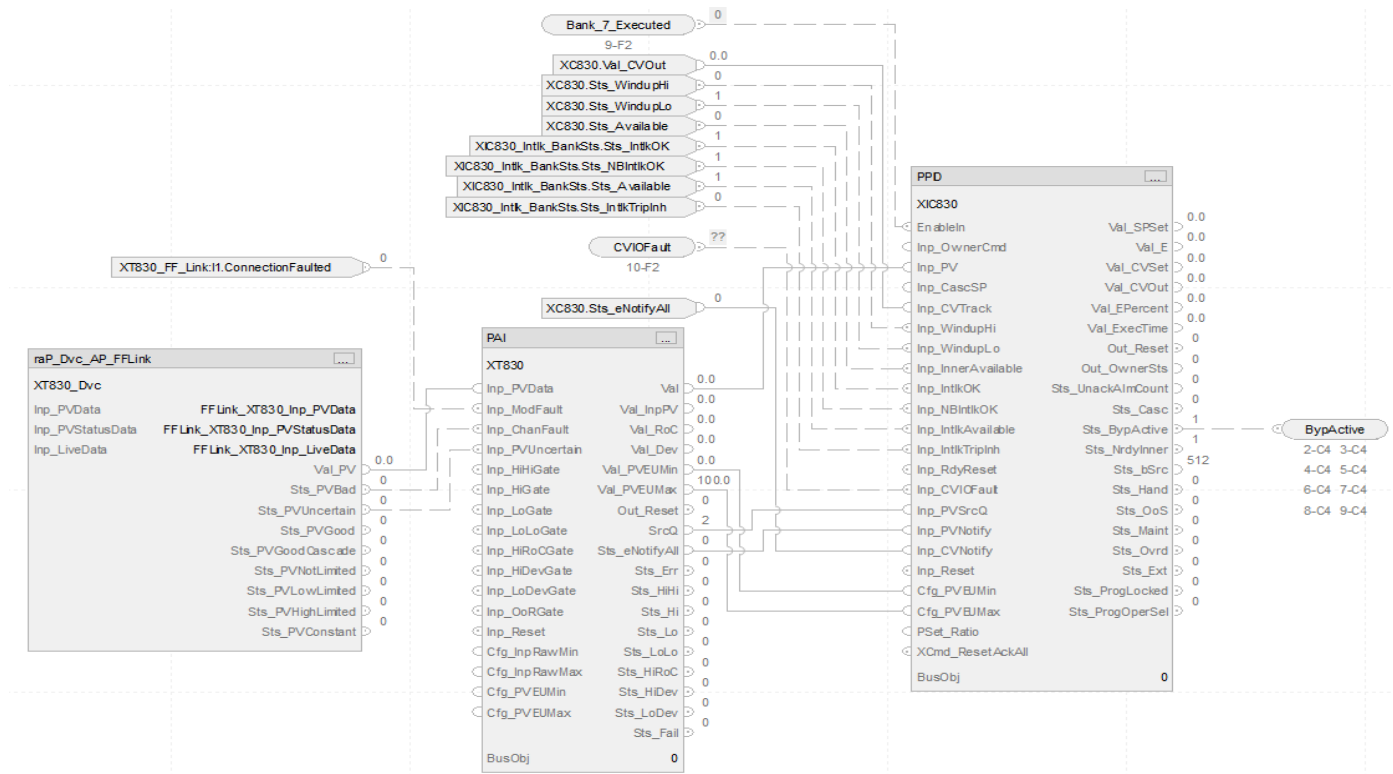


The CS\_PPID\_PAO\_PID EtherNet/IP NoHB sheet operates the same as the CS\_PPID\_PAO\_PID sheet but relies on EtherNet/IP input data without heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XT631 for XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_PAO\_PID FF Sheet (XIC830)

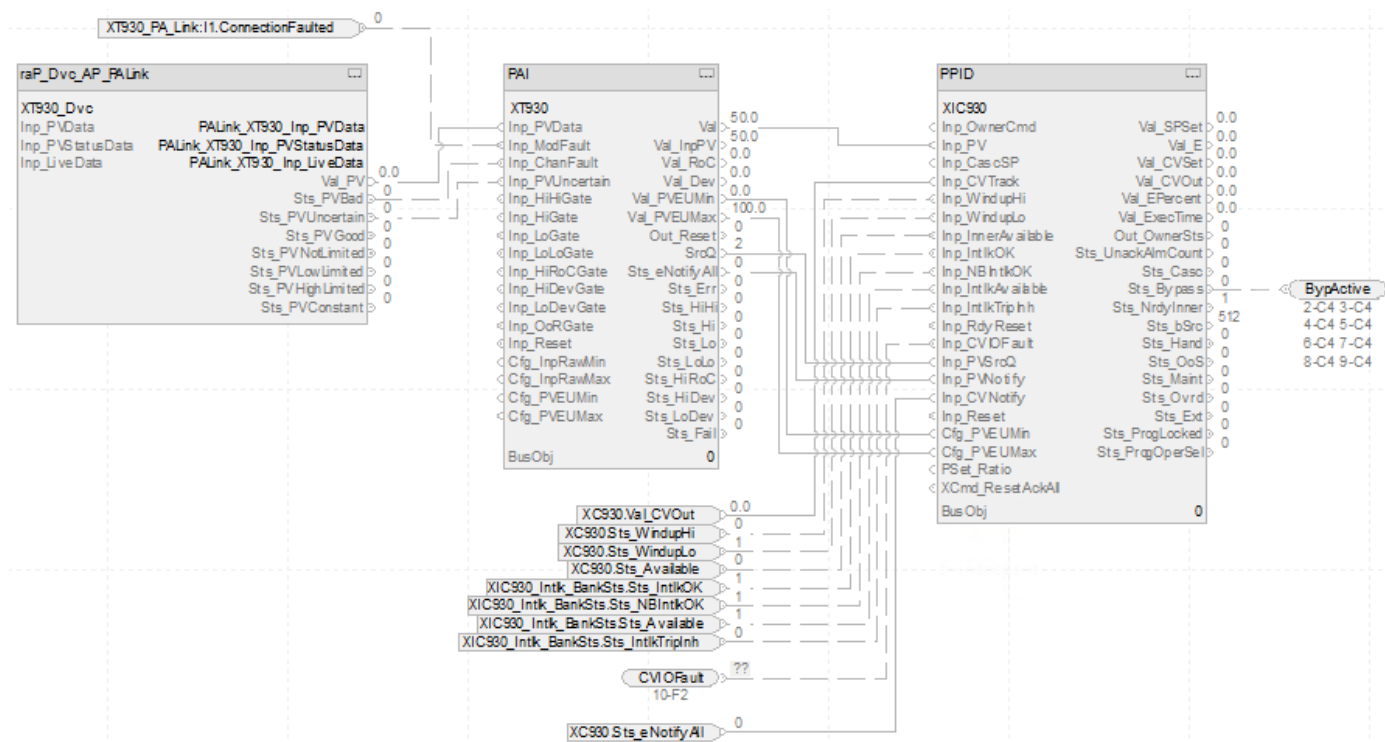


The CS\_PPID\_PAO\_PID FOUNDATION Fieldbus sheet operates the same as the CS\_PPID\_PAO\_PID sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XT830 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_PAO\_PID Profibus PA Sheet (XIC930)

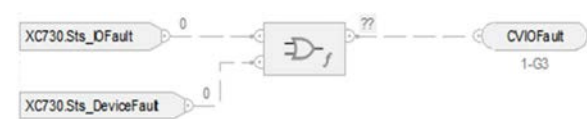


The CS\_PPID\_PAO\_PID Profibus PA sheet operates the same as the CS\_PPID\_PAO\_PID sheet but relies on Profibus PA input data.

- For information on Profibus PA outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XT930 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

IO Faults Sheet



Faults Input References

Parameter	Description
XCxxx.Sts_IOFault	1 = IO Fault Status Bad 0 = OK There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PAOTag.@Alarms.Alm_IOFault.AlarmElement Source: PAO instruction
XCxxx.Sts_DeviceFault	Device Fault status: 1 = Bad, 0 = OK. There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PAOTag.@Alarms.Alm_DeviceFault.AlarmElement Source: PAO instruction

Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

ACM Considerations for PPID with PAO Control

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the type of analog input via the PAI\_Type
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type and CV\_IO\_Signal\_Type parameters

## ACM-Based Parameters for a PPID Instance with PAO Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
CV_IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasCVIntlkObj	always	Set to create an instance of the PINTLK instruction
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Bus_InstanceCV	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
<b>02.01 - Device Configuration Feedback</b>		
Cfg_HasOpenedFdbk	always	Set if the device has an open feedback input
Cfg_HasClosedFdbk	always	Set if the device has a closed feedback input
Cfg_HasPosFdbk	always	Set if the device has a position feedback input
Cfg_HasCombIndFdbk	always	Set if the device has combined feedback providing open, closed, and position.
<b>03.00 - Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PAO_RefTag	always	Link to the analog output reference
Cfg_CVUSeHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Hart_CVType	IO_Signal_Type=HART Cfg_CVUSeHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_CVHartDevice	IO_Signal_Type=HART Cfg_CVUSeHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects



Parameter	Visible When	Details
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.01 - IO Configuration**

PAI_RefTag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the PV input reference
Inp_PosFdbk	PAO_RefTag is linked to an analog output reference Cfg_HasPosFdbk=True	Link to the input position reference
	PAO_RefTag is linked to an analog output reference Cfg_HasCombindFdbk=True	
Inp_OpenLS	PAO_RefTag is linked to an analog output reference Cfg_HasOpenedFdbk=True	Link to the open limit switch feedback input reference
	PAO_RefTag is linked to an analog output reference Cfg_HasCombindFdbk=True	
Inp_ClosedLS	PAO_RefTag is linked to an analog output reference Cfg_HasClosedFdbk=True	Link to the closed limit switch feedback input reference
	PAO_RefTag is linked to an analog output reference Cfg_HasCombindFdbk=True	
Out_CV	always	Link to the CV output reference

**03.02 - Ref PAI Alarm Configuration**

Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Parameter	Visible When	Details
Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

## Additional Sub-Objects for a PPID with PAO Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CVIntlk	Configure interlocks for the CV instance See <a href="#">Interlocks on page 49</a>
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CVEvents	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>

## Process Proportional + Integral + Derivative (PPID) Basic PPID with Process Variable Speed Drive (PVSD) Control Strategies

This control strategy differs from the Basic PPID control strategy in that the output of the PPID is used to set the speed reference of a drive, rather than an analog output channel, using the PVSD. The PVSD instruction is used to monitor and control a variable speed motor using an AC (variable frequency) or DC drive.

The PPID and PVSD control strategy can be used to control a process variable by manipulating the speed reference of the PVSD. Typical process variables that can be controlled with this strategy include speed, flow, and pressure.

The following PPID with PVSD control strategies are available as routines in the process library:

### PPID with PVSD Control Strategies

Drive Type	Control Strategy	Routine
Generic variable speed drive	CS_PPID_PVSD CS_PPID_PVSD_HART CS_PPID_PVSD_EtherNetIP CS_PPID_PVSD_FF CS_PPID_PVSD_PA	<ul style="list-style-type: none"> <li>▷ <a href="#">CS_PPID_PVSD</a></li> <li>▷ <a href="#">CS_PPID_PVSD_EtherNetIP</a></li> <li>▷ <a href="#">CS_PPID_PVSD_FF</a></li> <li>▷ <a href="#">CS_PPID_PVSD_HART</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PA</a></li> </ul>
PowerFlex® 525	CS_PPID_PVSD_PF525 CS_PPID_PVSD_PF525_HART CS_PPID_PVSD_PF525_EtherNetIP CS_PPID_PVSD_PF525_FF CS_PPID_PVSD_PF525_PA	<ul style="list-style-type: none"> <li>▷ <a href="#">CS_PPID_PVSD_PF525</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF525_EtherNetIP</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF525_FF</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF525_HART</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF525_PA</a></li> </ul>
PowerFlex 753	CS_PPID_PVSD_PF753 CS_PPID_PVSD_PF753_HART CS_PPID_PVSD_PF753_EtherNetIP CS_PPID_PVSD_FF CS_PPID_PVSD_PF753_PA	<ul style="list-style-type: none"> <li>▷ <a href="#">CS_PPID_PVSD_PF753</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF753_EtherNetIP</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF753_FF</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF753_HART</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF753_PA</a></li> </ul>
PowerFlex 755	CS_PPID_PVSD_PF755 CS_PPID_PVSD_PF755_HART CS_PPID_PVSD_PF755_EtherNetIP CS_PPID_PVSD_PF755_FF CS_PPID_PVSD_PF755_PA	<ul style="list-style-type: none"> <li>▷ <a href="#">CS_PPID_PVSD_PF755</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755_EtherNetIP</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755_FF</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755_HART</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755_PA</a></li> </ul>
PowerFlex 755T	CS_PPID_PVSD_PF755T CS_PPID_PVSD_PF755T_HART CS_PPID_PVSD_PF755T_EtherNetIP CS_PPID_PVSD_PF755T_PA	<ul style="list-style-type: none"> <li>▷ <a href="#">CS_PPID_PVSD_PF755T</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755T_EtherNetIP</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755T_FF</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755T_HART</a></li> <li>▷ <a href="#">CS_PPID_PVSD_PF755T_PA</a></li> </ul>

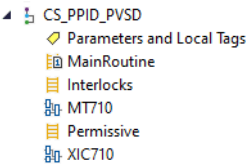
PPID with PVSD Control Strategies

Drive Type	Control Strategy	Routine
PowerFlex 6000T	CS_PPID_PVSD_PF6000T CS_PPID_PVSD_PF6000T_HART CS_PPID_PVSD_PF6000T_EtherNetIP CS_PPID_PVSD_PF6000T_FF CS_PPID_PVSD_PF6000T_PA	<div> ▶ CS_PPID_PVSD_PF6000T </div> <div> ▶ CS_PPID_PVSD_PF6000T_EtherNetIP </div> <div> ▶ CS_PPID_PVSD_PF6000T_FF </div> <div> ▶ CS_PPID_PVSD_PF6000T_HART </div> <div> ▶ CS_PPID_PVSD_PF6000T_PA </div>
PowerFlex 7000	CS_PPID_PVSD_PF7000 CS_PPID_PVSD_PF7000_HART CS_PPID_PVSD_PF7000_EtherNetIP CS_PPID_PVSD_PF7000_FF CS_PPID_PVSD_PF7000_PA	<div> ▶ CS_PPID_PVSD_PF7000 </div> <div> ▶ CS_PPID_PVSD_PF7000_EtherNetIP </div> <div> ▶ CS_PPID_PVSD_PF7000_FF </div> <div> ▶ CS_PPID_PVSD_PF7000_HART </div> <div> ▶ CS_PPID_PVSD_PF7000_PA </div>

Import the appropriate control strategy as a routine in your controller project.

Also, import the appropriate device object as a routine in your controller project. These objects are from the Power Device Library and must be downloaded separately from the PlantPax<sup>®</sup> Process Library.

Each control strategy contains a PPID routine (XICxxx) and a PVSD routine (MTxxx). Substitute the correct tags for PPID and variable speed drive.



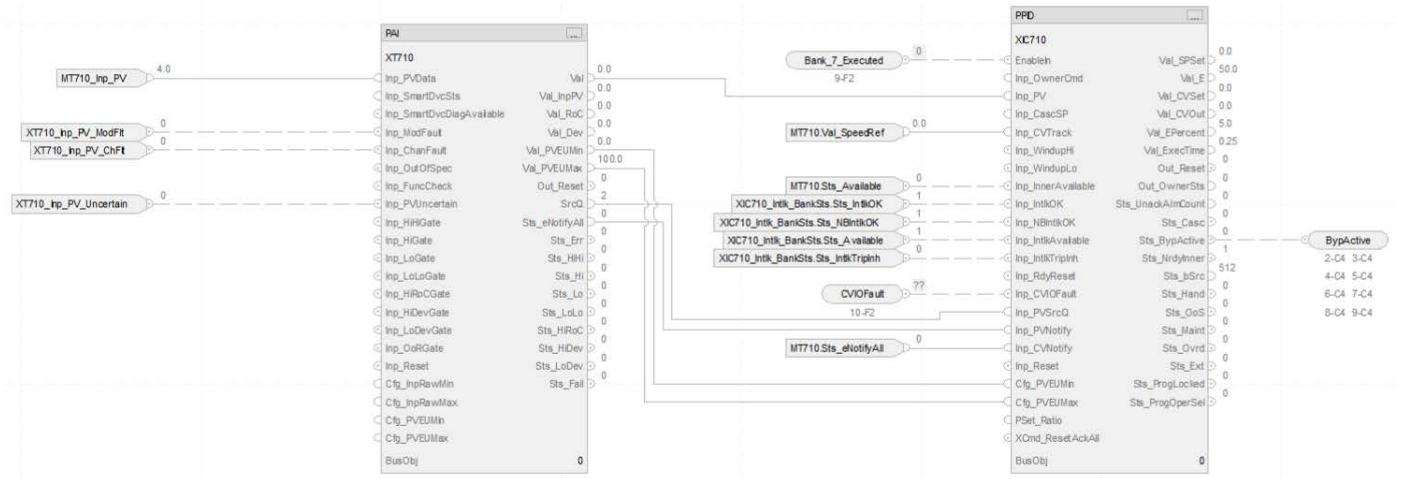
For more details on the PVSD routine, See [Process Variable Speed Drive \(PVSD\) Control Strategies on page 459](#).

ROUTINE PPID\_PVSD

Each routine contains these sheets.

Sheet	Description
CS_PPID_PVSD	PPID instructions
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	<p>The PVSD instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable).</p> <p>There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default.</p> <p>Use the sheets and interlocks that you need and delete the remainder.</p>
IO Faults	The logic monitors Control Variable faults.

## CS\_PPID\_PVSD\_PID Controller Sheet (XIC710)



### PAI Input References

See the [CS\\_PAI Sheet on page 148](#) for more details.

- Substitute XIC710 for the PV data instance of XT101
- Substitute MT710 for the remaining instances of XT101

## PAI Outputs to PPID Inputs

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUmin parameter PV minimum value in engineering units (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value in engineering units (PVEU). Valid any float greater than Cfg_PVEUmin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: <div> <div> 0 = Good, live, confirmed good  1 = Good, live, assumed good  2 = Good, no feedback, assumed good  8 = Test, simulated  9 = Test, loopback  10 = Test, manually entered  16 = Uncertain, live, off-spec  17 = Uncertain, substituted at device </div> <div> 18 = Uncertain, substituted at instruction  19 = Uncertain, using last known good  20 = Uncertain, using replacement value  32 = Bad, signal failure  33 = Bad, channel fault  34 = Bad, module/communications fault  35 = Bad, invalid configuration </div> </div>
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: <div> <div> 0 = Not in alarm, acknowledged  1 = Not in alarm, unacknowledged or reset required  2 = Low severity alarm, acknowledged  3 = Low severity alarm, unacknowledged  4 = Medium severity alarm, acknowledged </div> <div> 5 = Medium severity alarm, unacknowledged  6 = High severity alarm, acknowledged  7 = High severity alarm, unacknowledged  8 = Urgent severity alarm, acknowledged  9 = Urgent severity alarm, unacknowledged </div> </div>

## PPID Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
MT710.Val_SpeedRef	Speed Reference (target) to drive
MT710.Sts_Available	1 = Device has been acquired by Program and is now available for start/stop control
XIC710_Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
XIC710_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status, 1 = All non-bypassable interlocks OK to run
XIC710_Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
XIC710_Intlk_BankSts.Sts_IntlkInh	Interlock bank status, 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	
MT710.Sts_eNotify	

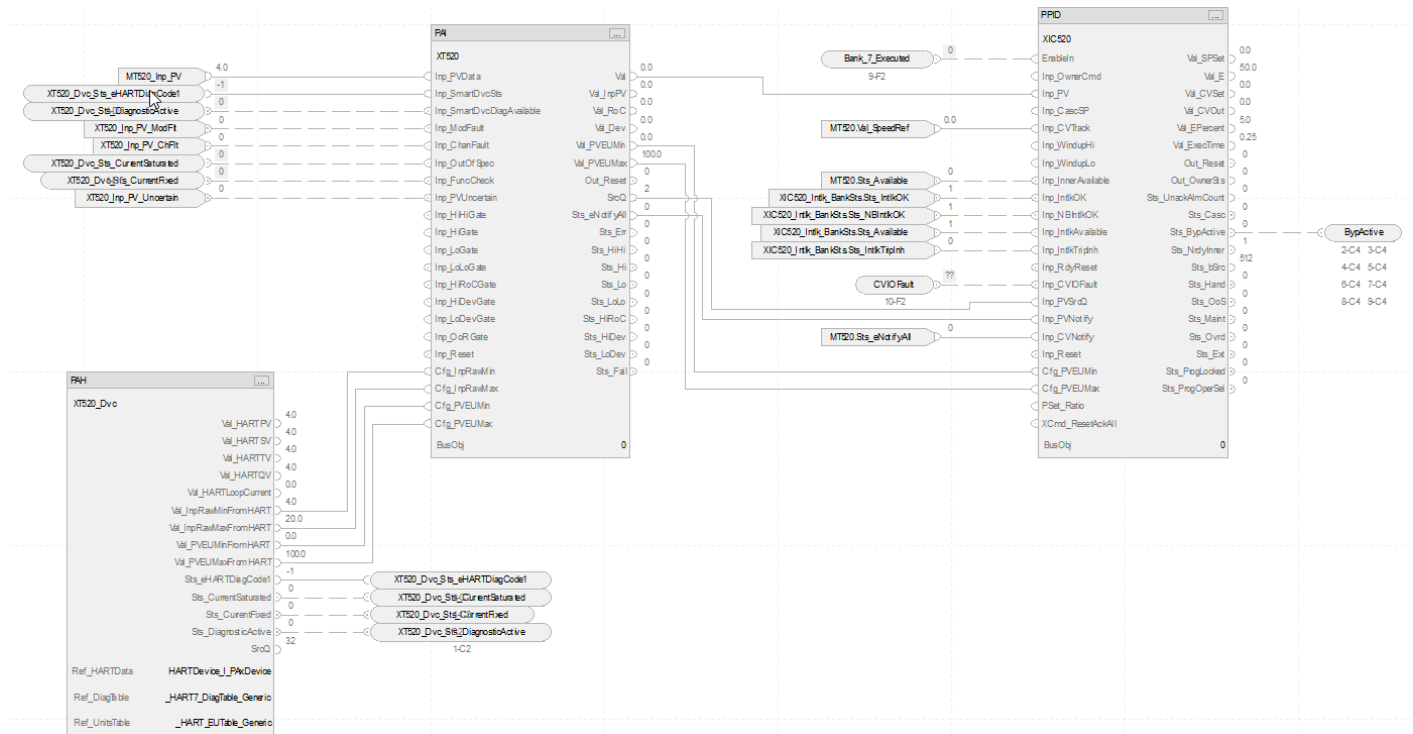
## PPID Output Reference

Parameter	Description
BypActive	Output connection to interlock bank sheet

## PPID Configuration Considerations

Operand	Type	Description
PlantPax control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>0 if not using organization</li> <li>Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID\_PVSD\_HART\_PID Controller Sheet (XIC520)

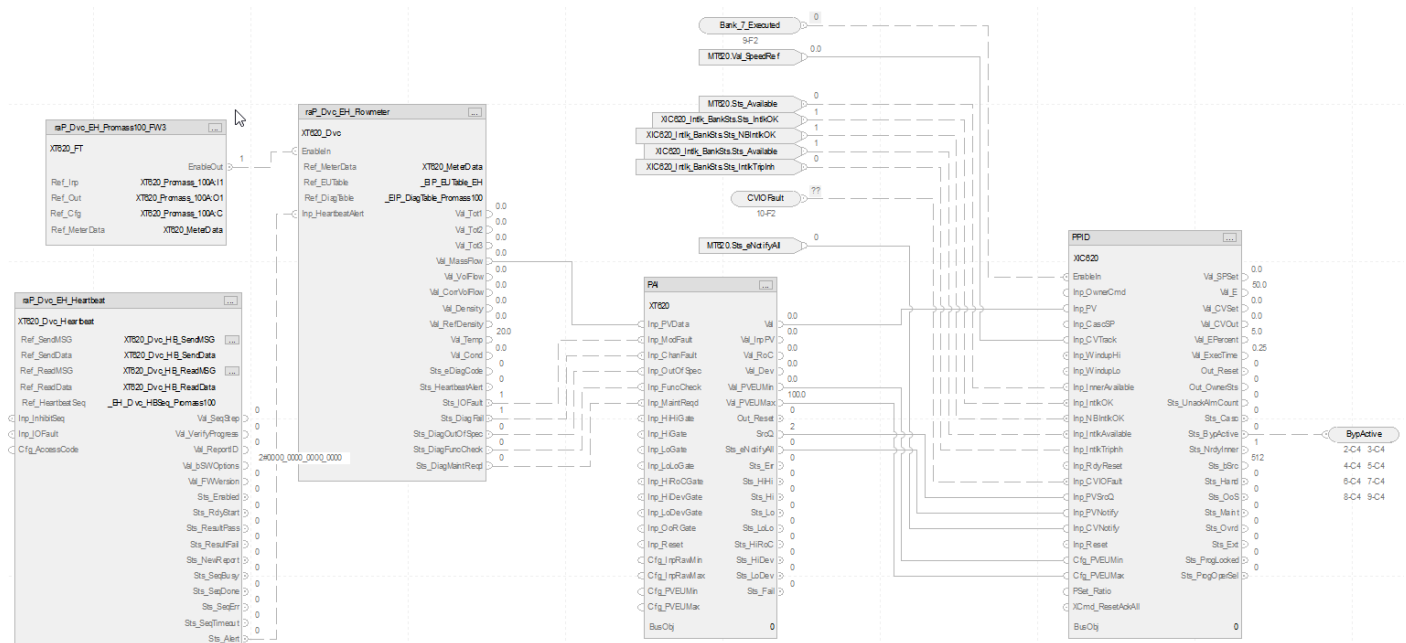


The CS\_PPID\_PVSD\_HART sheet operates the same as the CS\_PPID\_PVSD sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XIC520 for the PV data instance of XT101
- Substitute XT520 for the remaining instances of XT100

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_EH\_Promass100\_P VSD\_PID\_Controller Sheet

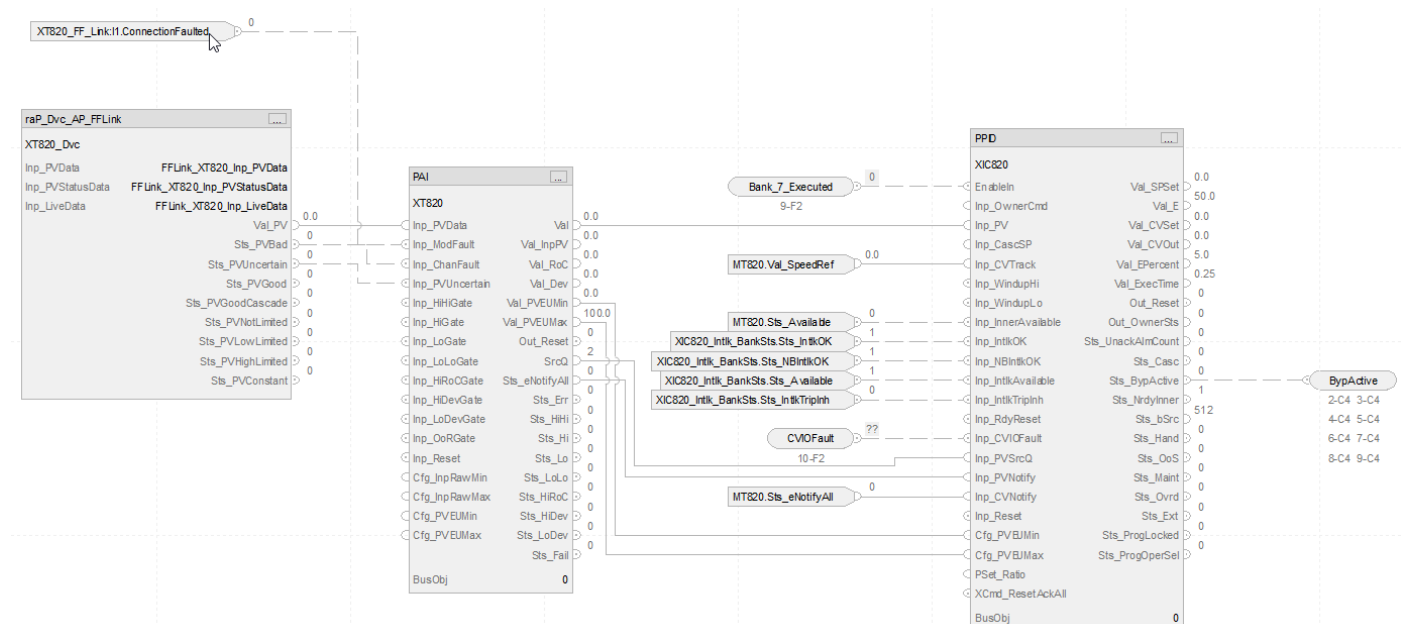


The CS\_PPID\_EH\_Promass100\_PVSD\_PID\_Controller sheet operates the same as the CS\_PPID\_PVSD sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XIC620 for the PV data instance of XT101
- Substitute XT620 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_FFLink\_PVSD\_PID\_Controller Sheet



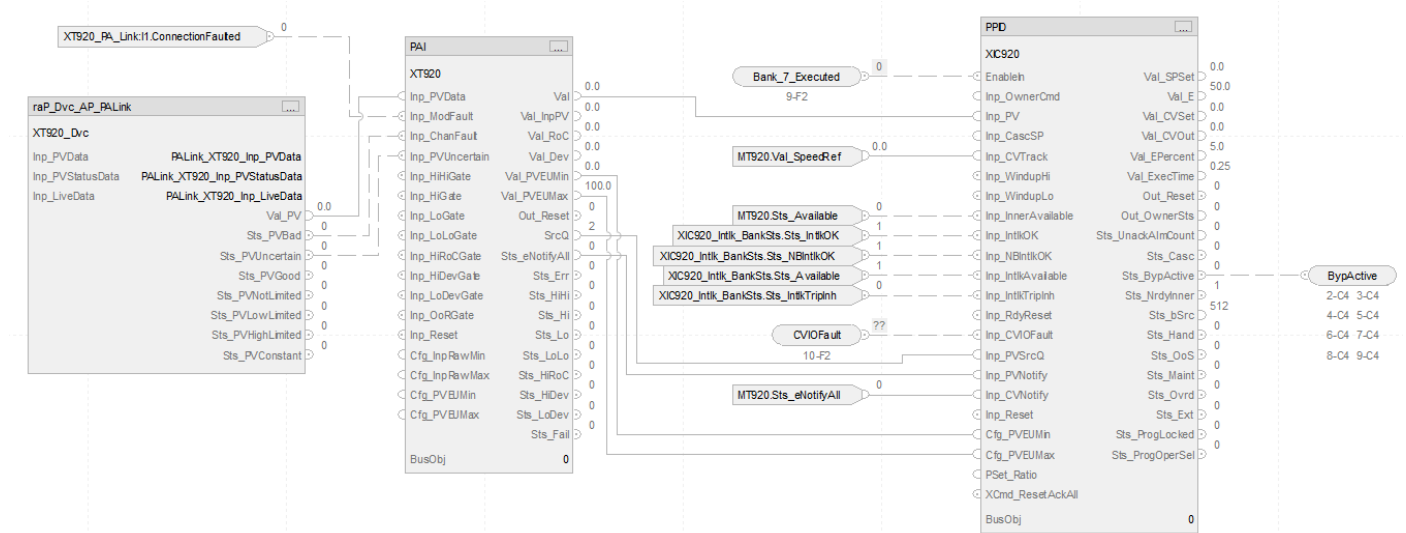


The CS\_PPID\_FFLink\_PVSD sheet operates the same as the CS\_PPID\_PVSD sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XIC820 for the PV data instance of XT101
- Substitute XT820 for the remaining instances of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_PALink\_PVSD\_PID \_Controller Sheet



The CS\_PPID\_PALink\_PVSD sheet operates the same as the CS\_PPID\_PVSD sheet but relies on Profibus-PA input data.

- For information on Profibus-PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XIC920 for the PV data instance of XT101
- Substitute XT920 for the remaining instances of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## IO Faults Sheet



### Fault Input References

Parameter	Description
MTxxx.Sts_IOFault	1 = IO Fault Status Bad 0 = OK There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PVSDTag.@Alarms.Alm_IOFault.AlarmElement Source: PVSD instruction
MTxxx.Sts_DriveFault	Drive Fault status: 1 = Bad, 0 = OK. There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PVSDTag.@Alarms.Alm_DriveFault.AlarmElement Source: PVSD instruction

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_PVSD sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID with PVSD

Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.

- Specify the type of analog input via the PAI\_Type
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
CV_IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping

Parameter	Visible When	Details
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset and the Inp_Rest objects
Cfg_HasCVIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
Cfg_HasReverse	always	Set if drive can run or jog in reverse
Cfg_HasFwdPermObj	always	Set to create an instance of the PPERM instruction to allow a run forward command See <a href="#">Permissives on page 50</a>
Cfg_HasRevPermObj	Cfg_HasReverse=True	Set to create an instance of the PPERM instruction to allow a run reverse command See <a href="#">Permissives on page 50</a>
Cfg_HasResInhObj	always	Set to create an instance of the restart inhibit (PRI) instruction See <a href="#">Statistics Objects on page 57</a>
Cfg_HasRunTimeObj	always	Set to create an instance of a runtime (PRT) instruction See <a href="#">Statistics Objects on page 57</a>
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Bus_InstanceCV	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device

**02.01 – Device Configuration Feedback**

Cfg_HasCVRunFdbk	always	Set if drive provides feedback signal when running
Cfg_HasCVSpeedFdbk	always	Set if drive provides speed feedback
Cfg_HasHand	always	Set to enable a hand switch input (Inp_Hand)

**03 – IO Selection**

PVSD_RefTag	always	Link to the PVSD instance.
Cfg_HasDvcObj	always	Set if device has connection to a device object
Cfg_DvcObj_Tag	Cfg_HasDvcObj=True	Link to the device object. See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>

**03.00 – IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus-PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.01 – IO Configuration**

PAI_RefTag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the PV input reference
Inp_Running	CfgHasDvcObj=False Cfg_HasRunFdbk=True	Link to run feedback input reference
Inp_SpeedFdbk	CfgHasDvcObj=False Cfg_HasSpeedFdbk=True	Link to speed feedback input reference

Parameter	Visible When	Details
Out_Start	always	Link to the start output reference
Out_Stop	always	Link to the stop output reference
OutSpeedRef	always	Link to the speed reference input reference
Out_Horn	always	Link to the horn output reference
Out_Reset	always	Link to the reset output reference
Out_Fwd	always	Link to the forward output reference
Out_Rev	always	Link to the reverse output reference
Out_ClearFaultData	always	Link to the clear fault data output reference
Inp_Accelerating	CfgHasDvcObj=False	Link to the accelerating input reference
Inp_ActualDir	CfgHasDvcObj=False	Link to the actual direction input reference
Inp_Alarm	CfgHasDvcObj=False	Link to the alarm input reference
Inp_AtSpeed	CfgHasDvcObj=False	Link to the at-speed input reference
Inp_CommandDir	CfgHasDvcObj=False	Link to the command direction input reference
Inp_Decelerating	CfgHasDvcObj=False	Link to the decelerating input reference
Inp_Faulted	CfgHasDvcObj=False	Link to the fault input reference
Inp_Ready	CfgHasDvcObj=False	Link to the ready input reference
Inp_LastFaultCode	CfgHasDvcObj=False	Link to the last fault code input reference
Inp_Hand	Cfg_HasHand=True	Link to the hand switch input reference

**03.02 - Ref PAI Alarm Configuration**

Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

## Additional Sub-Objects for a PPID with PVSD Instance

- Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CVIntlk	Configure interlocks for the CV instance See <a href="#">Interlocks on page 49</a>
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
CV_Fwd_Permissive CV_Rev_Permissive	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CVEvents	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>
Linked Libraries	Configure device libraries needed for your project See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>

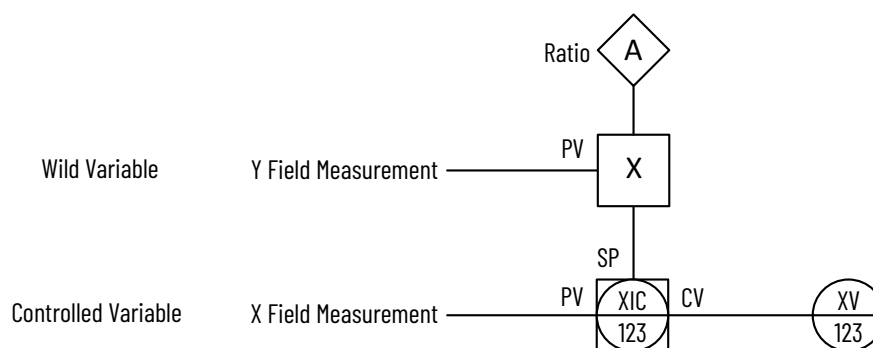
## Notes:

## Process Proportional + Integral + Derivative (PPID) Ratio Control Strategies

Use the PPID Ratio control strategy to add a material in a set proportion to another material.

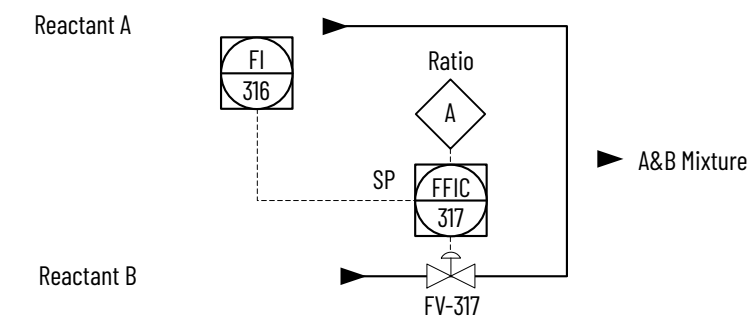
To scale the CV to align with the associated I/O module channel range or to accommodate a fail-open (FO) valve (or air to close) use either of the following options:

- Use a basic PPID with Analog Output control strategy
- Insert a scalar instruction between the PPID CV and the analog output channel reference



# PPID Ratio Example

In this example, two reactants (A and B) are added to a tank in a constant ratio. The flow rate of reactant A might change over time because of some upstream process upsets. Use a PPID Ratio control strategy to automatically adjust the rate of the reactant B addition. In this example, reactant A is the uncontrolled or wild flow because it is not controlled by the PPID instruction. The flow of reactant B is the controlled flow.



To perform ratio control with a PPID instruction, set the Cfg.HasCasc and Cfg.HasRatio input parameters. Wire the uncontrolled flow into the Inp\_CascSP input parameter. When in Cascade/Ratio mode, the uncontrolled flow is multiplied by either the OSet\_Ratio, when in Operator control, or the PSet\_Ratio, when in Program control, and the resulting value is used by the PPID instruction as the setpoint.

The following PPID control strategies are available as routines in the process library:

- CS\_PPID\_RATIO
- CS\_PPID\_RATIO\_HART
- CS\_PPID\_RATIO\_EtherNetIP
- CS\_PPID\_RATIO\_EtherNetIP\_NoHB
- CS\_PPID\_RATIO\_FF
- CS\_PPID\_RATIO\_PA

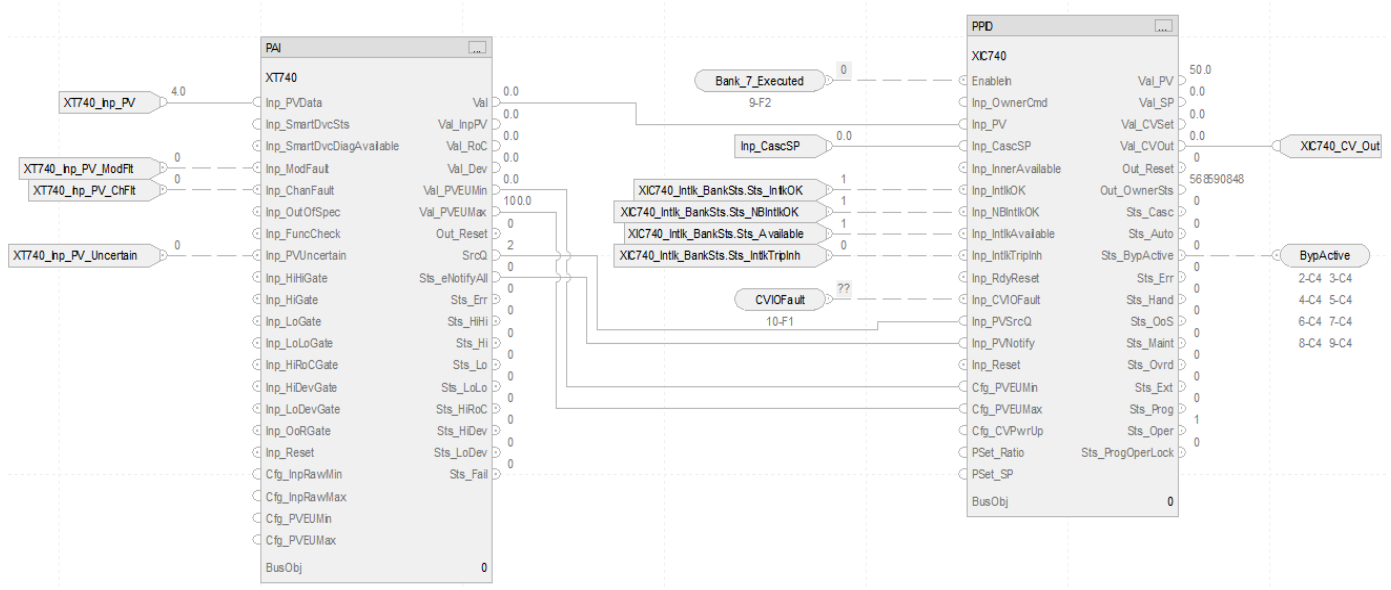
Import the appropriate control strategy as a **routine** in your controller project.

Each PPID Ratio control strategy contains these sheets:

Sheet	Description
CS_PPID_RATIO	PPID instruction
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PPID instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors Control Variable faults.



## CS\_PPID\_RATIO Sheet



### PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

- Substitute XIC740 for the PV data instance of XT101
- Substitute XT740 for the remaining instances of XT101

### PAI Outputs to PPID Inputs

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUMin parameter PV minimum value in engineering units (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value in engineering units (PVEU). Valid any float greater than Cfg_PVEUMin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: 0 = Good, live, confirmed good 1 = Good, live, assumed good 2 = Good, no feedback, assumed good 8 = Test, simulated 9 = Test, loopback 10 = Test, manually entered 16 = Uncertain, live, off-spec 17 = Uncertain, substituted at device 18 = Uncertain, substituted at instruction 19 = Uncertain, using last known good 20 = Uncertain, using replacement value 32 = Bad, signal failure 33 = Bad, channel fault 34 = Bad, module/communications fault 35 = Bad, invalid configuration
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: 0 = Not in alarm, acknowledged 1 = Not in alarm, unacknowledged or reset required 2 = Low severity alarm, acknowledged 3 = Low severity alarm, unacknowledged 4 = Medium severity alarm, acknowledged 5 = Medium severity alarm, unacknowledged 6 = High severity alarm, acknowledged 7 = High severity alarm, unacknowledged 8 = Urgent severity alarm, acknowledged 9 = Urgent severity alarm, unacknowledged

## PPID Input References

Parameter	Description
XIC740_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XIC740_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XIC740_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XIC740_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet

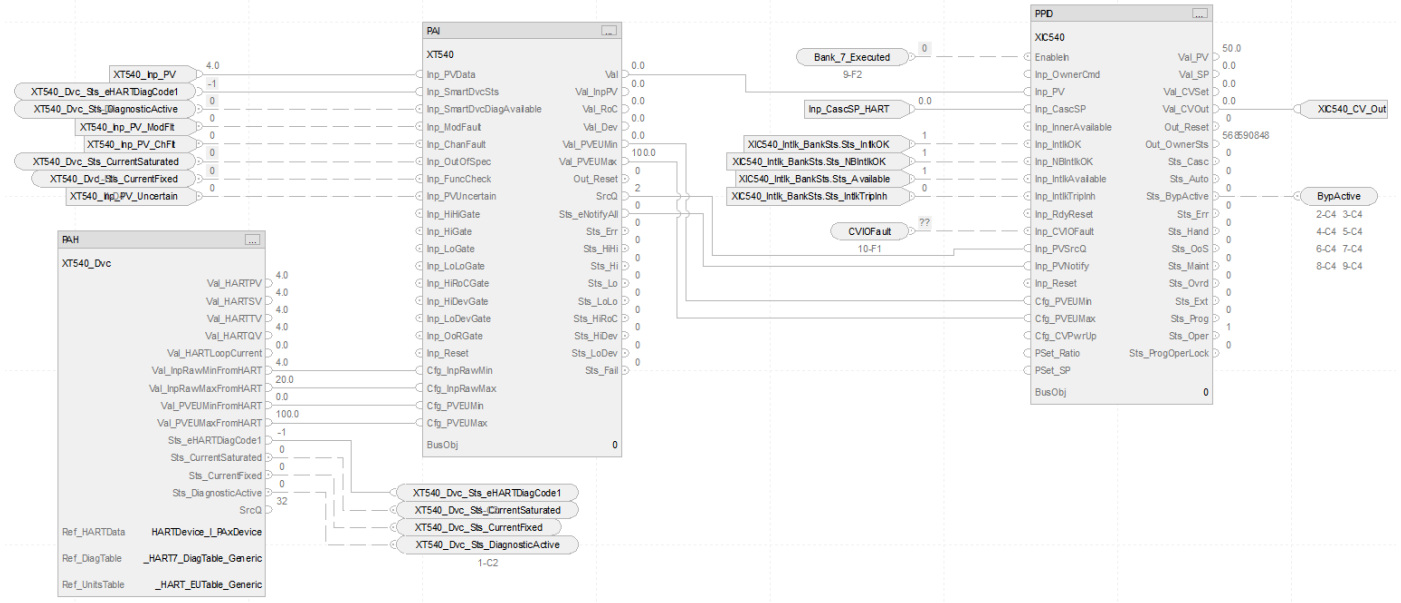
## PPID Output References

Parameter	Description
XIC740_Out_CV	Control Variable output Loop CV after clamping and ramping (CVEU)
BypActive	Output connection to interlock bank sheet

## PPID Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>0 if not using organization</li> <li>Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID\_RATIO HART Sheet

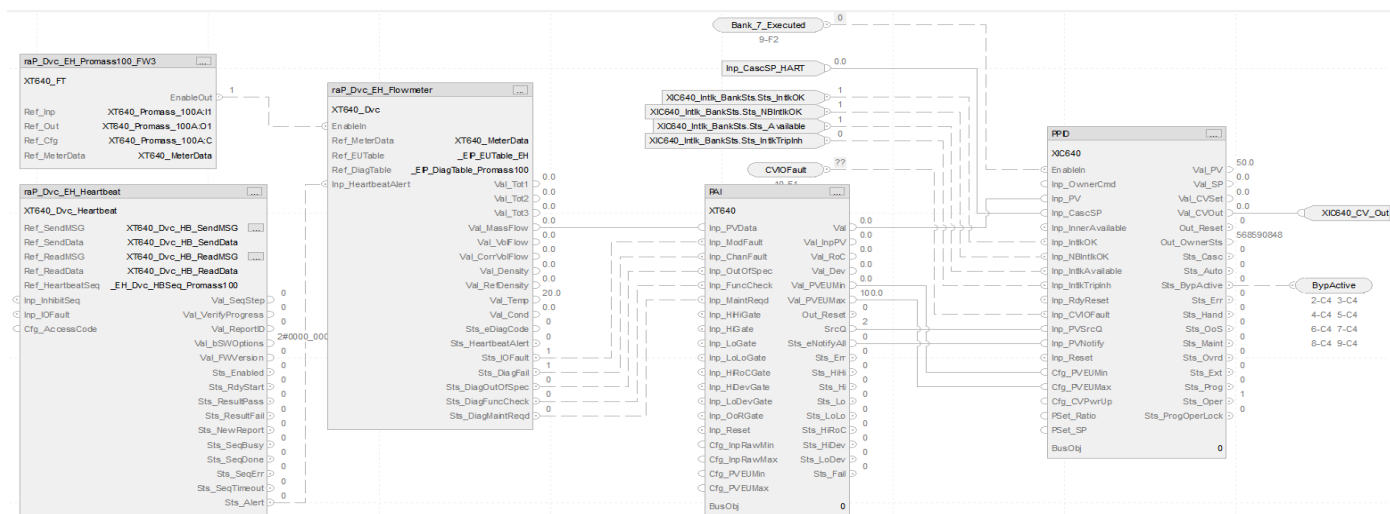


The CS\_PPID\_RATIO HART sheet operates the same as the CS\_PPID\_RATIO sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute XIC540 for the PV data instance of XT101
- Substitute XT540 for the remaining instances of XT100

For more information, see [HART Integration on page 61](#).

# CS\_PPID\_RATIO\_EtherNetIP Sheet

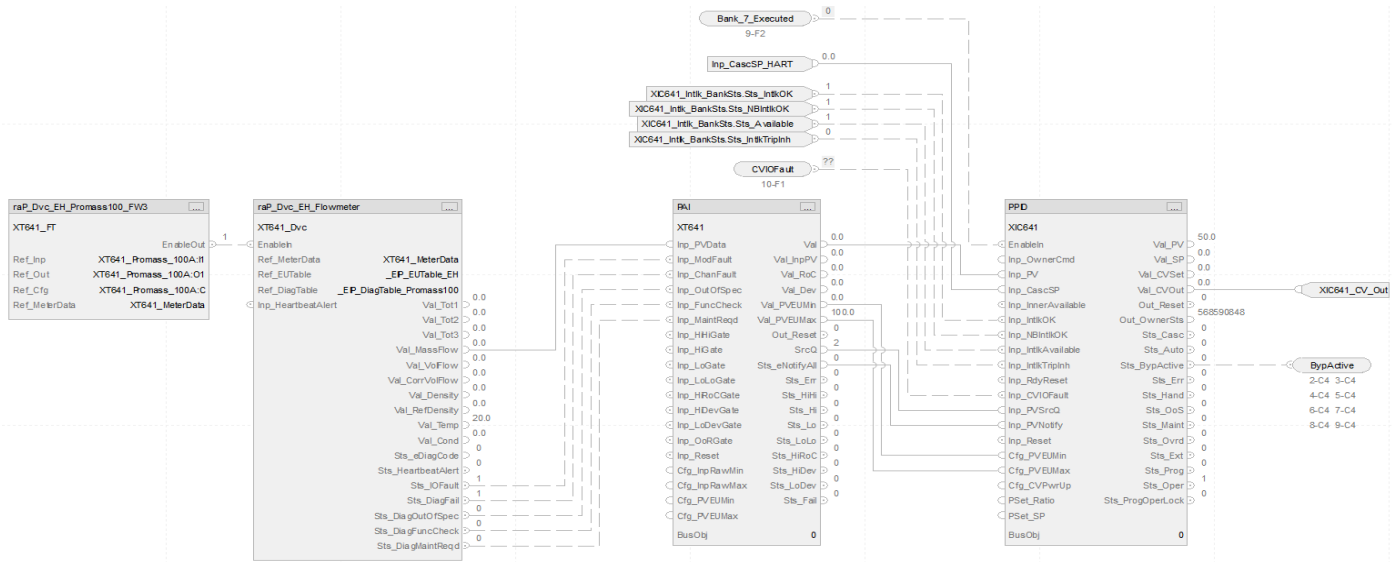


The CS\_PPID\_RATIO EtherNet/IP™ sheet operates the same as the CS\_PPID\_RATIO sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XIC640 for the PV data instance of XT101
- Substitute XT640 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_RATIO\_EtherNetIP \_NoHB Sheet

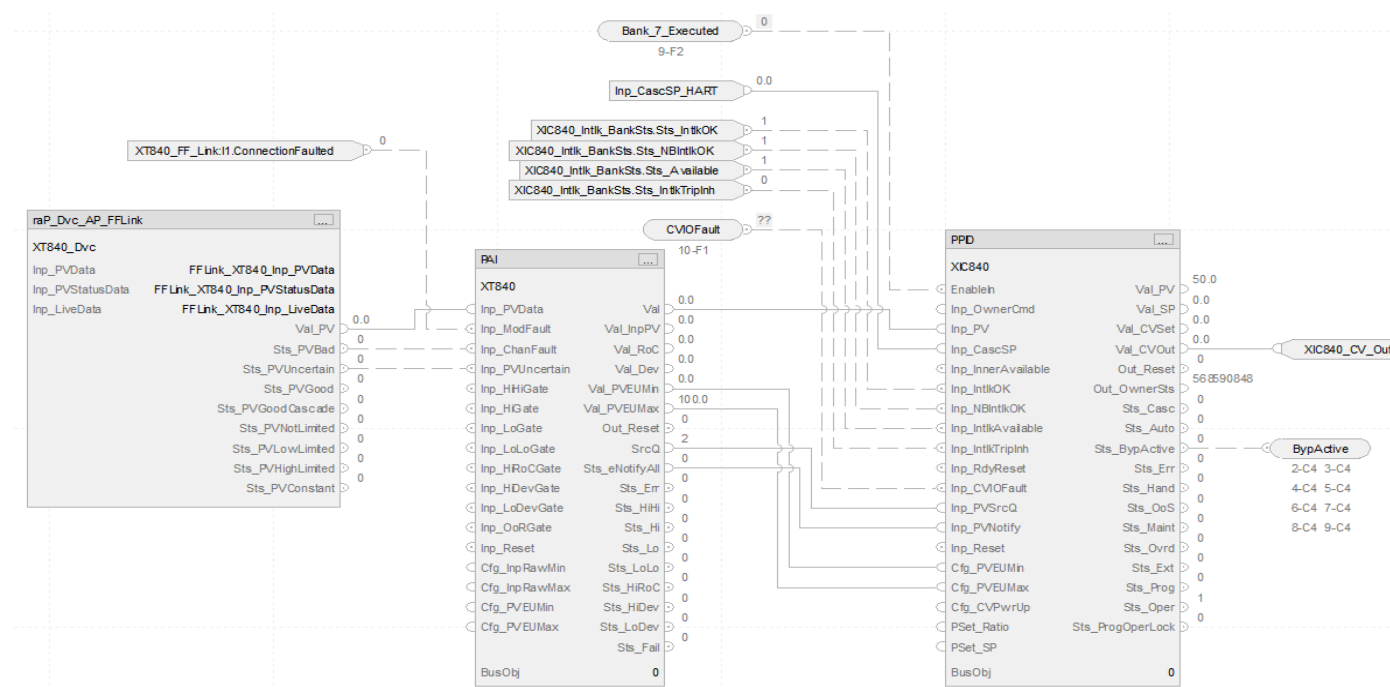


The CS\_PPID\_RATIO EtherNet/IP NoHB sheet operates the same as the CS\_PPID\_RATIO sheet but relies on EtherNet/IP input data with no heartbeat.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP\\_NoHB Sheet on page 153](#).
- Substitute XIC641 for the PV data instance of XT101
- Substitute XT641 for the remaining instances of XT100

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_RATIO\_FF

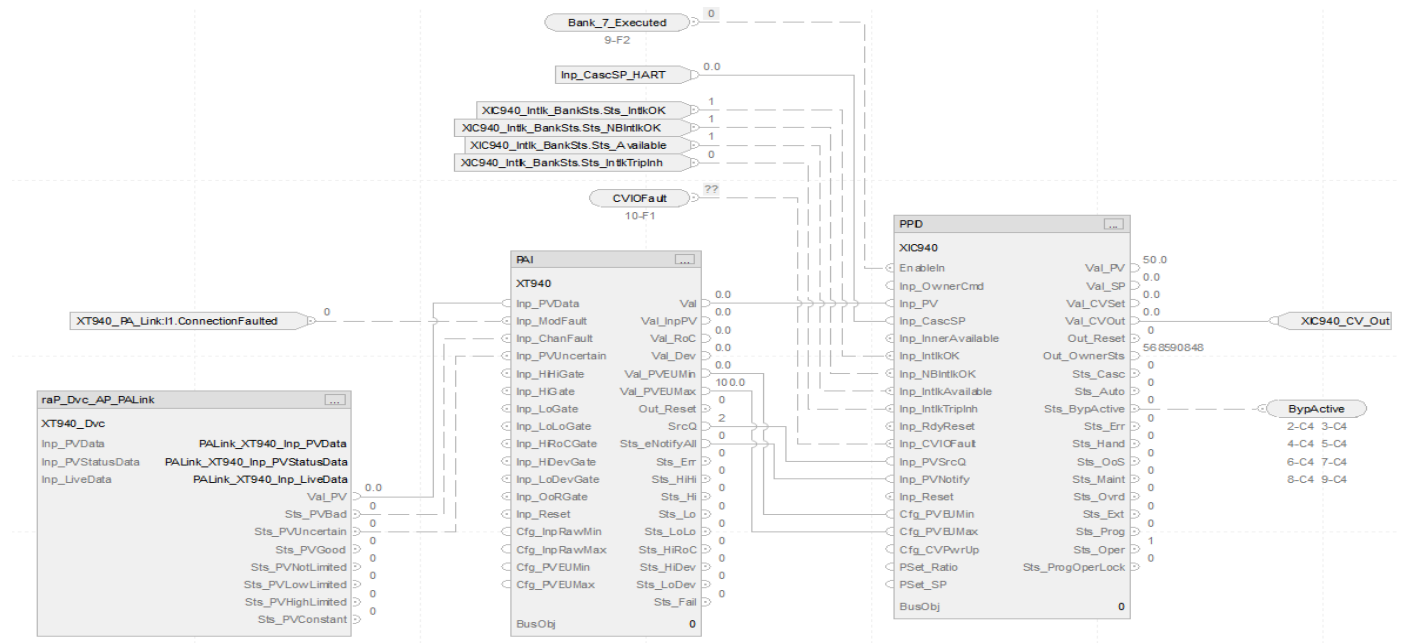


The CS\_PPID\_RATIO\_FF sheet operates the same as the CS\_PPID\_RATIO sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XIC840 for the PV data instance of XT101
- Substitute XT840 for the remaining instances of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PPID\_RATIO\_PA



The CS\_PPID\_RATIO\_PA sheet operates the same as the CS\_PPID\_RATIO sheet but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute XIC940 for the PV data instance of XT101
- Substitute XT940 for the remaining instances of XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

# I/O Faults Sheet



## Fault Input References

Parameter	Description
XICxxx_Out_CV_ChFlt	Channel fault 1 = I/O channel fault or failure 0 = OK Source: PAI instruction
XICxxx_Out_CV_ModFlt	Module fault 1 = I/O module failure or module communication status bad 0 = OK Source: PAI instruction

## Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_RATIO sheet

For examples on how to map data to input tags, see [PlantPAx Control Strategies on page 21](#).

## ACM Considerations for PPID with Ratio Control

- Configure these parameters first because they affect the visibility of the remaining parameters in the PPID object.
- Specify the type of analog input via the PAI\_Type parameter
  - If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter



## ACM-Based Parameters for a PPID Instance with Ratio Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
PAI_RefTag	always	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Wild_Variable_PV	always	Link to tag that will be connected to the Inp_CascSP parameter of the PPID
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the PV input reference
CV_Out	always	Link to the CV output reference
<b>03.00 - IO Configuration</b>		
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03.01 - Ref PAI Alarm Configuration</b>		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_LoGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAL_RefTag is linked to an analog input reference PAL_Type=PAI(Single_channel)	Link to the gate reference

**04 - Alarm Configuration**

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

**Additional Sub-Objects for a PPID with Ratio Control Instance**

Each sub-object has a tab on the configuration dialog for the control strategy object.

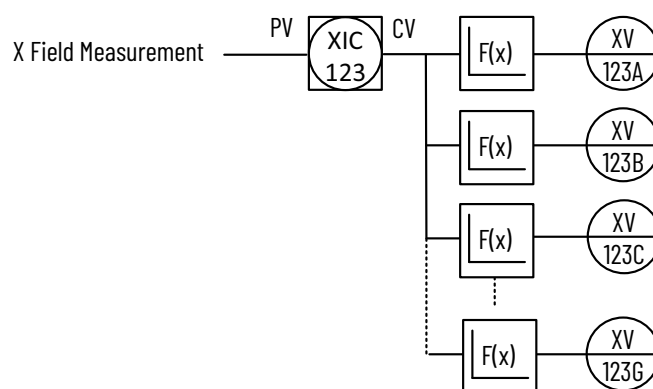
Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Notes:



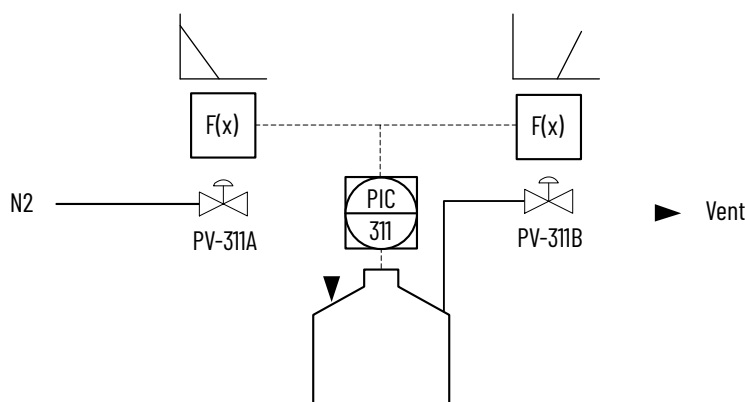
## Process Proportional + Integral + Derivative (PPID) Split Range Control Strategies

This PPID Split Range control strategy as provided, manipulates two field devices to maintain one process variable (PV) at setpoint (SP). You can modify this strategy to manipulate up to eight field devices in one Split Range PPID control strategy.



A feature built into the Split Range control strategies is that the PPID instruction receives an indication whether its downstream objects can be controlled. If no downstream object is available for manipulation, the PPID tracks a configured selection (CV1 Initial Value or a fixed value).

### PPID Split Range Example



Blanket gas (an inert gas) often pressurizes tanks containing combustible material to ensure no admission of air. In the example above, when PIC-311 CV = 50%, both valves are closed. When the CV is less than 50%, the pressurizing valve (PV-311A) opens, and the vent valve (PV-311B) is kept closed. When the CV is greater than 50%, the vent valve (PV-311B) opens, and the pressurizing valve is kept closed.

In practice, a gap can be used in the characterization to keep the valves from continuously cycling when the CV is near 50% (such as keep both valves closed when 48% < CV < 52%).

The valves characterization (CV splitting) is done with the Process Analog Fanout (PFO) instruction. The action of the PPID is direct acting to accommodate the valves characterization. If pressure is above setpoint, the CV increases; if pressure is below setpoint, the CV decreases.

### PPID Split Range Control Strategies

Control Strategy	Routines
CS_PPID_SPLITRANGE	<ul style="list-style-type: none"> <li>▾ CS_PPID_SPLITRANGE               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>XC780A</li> <li>XC780B</li> <li>XIC780</li> </ul> </li> </ul>
CS_PPID_SPLITRANGE_HART	<ul style="list-style-type: none"> <li>▾ CS_PPID_SPLITRANGE_HART               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>XC580A</li> <li>XC580B</li> <li>XIC580</li> </ul> </li> </ul>
CS_PPID_SPLITRANGE_EtherNetIP	<ul style="list-style-type: none"> <li>▾ CS_PPID_SPLITRANGE_EtherNetIP               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>XC680A</li> <li>XC680B</li> <li>XIC680</li> </ul> </li> </ul>
CS_PPID_SPLITRANGE_FF	<ul style="list-style-type: none"> <li>▾ CS_PPID_SPLITRANGE_FF               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>FFLinkMap</li> <li>Interlocks</li> <li>XC880A</li> <li>XC880B</li> <li>XIC880</li> </ul> </li> </ul>
CS_PPID_SPLITRANGE_PA	<ul style="list-style-type: none"> <li>▾ CS_PPID_SPLITRANGE_PA               <ul style="list-style-type: none"> <li>Parameters and Local Tags</li> <li>MainRoutine</li> <li>Interlocks</li> <li>PALinkMap</li> <li>XC980A</li> <li>XC980B</li> <li>XIC980</li> </ul> </li> </ul>

Import the **routines** for the appropriate control strategy in your controller project. Each control strategy contains multiple routines; each routine contains multiple Function Block sheets. The control strategy, as supplied, uses only two analog outputs. The control strategy can support as many as eight analog outputs by exposing additional parameters in the PFO instruction and adding PAO routines.

Each PPID Split Range control strategy Program is built from multiple Routines:

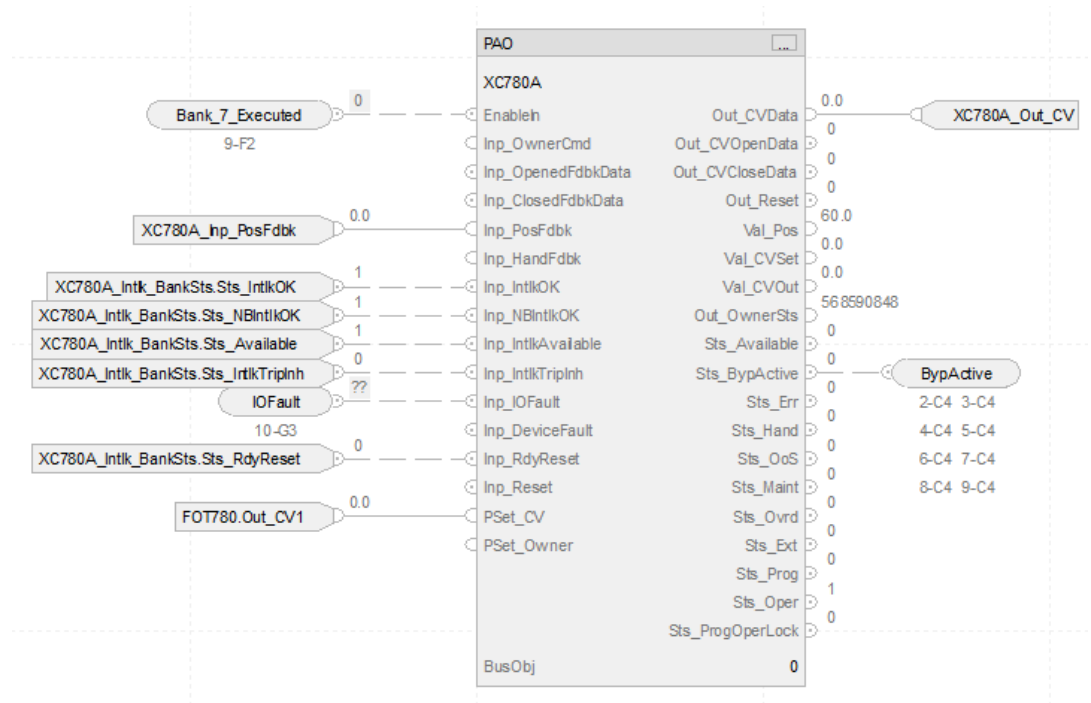
- Process Analog Output A
- Process Analog Output B
- Process Analog Input to Process PID with Fanout

## ROUTINE: Process Analog Output

There are two routines; each routine contains these sheets:

Sheet	Description
CS_PAO	Process High or Low Selector instruction <ul style="list-style-type: none"> <li>XC780A and XC780B analog</li> <li>XC580A and XC580B HART</li> <li>XC680A and XC680B EtherNet/IP™</li> <li>XC880A and XC880B FOUNDATION Fieldbus</li> <li>XC980A and XC980B Profibus PA</li> </ul>
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PAO instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors one analog output channel for I/O fault input and raises alarm on an I/O fault.

### CS\_PAO Sheet



#### Input References to PAO

See the [CS\\_PAO Sheet on page 180](#) for details.

Substitute:

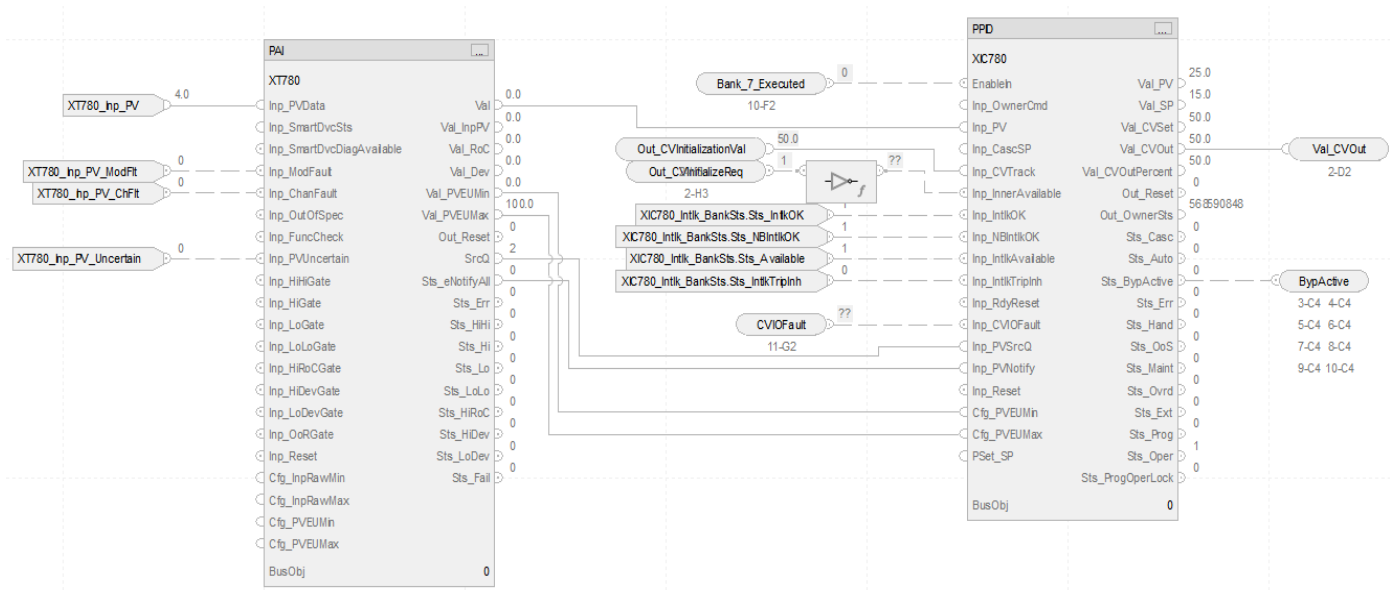
Input	First Instance of XC101 =	Second Instance of XC101 =	XC100 =
Analog	XC780A	XC780B	FOT780
HART	XC580A	XC580B	FOT580
EtherNet/IP	XC680A	XC680B	FOT680
FOUNDATION Fieldbus	XC880A	XC880B	FOT880
Profibus PA	XC980A	XC980B	FOT980

ROUTINE: Process Analog Input to Process PID with Fanout

There are two routines (One without HART and another with HART); each routine contains these sheets.

Sheet	Description
CS_PPID	Process PID instruction <ul style="list-style-type: none"><li>• XIC780 analog</li><li>• XIC580 HART</li><li>• XIC680 EtherNet/IP</li><li>• XIC880 FOUNDATION Fieldbus</li><li>• XIC980 Profibus PA</li></ul>
PFO	Process fan out
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PPID instruction monitors interlock conditions which cause output CV and SP to shed. CV shed can be configured to hold the last good CV value or to use the configured safe value. SP is shed to current PV. There are 8 interlock bank sheets; each bank exposes 16 interlocks but supports as many as 32 interlocks.  Use the sheets and interlocks that you need and delete the remainder.
IO Faults	The logic monitors Control Variable faults.

CS\_PPID Sheet



PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

- Substitute XIC780 for the PV data instance of XT101
- Substitute XT780 for the remaining instances of XT101



*PAI Outputs to PPID Inputs*

Parameter	Description
Val	Value for PPID Inp_PV parameter Process Variable (PVEU)
Val_PVEUmin	Value for PPID Cfg_PVEUmin parameter PV minimum value in engineering units (PVEU). Valid any float less than Cfg_PVEUMax.
Val_PVEUMax	Value for PPID Cfg_PVEUMax parameter PV maximum value in engineering units (PVEU). Valid any float greater than Cfg_PVEUmin.
SrcQ	Value for PPID Inp_PVSrcQ parameter Inp_PV source status and quality: <div style="display: flex; justify-content: space-between;"> <div> 0 = Good, live, confirmed good  1 = Good, live, assumed good  2 = Good, no feedback, assumed good  8 = Test, simulated  9 = Test, loopback  10 = Test, manually entered  16 = Uncertain, live, off-spec  17 = Uncertain, substituted at device </div> <div> 18 = Uncertain, substituted at instruction  19 = Uncertain, using last known good  20 = Uncertain, using replacement value  32 = Bad, signal failure  33 = Bad, channel fault  34 = Bad, module/communications fault  35 = Bad, invalid configuration </div> </div>
Sts_eNotifyAll	Value for PPID Inp_PVNotify parameter Related PV object alarm priority and acknowledgment status: <div style="display: flex; justify-content: space-between;"> <div> 0 = Not in alarm, acknowledged  1 = Not in alarm, unacknowledged or reset required  2 = Low severity alarm, acknowledged  3 = Low severity alarm, unacknowledged  4 = Medium severity alarm, acknowledged </div> <div> 5 = Medium severity alarm, unacknowledged  6 = High severity alarm, acknowledged  7 = High severity alarm, unacknowledged  8 = Urgent severity alarm, acknowledged  9 = Urgent severity alarm, unacknowledged </div> </div>

*Input References to PPID*

Parameter	Description
Out_CVInitializationVal	Initialization value to PPID Source: PFO instruction
Out_CVInitializeReq	Initialization request to PPID Source: PFO instruction
XIC780_Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
XIC780_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status, 1 = All non-bypassable interlocks OK to run
XIC780_Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
XIC780_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status, 1 = Interlock trip inhibit - stops equipment but does not trip
CVIOFault	Input connection from IO Faults sheet
XIC780_PSet_SP	Program setting for SP, loop mode Auto (PVEU). Valid any float.

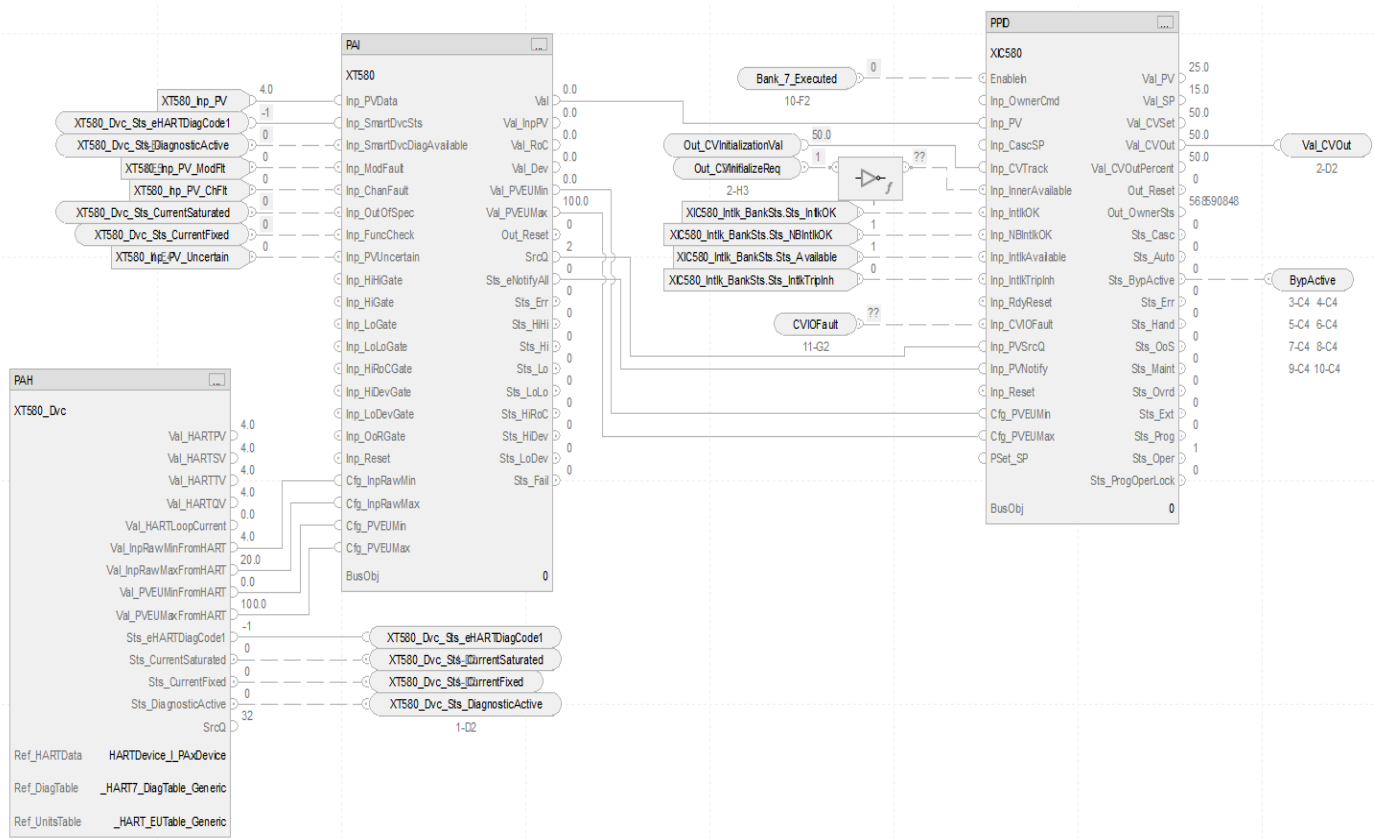
*PPID Output References*

Parameter	Description
Val_CVOut	Control Variable output Loop CV after clamping and ramping (CVEU) Destination: PFO instruction
BypActive	Output connection to interlock bank sheet

*PPID Configuration Considerations*

Operand	Type	Description
PlantPAX® control	P_PID	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>0 if not using organization</li> <li>Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## CS\_PPID HART Sheet

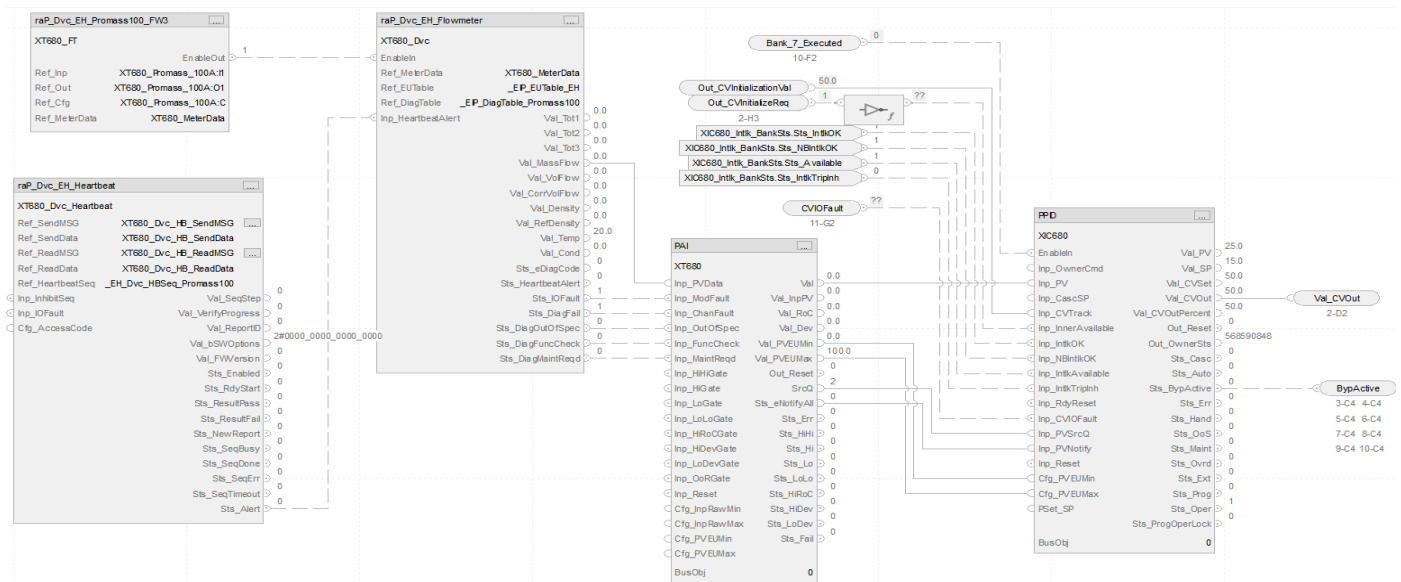


The CS\_PPID HART sheet operates the same as the CS\_PPID sheet but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAH\\_HART Sheet on page 149](#).
- Substitute XIC580 for the PV data instance of XT101
- Substitute XT580 for the remaining instances of XT101

For more information, see [HART Integration on page 61](#).

## CS\_PPID\_EtherNetIP Sheet

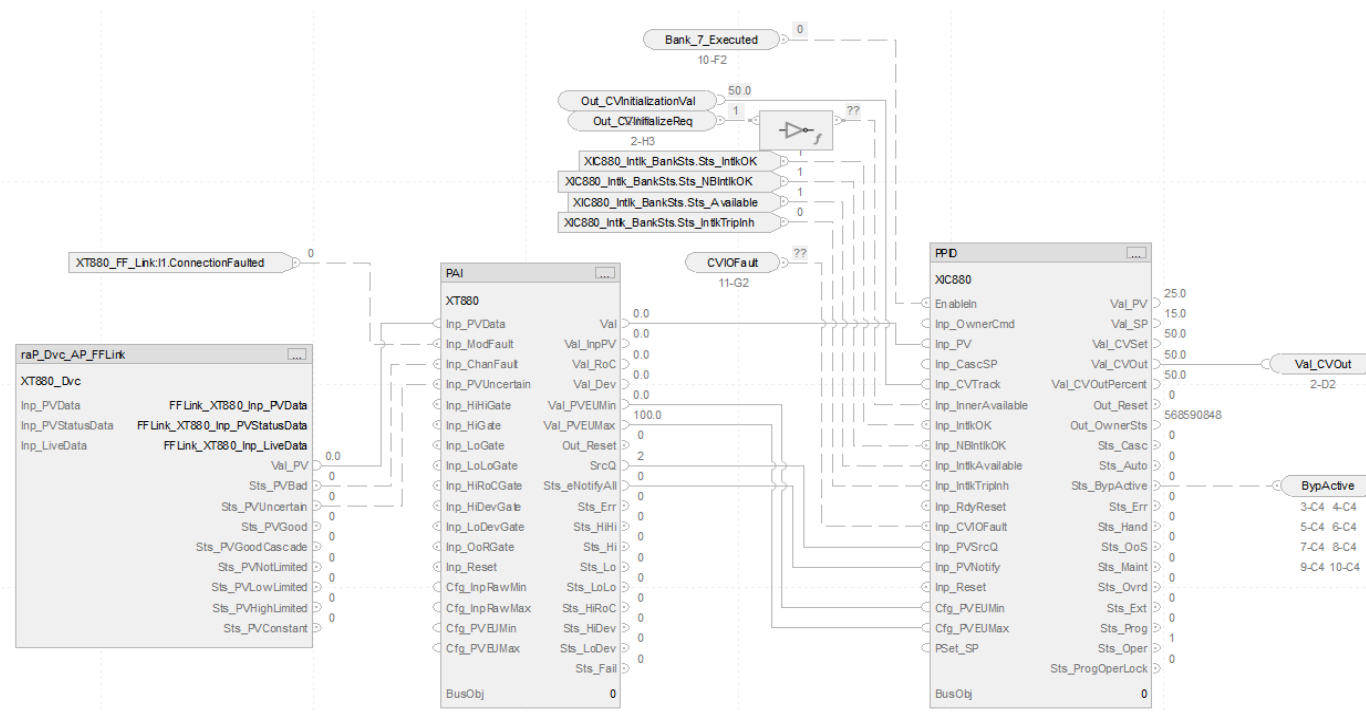


The CS\_PPID\_EtherNetIP sheet operates the same as the CS\_PPID sheet but relies on EtherNet/IP input data.

- For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).
- Substitute XIC680 for the PV data instance of XT101
- Substitute XT680 for the remaining instances of XT101

For more information, see [EtherNet/IP Integration on page 85](#).

## CS\_PPID\_FF Sheet



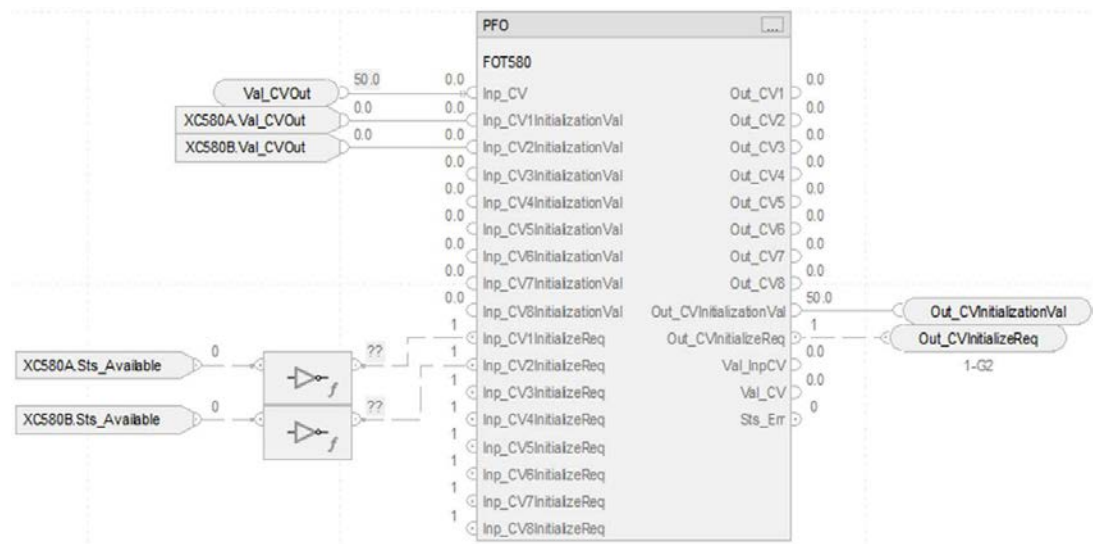
The CS\_PPID\_FF sheet operates the same as the CS\_PPID sheet but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).
- Substitute XIC880 for the PV data instance of XT101
- Substitute XT880 for the remaining instances of XT101

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).



PFO Sheet



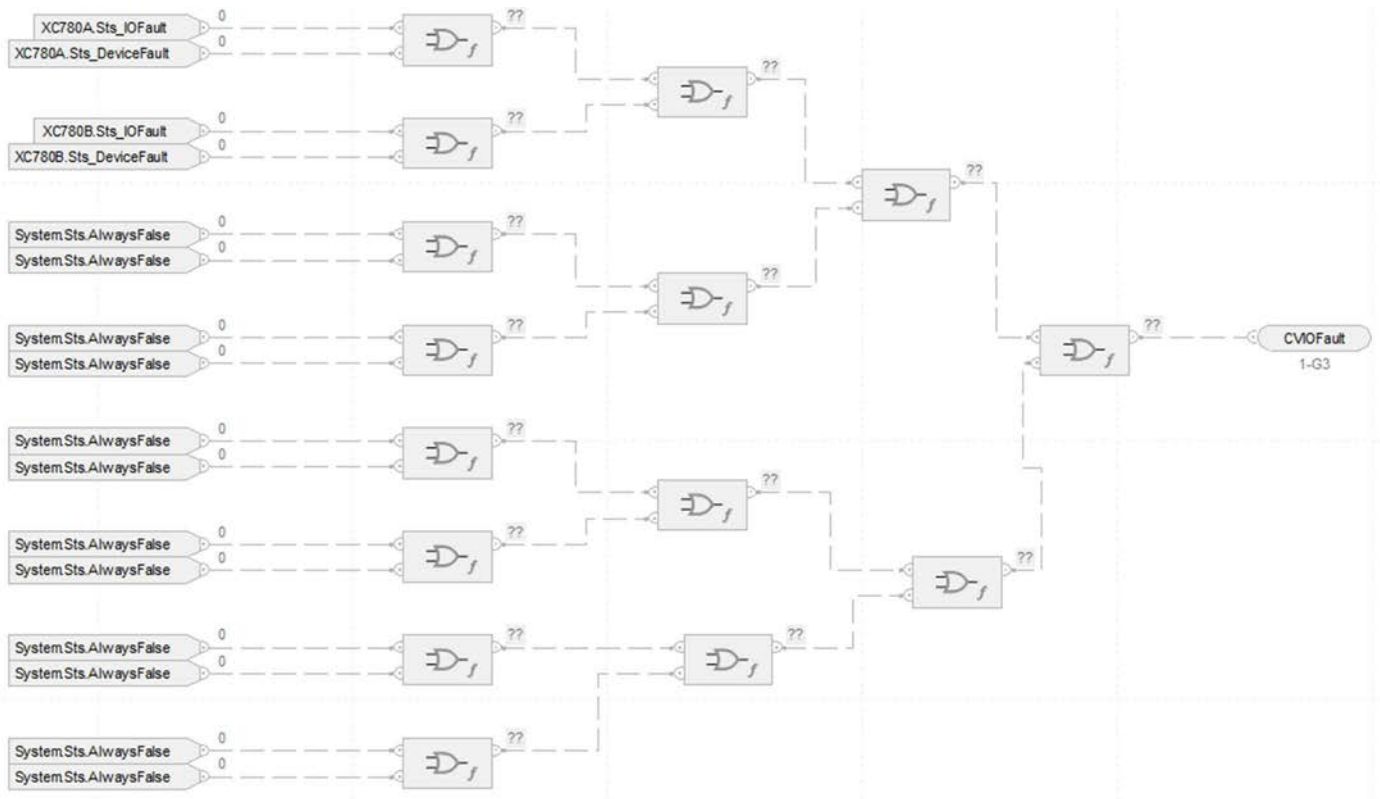
PFO Input References

Parameter	Description
Val_CVOut	Control Variable output Loop CV after clamping and ramping (CVEU)
XCxxxA.Val_CVOut XCxxxB.Val_CVOut	Value of CV Output after optional rate limiting, in engineering units. Extended Properties of this member: Engineering Unit - Engineering units (text) used for the analog output. Source: PAO instructions
XCxxxA.Sts_Available XCxxxB.Sts_Available	1 = Analog output available for control by program Source: PAO instructions

PFO Output References

Parameter	Description
Out_CVInitializationVal	Initialization value for PPID
Out_CVInitializeReq	Initialization request for PPID

## IO Faults Sheet



### Faults Input References

Parameter	Description
XCxxxA.Sts_IOFault XCxxxB.Sts_IOFault	1 = IO Fault Status Bad, 0 = OK  There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PAOTag.@Alarms.Alm_IOFault.AlarmElement
XCxxxB.Sts_DeviceFault XCxxxB.Sts_DeviceFault	1 = Device Fault Status Bad, 0=OK  There is a predefined default discrete Logix tag-based alarm for the status. Set standard configuration members of the discrete Logix tag-based alarm. Access alarm elements using this format: PAOTag.@Alarms.Alm_DeviceFault.AlarmElement
System.Sts.AlwaysFalse	raP_UDT_Opr_ System.Sts.AlwaysFalse

### Fault Output Reference

Parameter	Description
CVIOFault	Output connection to CS_PPID_SPLITRANGE sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for PPID with Split Range Control

Configure the I/O signal type via the IO.Signal.Type parameter first because the type affects the visibility of the remaining parameters in the PPID object.

## ACM-Based Parameters for a PPID Instance with Split Range Control

Parameter	Visible When	Details
<b>00 - Selection</b>		
IO_Signal_Type	always	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
CV_Selection_Type	always	Select the type of CV output as PAO or PVSD
Cfg_PAOSelection	CV_Selection_Type=PAO	Select the number of PAO objects (1...8) to connect.
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PFO_RefTag	always	Link to the analog fan output reference
PAI_Type	always	Select the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAI_RefTag	PAI_Type=PAI(Single_channel) PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	always	Link to the PV input reference
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference



Parameter	Visible When	Details
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PA(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PA(Single_channel)	Link to the gate reference
	PAI_RefTag is linked to an analog input reference PAI_Type=PA(Single_channel)	Link to the gate reference

**03.00 - IO Configuration**

Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.0x - IO Configuration**

Where x = 1...8

Inp_PosFdbkx	CV_Selection_Type = PAO Cfg_PAOSelction>=x	Link to the input position reference
PAOx_RefTag	CV_Selection_Type = PAO Cfg_PAOSelction>=x	Link to the analog output reference
Cfg_HasCVxIntlkObj	CV_Selection_Type = PAO Cfg_PAOSelction>=x	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
Bus_Instance_CVx	CV_Selection_Type = PAO Has_OOAP=True (controller parameter) Use_OOAP=True Cfg_PAOSelction>=x	Link to a bus array instance. This should be unique for each device

**04 - Alarm Configuration**

Cfg_HasHiHiDevAlm	always	If Cfg_HasHiHiDevAlm=True, ACM displays section 4.02 - Hi Hi Dev Alarm with additional parameters
Cfg_HasHiDevAlm	always	If Cfg_HasHiDevAlm=True, ACM displays section 4.03 - Hi Dev Alarm with additional parameters
Cfg_HasLoDevAlm	always	If Cfg_HasLoDevAlm=True, ACM displays section 4.04 - Lo Dev Alarm with additional parameters
Cfg_HasLoLoDevAlm	always	If Cfg_HasLoLoDevAlm=True, ACM displays section 4.05 - Lo Lo Dev Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.06 - Interlock Trip Alarm with additional parameters
Cfg_HasFailAlm	always	If Cfg_HasFailAlm=True, ACM displays section 4.01 - Input Failure Alarm with additional parameters

## Additional Sub-Objects for a PPID with Split Range Control Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
CV1Intlk CV2Intlk CV3Intlk CV4Intlk CV5Intlk CV6Intlk CV7Intlk CV8Intlk	Configure an interlock for the CV instance See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CV1Events CV2Events CV3Events CV4Events CV5Events CV6Events CV7Events CV8Events	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>

## Notes:



## Process Pressure/Temperature Compensated Flow (PPTC) Control Strategies

The Pressure/Temperature Compensated Flow (PPTC) instruction calculates the flow of a gas at standard / design temperature and pressure, essentially a mass flow rate, given a volumetric flow rate or differential pressure measurement. The design temperature and pressure are specific to each instrument, and since the thermodynamic conditions of the actual gas flow rarely align with the design conditions, temperature and pressure compensation is often used to compensate the actual flow measurement so that the measurement is adjusted to design conditions (essentially normalizing the gas flow to design conditions). This instruction requires measurements of the actual temperature and pressure of the flowing gas.

For the compensation to work correctly, like units for temperature and pressure must align and the final calculation is applied to absolute values. That is, `Inp_Pact` and `Cfg_PStd` must both be in the same units (`Cfg_POffset` is added to both values to convert to absolute pressure). Also, `Inp_Tact` and `Cfg_TStd` must both be in the same units (`Cfg_TOffset` is added to both values to convert to absolute temperature) \_

The PPTC Add-On Instruction is intended as a calculation function only, between other blocks, and no HMI components are provided. If a faceplate or alarms are needed, the calculated output from the instruction can be sent to a PAI (analog input) instruction for alarming and display.

The PPTC control strategy is available as two routines in the process library:







Routine	Description
FY101A	Linear flow transmitter
FY201A	DP Transmitter

### CS\_PPTC

- Logic and Tags
  - Parameters and Local Tags
    - MainRoutine
    - FY101A
    - FY201A







The PPTC HART control strategy is available as two routines in the process library:

Routine	Description
FY102A	HART linear flow transmitter
FY202A	HART DP Transmitter

-  CS\_PPTC\_HART
  -  Logic and Tags
    -  Parameters and Local Tags
    -  MainRoutine
    -  FY102A
    -  FY202A








The PPTC EtherNet/IP control strategy is available as two routines in the process library:

Routine	Description
FY103A	EtherNet/IP™ linear flow transmitter
FY203A	EtherNet/IP DP Transmitter

-  CS\_PPTC\_EtherNetIP
  -  Logic and Tags
    -  Parameters and Local Tags
    -  MainRoutine
    -  FY103A
    -  FY203A








The PPTC FOUNDATION Fieldbus control strategy is available as two routines in the process library:

Routine	Description
FY105A	FOUNDATION Fieldbus linear flow transmitter
FY205A	FOUNDATION Fieldbus DP Transmitter

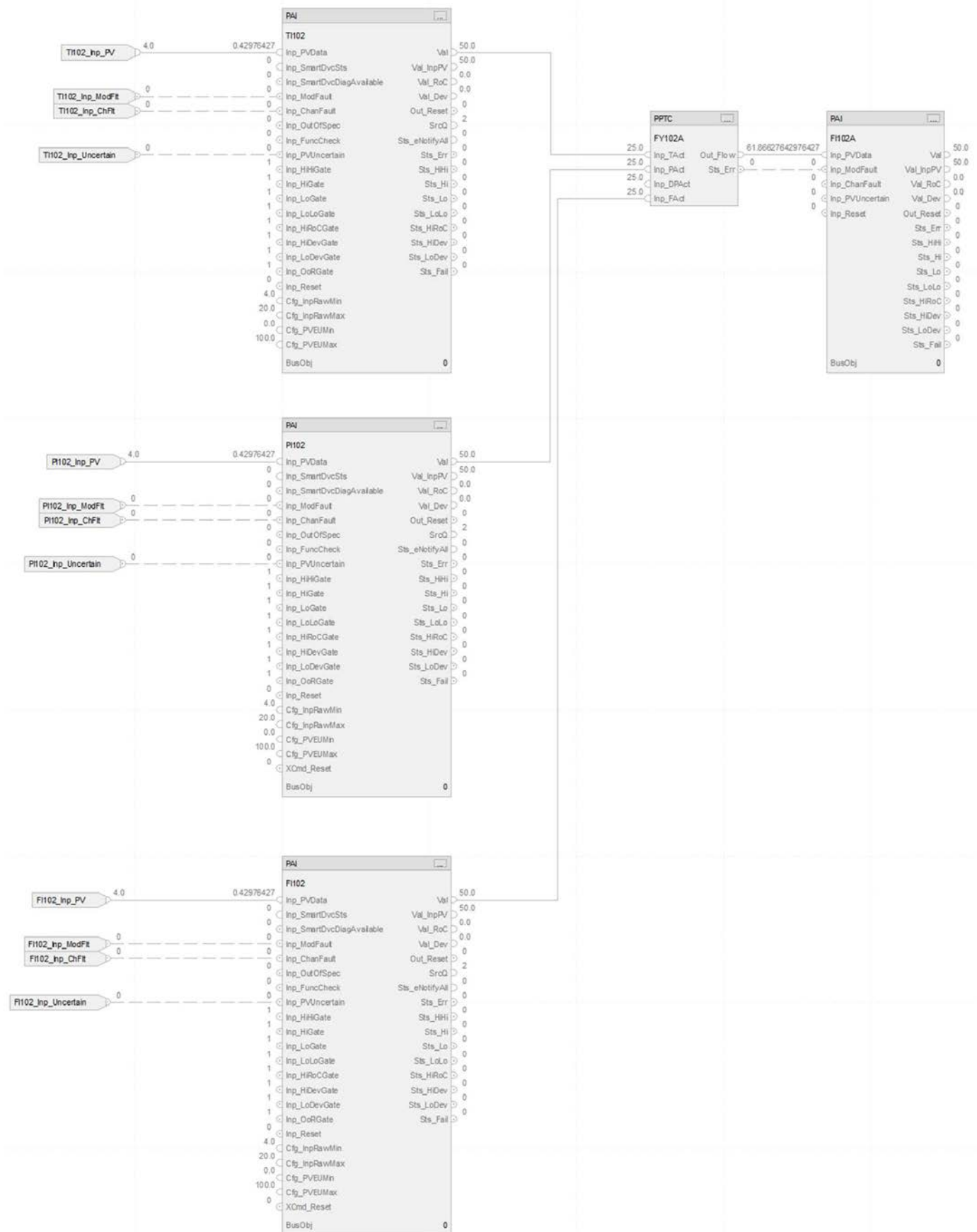
-  CS\_PPTC\_FF
  -  Logic and Tags
    -  Parameters and Local Tags
    -  MainRoutine
    -  FFLinkMap
    -  FY104A
    -  FY204A

The PPTC Profibus PA control strategy is available as two routines in the process library:

Routine	Description
FY104A	Profibus PA linear flow transmitter
FY204A	Profibus PA DP Transmitter

-  CS\_PPTC\_PA
  -  Logic and Tags
    -  Parameters and Local Tags
    -  MainRoutine
    -  FY105A
    -  FY205A
    -  PALinkMap

## CS\_PPTC Sheet



## PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

PAI Instruction	Description	Substitute the Desired Instrument Name for:
PPTC Inp_TAct	Actual (measured) temperature	TI101/TI201
PPTC Inp_PAct	Actual (measured) pressure	PI101/PI201
PPTC Inp_DPAct	Actual (measured) differential pressure (square root)	PDIT101/PDIT201
PPTC Inp_FAct	Actual (measured) uncompensated flow (linear)	FI101/FI201

To configure the flow calculation method, see the Advanced properties page for the PPTC instruction. Select one of the following:

- Differential pressure (PPTC Inp\_DPAct)
- Flow input (PPTC Inp\_FAct)

## PAI Outputs

Parameter	Description
Val for PPTC Inp_TAct	Actual (measured) temperature
Val for PPTC Inp_PAct	Actual (measured) pressure
Val for PPTC Inp_DPAct	Actual (measured) differential pressure
Val PPTC Inp_FAct	Actual (measured) uncompensated flow

## PPTC Outputs

Parameter	Description
Out_Flow	Compensated flow (at standard temperature and pressure: mass flow)
Sts_Err	1 = Error in configuration: See detail bits (Sts_Errxxx) for reason

## PPTC Configuration Considerations

Operand	Type	Description
PlantPax® control	P_PRESS_TEMP_COMPENSATED	Instance of data structure (backing tag) required for proper operation of instruction



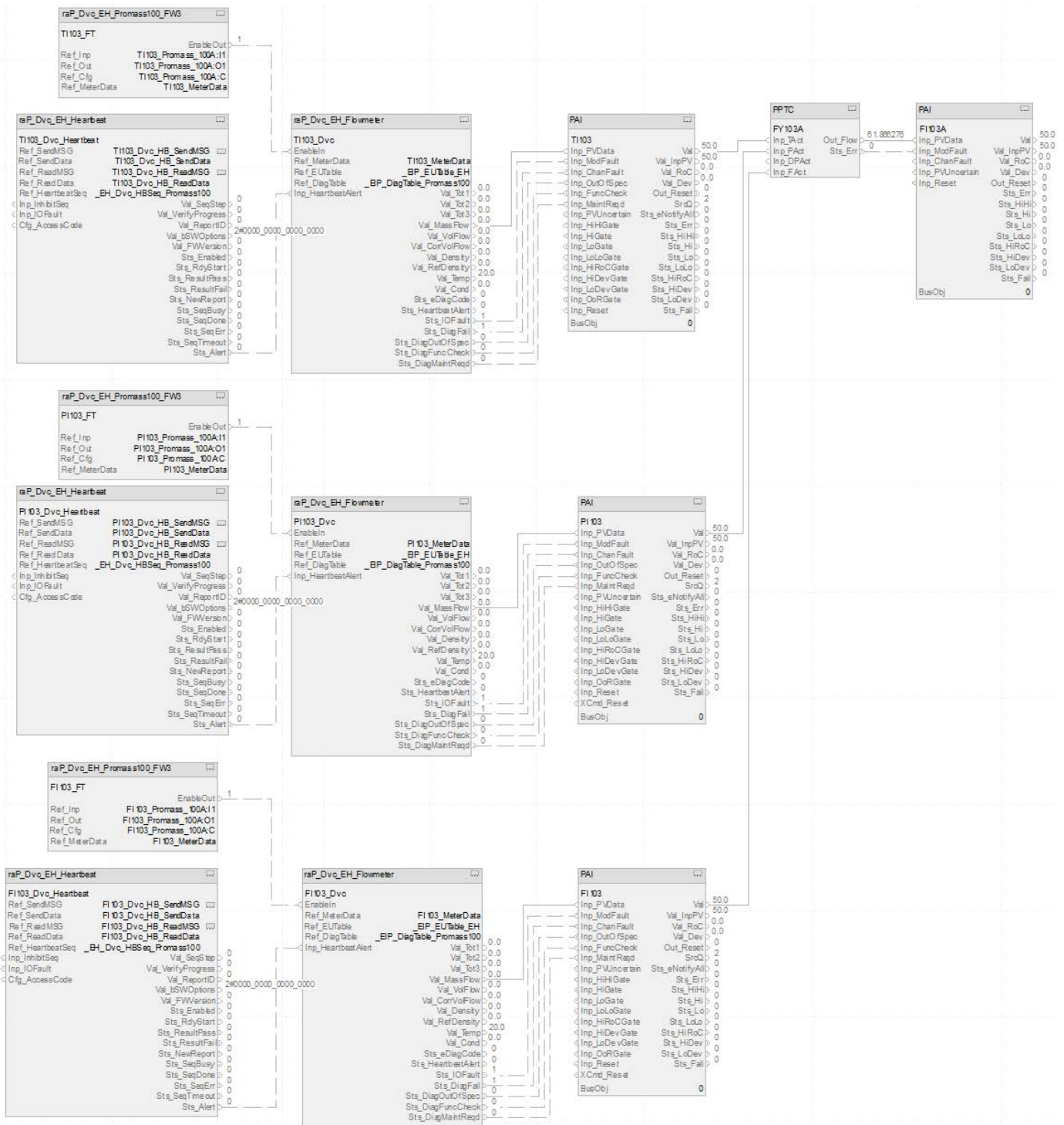


The CS\_PPTC\_HART control strategy operates the same as the CS\_PPTC control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- For more information, see [HART Integration on page 61](#).

PAI Instruction	Description	Substitute the Desired Instrument Name for:
PPTC Inp_TAct	Actual (measured) temperature	TI102/TI202
PPTC Inp_PAct	Actual (measured) pressure	PI102/PI202
PPTC Inp_DPAct	Actual (measured) differential pressure	PDTI102/PDIT202
PPTC Inp_FAct	Actual (measured) uncompensated flow	FI102/FI202

## CS\_PPTC\_EtherNetIP Sheet



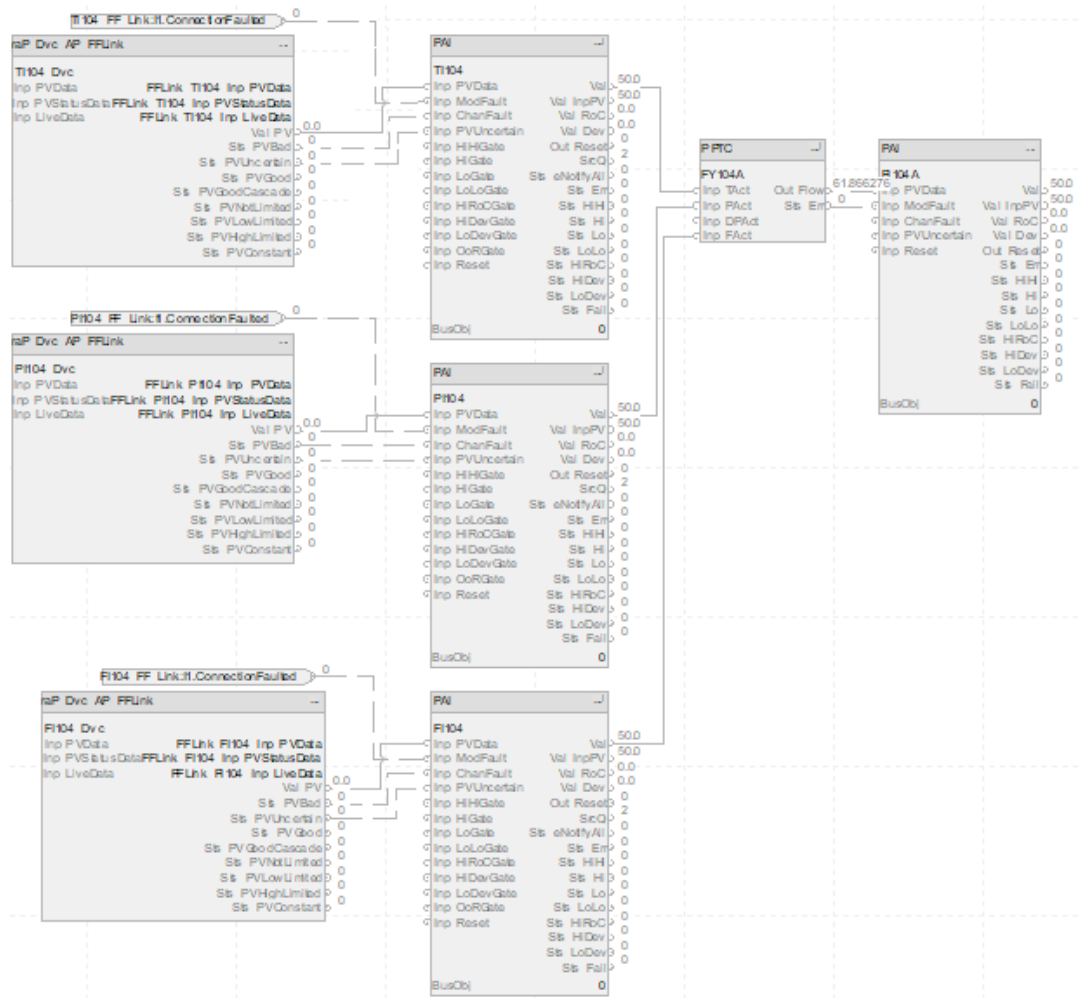
The CS\_PPTC\_EtherNetIP control strategy operates the same as the CS\_PPTC control strategy but relies on EtherNet/IP input data.

For information on EtherNet/IP device outputs to PAI inputs, see [CS\\_PAI\\_EtherNetIP Sheet on page 151](#).

For more information, see [EtherNet/IP Integration on page 85](#).

PAI Instruction	Description	Substitute the Desired Instrument Name for:
PPTC Inp_TAct	Actual (measured) temperature	TI103/TI203
PPTC Inp_PAct	Actual (measured) pressure	PI103/PI203
PPTC Inp_DPAct	Actual (measured) differential pressure	PDTI103/PDIT203
PPTC Inp_FAct	Actual (measured) uncompensated flow	FI103/FI203

## CS\_PPTC\_FF Sheet



The CS\_PPTC\_FF control strategy operates the same as the CS\_PPTC control strategy but relies on FOUNDATION Fieldbus input data.

For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PAI\\_FF Sheet on page 155](#).

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

PAI Instruction	Description	Substitute the Desired Instrument Name for:
PPTC Inp_TAct	Actual (measured) temperature	TI104/TI204
PPTC Inp_PAct	Actual (measured) pressure	PI104/PI204
PPTC Inp_DPAct	Actual (measured) differential pressure	PDT1104/PDIT204
PPTC Inp_FAct	Actual (measured) uncompensated flow	FI104/FI204



## ACM Considerations for PPTC

Configure these parameters first because they affect the visibility of the remaining parameters in the PPTC object.

- Specify the type of analog input via the PAI\_Type parameter
- Specify Cfg\_UseDP=True to use differential pressure to calculate flow or Cfg\_UseDP=False to use a flowmeter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

### ACM-Based Parameters for a PPTC Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters.  Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAIOut_RefTag	always	Link to the PAI output reference
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Cfg_UseDP	always	Set to use Inp_DPact (square root curve) to calculate flow Clear to use Inp_Fact (linear) to calculate flow
Cfg_HasHARTTI	PAI_Type = PAI(Single_channel) IO_Signal_Type=HART	Set to use a HART device PAI PVTI input
Cfg_HasHARTPI	PAI_Type = PAI(Single_channel) IO_Signal_Type=HART	Set to use a HART device PAI PVPI input
Cfg_HasHARTPDIT	PAI_Type = PAI(Single_channel) Cfg_UseDP=True IO_Signal_Type=HART	Set to use a HART device PAI PVDIT input
Cfg_HasHARTFI	PAI_Type = PAI(Single_channel) Cfg_UseDP=False IO_Signal_Type=HART	Set to use a HART device PAI PVFI input
Cfg_HasEHTI	PAI_Type = PAI(Single_channel) IO_Signal_Type=EH_EtherNetIP	Set to use an E+H EtherNet/IP device PAI PVTI input
Cfg_HasEHPI	PAI_Type = PAI(Single_channel) IO_Signal_Type= EH_EtherNetIP	Set to use an E+H EtherNet/IP device PAI PVPI input
Cfg_HasEHPDIT	PAI_Type = PAI(Single_channel) Cfg_UseDP=True IO_Signal_Type=EH_EtherNetIP	Set to use an E+H EtherNet/IP device PAI PVDIT input
Cfg_HasEHFI	PAI_Type = PAI(Single_channel) Cfg_UseDP=False IO_Signal_Type=EH_EtherNetIP	Set to use an E+H EtherNet/IP device PAI PVFI input
Cfg_HasFFTI	PAI_Type = PAI(Single_channel) IO_Signal_Type=FF	Set to use a FOUNDATION Fieldbus device PAI PVTI input
Cfg_HasFFPI	PAI_Type = PAI(Single_channel) IO_Signal_Type=FF	Set to use a FOUNDATION Fieldbus device PAI PVPI input
Cfg_HasFFPDIT	PAI_Type = PAI(Single_channel) Cfg_UseDP=True IO_Signal_Type=FF	Set to use a FOUNDATION Fieldbus device PAI PVDIT input
Cfg_HasFFFI	PAI_Type = PAI(Single_channel) Cfg_UseDP=False IO_Signal_Type=FF	Set to use a FOUNDATION Fieldbus device PAI PVFI input
Cfg_HasPATI	PAI_Type = PAI(Single_channel) IO_Signal_Type=PA	Set to use a Profibus PA device PAI PVTI input
Cfg_HasPAPI	PAI_Type = PAI(Single_channel) IO_Signal_Type=PA	Set to use a Profibus PA device PAI PVPI input
Cfg_HasPAPDIT	PAI_Type = PAI(Single_channel) Cfg_UseDP=True IO_Signal_Type=PA	Set to use a Profibus PA device PAI PVDIT input
Cfg_HasPAFI	PAI_Type = PAI(Single_channel) Cfg_UseDP=False IO_Signal_Type=PA	Set to use a Profibus PA device PAI PVFI input

Parameter	Visible When	Details
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
01 - Options		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART Cfg_HasHARTTI=True	Set to use HART Digital Data for the PV, SV, TV, and FV values
	IO_Signal_Type=HART Cfg_HasHARTPI=True	
	IO_Signal_Type=HART Cfg_HasHARTPDIT=True	
	IO_Signal_Type=HART Cfg_HasHARTFI=True	
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_HasHARTTI=True Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
	IO_Signal_Type=HART Cfg_HasHARTPI=True Cfg_UseHARTDigitalData=False	
	IO_Signal_Type=HART Cfg_HasHARTPDIT=True Cfg_UseHARTDigitalData=False	
	IO_Signal_Type=HART Cfg_HasHARTFI=True Cfg_UseHARTDigitalData=False	
Hart_Type	IO_Signal_Type=HART Cfg_HasHARTTI=True Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
	IO_Signal_Type=HART Cfg_HasHARTPI=True Cfg_UseHARTDigitalData=False	
	IO_Signal_Type=HART Cfg_HasHARTPDIT=True Cfg_UseHARTDigitalData=False	
	IO_Signal_Type=HART Cfg_HasHARTFI=True Cfg_UseHARTDigitalData=False	
03.01 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PAITL_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_TL_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_TL_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PVTI	PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV TI input reference
	PAI_Type=ExternalPAI(Single_channel) IO_SignalType=HART	
Ref_HartDeviceTI	PAI_Type=PAI(Single_channel) IO_Signal_Type=HART Cfg_UseHARTDigitalData=False Cfg_HasHARTTI=True	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModuleTI	IO_Signal_Type=EH_EthernetIP Cfg_HasEHTI=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_TL_Module	IO_Signal_Type=FF Cfg_HasFFTI=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_TL_Module	IO_Signal_Type=PA Cfg_HasPATI=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
03.01.10 - Ref PAI Alarm Configuration		



Parameter	Visible When	Details
Ref_HiHiGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_0oRGateTI	PAITI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

### 03.02 - IO Configuration

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

PAIPI_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_PL_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_PL_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PVPI	PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV PI input reference
	PAI_Type=PAI(Single_channel) IO_SignalType=HART	
Ref_HartDevicePI	PAI_Type=PAI(Single_channel) IO_SignalType=HART Cfg_UseHARTDigitalData=False Cfg_HasHARTPI=True	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModulePI	IO_SignalType=EH_EthernetIP Cfg_HasEHPI=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_PI_Module	IO_SignalType=FF Cfg_HasFFPI=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_PI_Module	IO_SignalType=PA Cfg_HasPAPI=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

### 03.02.10 - Ref PAI Alarm Configuration

Ref_HiHiGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_HiRoCGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGatePI	PAIPI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

**03.03 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

PAIPDIT_Ref_Tag	Cfg_UseDP=True PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_UseDP=True PAI_Type=ExternalPAI(Single_channel)	
PAID_PDIT_Ref_Tag	Cfg_UseDP=True PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_PDIT_Ref_Tag	Cfg_UseDP=True PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PVPDIT	PAI_Type=PAI(Single_channel) Cfg_UseDP=True IO_SignalType=None	Link to the PV PDIT input reference
	PAI_Type=PAI(Single_channel) Cfg_UseDP=True IO_SignalType=HART	
Ref_HartDevicePDIT	PAI_Type=PAI(Single_channel) Cfg_UseDP=True IO_SignalType=HART Cfg_UseHARTDigitalData=False Cfg_HasHARTPDIT=True	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModulePDIT	IO_Signal_Type=EH_EthernetIP Cfg_UseDP=True Cfg_HasEHPDIT=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_PDIT_Module	IO_Signal_Type=FF Cfg_UseDP=True Cfg_HasFFPDIT=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_PDIT_Module	IO_Signal_Type=PA Cfg_UseDP=True Cfg_HasPAPDIT=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.03.10 - Ref PAI Alarm Configuration**

Ref_HiHiGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_HiGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_LoGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_LoLoGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_HiRoCGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference

Parameter	Visible When	Details
Ref_HiDevGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_LoDevGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference
Ref_OoRGatePDIT	PAIPDIT_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=True	Link to the gate reference

**03.04 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See [I/O Mapping on page 38](#).

PAIFI_Ref_Tag	Cfg_UseDP=False PAI_Type=PAI(Single_channel)	Link to the analog input reference
	Cfg_UseDP=False PAI_Type=ExternalPAI(Single_channel)	
PAID_FI_Ref_Tag	Cfg_UseDP=False	Link to the analog input (dual channel) reference
	PAI_Type=PAI(Dual_channel)	
PAIM_FI_Ref_Tag	Cfg_UseDP=False PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PVFI	PAI_Type=PAI(Single_channel) Cfg_UseDP=False IO_SignalType=None	Link to the PV PDIT input reference
	PAI_Type=PAI(Single_channel) Cfg_UseDP=False IO_SignalType=HART	
Ref_HartDeviceFI	PAI_Type=PAI(Single_channel) IO_SignalType=HART Cfg_UseHARTDigitalData=False Cfg_UseDP=False Cfg_HasHARTFI=True	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModuleFI	IO_SignalType=EH_EthernetIP Cfg_UseDP=False Cfg_HasEHFI=True	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_FI_Module	IO_SignalType=FF Cfg_UseDP=False Cfg_HasFFFI=True	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_FI_Module	IO_SignalType=PA Cfg_UseDP=False Cfg_HasPAFI=True	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects

**03.03.10 - Ref PAI Alarm Configuration**

Ref_HiHiGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_HiGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_LoGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_LoLoGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_HiRoCGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference

Parameter	Visible When	Details
Ref_HiDevGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_LoDevGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference
Ref_OoRGateFI	PAIFI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) Cfg_UseDP=False	Link to the gate reference

## Additional Sub-Object for a PPTC Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

## Process Tank Strapping Table (PTST) Control Strategies

Use a PTST control strategy to calculate the volume of product in an upright cylindrical tank, given the level of the product and the tank calibration table. The instruction can compensate for:

- Free water at the bottom of the tank, given a product/water interface level.
- Thermal expansion of the tank shell, given the coefficient of linear expansion of the shell material and product and ambient temperatures.
- A floating tank roof, given the product density is provided.

The PTST instruction is intended only as a calculation function, between other blocks, and so no HMI components are provided.

The PTST control strategy is available as two routines in the process library:

Routine	Description
Q1101	Level input with no compensation
Q1201	Level input with compensation inputs exposed

### CS\_PTST

Parameters and Local Tags

MainRoutine

Q1101

Q1201

The PTST HART control strategy is available as two routines in the process library:

Routine	Description
Q1102	HART level input with no compensation
Q1202	HART level input with compensation inputs exposed

### CS\_PTST\_HART

Parameters and Local Tags

MainRoutine

Q1102

Q1202

The PTST FOUNDATION Fieldbus control strategy is available as two routines in the process library:

Routine	Description
QI104	FOUNDATION Fieldbus level input with no compensation
QI204	FOUNDATION Fieldbus level input with compensation inputs exposed

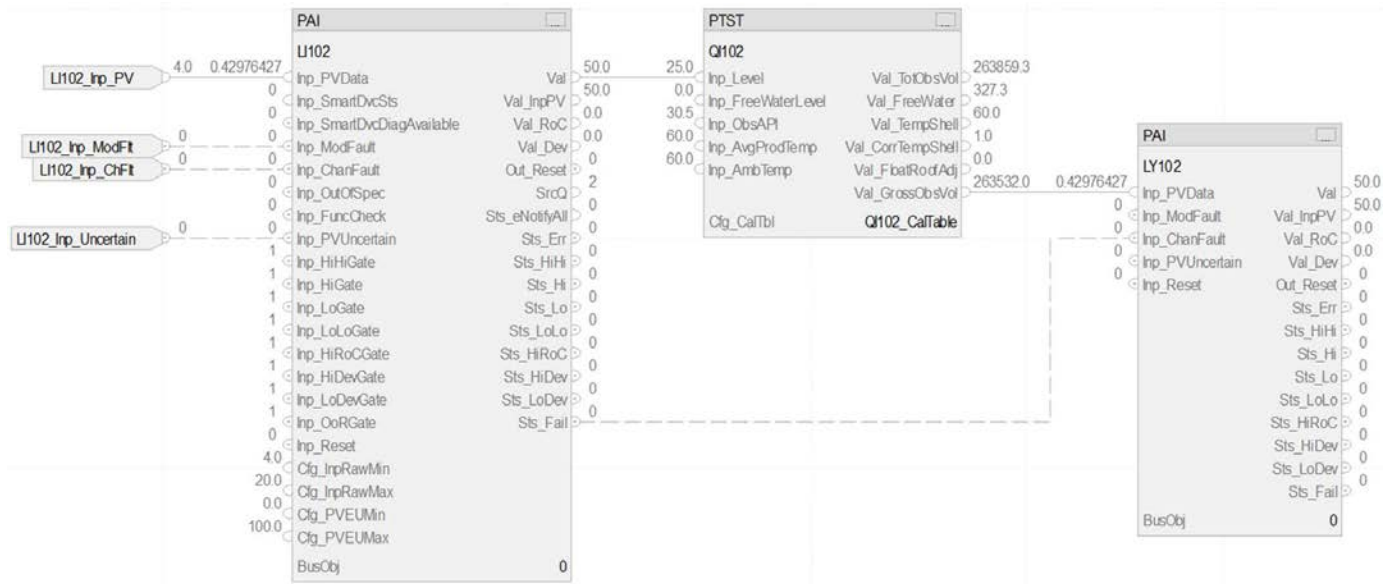
- CS\_PTST\_FF
  - Parameters and Local Tags
    - MainRoutine
    - FFLinkMap
    - QI104
    - QI204

The PTST Profibus PA control strategy is available as two routines in the process library:

Routine	Description
QI105	Profibus PA level input with no compensation
QI205	Profibus PA level input with compensation inputs exposed

- CS\_PTST\_PA
  - Parameters and Local Tags
    - MainRoutine
    - PALinkMap
    - QI105
    - QI205

## CS\_PTST Sheet



### PAI Input References

See [CS\\_PA1 Sheet on page 148](#) for details.

- Substitute Q101/Q1201 with the desired tag name
- Substitute LI101/LI201 with the desired instrument tag name
- Substitute LY101/LY201 with the desired tag name

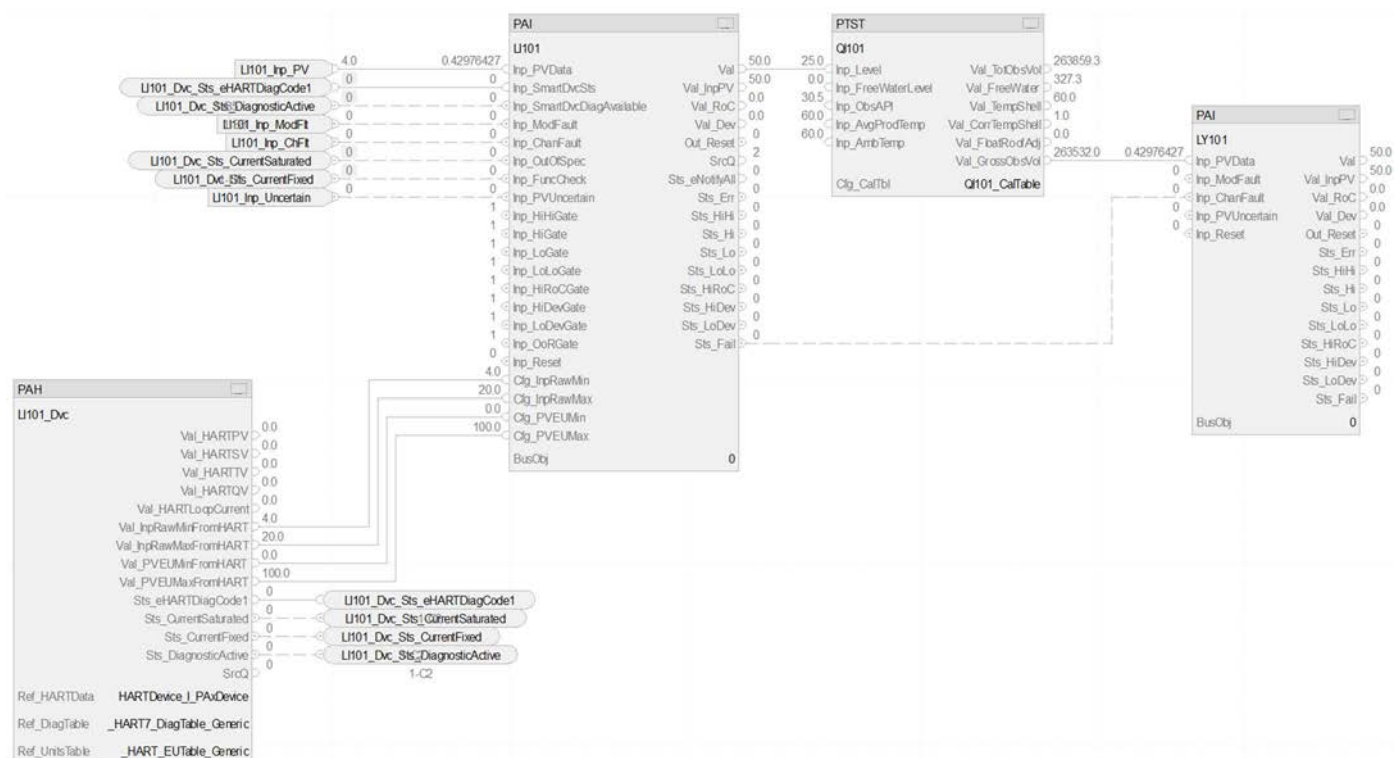
### PAI Outputs

Parameter	Description
Val	Input to Inp_Level of PTST instruction Tank innage level, in feet or meters
Sts_Fail	Input to Inp_ChanFault of secondary PAI instruction 1 = I/O channel fault or failure 0 = OK

### PTST Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_TANK_STRAPPING_TABLE	Instance of data structure (backing tag) required for proper operation of instruction
Cfg_CalTbl	P_STRAPPING_TABLE_ROW	Array for tank calibration table, level to volume

## CS\_PTST\_HART Sheet



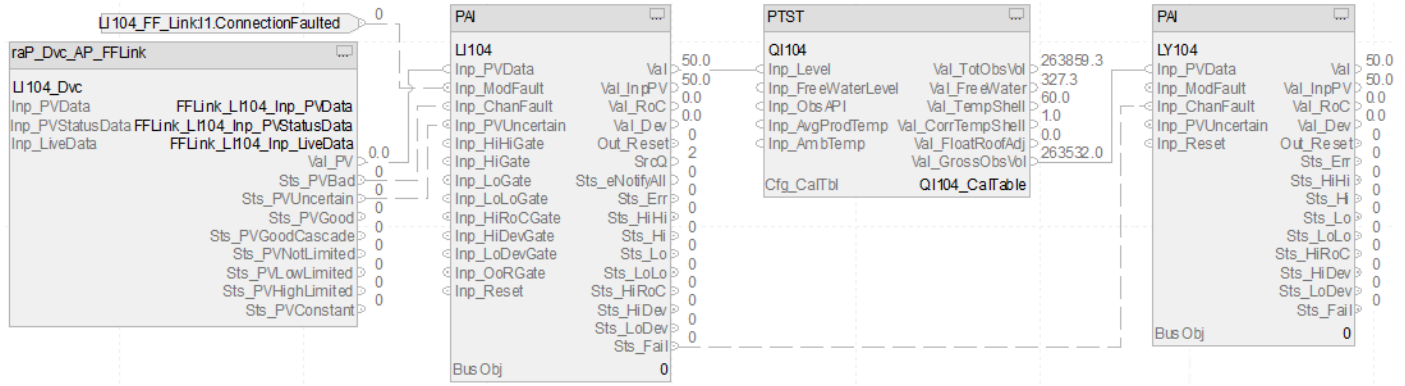
The CS\_PTST\_HART control strategy operates the same as the CS\_PTST control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute Q102/Q1202 with the desired tag name
- Substitute LI102/LI202 with the desired instrument tag name
- Substitute LY102/LY202 with the desired tag name.

For more information, see [HART Integration on page 61](#).



## CS\_PTST\_FF Sheet

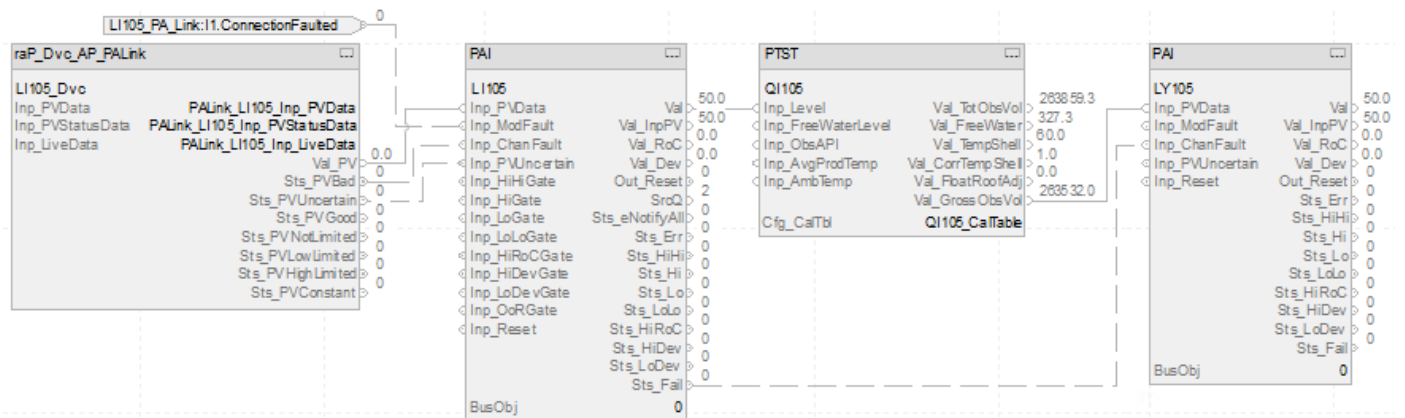


The CS\_PTST\_FF control strategy operates the same as the CS\_PTST control strategy but relies on FOUNDATION Fieldbus input data.

- For information on Foundation Fieldbus device outputs to PAI inputs, see [CS\\_PA\\_FF Sheet on page 155](#).
- Substitute QI103/QI203 with the desired tag name
- Substitute LI103/LI203 with the desired instrument tag name
- Substitute LY103/LY203 with the desired tag name.

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## CS\_PTST\_PA Sheet



The CS\_PTST\_PA control strategy operates the same as the CS\_PTST control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PA\\_PA Sheet on page 156](#).
- Substitute QI104/QI204 with the desired tag name
- Substitute LI104/LI204 with the desired instrument tag name
- Substitute LY104/LY204 with the desired tag name.

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

## ACM Considerations for PTST

Configure these parameters first because they affect the visibility of the remaining parameters in the PTST object.

- Specify the type of analog input via the PAI\_Type parameter
- If you use a specific I/O signal type, select the type for the IO\_Signal\_Type parameter

## ACM-Based Parameters for a PTST Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
PAI_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters.  Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
PAIOut_RefTag	always	Link to the PAI output reference
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH instruction
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
<b>02 - Device Configuration</b>		
Cfg_HasCorrTempShell	always	Set to enable correction for temperature
Ref_AmbTemp	PAI_Type=PAI(Single_channel) Cfg_HasCorrTempShell=True	Link to the ambient temperature input reference
Ref_AvgProdTemp	PAI_Type=PAI(Single_channel) Cfg_HasCorrTempShell=True	Link to the average product temperature input reference
Cfg_HasFloatRoofAdj	always	Set to use floating roof adjustment
Ref_ObsAPI	PAI_Type=PAI(Single_channel) Cfg_HasFloatRoofAdj=True	Link to the observed density reference
<b>03.00 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PAI_RefTag	PAI_Type=PAI(Single_channel) PAI_Type=ExternalPAI(Single_channel)	Link to the analog input reference
PAID_RefTag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_RefTag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None IO_SignalType=HART	Link to the PV input reference
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
<b>03.00.10 - Ref PAI Alarm Configuration</b>		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

## Additional Sub-Object for a PTST Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>

**Notes:**

## Process Valve Hand Operated (PVLVHO) Control Strategy

Use the PVLVHO control strategy to monitor a hand (locally) operated valve and display its current state. The valve can have any type of actuator – handwheel, lever, motor, solenoid, pneumatic, hydraulic – but it is normally operated at the valve and only monitored by the control system via open and closed limit switches.

This PVLV control strategy does not provide operator access to control the valve, but it does provide an optional Trip output. The Trip state is generated by interlock conditions not being met and the output can be used to de-energize a valve control circuit to drive the valve to its default (fail) position. If the trip function is used, the PVLV instruction checks to make sure that the valve reaches the configured trip position (open or closed) if a trip command is executed.

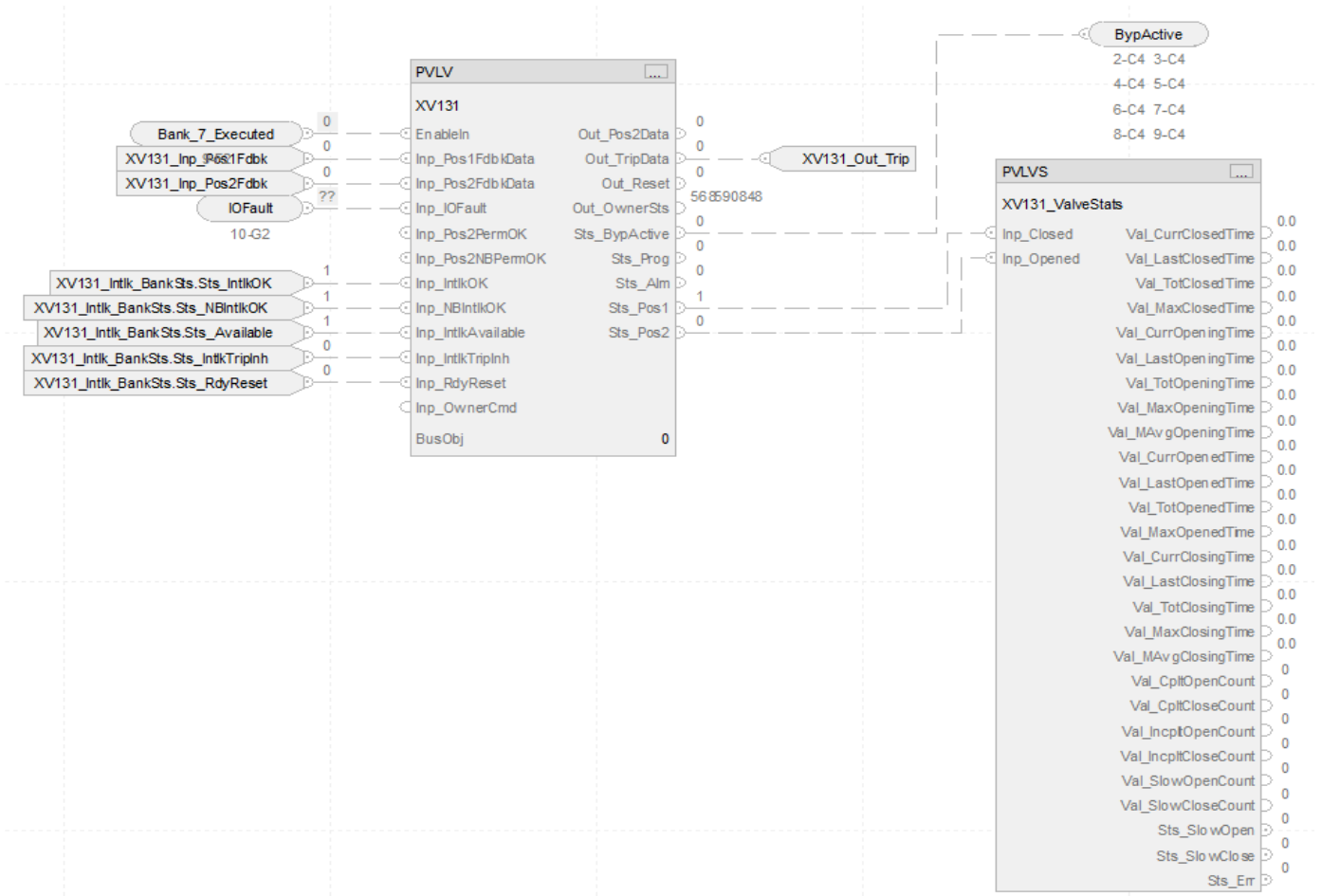
The CS\_PVLVHO control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PVLVHO control strategy contains these Function Block sheets:

Sheet	Description
CS_PVLVHO	Process Valve instruction, hand operated
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PVLV instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors one analog output channel for I/O fault input and raises an alarm on an I/O fault.

CS\_PVLVH0 Sheet



PVLV Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XV131_Inp_Pos1Fdbk	Feedback from Position 1 limit switch of the device 1 = Device confirmed Position 1
XV131_Inp_Pos2Fdbk	Feedback from Position 2 limit switch of the device 1 = Device confirmed Position 2
IOFault	Input connection from IO Faults sheet
XV131_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
XV131_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XV131_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XV131_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
XV131_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

## PVLV Output References

Parameter	Description
XV131_Out_Trip	1 = Trip valve to safe/fail state
BypActive	Output connection to interlock bank sheet

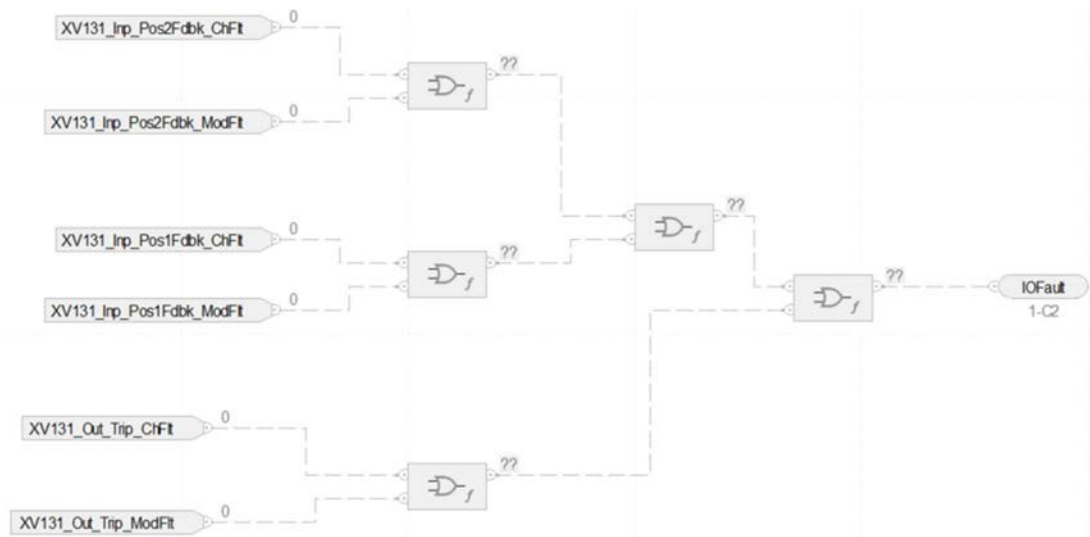
## PVLV Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_VALVE_DISCRETE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

## PVLV Output References to PVLVS

Parameter	Description
Sts_Pos1	1 = Valve requested to Position 1 and is confirmed Position 1
Sts_Pos2	1 = Valve requested to Position 1 and is confirmed Position 2

## IO Faults Sheet



## Fault Input References

Parameter	Description
XV131_Inp_Pos1Fdbk_ChFit	Tieback input 1 channel fault
XV131_Inp_Pos1Fdbk_ModFit	Tieback input 1 module fault
XV131_Inp_Pos2Fdbk_ChFit	Tieback input 2 channel fault
XV131_Inp_Pos2Fdbk_ModFit	Tieback input 2 module fault
XV131_Out_Trip_ChFit	Output channel fault
XV131_Out_Trip_ModFit	Output module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PVLVHO sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).



## ACM Considerations for a PVLV Instance

Configure the valve type via the ACM\_Type parameter:

- Solenoid-operated valve (P\_ValveS0)
- Motor-operated valve (P\_ValveM0)
- Hand-operated valve (P\_ValveH0)
- Mixer-proof valve (P\_ValveMP)

### ACM-Based Parameters for a PVLV Instance

Parameter	Visible When	Details
00 - Selection		
ACM_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters. Define the PVLV type: P_ValveS0, P_ValveM0, P_ValveH0 or P_ValveMP
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Has_OOAP=True (controller parameter)	Set to use the ArbitrationQ instruction for ownership queuing.
	Use_OOAP=True	See <a href="#">Process Controller on page 36</a>
01 - Options		
Bus_Instance	Has_OOAP=True (controller parameter)	Link to a bus array instance. This should be unique for each device
	Use_OOAP=True	See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasPos1PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Set to create an instance of the PPERM instruction to allow an Out_Pos1 command.
	ACM_Type= P_ValveM0	See <a href="#">Permissives on page 50</a>
Cfg_HasPos2PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Set to create an instance of the PPERM instruction to allow an Out_Pos2 command.
	ACM_Type= P_ValveM0	See <a href="#">Permissives on page 50</a>
Cfg_HasStop	ACM_Type=P_ValveM0	Set if the motorized valve has a stop output or if simultaneously de-energizing Out_Pos1 and Out_Pos2 is necessary.
Cfg_HasTrip	ACM_Type=P_ValveH0	Set if a trip output exists for the device (such as Horn or Light)
Cfg_HasOpenIntlkObj Cfg_HasUpperSeatIntlkObj Cfg_HasLowerSeatIntlkObj Cfg_HasCavityIntlkObj	ACM_Type=P_ValveMP	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Interlocks on page 49</a>
02 - Device Configuration		
Cfg_FailPos2	ACM_Type=P_ValveS0	Set if Position 2 is the Fail Position of the device
	ACM_Type= P_ValveM0	
Cfg_HasStatsObj	always	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Valve Statistics on page 58</a>
02.01 - Device Configuration		
Cfg_HasPos1Fdbk	ACM_Type=P_ValveS0	Set if Position 1 feedback exists
	ACM_Type= P_ValveM0	
	ACM_Type= P_ValveH0	
Cfg_HasPos2Fdbk	ACM_Type=P_ValveS0	Set if Position 2 feedback exists
	ACM_Type= P_ValveM0	
	ACM_Type= P_ValveH0	
03 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		

Parameter	Visible When	Details
Inp_Pos1Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos1Fdbk=True	Link the position 1 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos1Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos1Fdbk=True	
Inp_Pos2Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos2Fdbk=True	Link the position 2 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos2Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos2Fdbk=True	
Out_Pos1	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Link the position 1 output reference
	ACM_Type= P_ValveM0 Cfg_FailPos2=True	
Out_Pos2	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Link the position 2 output reference
	ACM_Type=P_ValveM0 Cfg_FailPos2=False	
Inp_OpenLS Inp_ClosedLS Inp_LowerLS Inp_UpperLS Inp_CavityInLS Inp_Cavity_OutLS	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding input configuration parameter to the value defined by each parameter
Out_Open Out_Close Out_LiftLower Out_LiftUpper Out_CavityIn Out_CavityOut	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding output configuration parameter to the value defined by each parameter
<b>03.01 - IO Configuration VLVM0</b>		
Out_Stop	ACM_Type=P_ValveM0 Cfg_HasStop=True	Link the Stop output reference
<b>03.02 - IO Configuration VLVH0</b>		
Out_Trip	ACM_Type=P_ValveH0 Cfg_HasTrip=True	Link the Trip output reference
<b>04 - Alarm Configuration</b>		
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasActuatorFaultAlm	ACM_Type=P_ValveS0	If Cfg_HasActuatorFaultAlm=True, ACM displays section 4.01- Interlock Trip Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasFullStallAlm	ACM_Type=P_ValveS0	If Cfg_HasFullStallAlm=True, ACM displays section 4.04 - Full Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTransitStallAlm	ACM_Type=P_ValveS0	If Cfg_HasTransitStallAlm=True, ACM displays section 4.05 - Transit Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTripFailAlm	ACM_Type=P_ValveH0	If Cfg_HasTripFailAlm=True, ACM displays section 4.07- Trip Fail Alarm with additional parameters
Cfg_HasLossPos1Alm	ACM_Type=P_ValveS0	If Cfg_HasLossPos1Alm=True, ACM displays section 4.06 - Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	

Parameter	Visible When	Details
Cfg_HasLossPos2Alm	ACM_Type=P_ValveSO	If Cfg_HasLossPos2Alm=True, ACM displays section 4.08- Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasFailAlm	ACM_Type=P_ValveMP	If Cfg_HasFailAlm=True, ACM displays section 4.09 - Fail Alarm with additional parameters

## Additional Sub-Objects for a PVLV Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_1 Permissive_2	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Interlock_Lower Interlock_Upper Interlock_Cavity	Configure interlocks for the positions See <a href="#">Interlocks on page 49</a>

**Notes:**

## Process Valve Motor Operated (PVLVMO) Control Strategy

Use the PVLVMO control strategy to operate (open and close) a motor-operated valve. Since a motor-operated valve has no spring return (to return the valve to the fail-safe state), two digital outputs are required (one to move the valve towards the open position; and another to move the valve to the closed position).

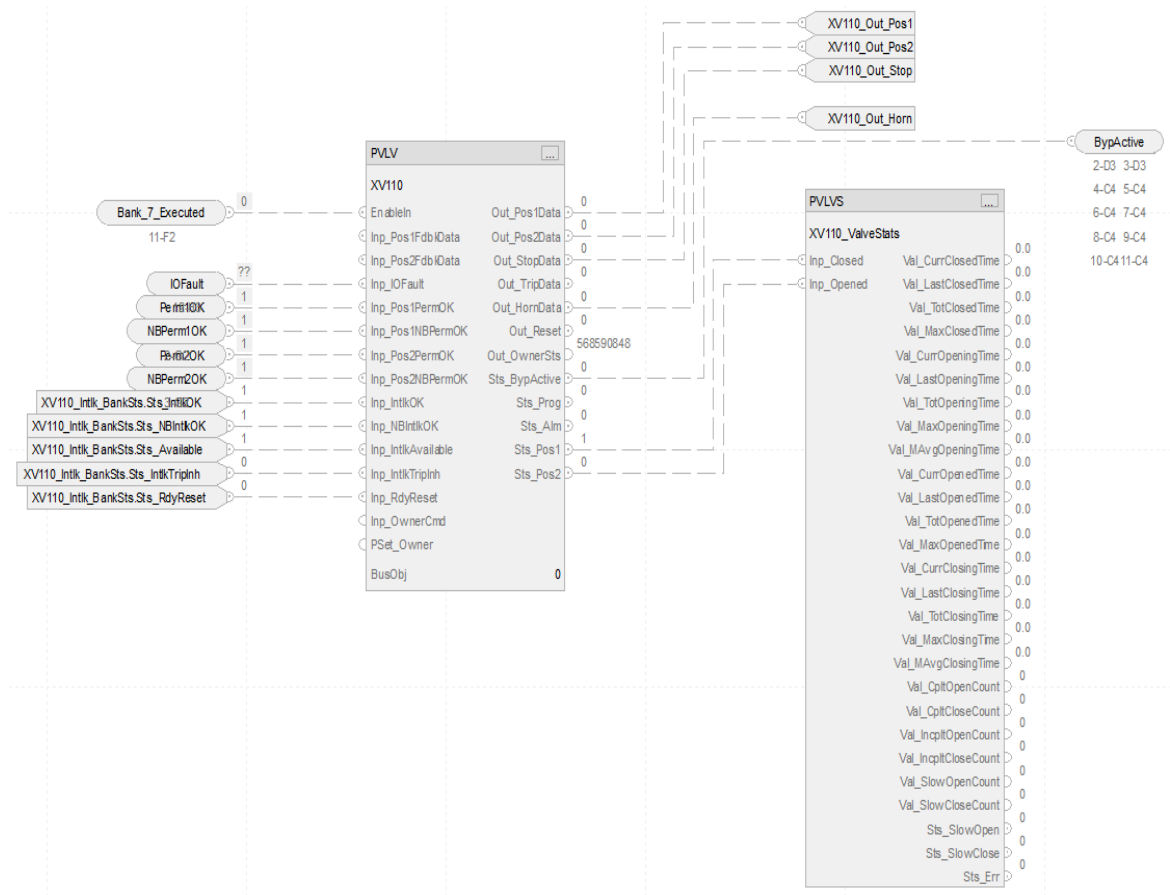
The CS\_PVLVMO control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PVLVMO control strategy contains these Function Block sheets:

Sheet	Description
CS_PVLVMO	Process Valve instruction, motor operated
Position 1 Permissives Position 2 Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. Position 1 and Position 2 permissives are applied to the commands to energize towards those positions. Permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PVLV instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors one analog output channel for I/O fault input and raises an alarm on an I/O fault.

## CS\_PVLVMO Sheet



## PVLV Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
IOFault	Input connection from IO Faults sheet
Perm10K	Input connection from Position 1 Permissives sheet 1 = On permissives OK, device can turn On
NBPerm10K	Input connection from Position 1 Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
Perm20K	Input connection from Position 2 Permissives sheet 1 = On permissives OK, device can turn On
NBPerm20K	Input connection from Position 2 Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
XV110_Intlk_BankSts_Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
XV110_Intlk_BankSts_Sts_NBIntlkOK	Interlock bank status, 1 = All non-bypassable interlocks OK to run
XV110_Intlk_BankSts_Sts_Available	Interlock bank status, 1 = Available
XV110_Intlk_BankSts_Sts_IntkTriphInh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
XV131_Intlk_BankSts_Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

## PVLV Configuration Considerations

## PVLV Output References to PVLVS

# Permissive Sheet

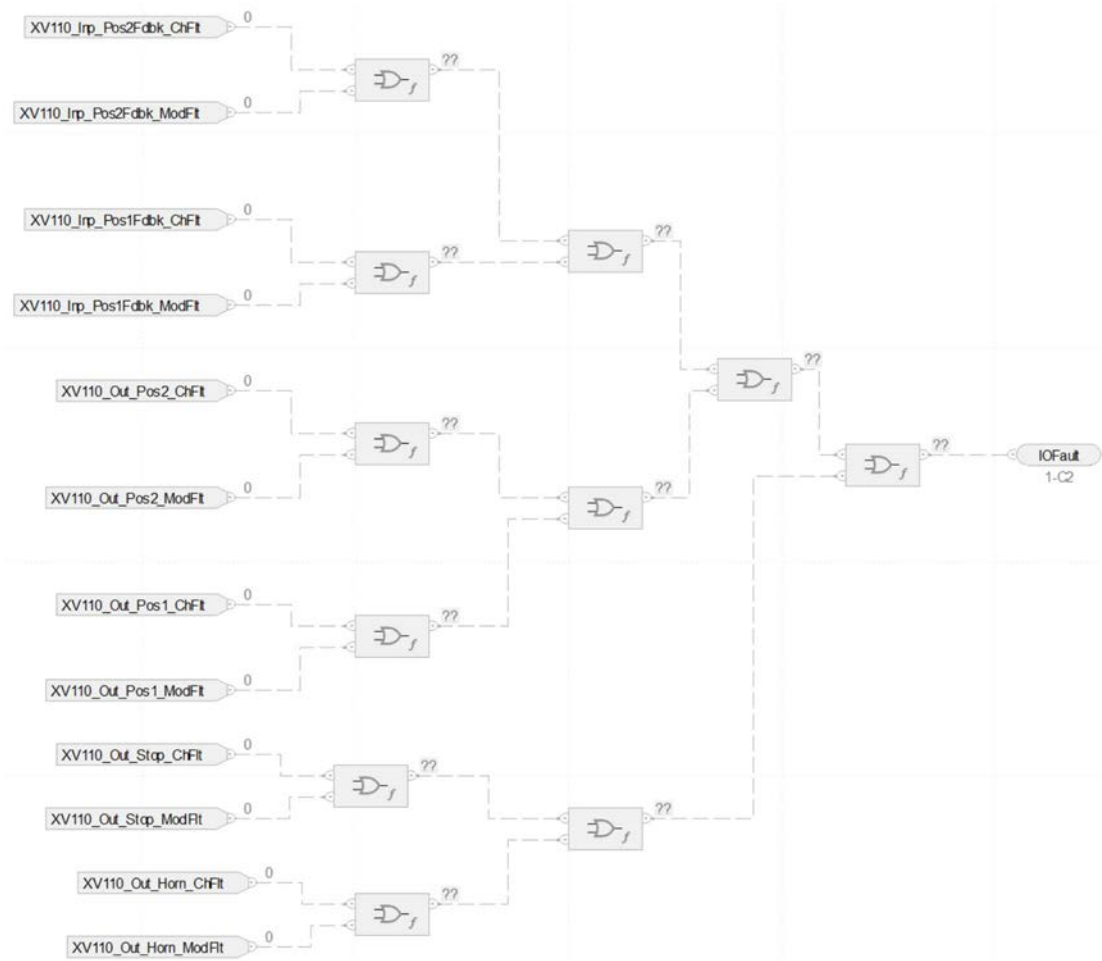
## PPERM Input References

431

PPERM Output References

Parameter	Description
Perm10K Perm20K	Overall permissive status (1 = OK to energize)
NBPerm10K NBPerm20K	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

IO Faults Sheet



Faults Input References

Parameter	Description
XV110_Inp_Pos1Fdbk_ChFlt	Tieback input 1 channel fault
XV110_Inp_Pos1Fdbk_ModFlt	Tieback input 1 module fault
XV110_Inp_Pos2Fdbk_ChFlt	Tieback input 2 channel fault
XV110_Inp_Pos2Fdbk_ModFlt	Tieback input 2 module fault
XV110_Out_Pos1_ChFlt	Position 1 channel fault
XV110_Out_Pos1_ModFlt	Position 1 module fault
XV110_Out_Pos2_ChFlt	Position 2 channel fault
XV110_Out_Pos2_ModFlt	Position 2 module fault
XV110_Out_Horn_ChFlt	Sound audible for output channel fault
XV110_Out_Horn_ModFlt	Sound audible for output module fault



## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PVLVM0 sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for a PVLV Instance

Configure the valve type via the ACM\_Type parameter:

- Solenoid-operated valve (P\_ValveS0)
- Motor-operated valve (P\_ValveM0)
- Hand-operated valve (P\_ValveH0)
- Mixer-proof valve (P\_ValveMP)

## ACM-Based Parameters for a PVLV Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
ACM_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters. Define the PVLV type: P_ValveS0, P_ValveM0, P_ValveH0 or P_ValveMP
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Has_OOAP=True (controller parameter)	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
	Use_OOAP=True	
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter)	Link to a bus array instance. This should be unique for each device
	Use_OOAP=True	See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction. See <a href="#">Interlocks on page 49</a>
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasPos1PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Set to create an instance of the PPERM instruction to allow an Out_Pos1 command. See <a href="#">Permissives on page 50</a>
	ACM_Type= P_ValveM0	
Cfg_HasPos2PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Set to create an instance of the PPERM instruction to allow an Out_Pos2 command. See <a href="#">Permissives on page 50</a>
	ACM_Type= P_ValveM0	
Cfg_HasStop	ACM_Type=P_ValveM0	Set if the motorized valve has a stop output or if simultaneously de-energizing Out_Pos1 and Out_Pos2 is necessary.
Cfg_HasTrip	ACM_Type=P_ValveH0	Set if a trip output exists for the device (such as Horn or Light)
Cfg_HasOpenIntlkObj Cfg_HasUpperSeatIntlkObj Cfg_HasLowerSeatIntlkObj Cfg_HasCavityIntlkObj	ACM_Type=P_ValveMP	Set to create an instance of the PINTLK instruction for each output type. See <a href="#">Interlocks on page 49</a>
<b>02 - Device Configuration</b>		
Cfg_FailPos2	ACM_Type=P_ValveS0	Set if Position 2 is the Fail Position of the device
	ACM_Type= P_ValveM0	
Cfg_HasStatsObj	always	Set to create an instance of the PINTLK instruction for each output type. See <a href="#">Valve Statistics on page 58</a>
<b>02.01 - Device Configuration</b>		

Parameter	Visible When	Details
Cfg_HasPos1Fdbk	ACM_Type=P_ValveSO	Set if Position 1 feedback exists
	ACM_Type= P_ValveMO	
	ACM_Type= P_ValveHO	
Cfg_HasPos2Fdbk	ACM_Type=P_ValveSO	Set if Position 2 feedback exists
	ACM_Type= P_ValveMO	
	ACM_Type= P_ValveHO	

**03 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See [I/O Mapping on page 38...](#)

Inp_Pos1Fdbk	ACM_Type=P_ValveSO Cfg_HasPos1Fdbk=True	Link the position 1 limit switch feedback input reference
	ACM_Type= P_ValveMO Cfg_HasPos1Fdbk=True	
	ACM_Type= P_ValveHO Cfg_HasPos1Fdbk=True	
Inp_Pos2Fdbk	ACM_Type=P_ValveSO Cfg_HasPos2Fdbk=True	Link the position 2 limit switch feedback input reference
	ACM_Type= P_ValveMO Cfg_HasPos2Fdbk=True	
	ACM_Type= P_ValveHO Cfg_HasPos2Fdbk=True	
Out_Pos1	ACM_Type=P_ValveSO Cfg_FailPos2=True	Link the position 1 output reference
	ACM_Type= P_ValveMO Cfg_FailPos2=True	
Out_Pos2	ACM_Type=P_ValveSO Cfg_FailPos2=False	Link the position 2 output reference
	ACM_Type=P_ValveMO Cfg_FailPos2=False	
Inp_OpenLS Inp_ClosedLS Inp_LowerLS Inp_UpperLS Inp_CavityInLS Inp_Cavity_OutLS	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding input configuration parameter to the value defined by each parameter
Out_Open Out_Close Out_LiftLower Out_LiftUpper Out_CavityIn Out_CavityOut	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding output configuration parameter to the value defined by each parameter

**03.01 - IO Configuration VLVMO**

Out_Stop	ACM_Type=P_ValveMO Cfg_HasStop=True	Link the Stop output reference
----------	--	--------------------------------

**03.02 - IO Configuration VLVHO**

Out_Trip	ACM_Type=P_ValveHO Cfg_HasTrip=True	Link the Trip output reference
----------	--	--------------------------------

**04 - Alarm Configuration**

Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasActuatorFaultAlm	ACM_Type=P_ValveSO	If Cfg_HasActuatorFaultAlm=True, ACM displays section 4.01- Interlock Trip Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasFullStallAlm	ACM_Type=P_ValveSO	If Cfg_HasFullStallAlm=True, ACM displays section 4.04 - Full Stall Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	

Parameter	Visible When	Details
Cfg_HasTransitStallAlm	ACM_Type=P_ValveSO	If Cfg_HasTransitStallAlm=True, ACM displays section 4.05 - Transit Stall Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasTripFailAlm	ACM_Type=P_ValveHO	If Cfg_HasTripFailAlm=True, ACM displays section 4.07- Trip Fail Alarm with additional parameters
Cfg_HasLossPos1Alm	ACM_Type=P_ValveSO	If Cfg_HasLossPos1Alm=True, ACM displays section 4.06 - Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasLossPos2Alm	ACM_Type=P_ValveSO	If Cfg_HasLossPos2Alm=True, ACM displays section 4.08- Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasFailAlm	ACM_Type=P_ValveMP	If Cfg_HasFailAlm=True, ACM displays section 4.09 - Fail Alarm with additional parameters

## Additional Sub-Objects for a PVLV Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_1 Permissive_2	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Interlock_Lower Interlock_Upper Interlock_Cavity	Configure interlocks for the positions See <a href="#">Interlocks on page 49</a>



## Process Mix Proof Valve (PVLVMP) Control Strategy

The Process Mix Proof Valve (PVLVMP) instruction controls and monitors feedback from a mix proof valve in various modes and states, and monitors for fault conditions. This instruction supports mix proof valves with or without additional connections for cleaning (CIP, clean-in-place) or steaming (SIP, sanitize in place).

Use the PVLVMP control strategy to control one mix proof valve in various modes and states, while monitoring position feedback inputs to verify that the valve reaches the commanded position. An alarm can be provided on failure to reach the commanded position.

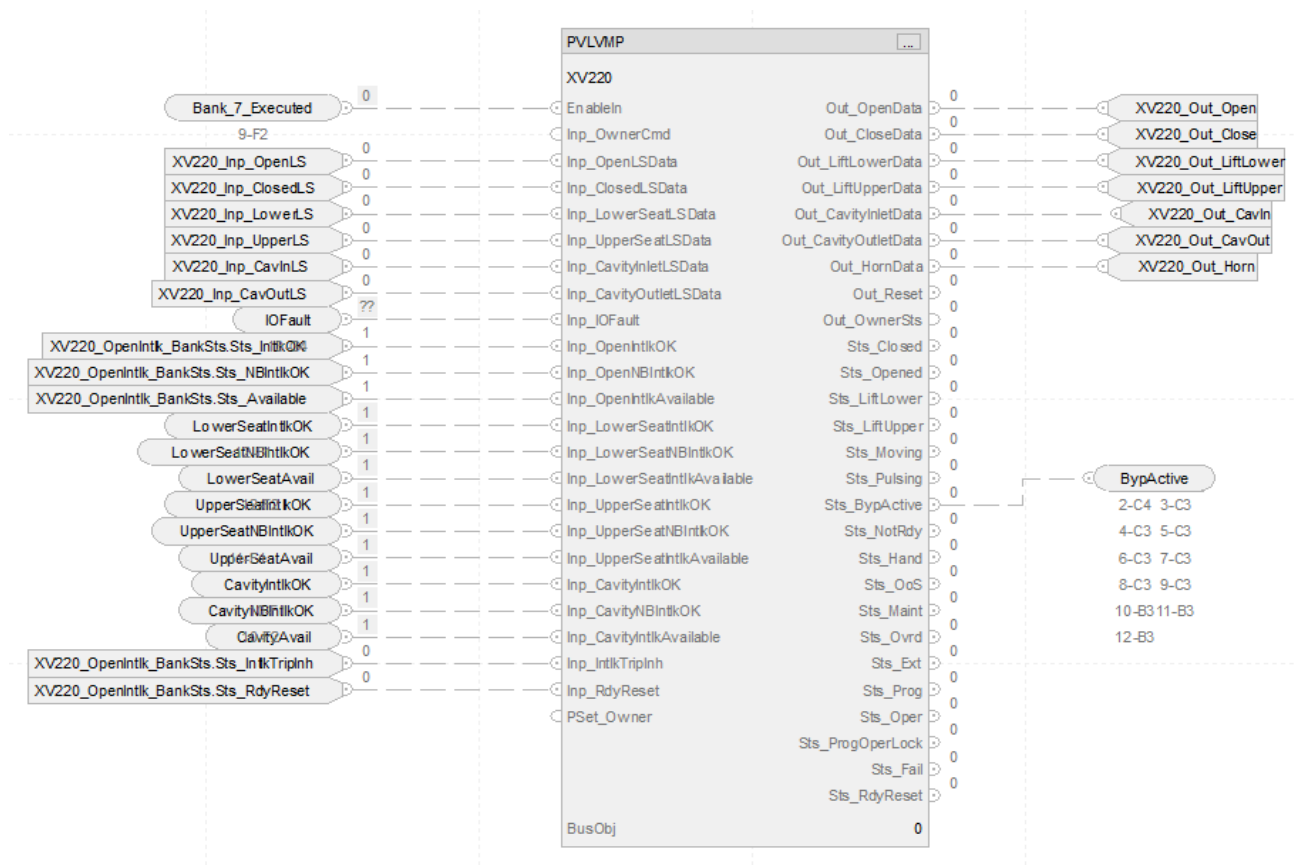
The CS\_PVLVMP control strategy is available as a routine in the process library.

Import the control strategy as a **routine** in your controller project.

The PVLVMP control strategy contains these Function Block sheets:

Sheet	Description
CS_PVLVMP	Process Mix Proof Valve instruction
Permissives	Process Permissives instruction The Process Permissives (PPERMP) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Open Interlock Cavity Interlocks Upper Seat Interlocks Lower Seat Interlocks	The PVLVMP instruction monitors bypassable and non-bypassable Interlocks that force the Output instead of 'analog output' and to the configured safe state. <ul style="list-style-type: none"> <li>Open Interlock has 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default.</li> <li>Cavity Interlocks, Upper Seat Interlocks, and Lower Seat Interlocks each have one interlock sheet that exposes 16 of the available 32 interlocks per bank by default.</li> <li>Use the sheets and interlocks that you need and delete the remainder.</li> </ul>
I/O Faults	The logic monitors the input and output modules and channels that are used to interface with the device for fault conditions and raises an alarm on an I/O fault.

## CS\_PVLVMP Sheet



### PVLVMP Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XV220_Inp_OpenLS	Valve Open Limit Switch, 1 = confirmed open
XV220_Inp_ClosedLS	Valve Closed Limit Switch, 1 = confirmed closed
XV220_Inp_LowerLS	Valve Lower Seat Lift Limit Switch, 1 = confirmed lower seat lifted
XV220_Inp_UpperLS	Valve Upper Seat Lift Limit Switch, 1 = confirmed upper seat lifted
XV220_Inp_CavInLS	Valve cavity inlet limit switch: 1 = Confirmed cavity inlet opened.
XV220_Inp_CavOutLS	Valve cavity outlet limit switch: 1 = Confirmed cavity output closed
IOFault	Input connection from I/O Faults sheet
PermOK	Input connection from Permissives sheet 1 = Permissives OK, valve can move from the closed position
NBPermOK	Input connection from Permissives sheet 1 = Non-bypassable permissives OK, valve can move from the closed position
XV220_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to move valve from the closed position, 0 = Close valve
XV220_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to move valve from the closed position 0 = Close valve
XV220_Intlk_BankSts.Sts_Available	Interlock bank status , 1 = Available
XV220_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - closes valve but does not raise trip alarm
XV220_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

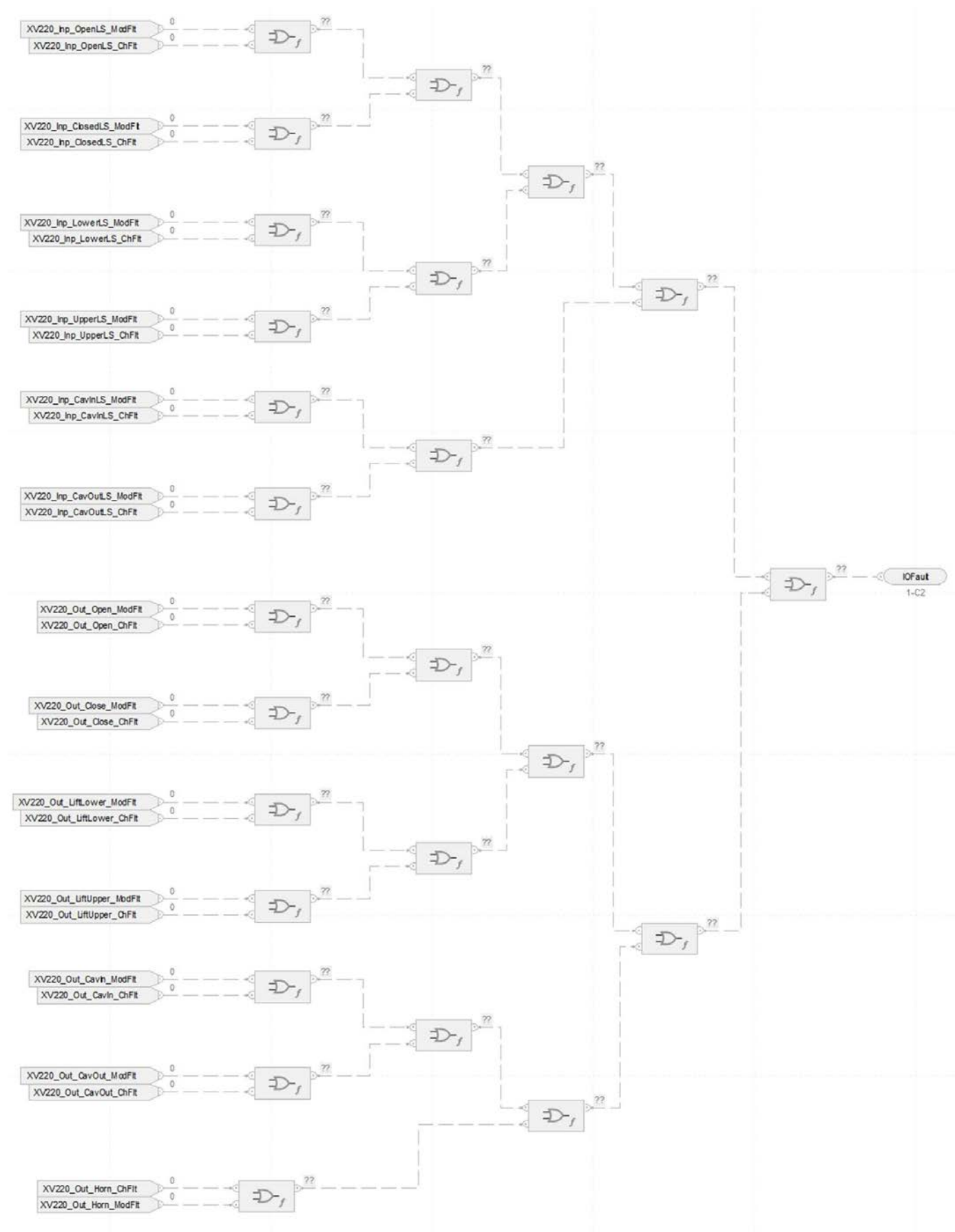
## PVLVMP Output References

Parameter	Description
XV220_Out_Open	Output to Open valve, 1 = Open
XV220_Out_Closed	Output to Close valve, 1 = Close
XV220_Out_LiftLower	Output to Lift lower valve seat, 1 = Lift
XV220_Out_LiftUpper	Output to Lift upper valve seat, 1 = Lift
XV220_Out_Horn	1 = Sound audible before commanded valve action
XV220_Out_CavIn	Cavity In Output
XV220_Out_CavOut	Cavity Out Output
BypActive	Output connection to permissive and interlock bank sheets

## PVLVMP Configuration Considerations

Operand	Type	Description
PlantPax® control	P_DISCRETE_MIX_PROOF	Instance of data structure (backing tag) required for proper operation of the instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

I/O Faults Sheet





## Fault Input References

Parameter	Description
XV220_Inp_OpenLS_ModFlt	Open Feedback Input Module Fault (Any Connected I/O Module Fault)
XV220_Inp_OpenLS_ChFlt	Open Feedback Input Channel Fault
XV220_Inp_Closed_LS_ModFlt	Closed Feedback Input Module Fault (Any Connected I/O Module Fault)
XV220_Inp_Closed_LS_ChFlt	Closed Feedback Input Channel Fault
XV220_Inp_LowerLS_ModFlt	Lower Limit Switch Input Module Fault (Any Connected I/O Module Fault)
XV220_Inp_LowerLS_ChFlt	Lower Limit Switch Input Channel Fault
XV220_Inp_UpperLS_ModFlt	Upper Limit Switch Input Module Fault (Any Connected I/O Module Fault)
XV220_Inp_UpperLS_ChFlt	Upper Limit Switch Input Channel Fault
XV220_Inp_CavInLS_ModFlt	Valve cavity inlet limit switch module fault
XV220_Inp_CavInLS_ChFlt	Valve cavity inlet limit switch channel fault
XV220_Inp_CavOutLS_ModFlt	Valve cavity outlet limit switch module fault
XV220_Inp_CavOutLS_ChFlt	Valve cavity outlet limit switch channel fault
XV220_Out_Open_ModFlt	Open limit switch module fault
XV220_Out_Open_ChFlt	Open limit switch channel fault
XV220_Out_Close_ModFlt	Closed limit switch module fault
XV220_Out_Close_ChFlt	Closed limit switch channel fault
XV220_Out_LiftLower_ModFlt	Lift lower limit switch module fault
XV220_Out_LiftLower_ChFlt	Lift lower limit switch channel fault
XV220_Out_LiftUpper_ModFlt	Lift upper limit switch module fault
XV220_Out_LiftUpper_ChFlt	Lift upper limit switch channel fault
XV220_Out_CavIn_ModFlt	Cavity Inlet Output Module Fault (Any Connected I/O Module Fault)
XV220_Out_CavIn_ChFlt	Cavity Inlet Output Input Channel Fault
XV220_Out_CavOut_ModFlt	Cavity Outlet Output Module Fault (Any Connected I/O Module Fault)
XV220_Out_CavOut_ChFlt	Cavity Outlet Output Input Channel Fault
XV220_Out_Horn_ChFlt	Audible output device channel fault
XV220_Out_Horn_ModFlt	Audible output device module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PVLVMP sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for a PVLV Instance

Configure the valve type via the ACM\_Type parameter:

- Solenoid-operated valve (P\_ValveS0)
- Motor-operated valve (P\_ValveM0)
- Hand-operated valve (P\_ValveH0)
- Mixer-proof valve (P\_ValveMP)

### ACM-Based Parameters for a PVLV Instance

Parameter	Visible When	Details
00 - Selection		
ACM_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters. Define the PVLV type: P_ValveSO, P_ValveMO, P_ValveHO or P_ValveMP
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Has_OOAP=True (controller parameter) Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
01 - Options		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasPos1PermObj	ACM_Type=P_ValveSO Cfg_FailPos2=True ACM_Type= P_ValveMO	Set to create an instance of the PPERM instruction to allow an Out_Pos1 command. See <a href="#">Permissives on page 50</a>
Cfg_HasPos2PermObj	ACM_Type=P_ValveSO Cfg_FailPos2=False ACM_Type= P_ValveMO	Set to create an instance of the PPERM instruction to allow an Out_Pos2 command. See <a href="#">Permissives on page 50</a>
Cfg_HasStop	ACM_Type=P_ValveMO	Set if the motorized valve has a stop output or if simultaneously de-energizing Out_Pos1 and Out_Pos2 is necessary.
Cfg_HasTrip	ACM_Type=P_ValveHO	Set if a trip output exists for the device (such as Horn or Light)
Cfg_HasOpenIntlkObj Cfg_HasUpperSeatIntlkObj Cfg_HasLowerSeatIntlkObj Cfg_HasCavityIntlkObj	ACM_Type=P_ValveMP	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Interlocks on page 49</a>
02 - Device Configuration		
Cfg_FailPos2	ACM_Type=P_ValveSO ACM_Type= P_ValveMO	Set if Position 2 is the Fail Position of the device
Cfg_HasStatsObj	always	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Valve Statistics on page 58</a>
02.01 - Device Configuration		
Cfg_HasPos1Fdbk	ACM_Type=P_ValveSO ACM_Type= P_ValveMO ACM_Type= P_ValveHO	Set if Position 1 feedback exists
Cfg_HasPos2Fdbk	ACM_Type=P_ValveSO ACM_Type= P_ValveMO ACM_Type= P_ValveHO	Set if Position 2 feedback exists
03 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38...</a>		

Parameter	Visible When	Details
Inp_Pos1Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos1Fdbk=True	Link the position 1 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos1Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos1Fdbk=True	
Inp_Pos2Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos2Fdbk=True	Link the position 2 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos2Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos2Fdbk=True	
Out_Pos1	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Link the position 1 output reference
	ACM_Type= P_ValveM0 Cfg_FailPos2=True	
Out_Pos2	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Link the position 2 output reference
	ACM_Type=P_ValveM0 Cfg_FailPos2=False	
Inp_OpenLS Inp_ClosedLS Inp_LowerLS Inp_UpperLS Inp_CavityInLS Inp_Cavity_OutLS	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding input configuration parameter to the value defined by each parameter
Out_Open Out_Close Out_LiftLower Out_LiftUpper Out_CavityIn Out_CavityOut	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding output configuration parameter to the value defined by each parameter
<b>03.01 - IO Configuration VLVM0</b>		
Out_Stop	ACM_Type=P_ValveM0 Cfg_HasStop=True	Link the Stop output reference
<b>03.02 - IO Configuration VL VH0</b>		
Out_Trip	ACM_Type=P_ValveH0 Cfg_HasTrip=True	Link the Trip output reference
<b>04 - Alarm Configuration</b>		
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasActuatorFaultAlm	ACM_Type=P_ValveS0	If Cfg_HasActuatorFaultAlm=True, ACM displays section 4.01- Interlock Trip Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasFullStallAlm	ACM_Type=P_ValveS0	If Cfg_HasFullStallAlm=True, ACM displays section 4.04 - Full Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTransitStallAlm	ACM_Type=P_ValveS0	If Cfg_HasTransitStallAlm=True, ACM displays section 4.05 - Transit Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTripFailAlm	ACM_Type=P_ValveH0	If Cfg_HasTripFailAlm=True, ACM displays section 4.07- Trip Fail Alarm with additional parameters
Cfg_HasLossPos1Alm	ACM_Type=P_ValveS0	If Cfg_HasLossPos1Alm=True, ACM displays section 4.06 - Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	

Parameter	Visible When	Details
Cfg_HasLossPos2Alm	ACM_Type=P_ValveSO	If Cfg_HasLossPos2Alm=True, ACM displays section 4.08- Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveMO	
	ACM_Type=P_ValveHO	
Cfg_HasFailAlm	ACM_Type=P_ValveMP	If Cfg_HasFailAlm=True, ACM displays section 4.09 - Fail Alarm with additional parameters

## Additional Sub-Objects for a PVLV Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_1 Permissive_2	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Interlock_Lower Interlock_Upper Interlock_Cavity	Configure interlocks for the positions See <a href="#">Interlocks on page 49</a>

## Notes:



## Process Valve Solenoid Operated (PVLVSO) Control Strategy

Use the PVLVSO control strategy to operate (open and close) one solenoid-operated valve. Generally, a solenoid-operated valve only requires one output to energize a solenoid providing pneumatic energy to an actuator that moves the valve from its fail-safe position. When this output is de-energized, a spring forces the valve back to its fail-safe position. When using this control strategy, one must consider whether the valve is Fail Closed (FC) or Fail Open (FO). For the more common FC valve, the output XV101\_Out\_Pos2 must be used to drive the field device. If the valve is a FO valve, the output XV101\_Out\_Pos1 must be used to drive the field device.

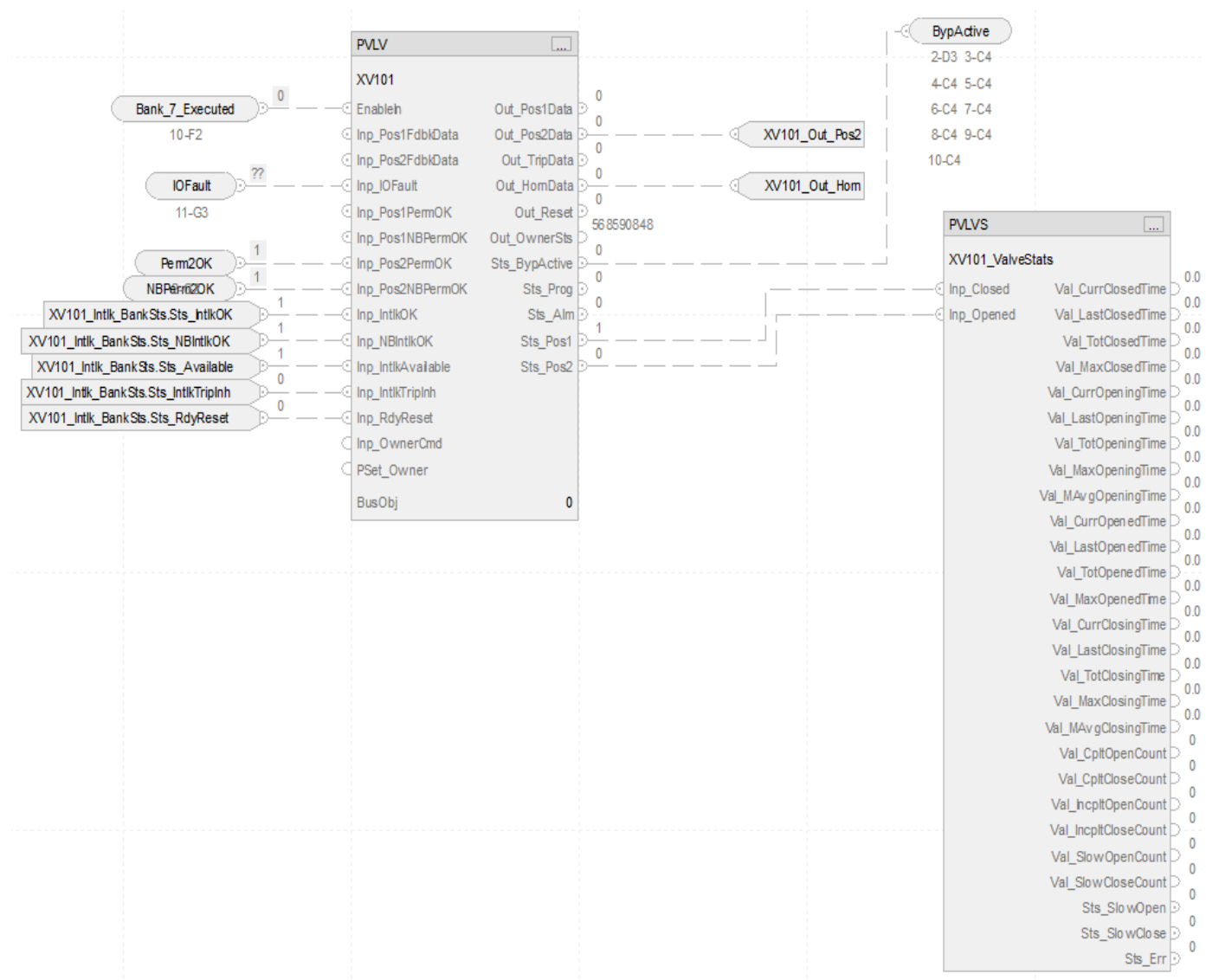
The CS\_PVLVSO control strategy is available as two routines in the process library:

- One routine for fail position 1, and one routine for fail position 2. Fail position 1 and 2 could be Close/Open, Up/ Down, Left/ Right depending on the application.
- Import the control strategy as a **routine** in your controller project.

The PVLVSO control strategy contains these Function Block sheets:

Sheet	Description
CS_PVLVSO	Process Valve instruction, solenoid operated
Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PVLV instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.
I/O Faults	The logic monitors one analog output channel for I/O fault input and raises an alarm on an I/O fault.

CS\_PVLVS0\_FailPos1 Sheet



PVLV Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XV101.Inp_Pos1Fdbk XV101.Inp_Pos2Fdbk	Feedback from Position limit switches of the device 1 = Device confirmed Position 1
IOFault	Input connection from IO Faults sheet
Perm2OK	Input connection from Position 2 Permissives sheet 1 = On permissives OK, device can turn On
NBPerm2OK	Input connection from Position 2 Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
XV101.Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run, 0 = Stop
XV101.Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run



Parameter	Description
XV101_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
XV101_Intlk_BankSts.Sts_IntlkTripInh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
XV101_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

### PVLV Output References

Parameter	Description
XV101_Out_Pos2	1 = Activate to move valve to Position 2
XV101_Out_Pos1	1 = Activate to move valve to Position 1
XV101_Out_Horn	1 = Sound audible before commanded valve start
BypActive	Output connection to permissives and interlock bank sheets

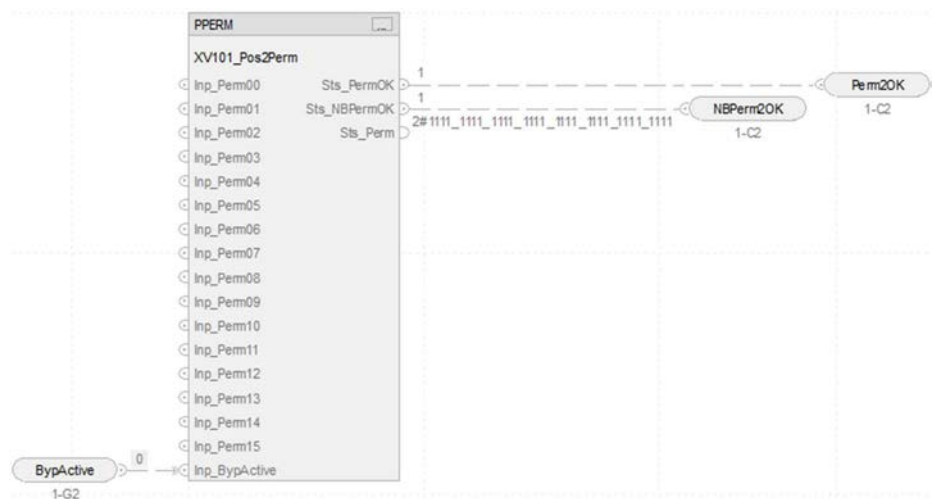
### PVLV Configuration Considerations

Operand	Type	Description
PlantPAX® control	P_VALVE_DISCRETE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

### PVLV Output References to PVLVS

Parameter	Description
Sts_Pos1	1 = Valve requested to Position 1 and is confirmed Position 1
Sts_Pos2	1 = Valve requested to Position 1 and is confirmed Position 2

## Position 2 Permissive Sheet



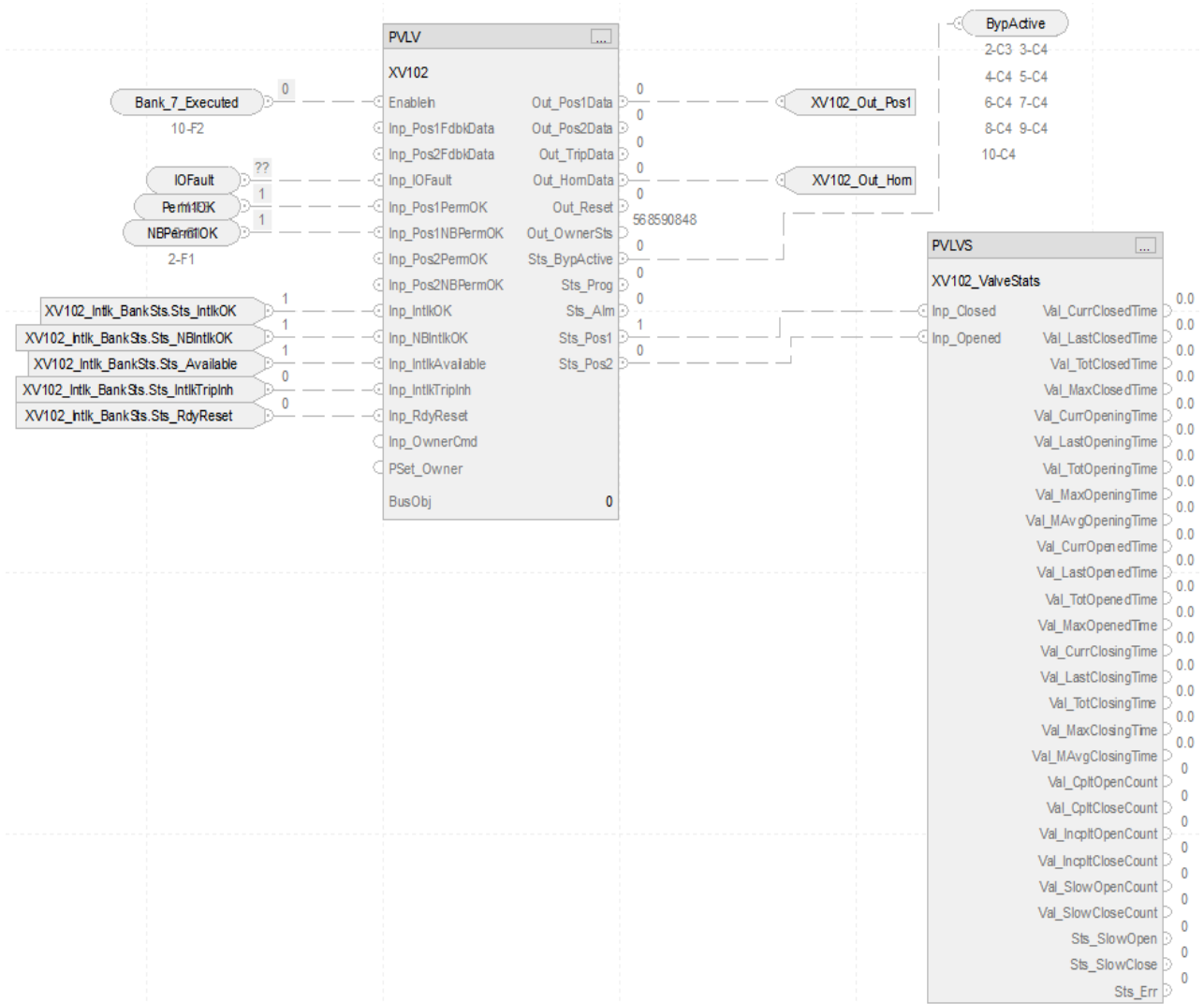
### PPERM Input References

Parameter	Description
BypActive	Input connection from the interlock bank sheet

## PPERM Output References

Parameter	Description
Perm20K	Overall permissive status (1 = OK to energize)
NBPerm20K	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

## CS\_PVLVS0\_FailPos2 Sheet



## PVLV Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
XV102.Inp_Pos1Fdbk XV102.Inp_Pos2Fdbk	Feedback from Position limit switches of the device 1 = Device confirmed Position 1
IOFault	Input connection from IO Faults sheet
Perm20K	Input connection from Position 2 Permissives sheet 1 = On permissives OK, device can turn On
NBPerm20K	Input connection from Position 2 Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
XV102.Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop

Parameter	Description
XV102_Intlk_BankSts.Sts.NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
XV102_Intlk_BankSts.Sts.Available	Interlock bank status 1 = Available
XV102_Intlk_BankSts.Sts.IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
XV102_Intlk_BankSts.Sts.RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

### PVLV Output References

Parameter	Description
XV102_Out_Pos2	1 = Activate to move valve to Position 2
XV102_Out_Pos1	1 = Activate to move valve to Position 1
XV102_Out_Horn	1 = Sound audible before commanded valve start
BypActive	Output connection to permissives and interlock bank sheets

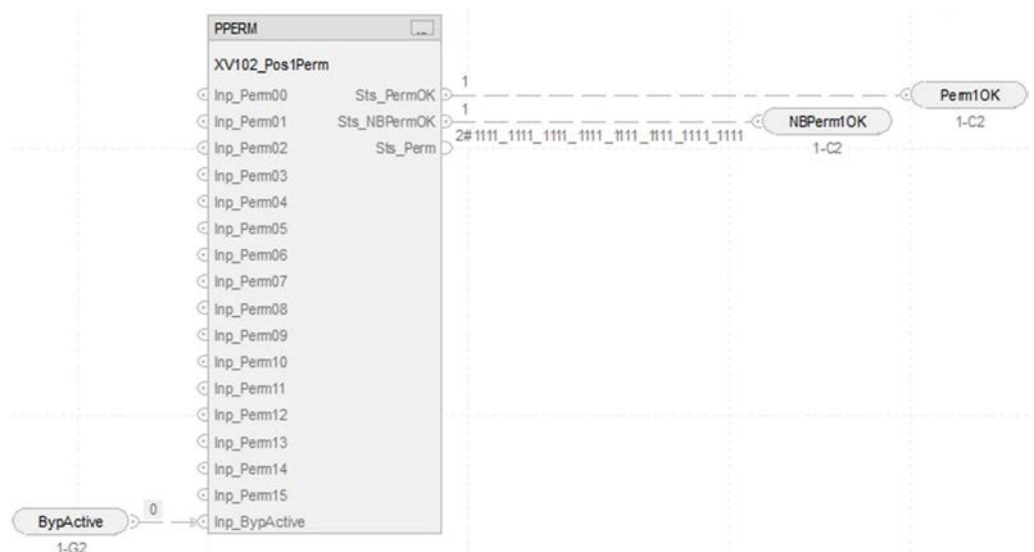
### PVLV Configuration Considerations

Operand	Type	Description
PlantPAx control	P_VALVE_DISCRETE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

### PVLV Output References to PVLVS

Parameter	Description
Sts_Pos1	1 = Valve requested to Position 1 and is confirmed Position 1
Sts_Pos2	1 = Valve requested to Position 1 and is confirmed Position 2

## Position 1 Permissive Sheet



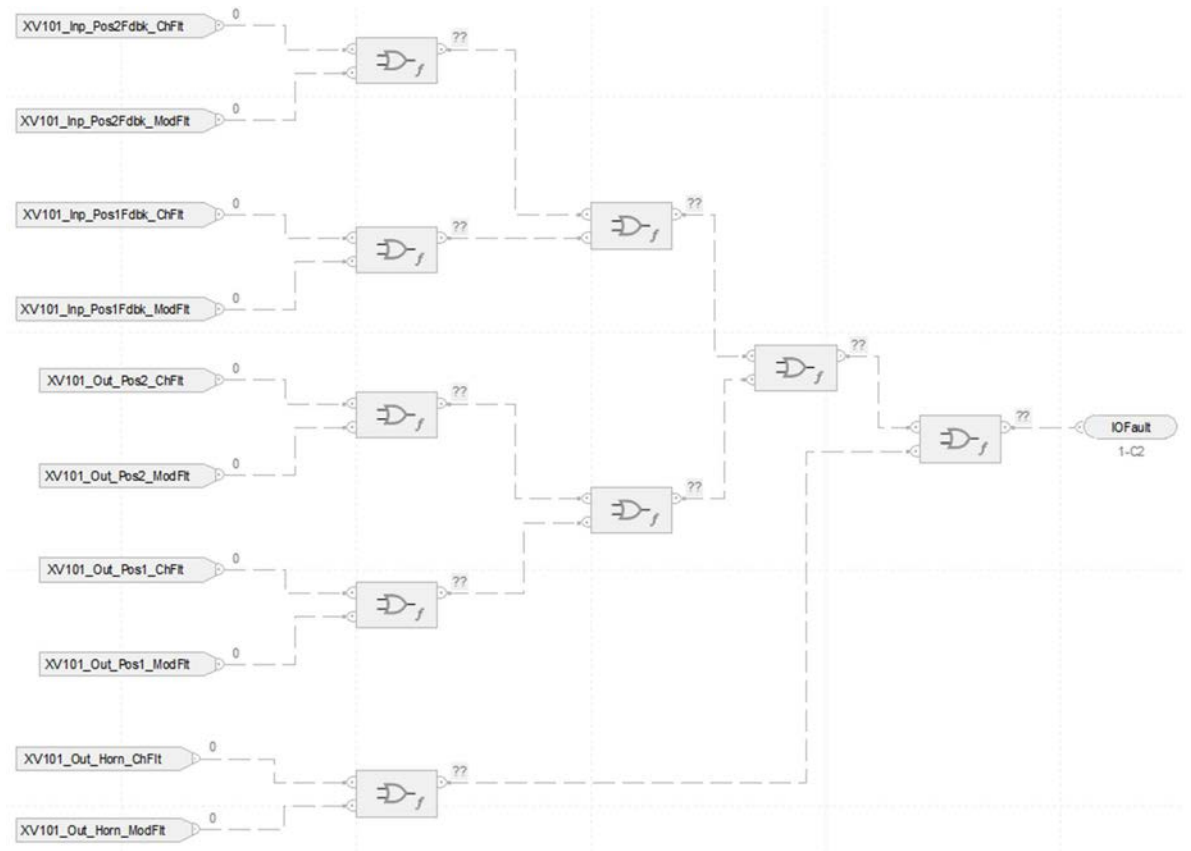
PPERM Input References

Parameter	Description
ByActive	Input connection from the interlock bank sheet

PPERM Output References

Parameter	Description
Perm10K	Overall permissive status (1 = OK to energize)
NBPerm10K	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

IO Faults Sheet-XV101



Fault Input References

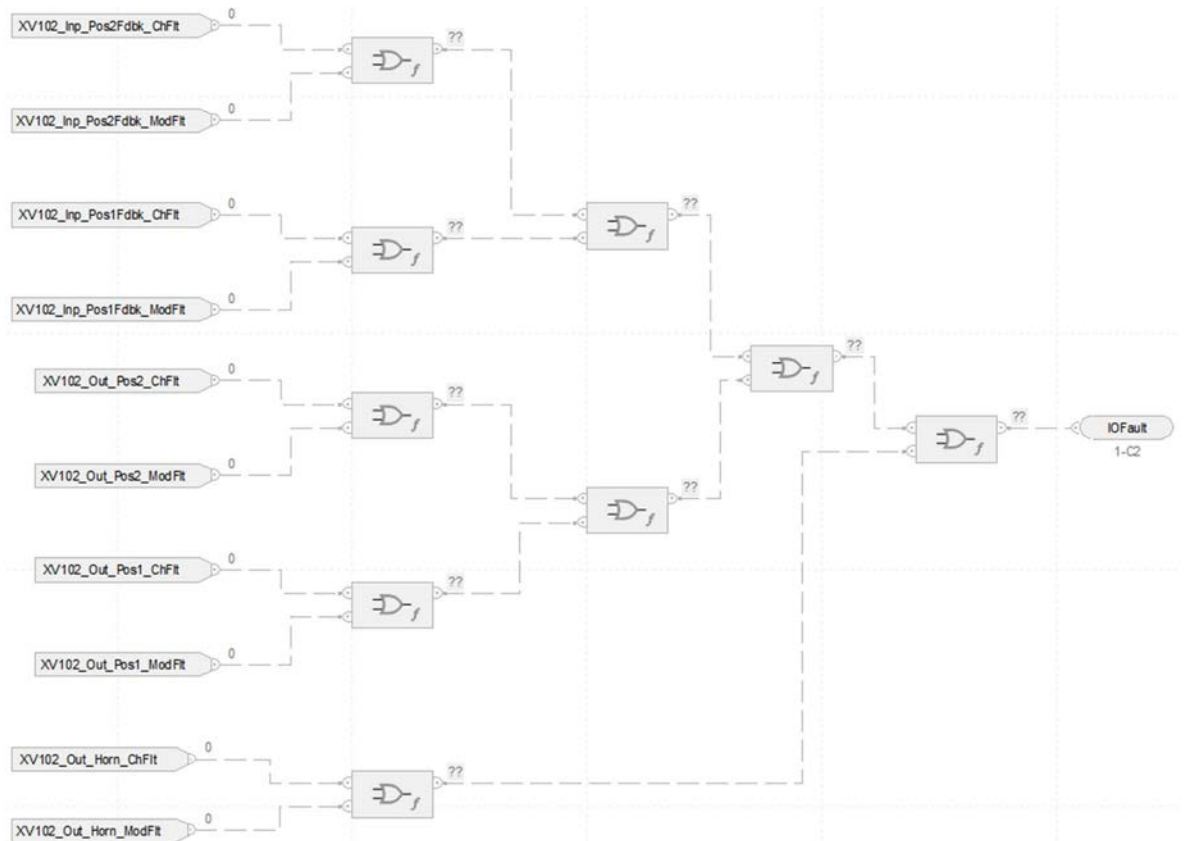
Parameter	Description
XV101_Inp_Pos2Fdbk_ChFlt	Tieback input 2 channel fault
XV101_Inp_Pos2Fdbk_ModFlt	Tieback input 2 module fault
Local:6:I.Fault.8	Discrete input fault
Local_06.Sts.IOFault	Discrete input communication faulted
Local:7:I.Fault.8	Discrete output fault
Local_07.Sts.IOFault	Discrete output communication faulted
XV101_Out_Pos1_ChFlt	Position 1 channel fault
XV101_Out_Pos1_ModFlt	Position 1 module fault
XV101_Out_Horn_ChFlt	Sound audible for output channel fault
XV101_Out_Horn_ModFlt	Sound audible for output module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PVLVS0 sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## IO Faults Sheet-XV102



## Fault Input References

Parameter	Description
XV102_Inp_Pos2Fdbk_ChFlt	Tieback input 2 channel fault
XV102_Inp_Pos2Fdbk_ModFlt	Tieback input 2 module fault
Local:6:I.Fault.8	Discrete input fault
Local_06.Sts.IOFault	Discrete input communication faulted
Local:7:I.Fault.8	Discrete output fault
Local_07.Sts.IOFault	Discrete output communication faulted
XV102_Out_Pos1_ChFlt	Position 1 channel fault
XV102_Out_Pos1_ModFlt	Position 1 module fault
XV102_Out_Horn_ChFlt	Sound audible for output channel fault
XV102_Out_Horn_ModFlt	Sound audible for output module fault

## Fault Output References

Parameter	Description
IOFault	Output connection to CS_PVLVS0 sheet

For examples on how to map data to input tags, see [PlantPax Control Strategies on page 21](#).

## ACM Considerations for a PVLV Instance

Configure the valve type via the ACM\_Type parameter:

- Solenoid-operated valve (P\_ValveS0)
- Motor-operated valve (P\_ValveM0)
- Hand-operated valve (P\_ValveH0)
- Mixer-proof valve (P\_ValveMP)

### ACM-Based Parameters for a PVLV Instance

Parameter	Visible When	Details
00 - Selection		
ACM_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters. Define the PVLV type: P_ValveS0, P_ValveM0, P_ValveH0 or P_ValveMP
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Has_OOAP=True (controller parameter)	Set to use the ArbitrationQ instruction for ownership queuing.
	Use_OOAP=True	See <a href="#">Process Controller on page 36</a>
01 - Options		
Bus_Instance	Has_OOAP=True (controller parameter)	Link to a bus array instance. This should be unique for each device
	Use_OOAP=True	See <a href="#">Process Controller on page 36</a>
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasPos1PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Set to create an instance of the PPERM instruction to allow an Out_Pos1 command.
	ACM_Type= P_ValveM0	See <a href="#">Permissives on page 50</a>
Cfg_HasPos2PermObj	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Set to create an instance of the PPERM instruction to allow an Out_Pos2 command.
	ACM_Type= P_ValveM0	See <a href="#">Permissives on page 50</a>
Cfg_HasStop	ACM_Type=P_ValveM0	Set if the motorized valve has a stop output or if simultaneously de-energizing Out_Pos1 and Out_Pos2 is necessary.
Cfg_HasTrip	ACM_Type=P_ValveH0	Set if a trip output exists for the device (such as Horn or Light)
Cfg_HasOpenIntlkObj Cfg_HasUpperSeatIntlkObj Cfg_HasLowerSeatIntlkObj Cfg_HasCavityIntlkObj	ACM_Type=P_ValveMP	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Interlocks on page 49</a>
02 - Device Configuration		
Cfg_FailPos2	ACM_Type=P_ValveS0	Set if Position 2 is the Fail Position of the device
	ACM_Type= P_ValveM0	
Cfg_HasStatsObj	always	Set to create an instance of the PINTLK instruction for each output type See <a href="#">Valve Statistics on page 58</a>
02.01 - Device Configuration		
Cfg_HasPos1Fdbk	ACM_Type=P_ValveS0	Set if Position 1 feedback exists
	ACM_Type= P_ValveM0	
	ACM_Type= P_ValveH0	
Cfg_HasPos2Fdbk	ACM_Type=P_ValveS0	Set if Position 2 feedback exists
	ACM_Type= P_ValveM0	
	ACM_Type= P_ValveH0	
03 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38...</a>		

Parameter	Visible When	Details
Inp_Pos1Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos1Fdbk=True	Link the position 1 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos1Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos1Fdbk=True	
Inp_Pos2Fdbk	ACM_Type=P_ValveS0 Cfg_HasPos2Fdbk=True	Link the position 2 limit switch feedback input reference
	ACM_Type= P_ValveM0 Cfg_HasPos2Fdbk=True	
	ACM_Type= P_ValveH0 Cfg_HasPos2Fdbk=True	
Out_Pos1	ACM_Type=P_ValveS0 Cfg_FailPos2=True	Link the position 1 output reference
	ACM_Type= P_ValveM0 Cfg_FailPos2=True	
Out_Pos2	ACM_Type=P_ValveS0 Cfg_FailPos2=False	Link the position 2 output reference
	ACM_Type=P_ValveM0 Cfg_FailPos2=False	
Inp_OpenLS Inp_ClosedLS Inp_LowerLS Inp_UpperLS Inp_CavityInLS Inp_Cavity_OutLS	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding input configuration parameter to the value defined by each parameter
Out_Open Out_Close Out_LiftLower Out_LiftUpper Out_CavityIn Out_CavityOut	ACM_Type=P_ValveMP	ACM generates code that pins the corresponding output configuration parameter to the value defined by each parameter
<b>03.01 - IO Configuration VLVMO</b>		
Out_Stop	ACM_Type=P_ValveM0 Cfg_HasStop=True	Link the Stop output reference
<b>03.02 - IO Configuration VLVH0</b>		
Out_Trip	ACM_Type=P_ValveH0 Cfg_HasTrip=True	Link the Trip output reference
<b>04 - Alarm Configuration</b>		
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters
Cfg_HasActuatorFaultAlm	ACM_Type=P_ValveS0	If Cfg_HasActuatorFaultAlm=True, ACM displays section 4.01- Interlock Trip Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasFullStallAlm	ACM_Type=P_ValveS0	If Cfg_HasFullStallAlm=True, ACM displays section 4.04 - Full Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTransitStallAlm	ACM_Type=P_ValveS0	If Cfg_HasTransitStallAlm=True, ACM displays section 4.05 - Transit Stall Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasTripFailAlm	ACM_Type=P_ValveH0	If Cfg_HasTripFailAlm=True, ACM displays section 4.07- Trip Fail Alarm with additional parameters
Cfg_HasLossPos1Alm	ACM_Type=P_ValveS0	If Cfg_HasLossPos1Alm=True, ACM displays section 4.06 - Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	



Parameter	Visible When	Details
Cfg_HasLossPos2Alm	ACM_Type=P_ValveS0	If Cfg_HasLossPos2Alm=True, ACM displays section 4.08- Loss Pos 1 Alarm with additional parameters
	ACM_Type=P_ValveM0	
	ACM_Type=P_ValveH0	
Cfg_HasFailAlm	ACM_Type=P_ValveMP	If Cfg_HasFailAlm=True, ACM displays section 4.09 - Fail Alarm with additional parameters

## Additional Sub-Objects for a PVLV Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissive_1 Permissive_2	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Interlock_Lower Interlock_Upper Interlock_Cavity	Configure interlocks for the positions See <a href="#">Interlocks on page 49</a>



## Process Variable Speed Drive (PVSD) Control Strategies

Use a PVSD control strategy to monitor and control a variable speed motor using an AC (variable frequency) or DC drive. Use the instruction to run or jog the motor forward or reverse. The drive interface can be through a Device Object Interface or through individual pins.

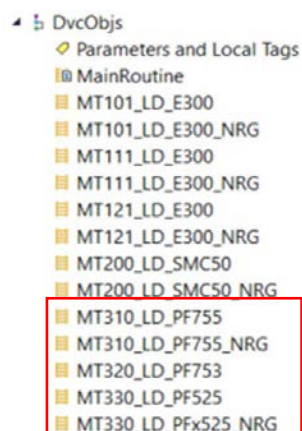
The following PVSD control strategies are available as routines in the process library:

Drive Type	Control Strategy	Description
Generic variable speed drive	CS_PVSD CS_PVSD_Hand	Basic Hand command source
PowerFlex® 525	CS_PVSD525 CS_PVSD525_Hand CS_PVSD525_Energy	Basic Hand command source Energy parameters
PowerFlex 753	CS_PVSD753 CS_PVSD753_Hand	Basic Hand command source
PowerFlex 755	CS_PVSD755 CS_PVSD755_Hand CS_PVSD755_Energy	Basic Hand command source Energy parameters
PowerFlex 755T	CS_PVSD_P755T CS_PVSD_P755T_Hand	Basic Hand command source
PowerFlex 6000T	CS_PVSD_P7000T CS_PVSD_P7000T_Hand	Basic Hand command source
PowerFlex 7000	CS_PVSD_P7000 CS_PVSD_P7000_Hand	Basic Hand command source

Import the appropriate control strategy as a **routine** in your controller project.

Also, import the appropriate device object as a routine in your controller project. These objects are from the Power Device Library and must be downloaded separately from the PlantPax® Process Library.

Each '\_NRG' object uses the Energy object to group energy parameters for the device. Use this object with the corresponding, energy-related control strategy.



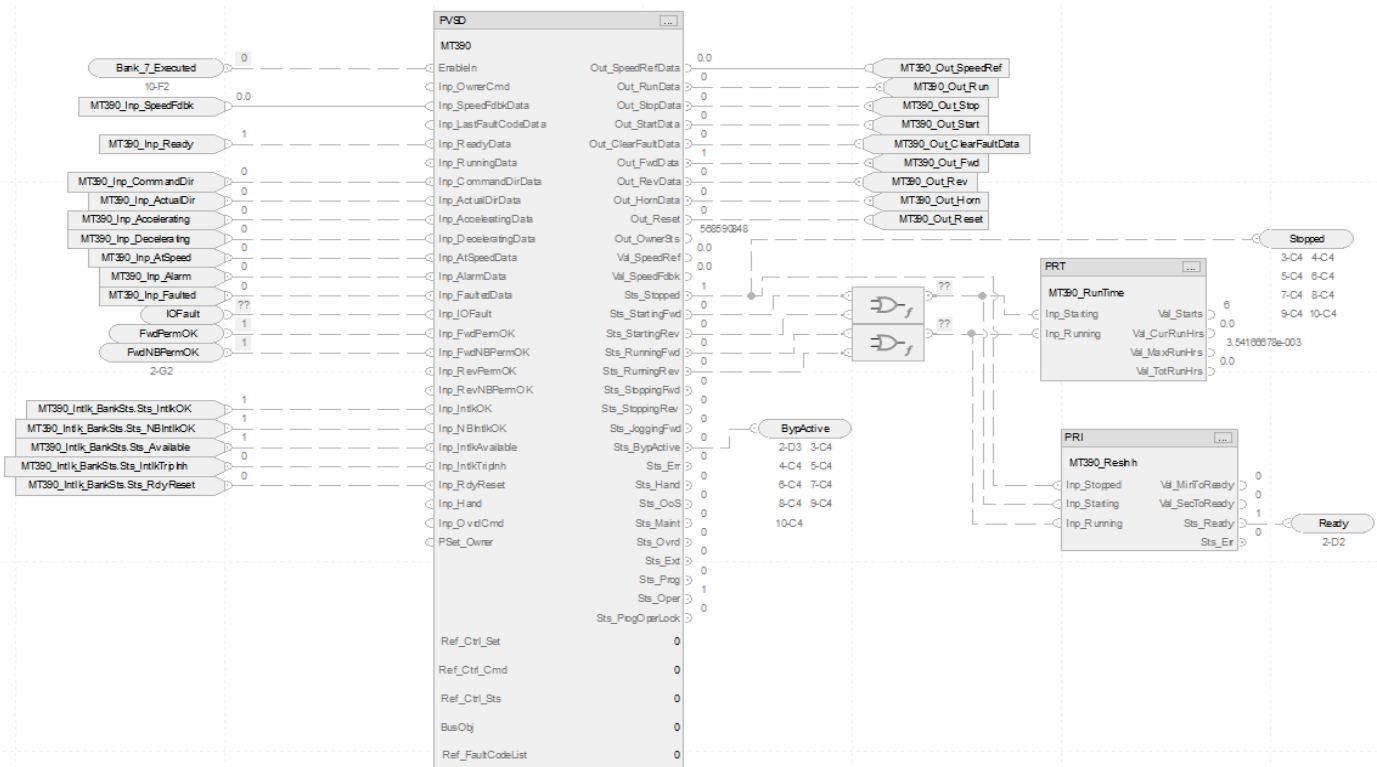
The PVSD control strategies contain these Function Block sheets:

Sheet	Description
CS_PVSD	Process Variable Speed Drive instruction
Permissive	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The PVSD instruction monitors bypassable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.

In the input and output reference descriptions on each sheet, [device] = one of the following:

Drive Type	Type
PowerFlex 525	MT330
PowerFlex 753	MT320
PowerFlex 755	MT310

## CS\_PVSD Sheet



## PVSD Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
FwdPermOK	Input connection from Forward Permissives sheet 1 = On permissives OK, device can turn On
FwdNBPermOK	Input connection from Forward Permissives sheet 1 = Non-bypassable On permissives OK, device can turn On
[device]_Intlk_BankSts.Sts_IntlkOK	Interlock bank status, 1 = OK to run, 0 = Stop
[device]_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
[device]_Intlk_BankSts.Sts_Available	Interlock bank status, 1 = Available
[device]_Intlk_BankSts.Sts_IntlkTriph	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
[device]_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset
[device]_Dvc_CtrlSts.ObjCtrl	Hand command source only 1 = Acquire Hand (typically hardwired local) 0 = Release Hand

## PVSD Output References

Parameter	Description
[device]_Out_Horn	1 = Sound audible before commanded state change
BypActive	Output connection to permissives and interlock bank sheets
Ready	Output connection to the permissive sheet
Stopped	Output connection to interlock bank sheet

The Boolean OR performs a bitwise OR based on these PVSD outputs:

- Sts\_Stopped
- Sts\_StartingFwd
- Sts\_StartingRev
- Sts\_RunningFwd
- Sts\_RunningRev

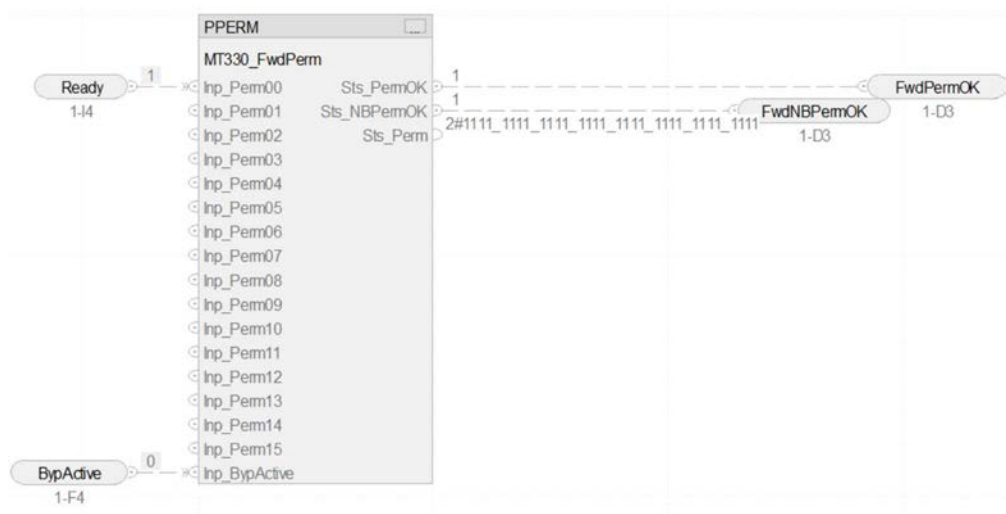
The result feeds these instructions:

Instruction	Description
Process Run Time and Start Counter (PRT)	The PRT instruction records the total run time and number of instances the drive starts.
Process Restart Inhibit (PRI)	The PRI instruction helps prevent the drive from starting repeatedly. Continual starts or start attempts in a short period overheat the motor windings and damage the motor.

## PVSD Configuration Considerations

Operand	Type	Description
PlantPAx control	P_VARIABLE_SPEED_DRIVE	Instance of data structure (backing tag) required for proper operation of instruction
BusObj	BUS_OBJ	Bus component for organization control <ul style="list-style-type: none"> <li>• 0 if not using organization</li> <li>• Bus[x].Obj when using organization</li> </ul> See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .
Ref_Ctrl_Set	RAC_ITF_DVC_PWRVELOCITY_SET	Velocity Automation Device Object Settings Interface Preconfigured in the device object ladder routine
Ref_Ctrl_Cmd	RAC_ITF_DVC_PWRVELOCITY_CMD	Velocity Automation Device Object Command Interface Preconfigured in the device object ladder routine
Ref_Ctrl_Sts	RAC_ITF_DVC_PWRVELOCITY_STS	Velocity Automation Device Object Status Interface Preconfigured in the device object ladder routine
Ref_FaultCodeList	RAC_CODEDESCRIPTION[400]	Fault Code to Fault Description lookup table for the drive Preconfigured in the device object ladder routine

Permissive Sheet



PPERM Input References

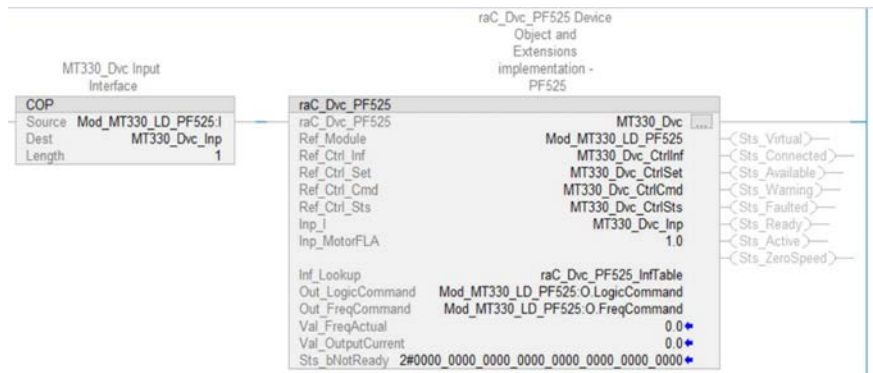
Parameter	Description
Ready	Input connection from the CS_PVSD sheet
BypActive	Input connection from the interlock bank sheet

PPERM Output References

Parameter	Description
FwdPermOK	Overall permissive status (1 = OK to energize)
FwdNBPemOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

# Drive Device Objects

## Drive



## Drive with Energy Parameters



## ACM Considerations for a PVSD Instance

Configure the PVSD parameters to monitor and control a variable speed motor.



## ACM-Based Parameters for a PVSD Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Cfg_UsedInOther	always	PVSD can be standalone (False) or used as part of another PPID_PVSD control strategy (True)
OTHER_RefTag	Cfg_UsedInOther=True	Link to the other control strategy that uses this PVSD instance.
Cfg_HasDvcObj	Cfg_UserInOther=False	Set if device has connection to a device object (Ex. PF525)
Cfg_DvcObj_Tag	Cfg_HasDvcObj=True	Link to the device object. See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>
Use_OOAP	Has_OOAP=True (controller parameter) Cfg_UselnOther=False	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
<b>01 - Options</b>		
Bus_Instance	Has_OOAP=True (controller parameter) Use_OOAP=True Cfg_UselnOther=False	Link to a bus array instance. This should be unique for each device See <a href="#">Process Controller on page 36</a>
Cfg_HasReverse	always	Set if drive can run or jog in reverse
Cfg_HasFwdPermObj	always	Set to create an instance of the PPERM instruction to allow a run forward command
Cfg_HasRevPermObj	Cfg_HasReverse=True	Set to create an instance of the PPERM instruction to allow a run reverse command
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasReslnhObj	always	Set to create an instance of the restart inhibit (PRI) instruction See <a href="#">Statistics Objects on page 57</a>
Cfg_HasRunTimeObj	always	Set to create an instance of a runtime (PRT) instruction See <a href="#">Statistics Objects on page 57</a>
Cfg_HasHand	always	Set to enable a hand switch input (Inp_Hand)
<b>02.00 - Device Configuration CmdSrc</b>		
Cfg_HasExt	always	Set if external commands exist for the PVSD. A typical use for external commands is a Hand-Off-Auto (HOA) switch.
<b>02.01 - Device Configuration Feedback</b>		
Cfg_HasRunFdbk	always	Set if drive provides feedback signal when running
Cfg_HasSpeedFdbk	always	Set if drive provides speed feedback
<b>02.03 - Device Configuration External Commands</b>		
XCmd_StartFwd	Cfg_UsedInOther=False CfgHasExt=True	Link to external start forward input reference
XCmd_StartRev	Cfg_UsedInOther=False CfgHasExt=True	Link to external start reverse input reference
XCmd_Stop	Cfg_UsedInOther=False CfgHasExt=True	Link to external stop input reference
XCmd_JogFwd	Cfg_UsedInOther=False CfgHasExt=True	Link to external jog forward input reference
XCmd_JogRev	Cfg_UsedInOther=False CfgHasExt=True	Link to external jog reverse input reference
XCmd_Reset	Cfg_UsedInOther=False CfgHasExt=True	Link to external reset input reference
XCmd_ResetAckAll	Cfg_UsedInOther=False CfgHasExt=True	Link to external reset/acknowledge all input reference

Parameter	Visible When	Details
<b>03 - IO Configuration</b>		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38...</a>		
Inp_Running	Cfg_UsedInOther=False CfgHasDvcObj=False Cfg_HasRunFdbk=True	Link to run feedback input reference
Inp_SpeedFdbk	Cfg_UsedInOther=False CfgHasDvcObj=False Cfg_HasSpeedFdbk=True	Link to speed feedback input reference
Out_Start	Cfg_UsedInOther=False	Link to the start output reference
Out_Stop	Cfg_UsedInOther=False	Link to the stop output reference
OutSpeedRef	Cfg_UsedInOther=False	Link to the speed reference input reference
Out_Horn	Cfg_UsedInOther=False	Link to the horn output reference
Out_Reset	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the reset output reference
Inp_Accelerating	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the accelerating input reference
Inp_ActualDir	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the actual direction input reference
Inp_Alarm	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the alarm input reference
Inp_AtSpeed	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the at-speed input reference
Inp_CommandDir	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the command direction input reference
Inp_Decelerating	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the decelerating input reference
Inp_Faulted	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the fault input reference
Inp_Ready	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the ready input reference
Inp_LastFaultCode	Cfg_UsedInOther=False CfgHasDvcObj=False	Link to the last fault code input reference
Out_Fwd	Cfg_UsedInOther=False	Link to the forward output reference
Out_Rev	Cfg_UsedInOther=False	Link to the reverse output reference
Out_ClearFaultData	Cfg_UsedInOther=False	Link to the clear fault data output reference
Out_Run	Cfg_UsedInOther=False	Link to the run output reference
Inp_Hand	Cfg_HasHand=True	Link to the hand switch input reference
<b>04 - Alarm Configuration</b>		
Cfg_HasDriveFaultAlm	always	If Cfg_HasDriveFaultAlm=True, ACM displays section 4.02 - Drive Fault Alarm with additional parameters
Cfg_HasFailToStartAlm	always	If Cfg_HasFailToStartAlm=True, ACM displays section 4.04 - Fail to Start Alarm with additional parameters
Cfg_HasFailToStopAlm	always	If Cfg_HasFailToStopAlm=True, ACM displays section 4.05 - Fail to Stop Alarm with additional parameters
Cfg_HasIOFaultAlm	always	If Cfg_HasIOFaultAlm=True, ACM displays section 4.03 - I/O Fault Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.01 - Interlock Trip Alarm with additional parameters

## Additional Sub-Objects for a PVSD Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Fwd_Permissive Rev_Permissive	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
Linked Libraries	Configure device libraries needed for your project See <a href="#">Device Object [Cfg_HasDvcObj] on page 51</a>










## Ramp/Soak (RMPS) Control Strategies

Ramp/Soak refers to the ramping of a controller setpoint to a final target at a predefined rate where it is held for a specified time. This strategy is typically used to control temperature where temperature is “ramped” (increased or decreased) at a predefined rate, and once reaching the target temperature the setpoint is “soaked” (held at temperature for a specified time). The RMPS control strategy can be used to manage multiple segments of alternating ramp and soak periods.








The RMPS control strategy is available as two routines in the process library:

Routine	Description
RMPS100	Ramp/Soak instruction
RMPS100_Out	Ramp/Soak instruction with analog output

- 
 CS\_RMPS
  -  Parameters and Local Tags
  -  MainRoutine
  -  Interlocks
  -  RMPS100
  -  RMPS100\_Out

The RMPS HART control strategy is available as two routines in the process library:

Routine	Description
RMPS101	HART Ramp/Soak instruction
RMPS101_Out	HART Ramp/Soak instruction with analog output

- 
 CS\_RMPS\_HART
  -  Parameters and Local Tags
  -  MainRoutine
  -  Interlocks
  -  RMPS101
  -  RMPS101\_Out

The RMPS EtherNet/IP™ control strategy is available as two routines in the process library:

Routine	Description
RMPS102	EtherNet/IP Ramp/Soak instruction
RMPS102_Out	EtherNet/IP Ramp/Soak instruction with analog output

- CS\_RMPS\_EtherNetIP
  - Parameters and Local Tags
  - MainRoutine
  - Interlocks
  - RMPS102
  - RMPS102\_Out

The RMPS Foundation Feildbus control strategy is available as two routines in the process library:

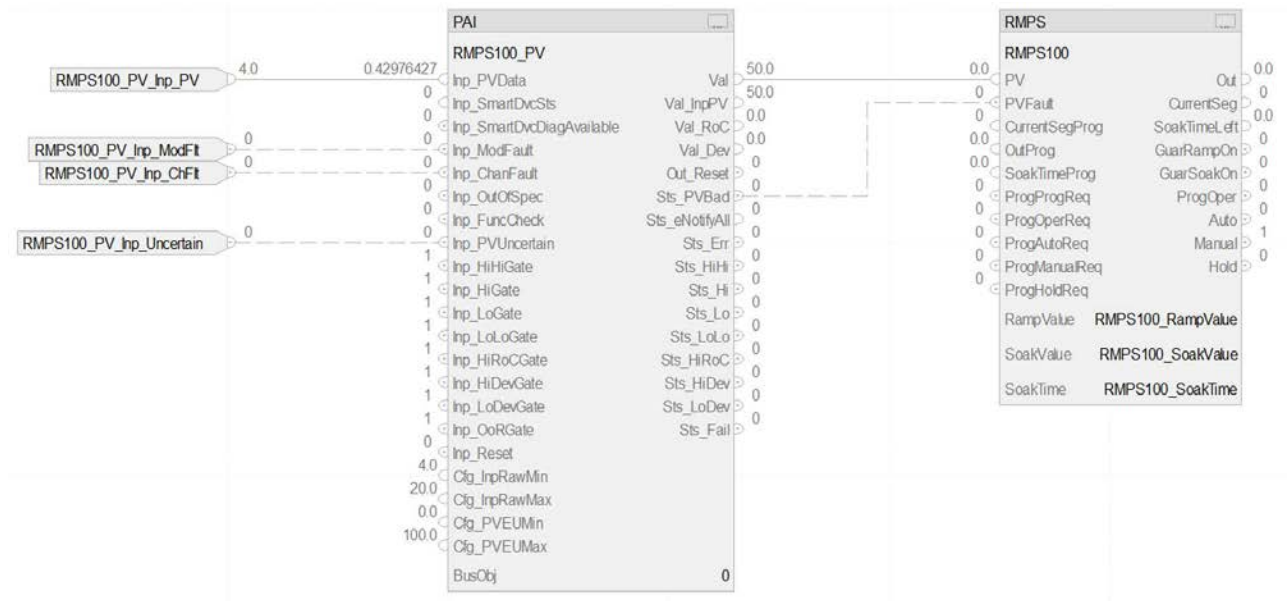
Routine	Description
RMPS103	FOUNDATION Fieldbus Ramp/Soak instruction
RMPS103_Out	FOUNDATION Fieldbus Ramp/Soak instruction with analog output

- CS\_RMPS\_FF
  - Parameters and Local Tags
  - MainRoutine
  - FFLinkMap
  - Interlocks
  - RMPS103
  - RMPS103\_Out

Routine	Description
RMPS104	Profibus PA Ramp/Soak instruction
RMPS104_Out	Profibus PA Ramp/Soak instruction with analog output

- CS\_RMPS\_PA
  - Parameters and Local Tags
  - MainRoutine
  - Interlocks
  - PALinkMap
  - RMPS104
  - RMPS104\_Out

## CS\_RMPS Sheet



### PAI Input References

See [CS\\_PAI Sheet on page 148](#) for details.

### PAI Outputs to RMPS Inputs

Parameter	Description
Val	Analog input value in engineering units (after Substitute PV, if used). Extended Properties of this member: Units - Engineering units (text) used for the analog input.
Sts_PVBad	Quality of PV value 1 = PV quality is flagged as Bad

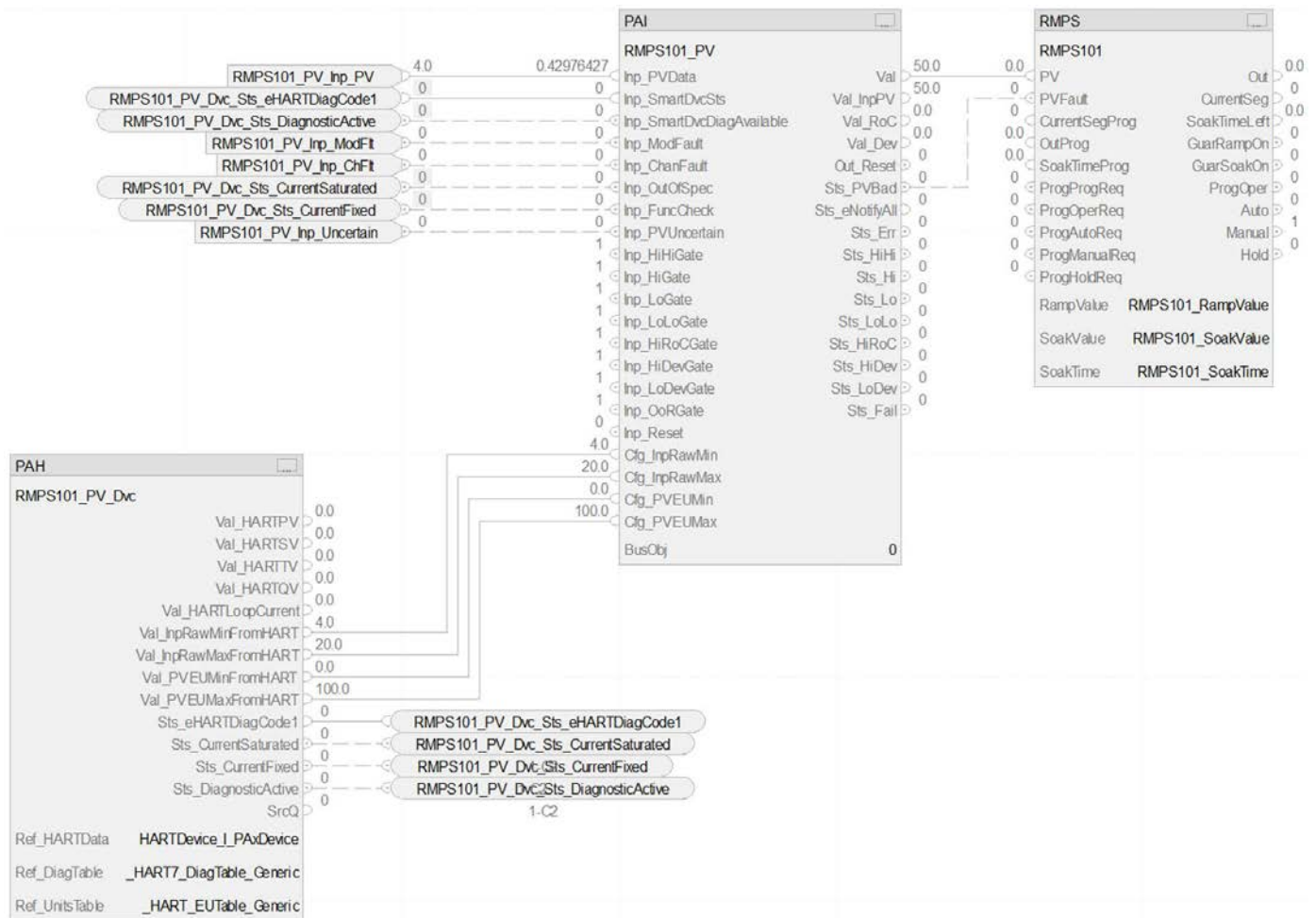
### RMPS Output Reference

Parameter	Description
RMPS100_Out	The output of the ramp/soak instruction

### RMPS Configuration Considerations

Operand	Type	Description
RMPS Tag	RAMP_SOAK	Instance of data structure (backing tag) required for proper operation of instruction. RMPS100 in this example corresponds to an instance of coordinating ramp/soak segments.
RampValue	REAL array	Enter a ramp value for each segment (0 to NumberOfSegs-1). Ramp values are entered as time in minutes or as a rate in units/minute. The TimeRate parameter reflects which method is used to specify the ramp. If a ramp value is invalid, the instruction sets the appropriate bit in Status and changes to Operator Manual or Program Hold mode. The array must be at least as large as NumberOfSegs. Valid = 0.0 to maximum positive float
SoakValue	REAL array	Enter a soak value for each segment (0 to NumberOfSegs-1). The array must be at least as large as NumberOfSegs. Valid = any float
SoakTime	REAL array	Enter a soak time for each segment (0 to NumberOfSegs-1). Soak times are entered in minutes. If a soak value is invalid, the instruction sets the appropriate bit in Status and changes to Operator Manual or Program Hold mode. The array must be at least as large as NumberOfSegs.

## CS\_RMPS\_HART Sheet



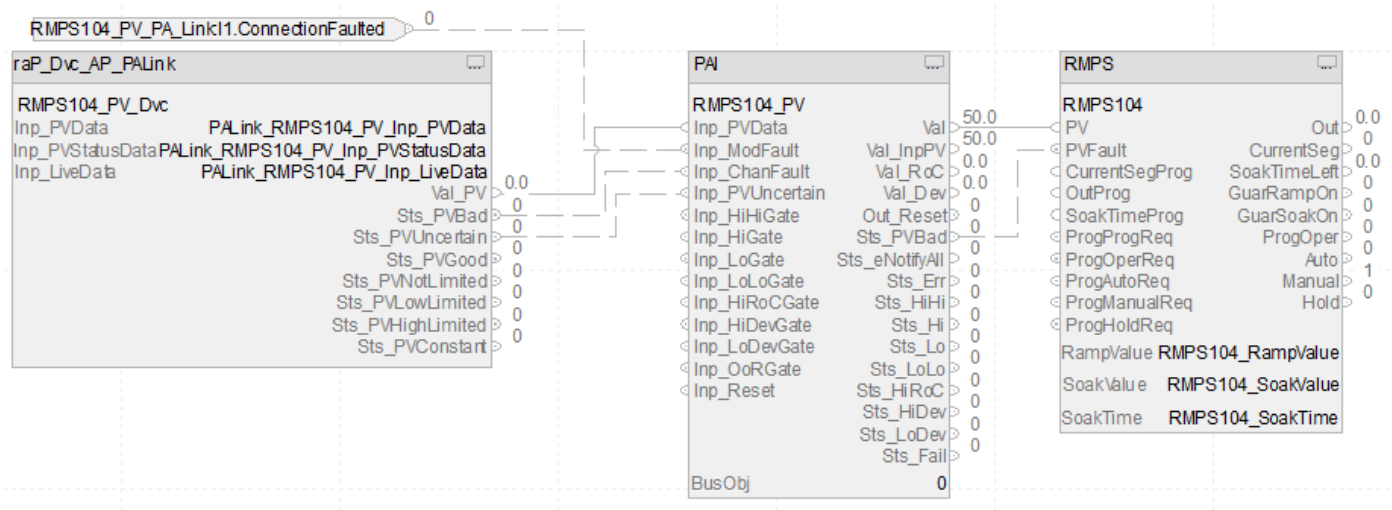
The CS\_RMPS\_HART control strategy operates the same as the CS\_RMPS control strategy but relies on HART input data.

- For information on PAH outputs to PAI inputs, see [CS\\_PAI\\_HART Sheet on page 149](#).
- Substitute for RMPS101 for XT100
- For more information, see [HART Integration on page 61](#).





CS\_RMPS\_PA Sheet



The CS\_RMPS\_PA control strategy operates the same as the CS\_RMPS control strategy but relies on Profibus PA input data.

- For information on Profibus PA device outputs to PAI inputs, see [CS\\_PAI\\_PA Sheet on page 156](#).
- Substitute for RMPS104 for XT100

For more information, see [FOUNDATION Fieldbus and Profibus PA Integration on page 99](#).

ACM Considerations for RMPS

Configure these parameters first because they affect the visibility of the remaining parameters in the RMPS object.

- Specify the type of analog input via the `PAI_Type` parameter
- If you use a specific I/O signal type, select the type for the `IO_Signal_Type` parameter

## ACM-Based Parameters for a RMPS Instance

Parameter	Visible When	Details
00 - Selection		
PAI_Type	always	<b>Important:</b> Select this parameter first as the option affects the remaining parameters.  Define the PAI type: PAI(Single_channel), PAID(Dual_channel), PAIM(Multi_channel), or External PAI(Single_channel)
Localize_PAO	always	Set to use a local routine in the program
IO_Signal_Type	PAI_Type = PAI(Single_channel)	Select the signal type: None, HART, EH_EthernetIP, FF, or PA.
Use_OOAP	Has_OOAP=True (controller parameter)	Set to use the bus for ownership and arbitration. See <a href="#">Process Controller on page 36</a>
Use_ArbitrationQ	Use_OOAP=True	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
01 - Options		
Cfg_UseHARTDigitalData	IO_Signal_Type=HART	Set to use HART Digital Data for the PV, SV, TV, and FV values
Cfg_UseHARTScaling	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Set to connect HART scaling from PAH object
Hart_Type	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Select the HART type (Generic, Hart5, Hart6, or Hart7) and the associated diagnostic table
Ref_HartDevice	IO_Signal_Type=HART Cfg_UseHARTDigitalData=False	Link to the HART device object. See (RA-LIB) Process > HART_Mapping > HART I/O Card Mapping
Ref_EtherNetIPModule	IO_Signal_Type=EH_EthernetIP	Link to the E+H EtherNet/IP device object. See (RA-LIB) Process > Module > Endress+Hauser for available objects
Ref_FF_Module	IO_Signal_Type=FF	Link to the FOUNDATION Fieldbus device object. See (RA-LIB) Process > Module > Foundation Fieldbus for available objects
Ref_PA_Module	IO_Signal_Type=PA	Link to the Profibus PA device object. See (RA-LIB) Process > Module > Profibus PA for available objects
03.00 - IO Configuration		
Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type and the configuration of the controller object I/O. See <a href="#">I/O Mapping on page 38</a> .		
PAI_Ref_Tag	PAI_Type=PAI(Single_channel)	Link to the analog input reference
	PAI_Type=ExternalPAI(Single_channel)	
PAID_Ref_Tag	PAI_Type=PAI(Dual_channel)	Link to the analog input (dual channel) reference
PAIM_Ref_Tag	PAI_Type=PAIM(Multi_channel)	Link to the analog input (multi channel) reference
Inp_PV	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=None	Link to the PV input reference
	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel) IO_SignalType=HART	
03.00.10 - Ref PAI Alarm Configuration		
Ref_HiHiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

Parameter	Visible When	Details
Ref_HiGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoLoGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiRoCGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_HiDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_LoDevGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference
Ref_OoRGate	PAI_RefTag is linked to an analog input reference PAI_Type=PAI(Single_channel)	Link to the gate reference

### 03.11 - IO Configuration

PAO_RefTag	always	Link to first analog output reference
Cfg_HasCVIntlkObj	Localize_PAO=True	Set if the analog output reference has an interlock CV
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Bus_Instance_CV	PAI_RefTag is linked to an analog input reference Has_OOAP=True (controller parameter) Use_OOAP=True Localize_PAO=True	Link to a bus array instance. This should be unique for each device
Inp_PosFdbk	Localize_PAO=True	Link to input position feedback
Out_CV	Localize_PAO=True	Link to the first output CV reference

## Additional Sub-Objects for a RMPS Control Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
CVIntlk	Configure an interlock for the CV instance See <a href="#">Interlocks on page 49</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
CVEvents	Configure an event to monitor for the CV instance See <a href="#">Event Logging on page 49</a>

## Organizational Control Strategies

This section includes information on several control strategies:

- CS\_raP\_Opr\_Area
- CS\_raP\_Opr\_Unit
- CS\_raP\_Opr\_EMGen
- CS\_raP\_Opr\_EPGen

### Area Control Strategy

The Process Area Add-On Instruction groups Units together, aggregates status from Unit objects, and broadcasts commands to Unit objects.

You can consolidate the status from groups of equipment, and display the consolidated status on an HMI. These status items include:

- Alarm Status
- Alarm Severity
- Mode
- Configuration Errors
- Prompt Status

You can also manage any of the following functions for a group of equipment with a global set of commands:

- Mode
- Alarm Acknowledge
- Alarm Reset
- Enable/Disable Alarms

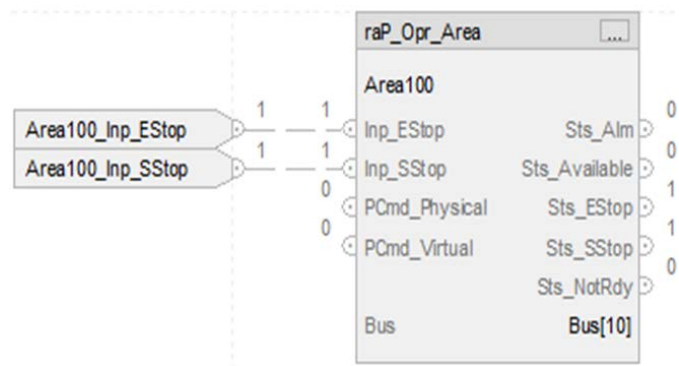
The CS\_raP\_Opr\_Area control strategy is available as a routine in the process library. Import the appropriate control strategy as a **routine** in your controller project.

The Process Area control strategy contains these routines:

Routine	Description
Area100	Function Block control strategy routine.
ExtdAlarms	Contains instances of external alarms and trigger logic. See <a href="#">Extended Alarms Routine on page 498</a>

### CS\_raP\_Opr\_Area Sheet

The Area100 routine contains the CS\_raP\_Opr\_Area sheet.



**CS\_raP\_Opr\_Area Input Reference**

Parameter	Description
Area100_Inp_EStop	1 = Emergency stop input ok.
Area100_Inp_SStop	1 = Software stop input ok.

**CS\_raP\_Opr\_Area Configuration Considerations**

Operand	Type	Description
PlantPAx® control	raP_Opr_Area	Instance of data structure (backing tag) required for proper operation of instruction
Bus	raP_UDT_Opr_Bus	Bus component for organization control 0 if not using organization Bus[x].Obj when using organization See the Rockwell Automation Library of Process Objects Reference Manual, publication <a href="#">PROCES-RM200</a> .

**ACM Considerations for an Area Instance**

Configure the Area parameters to manage the associated equipment.

## ACM-Based Parameters for an Area Instance

Parameter	Visible When	Details
<b>01 - Options</b>		
Bus_Instance	always	Link to a bus array instance. This should be unique for each device
<b>03 - IO Configuration</b> Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.		
Inp_EStopFdbk	always	Link to the EStop feedback reference
Inp_SStopFdbk	always	Link to the SStop feedback reference
<b>04 - Alarm Configuration</b>		
Cfg_HasEStopAlm	always	If Cfg_HasEStopAlm=True, ACM displays section 4.01 - Alarm EStop with additional parameters
Cfg_HasSStopAlm	always	If Cfg_HasSStopAlm=True, ACM displays section 04.02 - Alarm Soft Stop with additional parameters

## Additional Sub-Objects for an Area Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
ExtddAlm	Configure extended alarms for the control strategy. See <a href="#">Extended Alarms on page 54</a>
Events	Configure an event to monitor for the control strategy. See <a href="#">Event Logging on page 49</a>

## Unit Control Strategy

The Process Unit control strategy groups equipment together, and provides a propagation mechanism for aggregating status from equipment, and broadcasting commands to equipment. For example, each vessel, tank, mixer, machine within the control system would be considered a Unit. You can consolidate the status from groups of equipment, and display the consolidated status on an HMI.

These status items include:

- Alarm Status
- Alarm Severity
- Mode
- Configuration Errors
- Prompt Status

You can also manage any of the following functions for a group of equipment with a global set of commands:

- Mode
- Alarm Acknowledge
- Alarm Reset

The CS\_raP\_Opr\_Unit control strategy is available as a routine in the process library. Import the appropriate control strategy as a **routine** in your controller project.

The Process Unit control strategy contains these routines:

Routine	Description
Unit100	Function Block control strategy routine.

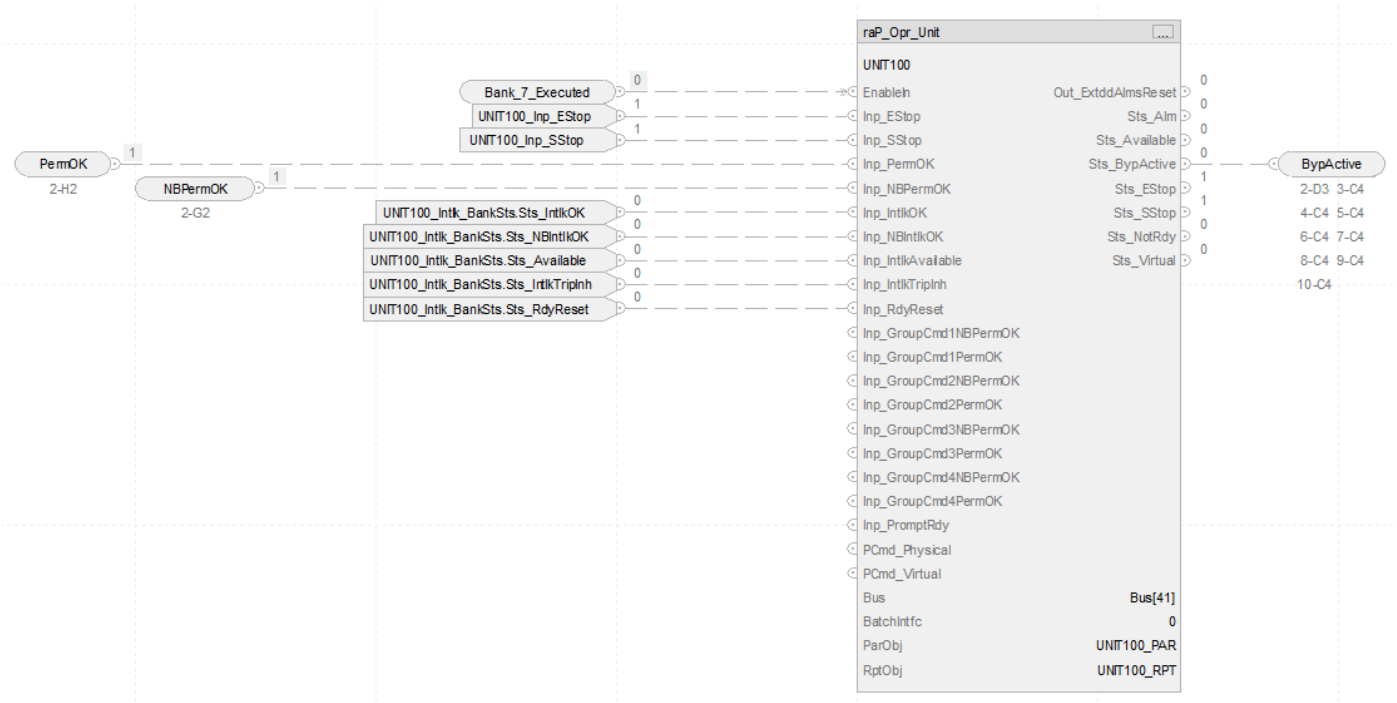
Routine	Description
ExtddAlarms	Contains instances of external alarms and trigger logic. See <a href="#">Extended Alarms on page 54</a>
Parameters	Contains raP_Opr_EMGen parameter mapping to and from Parameter blocks [_ParRpt (Enum, Integer, Real, String))] to raP_Opr_Unit instance. See <a href="#">Parameters on page 54</a>
Reports	Contains raP_Opr_EMGen report mapping to and from Parameter blocks [_ParRpt (Enum, Integer, Real, String))] to raP_Opr_Unit instance. See <a href="#">Reports on page 55</a>

The Unit100 routines contains these sheets:

Sheet	Description
CS_raP_Opr_Unit	Unit Add-On Instruction
Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The instruction monitors by passable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder. See <a href="#">Interlocks on page 49</a>

CS\_raP\_Opr\_Unit Sheet

The Unit100 routine contains theCS\_raP\_Opr\_Unit sheet





## CS\_raP\_Opr\_Unit Input Reference

Parameter	Description
BanI_7_Executed	[same content as we discussed in the meeting]
Unit100_Inp_EStop	1 = Emergency stop input ok.
Unit100_Inp_SStop	1 = Software stop input ok.
PermOK	Input connection from Permissives sheet 0 (State 1) 1 = On permissives OK, device can turn On
NBPermOK	Input connection from Permissives sheet 0 (State 1) 1 = Non-bypassable On permissives OK, device can turn On
Unit100_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
Unit100_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
Unit100_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
Unit100_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
Unit100_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset

## CS\_raP\_Opr\_Unit Output Reference

Parameter	Description
BypActive	Output connection to permissives and interlock bank sheets

## CS\_raP\_Opr\_Unit Configuration Considerations

Operand	Type	Description
PlantPAx control	raP_Opr_Unit	Instance of data structure (backing tag) required for proper operation of instruction
Bus	raP_UDT_Opr_Bus	Bus component
BatchIntfc	raP_UDT_Opr_PUnitFTBatch_Intfc	Batch interface
ParObj	raP_UDT_Opr_ParRpt_Intfc	Optional parameter object interface. Link to routine
RptObj	raP_UDT_Opr_ParRpt_Intfc	Optional report object interface. Link to routine

## ACM Considerations

Configure the Unit parameters to manage the associated equipment.

Parameter	Visible When	Details
<b>00.02 - Options</b>		
Bus_Instance	always	Link to a bus array instance. This should be unique for each device
Bus_PromptInstance	Cfg_HasPrompt=True	Link to a bus array instance for associated prompt instruction. This should be unique for each device
<b>01 - Programming Execution</b>		
Cfg_HasBatchObj	always	Set if the unit requires an external module to support batch data and commands
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction See <a href="#">Interlocks on page 49</a>

Parameter	Visible When	Details
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock.
Cfg_HasLatchDefeat	Cfg_HasIntlkObj=True	Set to connect a latched defeat to a state selection
Cfg_LatchDefeatConnection	Cfg_HasLatchDefeat=True	Set to connect a state to a latched defeat on an interlock
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow a state command See <a href="#">Permissives on page 50</a>
Cfg_HasPrompt	always	Set to create an instance of the PROMPT instruction for the HMI
Cfg_HasGroupCmd1PermObj	always	Set to create an instance of the PPERM instruction to allow a group command See <a href="#">Permissives on page 50</a>
Cfg_HasGroupCmd2PermObj	always	Set to create an instance of the PPERM instruction to allow a group command See <a href="#">Permissives on page 50</a>
Cfg_HasGroupCmd3PermObj	always	Set to create an instance of the PPERM instruction to allow a group command See <a href="#">Permissives on page 50</a>
Cfg_HasGroupCmd4PermObj	always	Set to create an instance of the PPERM instruction to allow a group command See <a href="#">Permissives on page 50</a>

**02.01 - Group Commands**

Cfg_HasGroupCmds	always	Set if the unit has group commands
Cfg_HasGroupSts	Cfg_HasGroupCmds=True	Set if there is group status

**03 - IO Configuration**

Assign a compatible I/O point to all applicable parameters in this section. The I/O connections depend on the I/O signal type.

Inp_EStopFdbk	always	Link to the EStop feedback reference
Inp_SStopFdbk	always	Link to the SStop feedback reference

**04 - Alarm Configuration**

Cfg_HasEStopAlm	always	If Cfg_HasEStopAlm=True, ACM displays section 4.01 - Alarm EStop Alarm with additional parameters
Cfg_HasSStopAlm	always	If Cfg_HasSStopAlm=True, ACM displays section 4.02 - Alarm SStop Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.03 - Interlock Trip Alarm with additional parameters
Cfg_HasGroupCmd1Fail	Cfg_HasGroupCmds=True	If Cfg_HasGroupCmd1Fail=True, ACM displays section 4.04 - Group Command 1 Fail Alarm with additional parameters
Cfg_HasGroupCmd2Fail	Cfg_HasGroupCmds=True	If Cfg_HasGroupCmd2Fail=True, ACM displays section 4.05 - Group Command 1 Fail Alarm with additional parameters
Cfg_HasGroupCmd3Fail	Cfg_HasGroupCmds=True	If Cfg_HasGroupCmd3Fail=True, ACM displays section 4.06 - Group Command 1 Fail Alarm with additional parameters
Cfg_HasGroupCmd4Fail	Cfg_HasGroupCmds=True	If Cfg_HasGroupCmd4Fail=True, ACM displays section 4.07 - Group Command 1 Fail Alarm with additional parameters

## Additional Sub-Objects for a Unit Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
ExtddAlm	Configure extended alarms for the control strategy See <a href="#">Extended Alarms on page 54</a>
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissives	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
UnitParameters	Configure parameters for the control strategy See <a href="#">Parameters on page 54</a>
UnitReports	Configure reports for the control strategy. See <a href="#">Reports on page 55</a>
GroupCmd1Permissive GroupCmd2Permissive GroupCmd3Permissive GroupCmd4Permissive	Configure permissives to allow group commands See <a href="#">Permissives on page 50</a>

## Generic Equipment Module (EMGEN) Control Strategy

An equipment module is a functional group of equipment that can carry out a finite number of specific minor processing activities. An equipment module is typically centered around a piece of process equipment (a weigh tank, a process heater, a scrubber, etc.). This term applies to both the physical equipment and the equipment entity.

The CS\_raP\_Opr\_EMGen control strategy controls an Equipment Module in a variety of modes and monitors for fault conditions.

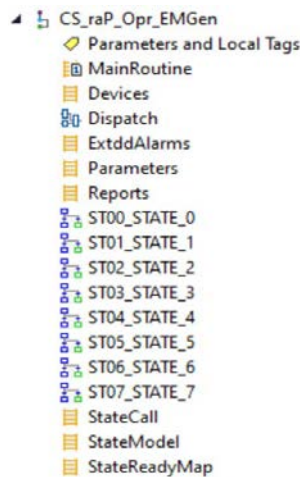
Use this control strategy when:

- You want to group equipment, and you want to apply a custom state model.
- You want to provide the following for a group of equipment:
  - Apply a mode model to the equipment group.
  - Definable Commands and states.
  - Apply interlocks and/or permissives to the group of equipment.
  - Parameters that define the behavior of the group of equipment.
  - Report resultant data from the group of equipment.
  - A faceplate that allows monitoring and control of the equipment grouping.
  - Alarm if any device fails.
  - Monitor step (description), and allow forcing of steps in maintenance mode.
  - Allow configurable alarms for certain process / equipment failure conditions.

Do NOT use this control strategy when:

- You need to sequence / coordinate a device, and do not require any of the above.
- You want to apply an ISA 88.01 state model to the equipment, use the CS\_raP\_Opr\_EPGen\_PHASE control strategy instead.
- You want to apply the PackML state model.

The EMGEN control strategy is available as a program in the process library:



Import the appropriate control strategy as a **program** in your controller project.

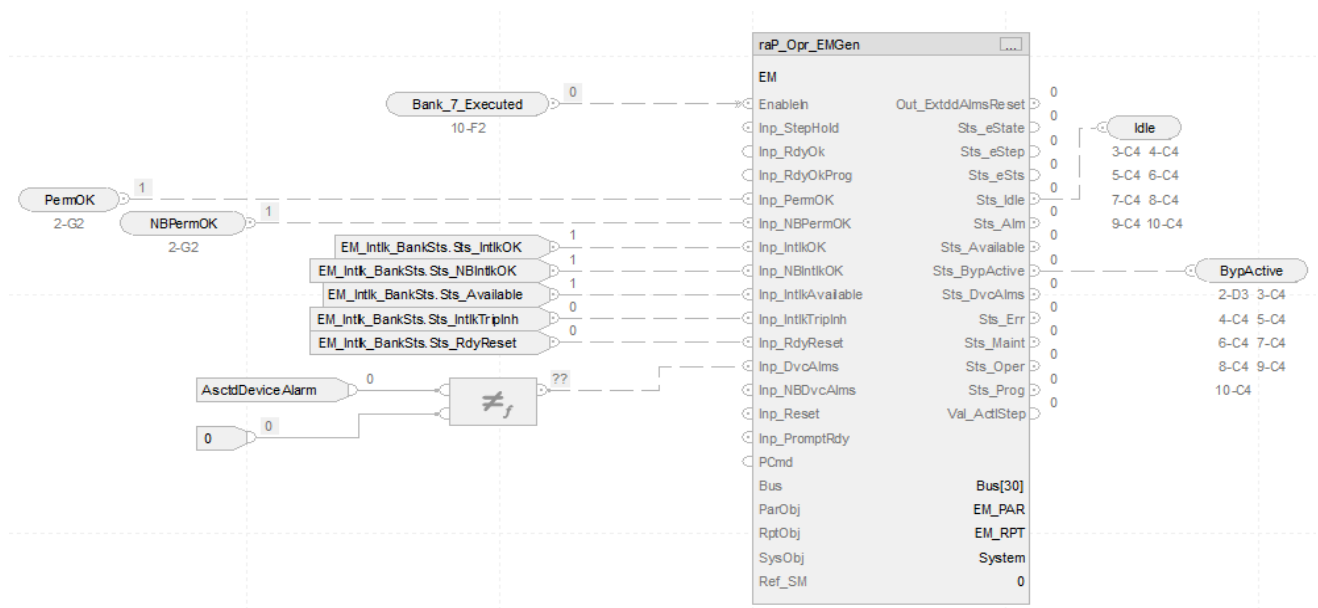
Routine	Description
Devices	Status of devices. Add logic appropriate to your application.
Dispatch	Contains raP_Opr_EMGen Add-On Instruction.
ExtddAlarms	Contains instances of external alarms and trigger logic. See <a href="#">Extended Alarms Routine on page 498</a>
Parameters	Contains raP_Opr_EMGen parameter mapping to and from Parameter blocks [_ParRpt (Enum, Integer, Real, String)] to raP_Opr_EMGen instance. See <a href="#">Parameters and Reports Routines on page 500</a>
Reports	Contains raP_Opr_EMGen report mapping to and from Parameter blocks [_ParRpt (Enum, Integer, Real, String)] to raP_Opr_EMGen instance. See <a href="#">Parameters and Reports Routines on page 500</a>
ST00_STATE_0...31	<p>32 available user-defined routines that contain logic which sequences and coordinates devices (implement states as required).</p> <p>You can rename these routines for your project.</p>
StateCall	Calls the routine for the associated state when that state is active.
StateModel	Sets which state is active based upon the state request for that state and any other required conditions.
StateReadyMap	<p>Equipment Module StateReadyMap Routine - Defines when each Equipment Module State is Available for both selection by the HMI and selection by Controller Logic.</p> <p>For each state number, 0 to 31:</p> <ul style="list-style-type: none"><li>• EM.Inp_RdyOk.0 to 31 needs to be true for that state to be available to select from the HMI</li><li>• EM.Inp_RdyOkProg.0 to 31 needs to be true for that state to be available to enter via Program Commands</li></ul>

Dispatch Routine

The Dispatch routine contains these Function Block sheets:

Sheet	Description
EMGEN100	Equipment Module Add-On Instruction
Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The instruction monitors by passable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.

## Dispatch EPGEN100 Sheet



### raP\_Opr\_EMGen Input References

Parameter	Description
Bank_7_Executed Where 7 = The total number of interlocks in your control strategy	1= All interlock banks have been evaluated
PermOK	Input connection from Permissives sheet 0 (State 1) 1 = On permissives OK, device can turn On
NBPermOK	Input connection from Permissives sheet 0 (State 1) 1 = Non-bypassable On permissives OK, device can turn On
EM_Intlk_BankSts.Sts_IntlkOK	Interlock bank status 1 = OK to run 0 = Stop
EM_Intlk_BankSts.Sts_NBIntlkOK	Interlock bank status 1 = All non-bypassable interlocks OK to run
EM_Intlk_BankSts.Sts_Available	Interlock bank status 1 = Available
EM_Intlk_BankSts.Sts_IntlkTriplnh	Interlock bank status 1 = Interlock trip inhibit - stops equipment but does not trip
EM_Intlk_BankSts.Sts_RdyReset	Interlock bank status 1 = A latched interlock (returned to OK) is ready to be reset
AsctdDeviceAlarm	Associated Device Alarm Active if any Bits are Logic 1

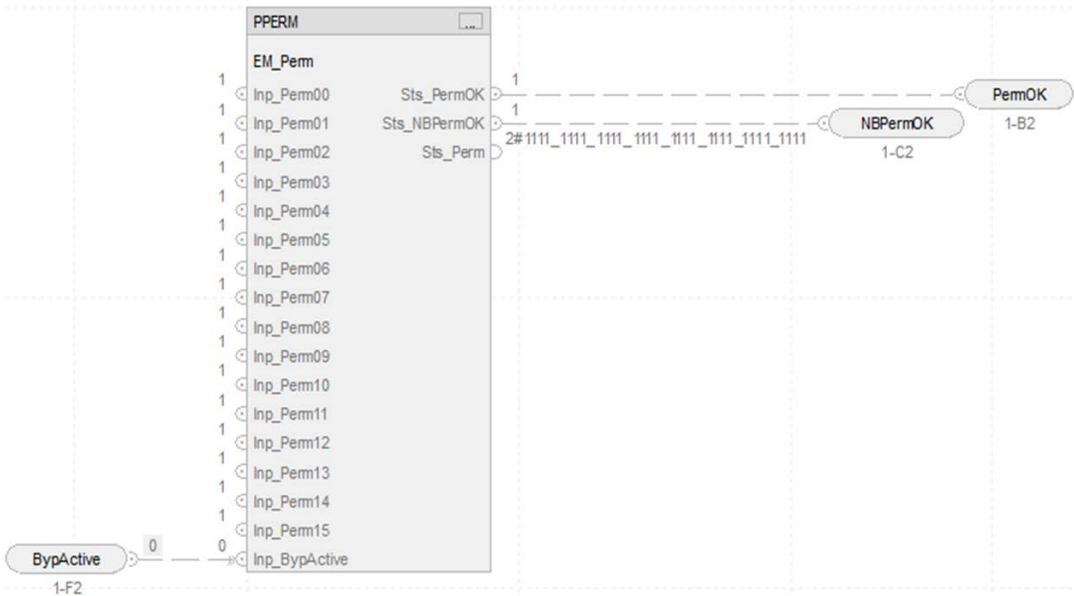
*raP\_Opr\_EMGen Output References*

Parameter	Description
BypActive	Output connection to permissives and interlock bank sheets

*raP\_Opr\_EMGen Configuration Considerations*

Operand	Type	Description
PlantPAX® control	raP_Opr_EMGen	Instance of data structure (backing tag) required for proper operation of instruction
Bus	raP_UDT_Opr_Bus	Bus component
ParObj	raP_UDT_Opr_ParRpt_Intfc	Optional parameter object interface. Link to routine
RptObj	raP_UDT_Opr_ParRpt_Intfc	Optional report object interface. Link to routine
SysObj	raP_UDT_Opr_System	System component.

**Dispatch Permissive Sheet**



*PPERM Input References*

Parameter	Description
BypActive	Input connection from the EPGEN100 sheet

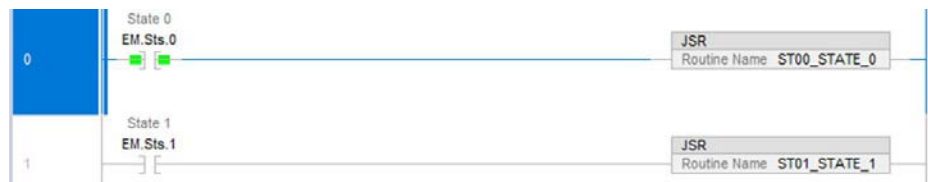
*PPERM Output References*

Parameter	Description
PermOK	Overall permissive status (1 = OK to energize)
NBPermOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

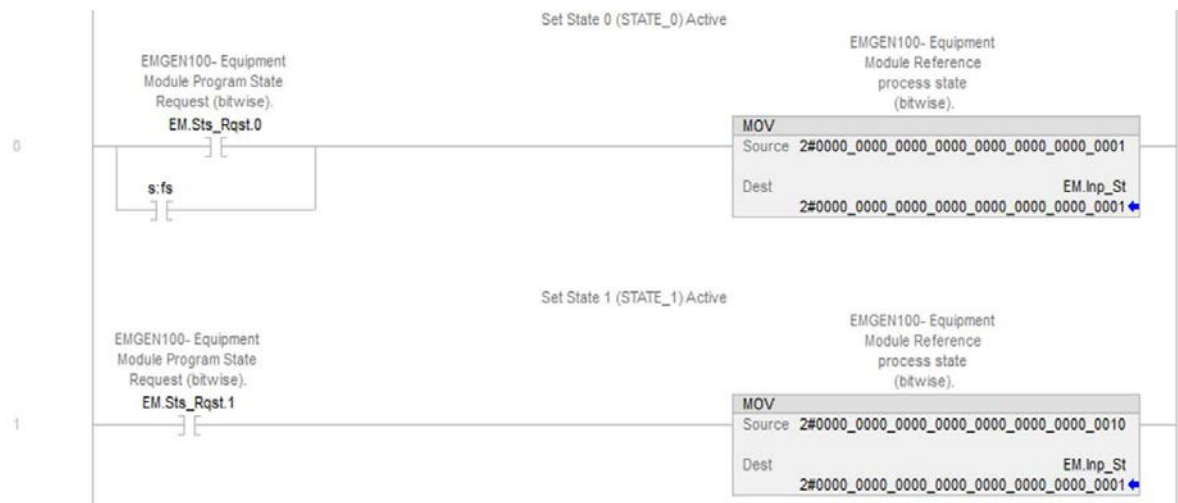
## State Routine Example



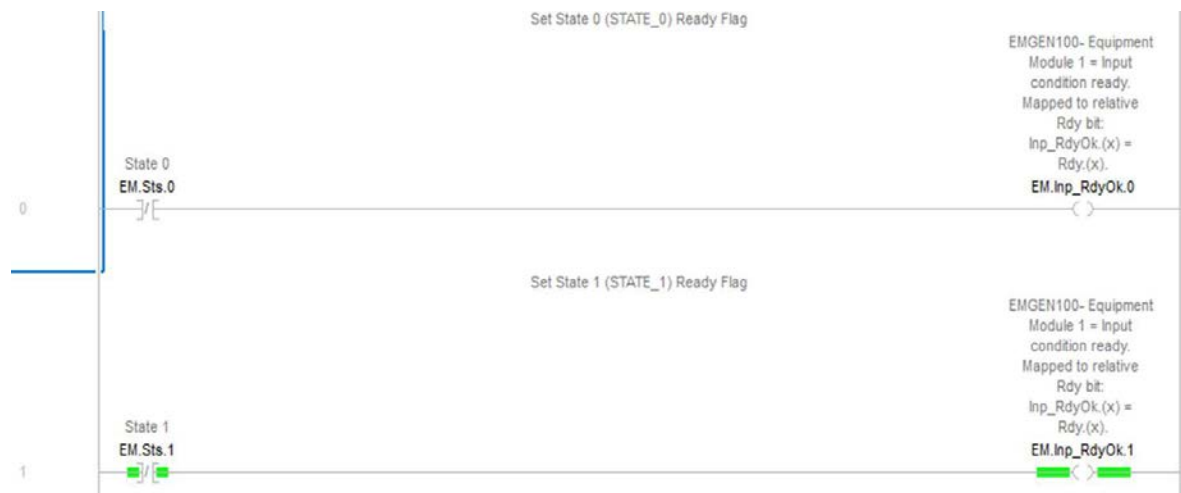
## StateCall Routine Example



## StateModel Routine Example



## StateReady Map Routine Example



## ACM Considerations for an EM Instance

Use the Cfg\_NumStates parameter to define the number of equipment module states.

## ACM-Based Parameters for an EM Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_ArbitrationQ	always	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
Use_VisualStateMachine_AOI	always	Set to deploy the visual state machine AOI. Clear to use deploy a state model
StateModelSelection	Use_VisualStateMachine_AOI=True	Select the type of state model: UserDefined S88 PackML NAMUR Equipment
Use_StepTransitionsDiagnostics	always	Select to deploy a state model with a visual state machine AOI
<b>00.02 - Options</b>		
Bus_Instance	always	Link to a bus array instance. This should be unique for each device
UseBus_Alias	always	0=Use Bus[x] in EM, 1=-Use EM_Bus Alias Tag
Bus_PromptInstance	Cfg_HasPrompt=True	Link to a bus array instance for associated prompt instruction. This should be unique for each device
<b>01 - Programming and Execution</b>		
BuildArbtrtnLogic	always	Set to include mode and ownership arbitration logic
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasLatchDefeat	Cfg_HasIntlkObj=True	Set to connect a latched defeat to a state selection
Cfg_LatchDefeatConnection	Cfg_HasLatchDefeat=True	Set to connect a state to a latched defeat on an interlock
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow a state command
Cfg_HasPrompt	always	Set to create an instance of the PROMPT instruction for the HMI



Parameter	Visible When	Details
<b>02 - Device Configuration</b>		
Cfg_NumStates	always	Enter the number of equipment module states (supports as many as 31 states)
Cfg_UseLegacyOwnership	always	Set to use legacy ownership
<b>03.x - State xx Configuration</b> Where x = 0...31		
State_xRoutineName	Cfg_NumStates=x	Enter the name of the routine (state name)
State_xRoutineType	Cfg_NumStates=x	Select the type of routine: RLL for ladder logic SFC for sequential function chart
<b>04 - Alarm Configuration</b>		
Cfg_HasDvcAlmsAlm	always	If Cfg_HasDvcAlmsAlm=True, ACM displays section 4.01 - Alarm Device Alarms Navigation with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.02 - Interlock Trip Alarm with additional parameters
Cfg_HasRptDataAlm	always	If Cfg_HasRptDataAlm=True, ACM displays section 4.03 - Alarm Report Data with additional parameters

## Additional Sub-Objects for an EM Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
ExtAlm	Configure extended alarms for the control strategy See <a href="#">Extended Alarms on page 54</a>
EPParameters	Configure parameters for the control strategy See <a href="#">Parameters on page 54</a>
EPReports	Configure reports for the control strategy. See <a href="#">Reports on page 55</a>
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissives	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
LocalTags	Configure locally-scoped tags for the control strategy. See <a href="#">Local Tags on page 55</a>
AsctdDevices	Configure associated control modules for the control strategy. See <a href="#">Associated Devices on page 56</a>
LocalLogic	Create custom rungs of ladder logic for the control strategy. See <a href="#">Local Logic on page 56</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
TransitionsDiag	Link a PPERM or PBL instruction to a transition of a state in the control strategy. See <a href="#">Transition Diagnostics on page 57</a>

## Generic Equipment Phase (EPGEN) Control Strategy

An equipment phase is a functional group of equipment that can conduct a finite number of specific minor processing activities when directed by a phase (recipe).

The CS\_raP\_Opr\_EPGen\_PHASE control strategy controls an Equipment Phase in various modes and monitors for fault conditions.

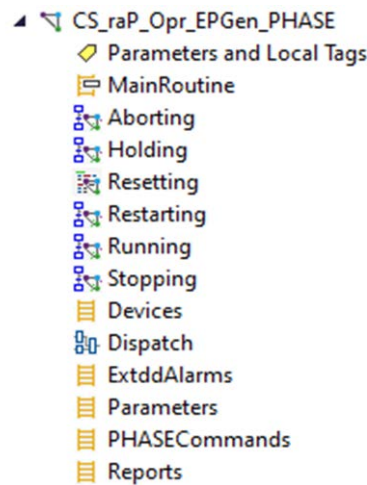
Use this control strategy when:

- You want to group equipment, and you want to apply the ISA 88.01 state model using PhaseManager™.
- You want to provide the following for a group of equipment.
  - Apply a mode model to the equipment group.
  - Apply interlocks and/or permissives to the group of equipment.
  - Parameters that define the behavior of the group of equipment.
  - Report resultant data from the group of equipment.
  - A faceplate that allows monitoring and control of the equipment grouping.
  - Monitor step (description), and allow forcing of steps in maintenance mode.
  - Allow alarms to be defined for certain process / equipment failure conditions.
  - Alarming function, including alarms based on device failure.

Do NOT use this control strategy when:

- You must sequence or coordinate a device, and do not require any of the above.
- You want to apply a custom state model to the equipment, use the CS\_raP\_Opr\_EMGen control strategy instead.
- You want to apply the PackML state model.

The EPGEN control strategy is available as a program in the process library:



Import the appropriate control strategy as a **program** in your controller project.

Routine	Description
Aborting	Used for shutting down equipment in an emergency situation. If you have implemented Stopping, you would at a minimum duplicate the stopping logic within Aborting. In some cases, the sequence in an emergency situation (Aborting) differs from the orderly shutdown of equipment (Stopping). Add logic appropriate to your application.
Holding	Used if equipment or a subset of equipment must be shut down when the phase enters the hold state. It can also be advantageous to release owned equipment if maintaining ownership while held constrains production by maintaining ownership of shared equipment. Add logic appropriate to your application.
Resetting	Used to perform “clean-up” activities such as release owned equipment. Add logic appropriate to your application.
Restarting	Generally implemented if Holding is implemented. Used to bring equipment from the state that it is in at the end of the Holding state back to the state it was in prior Holding. Add logic appropriate to your application.
Running	Use to start up equipment, and acquire ownership of equipment (if necessary). Add logic appropriate to your application.
Stopping	Use if equipment must be shut down in a given sequence.
Devices	Status of devices. Add logic appropriate to your application.
Dispatch	Contains the raP_Opr_EPGen Add-On Instruction.
ExtddAlarms	Contains instances of external alarms and trigger logic.
Parameters	Equipment Phase Parameters Routine - EP parameter mapping and logic
PHASECommands	Maps commands from EPGEN to PhaseManager commands
Report	Equipment Phase Reports Routine - EP Report mapping and logic

## Dispatch Routine

The Dispatch routine contains these Function Block sheets:

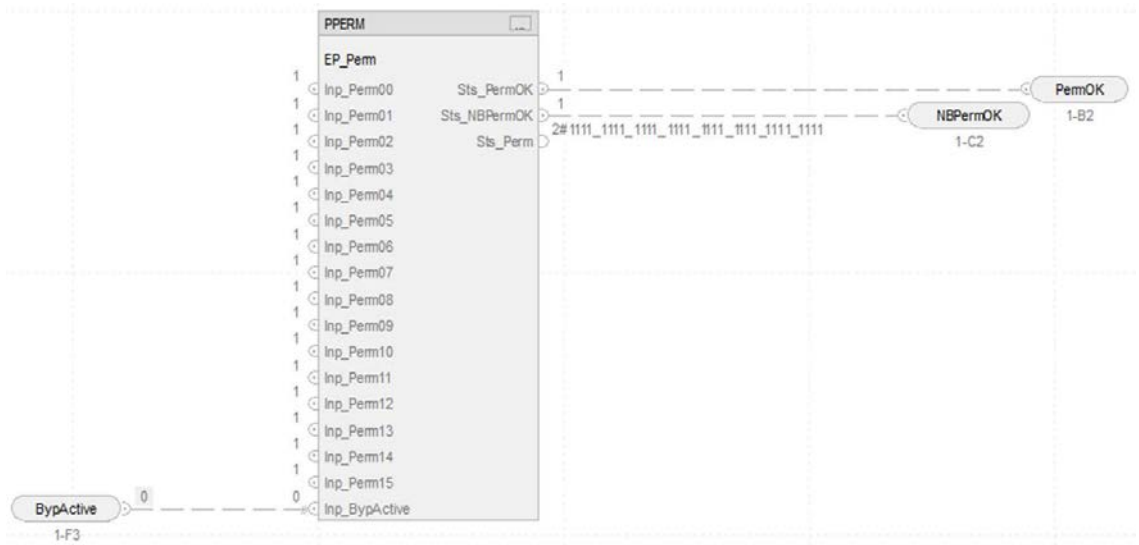
Sheet	Description
EPGEN100	Equipment Phase Add-On Instruction
Permissives	Process Permissives instruction The Process Permissives (PPERM) instruction collects, or sums up, the permissive conditions that let a piece of equipment energize. In most cases, permissive conditions must be true to energize equipment. Once the equipment is energized, permissives are ignored.
Interlock Bank 0 Interlock Bank 1 Interlock Bank 2 Interlock Bank 3 Interlock Bank 4 Interlock Bank 5 Interlock Bank 6 Interlock Bank 7	The instruction monitors by passable and non-bypassable Interlocks that force the analog output to a specific configured (safe) value or to maintain the current value (configurable). There are 8 interlock bank sheets; each sheet exposes 16 of the available 32 interlocks per bank by default. Use the sheets and interlocks that you need and delete the remainder.



raP\_Opr\_EPGen Configuration Considerations

Operand	Type	Description
PlantPAX® control	raP_Opr_EPGen	Instance of data structure (backing tag) required for proper operation of instruction
Ref_Phase	PHASE	Referenced phase.
Bus	raP_UDT_Opr_Bus	Bus component
ParObj	raP_UDT_Opr_ParRpt_Intfc	Optional parameter object interface
RptObj	raP_UDT_Opr_ParRpt_Intfc	Optional report object interface
SysObj	raP_UDT_Opr_System	System component.

Dispatch Permissive Sheet



PPERM Input References

Parameter	Description
BypActive	Input connection from the EPGEN100 sheet

PPERM Output References

Parameter	Description
PermOK	Overall permissive status (1 = OK to energize)
NBPermOK	Non-bypassable permissive status (1 = all non-bypassable permissives OK to energize)

PHASECommands Routine

Maps commands from the EP\_GEN instance to PhaseManager commands.

Steps required to map EP\_GEN Phase Commands in Ladder:

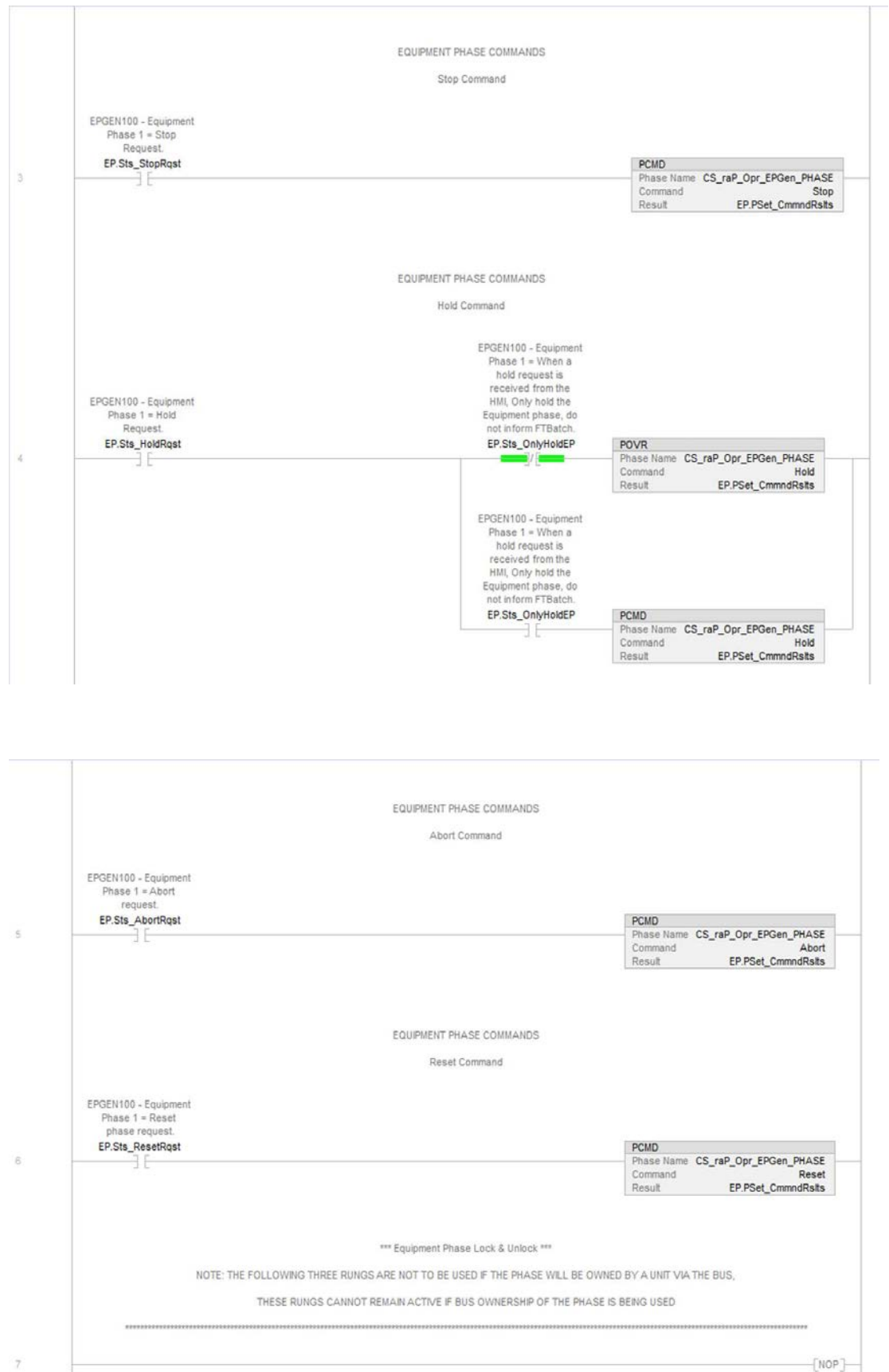
1. Open the CS\_raP\_Opr\_EPGen PhaseCommands Routine.
2. Modify the selected phase requests as required.

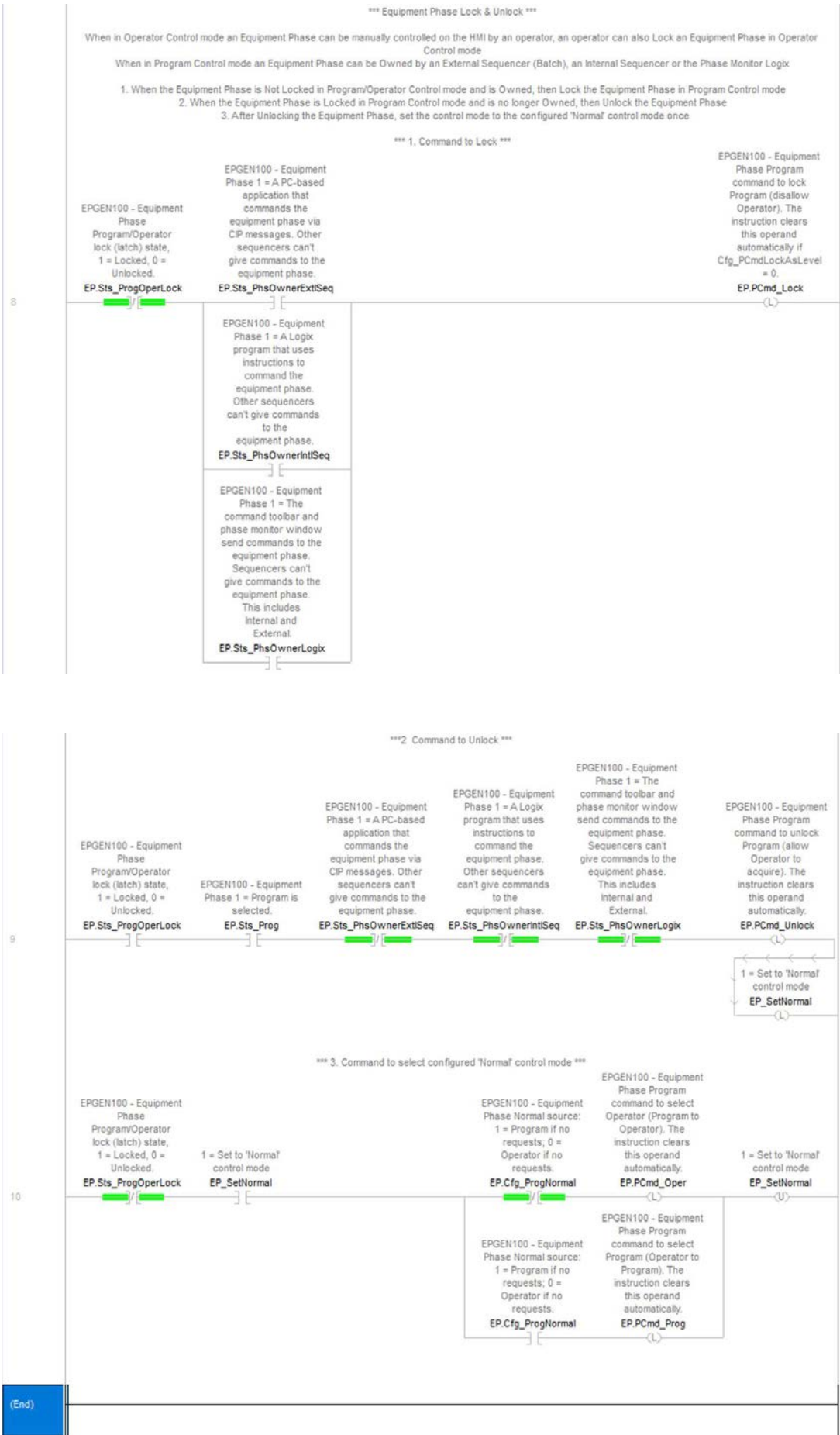
EP\_GEN Phase request interface points are as follows:

- Sts\_StartRqst
- Sts\_HoldRqst
- Sts\_RestartRqst
- Sts\_StopRqst
- Sts\_AbortRqst
- Sts\_ResetRqst
- Sts\_PauseRqst
- Sts\_ResumeRqst
- Sts\_StateCmpltRqst

3. EP\_GEN Request interface points are defined so that a 1 = Requested.
- Typically a command rung would contain an XIC that represents the EP\_GEN phase request (.Sts\_<state>Rqst), and a PCMD or POVR instruction to issue the corresponding command to the PhaseManager Phase
  - Phase Commands (PCMD) exist for Start, ReStart, Reset, Pause, and Resume states.
  - Phase Override Commands (POVR) exist for Stop, Hold, and ABORT states.
  - A Phase Command required for each Phase State routine you have defined within your PhaseManager Phase.
4. The PCMD and POVR require definition of several reference tags:
- Phase Name <tag>\_Phase
  - Command <phase command>
  - Result <tag>\_PSet\_CmmndRslts









## ACM Considerations for an EP Instance

Configure the EP parameters to define equipment phases.

### ACM-Based Parameters for an EP Instance

Parameter	Visible When	Details
<b>00 - Selection</b>		
Use_ArbitrationQ	always	Set to use the ArbitrationQ instruction for ownership queuing. See <a href="#">Process Controller on page 36</a>
Use_StepTransitionsDiagnostics	always	Select to deploy a state model with a visual state machine AOI
<b>00.02 - Options</b>		
Bus_Instance	always	Link to a bus array instance. This should be unique for each device
UseBus_Alias	always	0=Use Bus[x] in EM, 1=Use EM_Bus Alias Tag
Bus_PromptInstance	Cfg_HasPrompt=True	Link to a bus array instance for associated prompt instruction. This should be unique for each device
<b>01 - Programming and Execution</b>		
BuildArbtrtnLogic	always	Set to include mode and ownership arbitration logic.
Cfg_HasIntlkObj	always	Set to create an instance of the PINTLK instruction
UseResetWireConnectors	Cfg_HasIntlkObj=True	Set to connect the Out_Reset of the device to the Inp_Reset of the associated interlock
Cfg_HasLatchDefeat	Cfg_HasIntlkObj=True	Set to connect a latched defeat to a state selection
Cfg_LatchDefeatConnection	Cfg_HasLatchDefeat=True	Set to connect a state to a latched defeat on an interlock
Cfg_HasPermObj	always	Set to create an instance of the PPERM instruction to allow a state command
Cfg_HasPrompt	always	Set to create an instance of the PROMPT instruction for the HMI
<b>02 - Device Configuration</b>		
Cfg_ExtrenalSqncr	always	Set if the phase has an external sequencer (FactoryTalk Batch)
Cfg_IntrnalSqncr	always	Set if the phase has an internal sequencer (Sequence Manager)
Cfg_HasAbortingState	always	Set if there is an aborting phase
Cfg_HasHoldingState	always	Set if there is a holding state
Cfg_HasReSettingState	always	Set if there is a resetting phase
Cfg_HasReStartingState	always	Set if there is a restarting phase
Cfg_HasRunningState	always	Set if there is a running phase
Cfg_HasStoppingState	always	Set if there is a stopping phase
Cfg_HasPause	always	Set if there is a pause option
Cfg_UseLegacyOwnership	always	Set to use legacy ownership
<b>02.02 - External Sequence Configuration</b>		
Cfg_HasPhaseOwnership	Cfg_ExtrenalSqncr=True	Set if the phase has ownership logic
Cfg_ClearFailure	Cfg_ExtrenalSqncr=True	Set if an alarm acknowledgment or reset clears a phase failure
Cfg_MapAlmToPhs	Cfg_ExtrenalSqncr=True	Set to map alarm codes from the phase to the phase PFL codes
Cfg_Parameter_TOC	Cfg_ExtrenalSqncr=True	Set to download parameters upon transfer of control (PXRQ routine)
<b>02.03.00 - External Sequence Configuration</b>		

Cfg_ShedOnAlm	always	Set if an alarm issues a stop phase request
<b>03.00 - Aborting Phase Configuration</b>		
AbortingRoutineType	Cfg_HasAbortingState=True	Set to create a routine for the phase logic
<b>03.01 - Holding Phase Configuration</b>		
HoldingRoutineType	Cfg_HasHoldingState=True	Set to create a routine for the phase logic
<b>03.02 - ReSetting Phase Configuration</b>		
ReSettingRoutineType	Cfg_HasReSettingState=True	Set to create a routine for the phase logic
<b>03.03 - ReStarting Phase Configuration</b>		
ReStartingRoutineType	Cfg_Has_ReStartingState=True	Set to create a routine for the phase logic
<b>03.04 - Running Phase Configuration</b>		
RunningRoutineType	Cfg_HasRunningState=True	Set to create a routine for the phase logic
<b>03.05 - Stopping Phase Configuration</b>		
StoppingRoutineType	Cfg_HasStoppingState=True	Set to create a routine for the phase logic
<b>04 - Alarm Configuration</b>		
Cfg_HasDvcAlmsAlm	always	If Cfg_HasDvcAlmsAlm=True, ACM displays section 4.01 - Alarm Device Alarm with additional parameters
Cfg_HasIntlkTripAlm	always	If Cfg_HasIntlkTripAlm=True, ACM displays section 4.02 - Interlock Trip Alarm with additional parameters
Cfg_HasRptDataAlm	always	If Cfg_HasRptDataAlm =True, ACM displays section 4.03 - Alarm Report Data with additional parameters

## Additional Sub-Objects for an EP Instance

Each sub-object has a tab on the configuration dialog for the control strategy object.

Sub-Object	Description
ExtAlm	Configure extended alarms for the control strategy See <a href="#">Extended Alarms on page 54</a>
EPParameters	Configure parameters for the control strategy See <a href="#">Parameters on page 54</a>
EPReports	Configure reports for the control strategy. See <a href="#">Reports on page 55</a>
Interlocks	Configure interlocks for the control strategy See <a href="#">Interlocks on page 49</a>
Permissives	Configure permissives to allow output commands See <a href="#">Permissives on page 50</a>
LocalTags	Configure locally-scoped tags for the control strategy. See <a href="#">Local Tags on page 55</a>
AsctdDevices	Configure associated control modules for the control strategy. See <a href="#">Associated Devices on page 56</a>
LocalLogic	Create custom rungs of ladder logic for the control strategy. See <a href="#">Local Logic on page 56</a>
Events	Configure an event to monitor for the control strategy See <a href="#">Event Logging on page 49</a>
TransitionsDiag	Link a PPERM or PBL instruction to a transition of a state in the control strategy. See <a href="#">Transition Diagnostics on page 57</a>

## Extended Alarms Routine

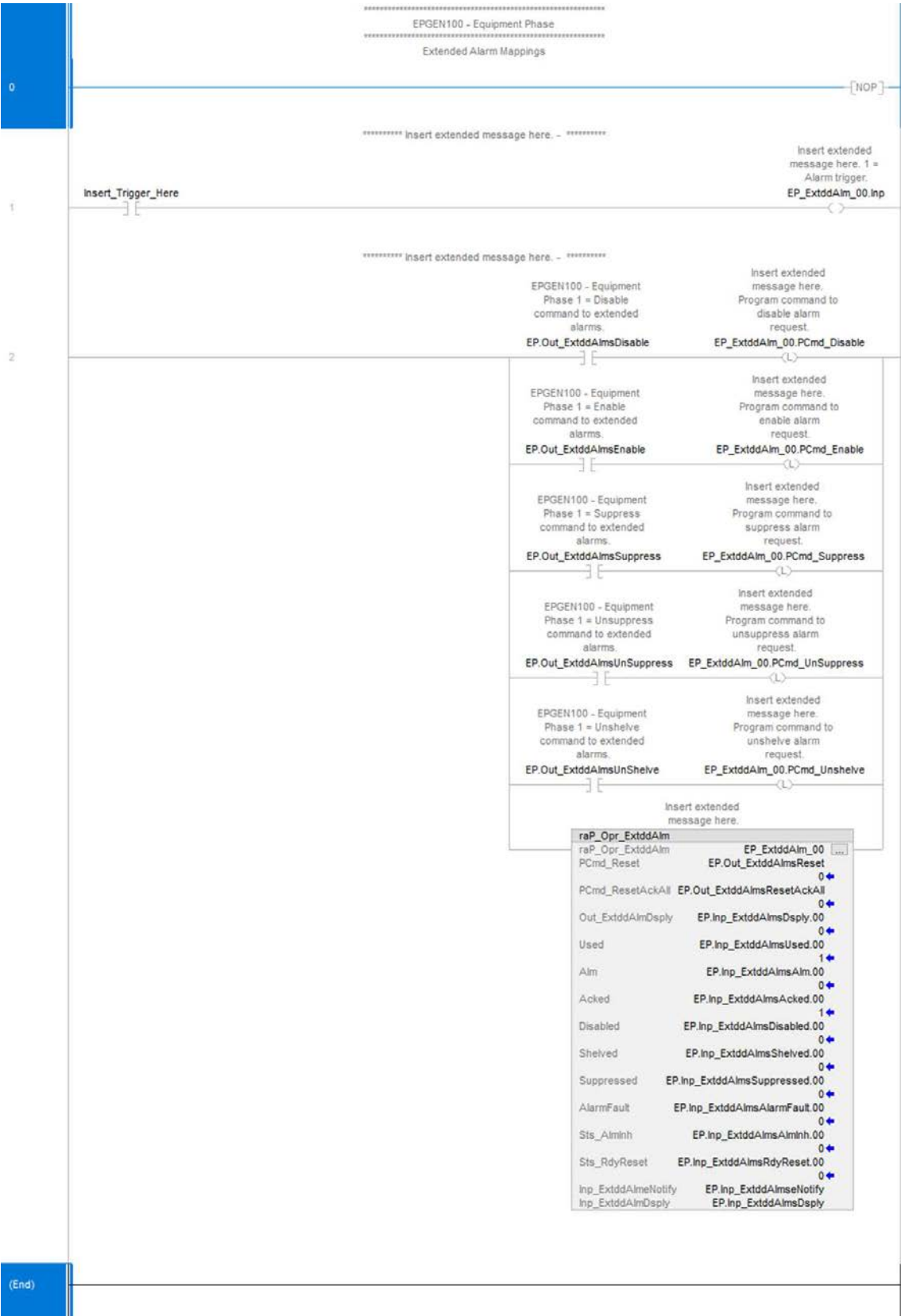
The raP\_Opr\_ExtddAlm (Extended Alarm Block) Add-On Instruction is used to provide notification to operators of abnormal conditions or events for up to 32 additional items external to a parent object. For more information, see the Rockwell Automation Library of Process Objects, publication [PROCES-RM200](#).

This instruction handles these connections.

Commands from the Parent Object	Status from raP_Opr_ExtddAlm
<ul style="list-style-type: none"> <li>• Acknowledge</li> <li>• Reset</li> <li>• Enabling/Disabling</li> <li>• Suppress/Unsuppress</li> <li>• UnShelve</li> </ul>	<ul style="list-style-type: none"> <li>• Used</li> <li>• Alarm</li> <li>• Acknowledged</li> <li>• Disabled</li> <li>• Suppressed</li> <li>• Shelved</li> <li>• Alarm Fault</li> <li>• Ready for Reset</li> <li>• Notify value</li> </ul>

## raP\_Opr\_ExtddAlm Parameters

Parameter	Description
PCmd_Reset	Program command to reset alarm request.
PCmd_ResetAckAll	Program command to reset and acknowledges all alarms. The instruction clears this operand automatically.
Out_ExtddAlmDsply	1 = Extended alarm severity value is greatest of all extended alarms active. Each bit represents an individual alarm (0...31).
Used	1 = Used.
Alm	1 = Alarm is active.
Acked	1 = In alarm acknowledged.
Disabled	1 = Alarm disabled.
Shelved	1 = Alarm shelved.
Suppressed	1 = Alarm suppressed.
AlarmFault	1 = Alarm fault.
Sts_Almlnh	1 = One or more alarms shelved, disabled, or suppressed.
Sts_Rdy_Reset	1 = A latched alarm condition is ready to be reset.
Inp_ExtddAlmeNotify	Extended alarms status enumerate values: 0 = Not in alarm 1 = Not in alarm unacknowledged or reset requires 2 = Low severity alarm acknowledged 3 = Low severity alarm unacknowledged 4 = Medium severity alarm acknowledged 5 = Medium severity alarm unacknowledged 6 = High severity alarm acknowledged 7 = High severity alarm unacknowledged 8 = Urgent severity alarm acknowledged 9 = Urgent severity alarm unacknowledged
Inp_ExtddAlmDsply	1 = Extended alarm severity value is greatest of all extended alarms active. Each bit represents an individual alarm (0...31).



# Parameters and Reports Routines

The raP\_Tec\_ParRpt Add-On Instruction is used to implement parameter data items. Use when:

- You need the ability to view or modify a parameter from either the HMI or from logic.
- You must arbitrate parameter input based on mode.

- You need the ability to limit the value of a parameter, from either the HMI or logic.
- You need the ability to capture an initial parameter value (based on a trigger), and provide an indication if the parameter was adjusted from the initial value.
- You must limit the adjustment of a parameter within a deadband relative to an initial value.
- You must apply command confirmation (that is, Electronic Signature) to parameter entry from the HMI.
- Your parameter is read-only or read/write.
- You need a Parameter (recipe) or Report (resultant) parameter.
- Your parameter is of data type: Integer, Real, String, or is an Enumeration.

For more information, see the Rockwell Automation Library of Process Objects, publication [PROCES-RM200](#).

## raP\_Tec\_ParRpt References

Parameter	Description
PSet_E	Program issued setting of enumeration parameter value.
PSet_I	Program issued setting of integer parameter value.
PSet_R	Program issued setting of real parameter value.
PSet_S	Program issued setting of string parameter value.

## raP\_Tec\_ParRpt Configuration Considerations

Operand	Type	Description
ParObj	raP_UDT_Opr_ParRpt_INTfC	Parameter object link to equipment
RptObj	raP_UDT_Opr_ParRpt_INTfC	Report object link to equipment

**IMPORTANT** You cannot set both ParObj and RptObj in the same Add-On Instruction.

- If you set ParObj, then RptObj must be zero.
- If you set RptObj, then ParObj must be zero.

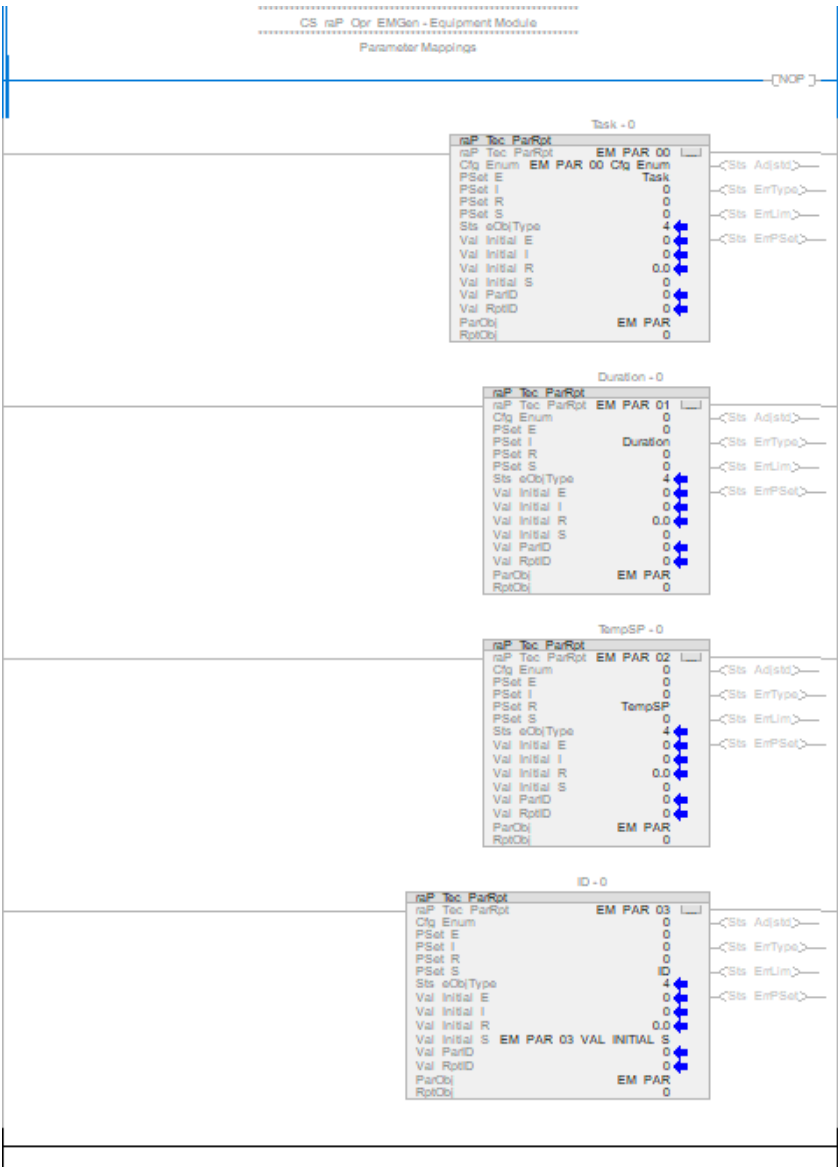
## CS\_raP\_Opr\_EMGen or CS\_raP\_Opr\_EPGen Parameters Routine

CS\_raP\_Opr\_EMGen and CS\_raP\_Opr\_EPGen have a similar structure for parameter routines.

Maps Parameters from Phase tags (input) to the standard parameter structure. The steps required to create this mapping logic are as follows:

1. First determine the parameters needed for your Equipment Phase, and confirm the associated tags (input) are defined within your PhaseManager program.
2. Open the Parameters Routine.
3. Start at parameter zero (EM\_Par\_00 or EP\_Par\_00), and determine the type of parameter block required (Enumeration, Integer, Real, or String).
4. Modify the rest of the reports routine as needed.

Typically the PSet variable in the instruction would represent the parameter tag (as defined in the program tags). For each additional Parameter required, increment the parameter number. Up to a maximum of 496 parameters and 496 reports can be included per instance.



## Notes:







# Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	<a href="http://rok.auto/support">rok.auto/support</a>
Local Technical Support Phone Numbers	Locate the telephone number for your country.	<a href="http://rok.auto/phonesupport">rok.auto/phonesupport</a>
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	<a href="http://rok.auto/techdocs">rok.auto/techdocs</a>
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	<a href="http://rok.auto/literature">rok.auto/literature</a>
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	<a href="http://rok.auto/pcdc">rok.auto/pcdc</a>

## Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at [rok.auto/docfeedback](http://rok.auto/docfeedback).

## Waste Electrical and Electronic Equipment (WEEE)



At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental compliance information on its website at [rok.auto/pec](http://rok.auto/pec).

Allen-Bradley, FactoryTalk, FLEX 5000, expanding human possibility, Integrated Architecture, Logix 5000, PlantPAx, PowerFlex, PowerMonitor, Rockwell Automation and Studio 5000 Logix Designer are trademarks of Rockwell Automation, Inc.  
EtherNet/IP is a trademark of ODVA, Inc.  
Trademarks not belonging to Rockwell Automation are property of their respective companies. Trademarks not belonging to Rockwell Automation are property of their respective companies.  
Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

rockwellautomation.com — expanding human possibility®

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000  
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2663 0600  
ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608  
UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800