Honeywell

Experion PKS

Control Builder Components Reference

EPDOC-XX15-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX15-en-431A	431	0	February 2015

Disclaimer

Contents

1 About This Document

This document provides a brief technical reference for function blocks configured through Control Builder. It includes a list of configurable parameters associated with a given block.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

2 Control Builder Components

Related topics

"Component Categories and Types" on page 14
"Hardware relation category" on page 15
"Functional relation category" on page 18
"Component Libraries" on page 20

2.1 Component Categories and Types

We divide the Control Builder components into these two major categories:

- “Functional relation category” on page 18
- “Hardware relation category” on page 15

2.2 Hardware relation category

The hardware relation category includes the physical equipment block types provided in Control Builder. These block types let you quickly integrate the related control hardware into your control strategy. Figure 1 shows the physical equipment that relates to the corresponding hardware relations covered in this document for a typical non-redundant system. Figure 2 shows the physical equipment that relates to the corresponding hardware relations for a typical redundant system.

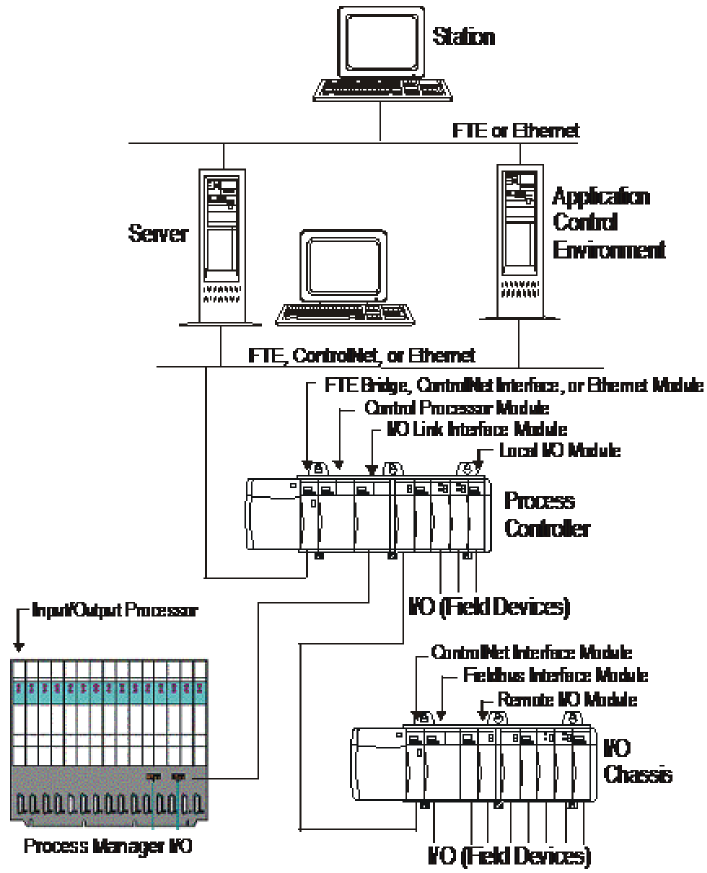

Figure 1: Physical Equipment reference for corresponding hardware component in typical non-redundant system architecture.

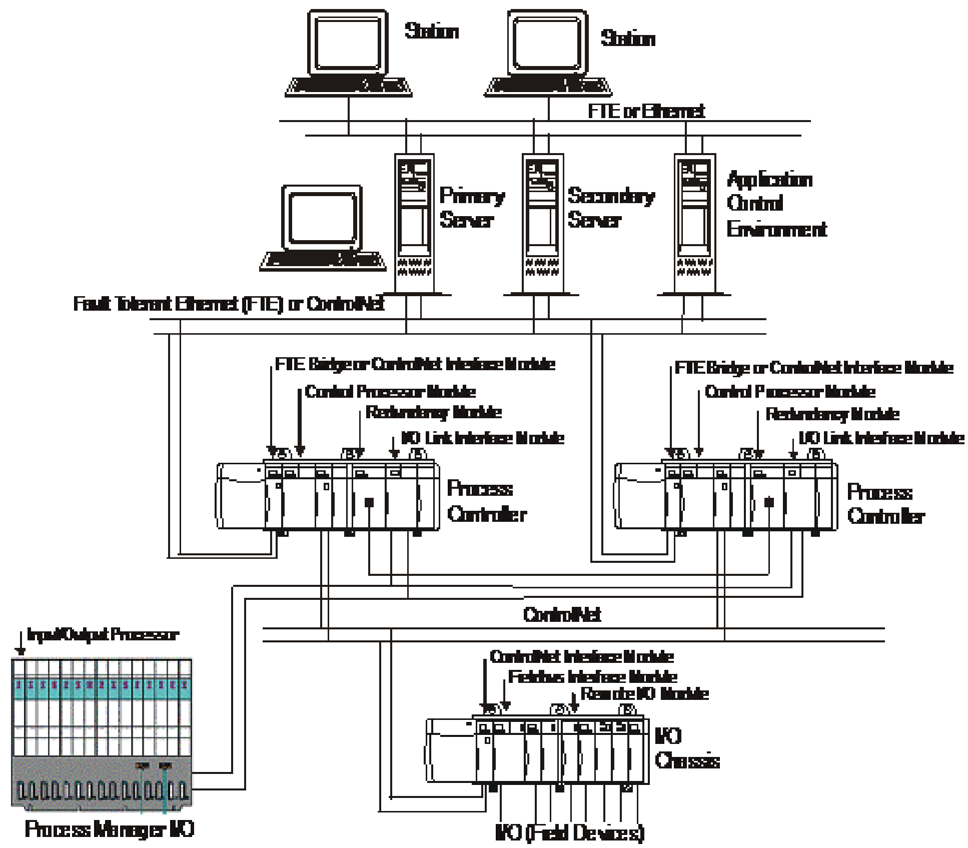

Figure 2: Physical Equipment reference for corresponding hardware component in typical redundant system architecture.

2.2.1 Physical equipment block types

Table 1 identifies the physical equipment block types provided to represent corresponding major control hardware components. The Control Execution Environment (CEE) block is included as one of these block types because of its relationship with the Control Processor, although it is a functional type more than a physical one.

Table 1: Physical equipment block types.

Physical Equipment Blocks	
Type	**Description**
"Control Processor Module Block" on page 248	Defines name/location and Control Execution Environment (CEE) assignment for Primary and Secondary CPMs in connected C200 Controllers. This CPM is redundancy compliant.
"Control Execution Environment Block" on page 256	Supports block execution and communications in given CPM.
"Input Type I/O Module Blocks" on page 287	Provides links for I/O channels to interface physical I/O module to given Control Processor Module. This includes Series R, Series H, Series A and HART Input modules.
"Output Type I/O Module Blocks" on page 300	Provides links for I/O channels to interface physical I/O module to given Control Processor Module. This includes Series R, Series H, Series A and HART Output modules.

Physical Equipment Blocks	
Type	**Description**
"Pulse Input Module Block TC-MDP081/TK-MDP081" on page 75	Serves as the interface board between the C200 Process Controller and field transducers such as tachometers, flow meters, and magnetic pickups.
"Pulse Input Module Block CC-PPIX01" on page 76	Serves as an interface between the C300 Controller and pulsed output transducers such as tachometers, flow meters, and magnetic pickups.
"Redundancy Module Block" on page 282	Defines name/location of Primary and Secondary Redundancy Modules in Redundant Chassis Pair. This module is redundancy compliant.
"Fault Tolerant Ethernet Bridge Module Block" on page 284	Provides link to Fault Tolerant Ethernet (FTE) supervisory network. This includes C200 Process Controller and Fieldbus Interface Module chassis.
"Serial Interface Module (SIM) I/O Module Block" on page 312	Provides configuration and communication software to enable devices to communicate via an ASCII serial protocol to perform bi-directional data exchange directly with the Control Processor.
"Input/Output Link Interface Module Block" on page 269	Provides links for Process Manager I/O channels to interface physical Input/Output Processors (IOPs) to given controller.
"Input/Output Link Block" on page 270	Supports interface communications in given IOLIM.
"IOLINK Block (C300 - 20mS CEE)" on page 275	Defines communication path to external OPC server.
"Inter Cluster Gateway Block" on page 279	Makes CDA data from one Experion cluster available to a second Experion cluster, allowing regulatory control cascades to span separate Experion clusters
"Process Manager Input/Output (PMIO) Blocks" on page 313	Provides links for I/O channels to interface physical I/O Processors to given I/O Link Interface Module.
"Series C Input/Output (I/O) Blocks " on page 332	Provides links for Series C I/O channels to interface Series C I/O modules with the C300.
"Speed Protection Module (SPM)" on page 342	Provides links to Honeywell's Safety Manager as well as Modbus TCP native devices and serial RTU devices through a Modbus TCP gateway/bridge.

2.3 Functional relation category

Our Control Builder application includes comprehensive libraries of function blocks that streamline the control strategy configuration process. You simply “drag and drop” selected blocks into a Control Module and/or Sequential Control Module container to emulate the necessary functional requirements of your process.

Please refer to *Control Building Guide* for details about how to configure a control strategy.

The functional relation category conveniently groups function blocks according to a related functional block type or component library. The component libraries provide a convenient way to group related function blocks for easy access and reference.

Figure 3 shows the general graphic orientation and Windows look-and-feel of the Control Builder application for reference.

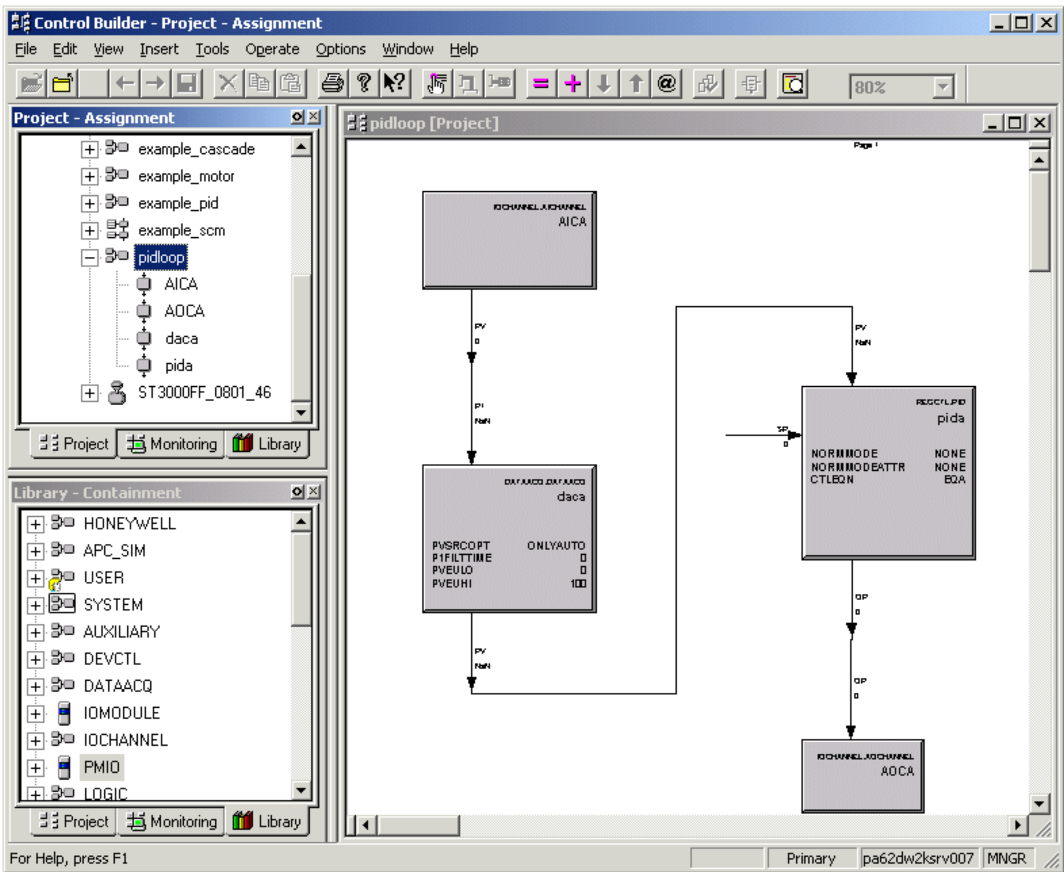

Figure 3: Typical view of control module configuration in Control Builder.

2.3.1 Functional block types

Table 2 identifies the functional block types used to represent a group of corresponding function blocks provided in Control Builder. These block types are used as a way to simplify information retrieval for a given function block, and do not necessarily correspond to an actual Control Builder function.

Table 2: Functional block types

Functional Block	
Type	**Description**
"Auxiliary Blocks" on page 34	Includes block types for performing auxiliary control functions, such as: calculation, general linearization and totalization.
"Data Acquisition Block" on page 57	Provides signal conditioning for a process input value from another function block.
"Device Control Block" on page 54	Provides a multi-input/multi-output function that provides an interface to discrete devices such as motors, pumps, solenoid valves, and motor-operated valves.
"IO Channel Blocks" on page 61	Includes channel block types (analog input, analog output, digital input, digital output, pulse width modulator) to represent I/O points that are device independent; each I/O channel type has a standard interface with control function blocks. This category also includes array channel blocks to support communications with the associated Serial Interface Module and the connected Field Terminal Assembly (FTA) device. You assign an array channel block to one of the SIM block's 32 channels as well as designating which of the two FTAs it is associated with. The array channel block types are flag, numeric, and text.
"HART DEVICE Block" on page 71	Provides a standard interface to the HART AI module TC-HAI081and HART AO module TC-HAO081.
"Pulse Input Channel/Module Blocks" on page 73	Provides a standard interface to the Pulse Input Module TC-MDP081/TK-MDP081.
"Exchange Blocks (ControlNet Interoperability)" on page 67	Includes block types for performing ControlNet Interoperability functions, such as: Flag, Numeric and Text storage.
"Logic Blocks" on page 79	Provides a set of Boolean, selection and comparison functions to be used as a basis for integrated logic control.
"Math Blocks" on page 101	Provides a set of math functions to be used for basic calculations.
"Power Generation Blocks" on page 108	Includes block types for building control strategies that support power generation applications.
"Regulatory Control Blocks" on page 123	Includes block types for building internal control loops.
"Sequential Control Module Blocks" on page 203	Includes block types for building a sequential control function.
"System Blocks" on page 208	Control Module (CM) is a Control Builder "container" that uses predefined continuous (discrete) control function blocks to define a given process control strategy. Sequential Control Module (SCM) is a Control Builder "container" that uses predefined sequential control function blocks to define the sequential operation for a given process control strategy.
"Universal Control Network Interface (UCNIF) Block" on page 215	Provides a configurable function for creating regulatory control cascade strategies between the Application Control Environment (ACE) supervisory controller and Process Manager controllers residing on a Universal Control Network in a connected TPS system.
"Hiway Interface (HIWAYIF) Blocks" on page 217	Includes block types for creating regulatory control cascade strategies between the Application Control Environment (ACE) supervisory controller and Data Hiway controllers residing on a Data Hiway in a connected TPS system.
"Utility Blocks" on page 222	Includes block types for performing utility control functions, such as: status flag, numeric storage and timer capabilities. Blocks have been added for message, data array, and parameter type convert support.

2.4 Component Libraries

Related topics

"Auxiliary (AUXILIARY) Library" on page 20
"Device Control (DEVCTL) Library" on page 21
"Data Acquisition (DATAACQ) Library" on page 21
"Input/Output Channel (IOCHANNEL) Library" on page 21
"Exchange Library (ControlNet Interoperability)" on page 21
"Pulse Input Channel/Module Library" on page 21
"Input/Output Module (IOMODULE) library" on page 21
"Logic (LOGIC) Library" on page 23
"Math (MATH) Library" on page 24
"Peer Control Data Interface (PCDI)" on page 24
"Power Generation (POWERGEN) library" on page 25
"Process Manager Input/Output (PMIO)" on page 25
"Series C Input/Output Library" on page 26
"Regulatory Control (REGCTL) library" on page 29
"Sequential Control Module (SCM) library" on page 30
"System (SYSTEM) Library" on page 30
"Universal Control Network Interface (UCNIF) Library" on page 30
"Hiway Interface (HIWAYIF) Library" on page 30
"Utility (UTILITY) Library" on page 30
"Rail I/O Modules -Series H (RAIL_IO_HAZ) Library" on page 31
"HART I/O Modules (HARTIO) Library" on page 31
"DeviceNet Interface (DNETIF) Library" on page 31
"PROFIBUS Interface (PBUSIF) Library" on page 31
"Rail I/O Modules - Series A (RAIL_IO) Library" on page 31

2.4.1 Auxiliary (AUXILIARY) Library

The Auxiliary Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled "Reference Data for Functional Block Types" on page 33.

• "AUXCALC (Auxiliary Calculation) block" on page 34	• "Reference Data for Functional Block Types" on page 33	• "CTUD (Counter Up/Down) Block" on page 38
• "CTUD (Counter Up/Down) Block" on page 38	• "ENHAUXCALC (Enhanced Auxiliary Calculation) block" on page 41	• "ENHGENLIN (Enhanced General Linearization) block" on page 43
• "ENHGENLIN (Enhanced General Linearization) block" on page 43	• "GENLIN (General Linearization) block" on page 45	• "LEADLAG (Lead Lag) block" on page 46
• "ROC (Rate of Change) block" on page 47	• "SIGNALSEL (Signal Selector) block" on page 47	• "TOTALIZER block" on page 52

2.4.2 Device Control (DEVCTL) Library

The Device Control Library includes the “DEVCTL (Device Control) block” on page 54. A detailed description is given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

2.4.3 Data Acquisition (DATAACQ) Library

The Data Acquisition Library includes the “DATAACQ (Data Acquisition) block” on page 57. A detailed description is given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

2.4.4 Input/Output Channel (IOCHANNEL) Library

The Input/Output Channel Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “AICHANNEL” on page 61	• “DOCHANNEL” on page 62
• “AOCHANNEL” on page 61	• “PWMCHANNEL” on page 63
• “DICHANNEL” on page 62	• “SIFLAGARRCH” on page 64
• “SINUMARRCH” on page 64	• “SITEXTARRCH” on page 65

2.4.5 Exchange Library (ControlNet Interoperability)

The Exchange Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “REQFLAGARRAY (Request Flag Array) block” on page 67	• “RSPFLAGARRAY (Response Flag Array) block” on page 69
• “REQNUMARRAY (Request Number Array) block” on page 67	• “RSPNUMARRAY (Response Number Array) block” on page 69
• “REQTEXTARRAY (Request Text Array) block” on page 68	• “RSPTEXTARRAY (Response Text Array) block” on page 70

2.4.6 Pulse Input Channel/Module Library

• “Pulse Input Channel with Fast Cutoff” on page 73
• “Pulse Input Channel” on page 74
• “Pulse Input Module Block TC-MDP081/TK-MDP081” on page 75
• “Pulse Input Totalizer” on page 74

2.4.7 Input/Output Module (IOMODULE) library

The Input/Output Module Library includes the Input/Output Module (IOM) function blocks listed below. Blocks are identified by the given Honeywell model number. The models with a TC prefix are not conformally coated and those with a TK prefix are conformally coated. Detailed descriptions are presented in the following section titled “Reference Data for Physical Equipment Block Types” on page 247.

IOM Function Blocks	Number of Channels	Type	Rating	Isolated
“TC-HAI081/TK-HAI081 (8 channel HART - 10V / 4 to 20mA - Analog Input)” on page 287	8	HART / Non-HART Analog Input	10 V and 4 to 20 mA	Yes
“TC-HAO081/TK-HAO081 (8 channel HART - 10V / 4 to 20mA - Analog Output)” on page 300	8	HART / Non-HART Analog Output	10 V and 4 to 20 mA	No
“TC-IDX081/TK-IDX081 (8 Channel - 120Vac Diagnostic Input)” on page 294	8	Diagnostic Input	120 Vac	Yes
“TC-IDA161/TK-IDA161 (16 Channel - 120Vac Non-Isolated - Digital Input)” on page 289	16	Digital Input	120 Vac	No
“TC-IDK161/TK-IDK161 (16 Channel - 120Vac Isolated - Digital Input)” on page 292	16	Digital Input	120 Vac	Yes
“TC-IDD321/TK-IDD321 (32 Channel - 24Vdc Non-Isolated - Digital Input)” on page 290	32	Digital Input	24 Vdc	No
“TC-IDX161/TK-IDX161 (16 Channel - 24Vdc Diagnostic Input)” on page 295	16	Diagnostic Input	24 Vdc	Yes
“TC-IDJ161/TK-IDJ161 (16 Channel - 24Vdc Isolated - Digital Input)” on page 291	16	Digital Input	24 Vdc	Yes
“TC-IAH061/TK-IAH061 (6 Channel - 10V / 4 to 20mA Isolated - Analog Input)” on page 287	6	Analog Input	10 V and 4 to 20 mA	Yes
“TC-IAH161/TK-IAH161 (16 Channel - 10V / 4 to 20mA Non-Isolated - Analog Input)” on page 288	16	Analog Input	10 V and 4 to 20 mA	No
“TC-IDW161/TK-IDW161 (16 Channel - 220Vac Isolated - Digital Input)” on page 293	16	Digital Input	220 Vac	Yes
“TC-IXR061/TK-IXR061 (6 Channel - RTD Input)” on page 298	6	Resistance Temperature Detector (RTD) Input	Resistance	Yes
“TC-IXL061/TK-IXL061 (6 Channel - Thermocouple Input)” on page 296	6	Thermocouple Input	Low level mV	No
“TC-IXL062/TK-IXL062 (6 Channel - Thermocouple Input)” on page 297	6	Thermocouple Input	Low level mV	No
“TC-ODX081/TK-ODX081 (8 Channel - 120Vac- Diagnostic Output)” on page 307	8	Diagnostic Output	120 Vac	Yes

IOM Function Blocks	Number of Channels	Type	Rating	Isolated
"TC-ODA161/TK-ODA161 (16 Channel - 120/220Vac Non-Isolated - Digital Output)" on page 303	16	Digital Output	120/220 Vac	No
"TC-ODK161/TK-ODK161 (16 Channel - 120/220Vac Isolated - Digital Output)" on page 306	16	Digital Output	120/220 Vac	Yes
"TC-ODX161/TK-ODX161 (16 Channel - 24Vdc- Diagnostic Output)" on page 308	16	Diagnostic Output	24 Vdc	Yes
"TC-ODD321/TK-ODD321 (32 Channel - 24Vdc Non-Isolated Digital Output)" on page 304	32	Digital Output	24 Vdc	No
"TC-ODJ161/TK-ODJ161 (16 Channel - 24Vdc Isolated Digital Output)" on page 305	16	Digital Output	24 Vdc	Yes
"TC-ORC081/TK-ORC081 (8 Channel - 8 n.c., 8 n.o. 5-150Vdc, 10-265Vac Isolated - Relay Output)" on page 309	8	Relay Output	24 Vdc and 120/220 Vac	Yes
"TC-ORC161/TK-ORC161 (16 Channel, 5-150Vdc, 10-265Vac Isolated - Contact Output)" on page 310	16	Contact Output	24 Vdc and 120/220 Vac	Yes
"TC-OAH061/TK-OAH061 (6 Channel - 4 to 20mA - Analog Output)" on page 301	6	Analog Output	4 to 20 mA	Yes
"TC-OAV061/TK-OAV061 (6 Channel - 10V - Analog Output)" on page 301	6	Analog Output	10 V	Yes
"TC-OAV081/TK-OAV081 (8 Channel - 10V / 4 to 20mA Non-Isolated - Analog Output)" on page 302	8	Analog Output	10 V and 4 to 20 mA	No
"TC-MUX021/TK-MUX021 (Up to 32 Array Channel Function Blocks)" on page 312	Up to 32 FTA Array Points	Bi-directional data exchange with devices using ASCII serial protocol communications	Modbus FTA or A-B FTA	

2.4.8 Logic (LOGIC) Library

The Logic Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled "Reference Data for Functional Block Types" on page 33.

• "AND block" on page 80	• "MIN block" on page 87	• "ROL (Rotate Output Left) block" on page 94
• "CHECKBAD block" on page 80	• "MINPULSE block" on page 88	• "ROR (Rotate Output Right) block" on page 95
• "CHECKBOOL block" on page 81	• "MUX (Multiplexer) block" on page 88	• "RS (Reset dominant SR-FLIP-FLOP) block" on page 95

2.4.9 Math (MATH) Library

The Math Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

2.4.10 Peer Control Data Interface (PCDI)

The Peer Control Data Interface Library includes the PCDI device and Array Request Channel function blocks listed below. Detailed descriptions are provided in the following section titled “Reference Data for Physical Equipment Block Types” on page 247.

2.4.11 Power Generation (POWERGEN) library

The Power Generation library includes the function blocks listed in the following table. Detailed descriptions are given in the subsequent functional entity block type headings in the section titled “Reference Data for Functional Block Types” on page 33.

• “GRPCAPRBK (Group Capability and Runback) block” on page 108	• “MAINIBV (Main IBV Logic) block” on page 116
• “HTMOTOR (HT Motor Drive Control) block” on page 109	• “SOLENOID (Solenoid Valve Drive Control) block” on page 117
• “LEVELCOMP (Drum Level Computation) block” on page 113	• “VALVEDAMPER (Valve/Damper Drive Control) block” on page 120
• “LTMOTOR (LT Motor Drive Control) block” on page 114	• “STEAMPROP (Steam Property) block” on page 213

2.4.12 Process Manager Input/Output (PMIO)

The Process Manager Input/Output Module (PMIO) Library includes the Input/Output Processor (IOP) function blocks listed below. Blocks are identified by block name and by Honeywell model number. The models with a MU prefix are not conformally coated and those with a MC prefix are conformally coated. Detailed descriptions are presented in the following section titled “Reference Data for Physical Equipment Block Types” on page 247.

IOP and Related Channel Function Blocks	Number of Channels	Type	Rating	CE Compliant
“AICHANNEL (PMIO) block” on page 313	n/a	Represents an AI point.	n/a	n/a
“HAICHANNEL (PMIO) block” on page 314	n/a	Represents a HART AI or non-HART AI point.	n/a	n/a
“AOCHANNEL (PMIO) block” on page 316	n/a	Represents an AO point.	n/a	n/a
“HAOCHANNEL (PMIO) block” on page 317	n/a	Represents a HART AO or non-HART AO point.	n/a	n/a
“DICHANNEL (PMIO) block” on page 319	n/a	Represents a DI point.	n/a	n/a
“DOCHANNEL (PMIO) block” on page 320	n/a	Represents a DO point.	n/a	n/a
“HLAI block” on page 320, MU-PAIH01, MU/MC-PAIH02, or MU/MC-PAIH03	16	High Level Analog Input	0 to 5 V 1 to 5 V 0.4 to 2 V 4 to 20 mA	Yes
“HLAIHART block” on page 322, MC-PHAI01	16	HART Analog Input	0 to 5 V 1 to 5 V 0.4 to 2 V 4 to 20 mA	Yes
LLAI, MU-PAIL01, MU/MC-PAIL02	8	Low Level Analog Input	T/C, RTD, or Voltage	Yes

IOP and Related Channel Function Blocks	Number of Channels	Type	Rating	CE Compliant
"LLMUX block" on page 323, MU/MC-PLAM02	32	Low Level Analog Multiplexer	T/C, RTD, or linear mV	Yes
RHMUX, MU/MCPRHM01	32	Remote Hardened Multiplexer	T/C, Linear mV	Yes
"STI_MV block" on page 324, MU/MC-PSTX03	16	Smart Transmitter Interface Multivariable	Honeywell DE protocol	Yes
"AO16 block" on page 325, MU/MC-PAOY22	16	Analog Output	4 to 20 mA	Yes
"AO16HART block" on page 327 MC-PHAO01	16	HART Analog Output	4 to 20 mA	Yes
AO8, MU-PAOX01, MU-PAOX02, or MU/MC-PAOX03	8	Analog Output	4 to 20 mA	Yes
"DI24V block" on page 328, MU-PDIX01, MU/MC-PDIX02, or MU/MC-PDIY22	32	Digital Input	24 Vdc	Yes
"DISOE block" on page 329, MU/MC-PDIS12	32	Digital Input Sequence of Events	24 Vdc, 120 Vac, or 240 Vac	Yes
"DI block" on page 330, MU-PDIX01, or MU/MCPDIX02	32	Digital Input	120 Vac or 240 Vac	Yes
"DO32 block" on page 331 MU/MC-PDOY22	32	Digital Output	24 Vdc isolated, or 240 Vac/125 Vdc relay	Yes
DO16, MU-PDOX01 or MU/MC-PDOX02	16	Digital Output	24 Vdc, 3-30 Vdc, 31-200 Vdc, 120/240 Vac, 120 Vac/125 Vdc relay, or 240 Vac/125 Vdc relay	Yes

2.4.13 Series C Input/Output Library

Attention

The CEE C300 - 20mS CEE controller does not support PMIO modules.

The Series C Input/Output Module (Series C I/O) Library includes the Series C Input/Output Module (IOM) function blocks, Speed Protection Module (SPM), and Servo Valve Positioner Module (SVPM) listed below. Blocks are identified by block name and by Honeywell model number. The models with a CU prefix are not conformally coated and those with a CC prefix are conformally coated. Detailed descriptions are presented in the following section titled "Reference Data for Physical Equipment Block Types" on page 247.

IOM and Related Channel Function Blocks	Number of Channels	Type
AICHANNEL (Series C)	n/a	The AI channel block represents a single analog input point on one of the following Series C Processors: "AI-HART" on page 336; "AI-LLMUX" on page 337,"AI-LLAI" on page 337

IOM and Related Channel Function Blocks	Number of Channels	Type
AOCHANNEL (Series C)	n/a	The AO channel block represents a single analog input point on the Series C “AO-HART” on page 339 Processor
DICHANNEL (Series C)	n/a	The DI channel block represents a single discrete input point on a Series C “DI-HV” on page 339, or “DI-24” on page 340 Processor.
DOCHANNEL (Series C)	n/a	The DO channel block represents a single discrete input point on a Series C“DO-24B” on page 340 I/O Processor
PI (Series C)	n/a	The PI channel block represents a single pulse input point on a Series C Pulse Input Module.
“AI-HART” on page 336 CU-PAIH01 CC-PAIH01	16	High Level Analog Input with HART
“AI-HART” on page 336 8U-PAIH51 8C-PAIH51	16	High Level Analog Input with HART
AI-HL 8U-PAIN01 8C-PAIN01	16	High Level Analog Input
“AI-LLMUX” on page 337 CU-PAIM01 CC-PAIM01	64	Low Level Mux Input
“AI-LLMUX” on page 337 8U-PAIM01 8C-PAIM01	64	Low Level Analog Input Mux
“AI-LLAI” on page 337 CC-PAIM51	16	Low Level Analog Input
“AO-HART” on page 339 CU-PAOH01 CC-PAOH01	16	Analog Output with HART IOP
AO 8U-PAON01 8C-PAON01	16	Analog Output
AO-HART 8U-PAOH51 8C-PAOH51	16	Analog Output, HART Capable

IOM and Related Channel Function Blocks	Number of Channels	Type
DI-24 8U-PDIL51 8C-PDIL51	32	Low Voltage (24DC) Digital Input
DI-SOE 8U-PDIS01 8C-PDIL51	32	Digital Input – Sequence of Events
DO-24B 8U-PDOD51 8C-PDOD51	32	Bussed Low Voltage Digital Output
PA 8U- PDIP51 8C- PDIP51	32	Low Voltage Pulse Accumulation Module (24volts DC)
"DI-HV" on page 339 CU-PDIH01 CC-PDIH01	32	High Voltage Digital Input (IOM supports both 120 and 240 volts AC)
"DI-24" on page 340 CU-PDIL01 CC-PDIL01	32	Low Voltage Digital Input (24 volts DC)
"DO-24B" on page 340 CU-PDOB01 CC-PDOB01	32	Bussed Low Voltage Digital Output (24 volts DC)
SP_AI (Series C)	n/a	The AI channel block represents a single analog input point on Speed Protection (SP) Module.
SP_AO (Series C)	n/a	The AO channel block represents a single analog output point on the Speed Protection (SP) Module.
SP_DI (Series C)	n/a	The DI channel block represents a single discrete input point on the Speed Protection (SP) Module.
SP_DO (Series C)	n/a	The DO channel block represents a single discrete output point on the Speed Protection (SP) Module.
SP_SPEED (Series C)	n/a	The SP_SPEED channel represents a pulse input on the Speed Protection (SP) Module.
SP_SPDVOTE (Series C)	n/a	The SP_SPDVOTE channel performs voting logic functionality on the speed inputs.
SVP_AI (Series C)	n/a	The AI channel represents an LVDT/RVDT input or a single analog input on the Servo Valve Positioner (SVP) Module.
SVP_AO (Series C)	n/a	The AO channel supports unipolar and bipolar current output besides the standard 4-20 mA analog output supported by Series C AO-IOM.
SVP_DI (Series C)	n/a	The DI channel block represents a single discrete input point on the Servo Valve Positioner (SVP) Module.

IOM and Related Channel Function Blocks	Number of Channels	Type
SVP_REGCTL (Series C)	n/a	The SVP_REGCTL channel block performs the regulatory control function for valve positioning.
SP IOM CC-PSP401	26	Speed Protection Module (SPM) supports the following channels. • 8 AI channels (SP_AI) • 1 AO channel (SP_AO) • 8 DI channels (SP_DI) • 4 DO channel (SP_DO) • 4 Speed channel (SP_SPEED) • 1 voting logic channel (SP_SPDVOTE)
SVP IOM CC-PSV201	8	Servo Valve Positioner Module (SVPM) supports the following channels. • 2 LVDT / RVDT / Analog input channels (SVP_AI) • 2 Digital input channels (SVP_DI) • 2 Servo / Analog output channels (SVP_AO) • 2 Regulatory control channels (SVP_REGCTL)
UIO IOM CC-PUIO01	32	In the Universal Input/Output Module, each channel can be configured as one of the following channels. • Analog Input channel • Analog Output channel • Digital Input channel • Digital Output channel

2.4.14 Regulatory Control (REGCTL) library

The Regulatory Control Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “AUTOMAN (Auto Manual) block ” on page 123	• “ENHREGCALC (Enhanced Regulatory Control Calculator) block” on page 127	• “FANOUT block” on page 133
• “PIDER block” on page 145	• “OVRDSEL(Override Selector) block” on page 136	• “PID block” on page 140
• “POSPROP (Position Proportional) block” on page 161	• “PID-PL block” on page 150	• “PIDFF (PID Feedforward) block” on page 155
• “RAMPSOAK block” on page 169	• “PULSECOUNT block” on page 166	• “PULSELENGTH block” on page 167
• “REEOUT (Remote EEOUT) block” on page 182	• “RATIOBIAS block” on page 173	• “RATIOCTL block” on page 178
• “REMCAS block” on page 194	• “REGCALC (Regulatory Control Calculator) block” on page 183	• “REGSUMMER (Regulatory Control Summer)” on page 189
• “SWITCH block” on page 198		

2.4.15 Sequential Control Module (SCM) library

The Sequential Control Module Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “HANDLER Block” on page 203	• “STEP Block” on page 204	• “TRANSITION Block” on page 206

2.4.16 System (SYSTEM) Library

The System Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “CONTROL MODULE block (Continuous Control)” on page 208	• “SEQUENTIAL CONTROL MODULE block (Sequential Control)” on page 209

2.4.17 Universal Control Network Interface (UCNIF) Library

The Universal Control Network Interface Library includes the “UCNOUT block” on page 215 and the “EUCNOUT block” on page 216. A detailed description is given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

2.4.18 Hiway Interface (HIWAYIF) Library

The Hiway Interface Library includes the function block listed below. A detailed description is given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “HIWAYOUT block” on page 217		

2.4.19 Utility (UTILITY) Library

The Utility Library includes the function blocks listed below. Detailed descriptions are given in the subsequent functional entity block type headings in the following section titled “Reference Data for Functional Block Types” on page 33.

• “ALMWINDOW (Alarm Window - Alarm Annunciator) block” on page 222	• “ANNPANEL (Annunciator Panel - Alarm Annunciator) block” on page 223	• “DIGACQ (Digital Acquisition) block” on page 224
• “FIRSTOUT (First Out Detection) block” on page 226	• “FLAG block” on page 227	• “FLAGARRAY block” on page 228
• “MESSAGE block” on page 228	• “NUMERIC block” on page 230	• “NUMERICARRAY block” on page 230
• “PUSH block” on page 230	• “TEXTARRAY block” on page 231	• “TEXTCOMMENT (Text Comment) block” on page 232
• “TIMER block” on page 232	• “TYPECONVERT block” on page 233	

2.4.20 Rail I/O Modules -Series H (RAIL_IO_HAZ) Library

The Series H I/O Library includes the IOM blocks associated with the RIOM-H components designed for use in locations with potentially explosive atmospheres. Refer to the *Series H Rail I/O Implementation Guide* for complete details about the Series H I/O Modules.

2.4.21 HART I/O Modules (HARTIO) Library

The HART I/O Library includes the interface blocks for linking HART compatible devices with the Experion system through Chassis I/O. Refer to the *HART I/O Implementation Guide* for complete details about the Chassis HART I/O components.

2.4.22 DeviceNet Interface (DNETIF) Library

The DeviceNet Interface Library includes the interface blocks associated with linking DeviceNet devices with Experion system through the interface modules. Refer to the *DeviceNet Interface Implementation Guide* for complete details about the DeviceNet Interface components.

2.4.23 PROFIBUS Interface (PBUSIF) Library

The PROFIBUS Interface Library includes the interface blocks associated with linking PROFIBUS devices with the *Experion* system through the interface modules. Refer to the *PROFIBUS Interface Implementation Guide* for complete details about the PROFIBUS Interface components.

2.4.24 Rail I/O Modules - Series A (RAIL_IO) Library

The Series A I/O Module library includes the IOM blocks associated with the RIOM-A components designed for use in general purpose locations. Please refer to the *Series A Rail I/O Implementation Guide* for complete details about the Series A I/O Modules.

3 Reference Data for Functional Block Types

This section provides detailed reference data for each functional block type that is part of the functional relation category for the Control Builder. It presents the block types associated with a given component library. The reference data is organized alphabetically by component library/ block type, and then alphabetically within each type by the function block name.

The reference data for each block covers these topics, as applicable:

- description
- function
- inputs and input ranges
- outputs and output ranges
- parameters

Note that the data varies, based on what is pertinent for each block.

Related topics

"Auxiliary Blocks" on page 34
"Device Control Block" on page 54
"Data Acquisition Block" on page 57
"IO Channel Blocks" on page 61
"Exchange Blocks (ControlNet Interoperability)" on page 67
"HART DEVICE Block" on page 71
"Pulse Input Channel/Module Blocks" on page 73
"Logic Blocks" on page 79
"Math Blocks" on page 101
"Power Generation Blocks" on page 108
"Regulatory Control Blocks" on page 123
"Sequential Control Module Blocks" on page 203
"System Blocks" on page 208
"Thermodynamic Utility Function Block" on page 213
"Universal Control Network Interface (UCNIF) Block" on page 215
"Hiway Interface (HIWAYIF) Blocks" on page 217
"Hiway Responder Block (HRB)" on page 218
"Utility Blocks" on page 222
"EtherNet/IP channel blocks" on page 235
"IOREFERENCES Blocks" on page 242

3.1 Auxiliary Blocks

Related topics

"AUXCALC (Auxiliary Calculation) block" on page 34
"AUXSUMMER (Auxiliary Summer) block" on page 37
"CTUD (Counter Up/Down) Block" on page 38
"DEADTIME block" on page 40
"ENHAUXCALC (Enhanced Auxiliary Calculation) block" on page 41
"ENHGENLIN (Enhanced General Linearization) block" on page 43
"FLOWCOMP (Flow Compensation) block" on page 44
"GENLIN (General Linearization) block" on page 45
"LEADLAG (Lead Lag) block" on page 46
"ROC (Rate of Change) block" on page 47
"SIGNALSEL (Signal Selector) block" on page 47
"Selection methods" on page 51
"TOTALIZER block" on page 52

3.1.1 AUXCALC (Auxiliary Calculation) block

Description	Lets you write up to eight expressions for computing a Process Variable (PV) value.
Function	Each expression can contain any valid combination of inputs, operators and functions and may perform arithmetic or logic operations, test conditions, etc. Status information is made available for input as well as the expression results. You can assign the result of an expression, a status, or an input to PV and PVSTS parameters which are then processed like the result of any other Auxiliary function block.
Inputs	Accepts up to six optional inputs (P[1] to P[6]) - none are required. • No inputs are required • All inputs must be fetched from other function blocks. • The number of process input connections are equal to the number of inputs; the default is 1. • Configure P inputs contiguously (without breaks) in arrays.
Outputs	Produces these outputs according to the values you assign to them. • PV and its status PVSTS, as well as a Boolean flag, PVSTSFL.BAD, to indicate to other function blocks, that this block's PV status is bad. • Up to eight expression results (C[1] to C[8])
Operators and Functions	Table 3 lists the expression operators and functions supported by this block for reference.
Parameter Identification	You must specify a parameter by its full tag name. For example, "CM25.PumpASelect.PVFL", or "CM57.PID100.MODE". In effect, tag names allow expressions to have an unlimited number of inputs and work with any data type. However, do not use more than six parameter references in an expression. The expression syntax has been expanded. Delimiters (') can be used in an expression containing an external reference component. The format for the delimiter usage is as follows: • TagName.'text'

Expression Rules	• Must include full tag.parameter name for P inputs in the expression and enclose identification number in brackets instead of parenthesizes. For example, CM151.AUXCALC BLOCK.P[1] * CM151.AUXCALC BLOCK.P[2] is valid. • Expressions cannot contain an assignment operation (a colon and equal sign with the current syntax) For example, “CM1.PID1.MODE:=X[1]” is invalid. • Each expression produces a single value (arithmetic or logical which is automatically stored in a “C” parameter. For example, if you write four expressions, the result of the first expression is stored in C[1], the result of the second is stored in C[2], etc. You can use these results, by name, in succeeding expressions. In this example, you could use C[1] as an input to expressions 2, 3, and 4. • You can mix and nest all operators and functions (including conditional assignments) in any order as long as types match or can be converted. • You can use blanks between operators and parameter names, but they are not required. • You can use all data types in expressions, including enumerations. They are all treated as numeric types. • You must configure calculator expressions contiguously (without breaks) in the arrays. • A short description can be provided for the expressions using the expression descriptor parameter (EXPRDESC[1..8]). The results of the expressions, which use the CONST[1…8] parameters, are affected if you change the values of these parameters on the Constants tab. • With R410 , non-CEE controllers such as PMD and Safety Manager and Experion server points such as TPS and SCADA, can be configured in the Expressions. • With R410, when you write the expressions using the TPS point's parameter references, ensure that the TPS reference parameter is configured using the parentheses “()”to specify array index. However, when you write the expressions using the other non-CEE points, you can use the brackets “[].”	
Parameters	C[1..8] CONFIGCODE[1..8] CONFIGDESC[1..8] CONFIGSTS[1..8] CONST[1…8] CONSTACCLOCK CONSTENABLE CSTS[1..8] DESC EUDESC EXECCODE[1..8] EXECDESC EXECDESC[1..8] EXECSTS EXECSTS[1..8] EXPR[1..8] EXPRDESC[1…8] EXPRPCODE[1..8] HIALM	NAME ORDERINCM P[1..6] PSTS[1..6] PV PVFORMAT PVSRC PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN PVSTSSRC PVVALSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the AUXCALC Block.	

Attention

Do not use equality operands = and <> to compare FLOAT64 and FLOAT32 floating point values in expressions. Use inequality operands Less Than (<), Less Than or Equal To (<=), Greater Than (>), or Greater Than or Equal To (>=) instead.

Table 3: Expression Operators and Functions Reference

Operators		Description	
Unary		+ -	
Binary Arithmetic		+ - * / MOD (x MOD y) ^ (x^y)	
Logical		AND OR NOT	
Relational		= <> <= >= < >	
Conditional		? : (For example, X ?Y : Z; similar to IF, THEN, ELSE)	
Parenthesis		()	
Array Syntax		[ ]	
Unary Functions			
ABS	absolute value	LOG	Base 10 logarithm of a number
ATN	arc tangent	RND	round value
COS[1]	cosine	SGN	sign of value (returns -1,0 or +1)
EXP	e to the power of x	SIN1	sine
INT	convert to integer	SQR	square of a number
ISFIN	is finite	SQRT	square root
ISNAN	is Not a Number	TAN[1]	tangent
LN	Natural logarithm of a number (log to the base of e)		
Multiple Argument Functions			
MIN	minimum of n arguments (ignore bad values). If this function has a NAN argument (bad value), it returns NaN.	MID	medium value of n arguments (average of middle values for even n). If this function has a NAN argument (bad value), it returns NaN.
MAX	maximum of n arguments (ignore bad values). If this function has a NAN argument (bad value), it returns NaN.	MUL	product of n arguments This function ignores NaN values. However, if all agruments are NaN, then it returns 1.
AVG	average of n arguments. This function ignores NaN values. However, if all arguments are NaN, then it returns NaN.	SUM	sum of n arguments. This function ignores NaN values. However, if all agruments are NaN, then it returns 0.
String Support Functions			
LEN	Returns an integer length of the string	NUMSTR	Takes the input parameter, casts it to a Float64 and converts it to a string
MIDS	Takes a string, an integer starting position and an integer length. The function returns the specified portion of the original string.	STRNUM	Takes the string input parameter and converts it to a Float64

[1] Be sure you specify the trigonometric functions cosine, sine, and tangent in radians and not degrees.

Operators		Description	
Time Support Functions			
ABSTOD	Takes an absolute time data type and strips off the year and date and returns a 64-bit float representing the time of day in milliseconds.	DTIMNUM	Takes a delta TIME data type and returns a 64-bit float representing the number of milliseconds.
NOW	Returns the current local date and time as an absolute time data type	NUMDTIM	Takes a 64-bit float representing some number of milliseconds and converts it to a delta TIME data type.
NUMTIM	Takes a 64-bit float representing the number of milliseconds since Jan 1, 1972 and converts it to absolute TIME data type.	STRTIM	Takes a string input parameter and converts it to an Absolute time. The string must be in the same format as an Absolute time constant.
TOD	Returns the current local time of day as Time of Day data type	TIMNUM	Takes an Absolute TIME data type and returns a 64-bit float representing the total number of milliseconds since Jan 1, 1972.
UTCTOD	Returns the current UTC time of day as Time of Day data type	UTCNOW	Returns the current UTC date and time of day as an absolute time data type

Case Sensitive Strings for Special Value Constants	
NAN	IEEE NaN value
+INF	IEEE + Infinity value
-INF	IEEE - Infinity value
PI	PI (3.14159. . .)
E	e (2.718. . .)

3.1.2 AUXSUMMER (Auxiliary Summer) block

Description	Lets you configure up to ten separate inputs to calculate a process variable (PV) value that can be scaled and biased.
Function	The AUXSUMMER block fetches values from other function blocks and determines their statuses in every execution cycle of the Control Module. It evaluates up to ten inputs and determines their statuses. It derives values for PV and PV status based on its calculation of the inputs and the configuration entries for the overall PV scale factor (CPV) and overall PV bias factor (DPV) parameters.
Inputs	This function block accepts as many as ten inputs (P[1…10]). • At least one input (P[i]) must be configured for the block to operate. • All inputs must be fetched from other function blocks • The number of process input connections (NUMPINPT) that can be made to other blocks is equal to the number of inputs. The default is 1.
Outputs	This block produces the following outputs: • PV and its status, PVSTS

Parameters	C[1..10] CPV D[1..10] DESC DPV EUDESC NUMPINPT NAME ORDERINCM P[1..10]	PDESC PENABLE PSTS[1..10] PSUB PV PVFORMAT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the AUXSUMMER Block.	

3.1.3 CTUD (Counter Up/Down) Block

Description	A new general purpose Up-Down Counter (CTUD) is introduced in the Auxiliary library to simplify event count strategies. The counter function block starts functioning based on the configured algorithm. The count inputs may be wired to other function blocks or stored by a program.
Function	The CTUD block is an up-down counter function block. The counter of the CTUD block can change its state (Up or Down) depending on the configuration of Count Up Flag (CNTUPFL) and Count Down Flag (CNTDNFL) parameter. The counting also depends on a valid IN (ININT32/INFLOAT64) configuration if the input is fed through wired connection. Up-down counting is evaluated as edge trigger quantity or level trigger quantity depending on the value configured for Count On Level (CNTLVLFL) Parameter. The CTUD Block supports pause (PAUSEFL), load (LOADFL) and reset (RESETFL) operation for the counter. If the Count Up and Count Down flags are set to “TRUE”, the same block execution results in a net internal counter change of zero.
Inputs	• The CTUD block accepts a combination of Integer 32, Boolean, and Float 64 inputs. • Either of ININT32 or INFLOAT64 can be used during block execution. Selection of which IN parameter to use is determined by Input Specifier (SELINT32FL) parameter. – If SELINT32FL is set to “TRUE”, value of ININT32 parameter is used. – If SELINT32FL is set to “FALSE”, value of INFLOAT64 parameter is used. • All inputs are processed synchronously with the block execution.

<table>
<tr><td>Outputs</td><td colspan="2">
• The current Counter output value is available in Float 64 (OUTFLOAT64) and Integer 32 (OUTINT32) formats.
• CARRYUPFL is set to TRUE for one block execution following a counter overflow.
• CARRYDNFL is set to TRUE for one block execution following a counter underflow.
• QUFL indicates count Up reached.
• QDFL indicates count Down reached
• The output values are fetched through a wired connection or read directly by a program.
• An output connection to an input that can be initialized does not create a back calculation or function block connections.

The flag outputs (QUFL, QDFL, CARRYUPFL, and CARRYDNFL) are transitory. Downstream blocks, which sample these outputs, should sample at a rate at least twice the execution rate of the counter block in order to recognize all transitions.
</td></tr>
<tr><td>Equations</td><td colspan="2">
You can configure CNTEQN to specify how the block must handle the overflow and underflow conditions.
• By default, Equation A is selected.
• Ideal configuration of the Equation D and H to start the counting is:
  – For equation D - the input value must be greater than zero.
  – For equation H - the input value must be less than zero.
    If not, then the counter excludes a gap of zero to the input value in both equations.
</td></tr>
<tr><td>Platforms supported</td><td colspan="2">
Counter block can be used with the following Control Execution Environments (CEE).
• C300
• C200E
• ACE
• ACE-T
• SIM-C300
• SIM-C200E
• SIM-ACE
</td></tr>
<tr><td rowspan="16">Parameters</td><td>BLOCKTYPENAME</td><td>EUDESC</td></tr>
<tr><td>BLOCKTYPNAME</td><td>GLOBSCMMON</td></tr>
<tr><td>BLCKCOMMENT1</td><td>INCLAMPOPT</td></tr>
<tr><td>BLCKCOMMENT2</td><td>INFLOAT64</td></tr>
<tr><td>BLCKCOMMENT3</td><td>ININT32</td></tr>
<tr><td>BLCKCOMMENT4</td><td>OUTFLOAT64</td></tr>
<tr><td>CARRYDNFL</td><td>OUTINT32</td></tr>
<tr><td>CARRYUPFL</td><td>LOADFL</td></tr>
<tr><td>CBBLOCKPROP</td><td>NUMBLOBS</td></tr>
<tr><td>CBBLOCKPROPI</td><td>ORDERINCM</td></tr>
<tr><td>CNTDNFL</td><td>PAUSEFL</td></tr>
<tr><td>CNTEQN</td><td>QDFL</td></tr>
<tr><td>CNTLVLFL</td><td>QUFL</td></tr>
<tr><td>CNTUPFL</td><td>RESETFL</td></tr>
<tr><td>DESC</td><td>SELINT32FL</td></tr>
<tr><td>DYNSTATE</td><td>USERSYMNAME</td></tr>
<tr><td>Reference</td><td colspan="2">
Refer to the Control Builder Parameter Reference for definitions of each parameter.

Refer to the Control Builder Component Theory for more information on the CTUD Block.
</td></tr>
</table>

3.1.4 DEADTIME block

<table>
<tr><td>Description</td><td colspan="2">Provides calculated output (PV) in which value changes may be delayed from the time that the corresponding change occurred in the P1 input. The delay time can be fixed, or it can vary as the inverse of another input (P2).</td></tr>
<tr><td>Function</td><td colspan="2">Applies a fixed or variable delay to a process input value</td></tr>
<tr><td>Inputs</td><td colspan="2">Requires on input value (P1); a second input (P2) is optional.
• P1 and P2 must be fetched from other function blocks</td></tr>
<tr><td>Outputs</td><td colspan="2">Produces the following output:
• PV and it's status, PVSTS and PVSTSFL</td></tr>
<tr><td>Operators and Functions</td><td colspan="2">Table 4 lists the expression operators and functions supported by this block for reference.</td></tr>
<tr><td>Delay Table</td><td colspan="2">Is used to accomplish the desired delays in the input (P1). P1 values are stored and shifted through the table at a rate that is calculated to produce the desired Deadtime. The table-shift rate is derived from the following information:
• The sample rate of the P1 value (TS). This is the execution rate of the function block.
• The delay time (DELAYTIME). If fixed delay is selected, user specifies the delay; if variable delay is selected, the delay is derived from P2.
• The number of entries to use in the delay table (NUMLOC). The table has a maximum of 60 entries, but the user may request to use fewer than that (by sorting to NUMLOC).</td></tr>
<tr><td>Delay Type</td><td colspan="2">Two types of delay are supported:
• Fixed Delay
• Variable Delay</td></tr>
<tr><td>Parameters</td><td>C1
C2
CPV
CUTOFF.LM
D1
D2
DELAYTABLE[1..60]
DELAYTIME
DELAYTYPE
DPV
EUDESC
INITREQ</td><td>NUMLOC
NAME
ORDERINCM
P1
P1STS
P2
P2STS
PV
PVFORMAT
PVSTS
PVSTSFL.BAD
PVSTSFL.NORM
PVSTSFL.UNCERTN
PVVALSTS</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.
Refer to the Control Builder Component Theory for more information on the DEADTIME Block.</td></tr>
</table>

3.1.5 ENHAUXCALC (Enhanced Auxiliary Calculation) block

Description	The ENHAUXCALC block provides the following enhancements over the AUXCALC block. • Expands existing arrayed input parameters PSTS and P from six to ten. • These arrayed parameters are added to correspond to each of the ten inputs. – Input Description – Scaling Factor – Enable/Disable Switch – PSUB Substitute Parameter – PP Scaled Input • Both the ENHAUXCALC and AUXCALC blocks are optimized so that expressions use memory based on the number of expressions configured, pcode size of each expression, the number of references in the expression and the offset needed for each expression.
Function	Each expression can contain any valid combination of inputs, operators and functions and may perform arithmetic or logic operations, test conditions, etc. Status information is made available for input as well as the expression results. You can assign the result of an expression, a status, or an input to PV and PVSTS parameters which are then processed like the result of any other Auxiliary function block.
Inputs	Accepts up to 10 optional inputs (P[1] to P[10]) - none are required. • No inputs are required • All inputs must be fetched from other function blocks. • The number of process input connections are equal to the number of inputs; the default is 1. • Configure P inputs contiguously (without breaks) in arrays.
Outputs	Produces these outputs according to the values you assign to them. • PV and its status PVSTS, as well as a Boolean flag, PVSTSFL.BAD, to indicate to other function blocks, that this block's PV status is bad. • Up to eight expression results (C[1] to C[8])
Operators and Functions	“AUXCALC (Auxiliary Calculation) block” on page 34 lists the expression operators and functions supported by this block for reference.
Parameter Identification	You must specify a parameter by its full tag name. For example,“CM25.PumpASelect.PVFL”, or “CM57.PID100.MODE”. In effect, tag names allow expressions to have an unlimited number of inputs and work with any data type. However, do not use more than six parameter references in an expression. The expression syntax has been expanded. Delimiters (') can be used in an expression containing an external reference component. The format for the delimiter usage is as follows: • TagName.'text' The size of each expression in the ENHAUXCALC block is limited to 255 characters. You can use the following additional arrayed parameters in expressions. • CP[1..10] • PP[1..10] • PENABLE[1..10] • PSUB[1..10] • PCODESIZE[1..8] • NUMSRCCONN[1..8]

<table>
<tr><td>Expression Rules</td><td colspan="2">

Must include full tag.parameter name for P inputs in the expression and enclose identification number in brackets instead of parenthesizes. For example, CM151.AUXCALC BLOCK.P[1] * CM151.AUXCALC BLOCK.P[2] is valid.
Expressions cannot contain an assignment operation (a colon and equal sign with the current syntax) For example,“CM1.PID1.MODE:=X[1]” is invalid.
Each expression produces a single value (arithmetic or logical which is automatically stored in a“C” parameter. For example, if you write four expressions, the result of the first expression is stored in C[1], the result of the second is stored in C[2], etc. You can use these results, by name, in succeeding expressions. In this example, you could use C[1] as an input to expressions 2, 3, and 4.
You can mix and nest all operators and functions (including conditional assignments) in any order as long as types match or can be converted.
You can use blanks between operators and parameter names, but they are not required.
You can use all data types in expressions except for Time data types. They are all treated as numeric types.
You must configure calculator expressions contiguously (without breaks) in the arrays.
A short description can be provided for the expressions using the expression descriptor parameter (EXPRDESC[1..8]). The results of the expressions, which use the CONST[1…8] parameters, are affected if you change the values of these parameters on the Constants tab.
With R410, non-CEE controllers such as PMD and Safety Manager, and Experion server points such as TPS and SCADA, can be configured in the Expressions.
With R410, when you write the expressions using the TPS point's parameter references, ensure that the TPS reference parameter is configured using the parentheses “()”to specify array index. However, when you write the expressions using the other non-CEE points you can use the brackets “[].”

</td></tr>
<tr><td>Parameters</td><td>C[1..8]
CONFIGCODE[1..8]
CONFIGDESC[1..8]
CONFIGSTS[1..8]
CONST[1…8]
CONSTACCLOCK
CONSTENABLE
CP[1..10]
CSTS[1..8]
DESC
EUDESC
EXECCODE[1..8]
EXECDESC[1..8]
EXECSTS[1..8]
EXPR[1..8]
EXPRDESC[1…8]
NAME
NUMSRCCONN[1..8]
ORDERINCM
P[1..10]</td><td>PCODESIZE[1..8]
PCONN[1..10]
PDESC[1..10]
PENABLE[1..10]
PP[1..10]
PSTS[1..10]
PSUB[1..10]
PV
PVFORMAT
PVSRC
PVSTS
PVSTSFL.ALL
PVSTSFL.BAD
PVSTSFL.MAN
PVSTSFL.NORM
PVSTSFL.UNCERTN
PVSTSSRC
PVVALSTS
SRC</td></tr>
</table>

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ENHAUXCALC Block.

3.1.6 ENHGENLIN (Enhanced General Linearization) block

Description	Calculates an output value (PV) as a function of the input value (P1) and ACTLINSEG parameter value based on configured linear coefficients. It can be any function represented by up to 12 continuous, linear segments defined with 13 monotonic value pairs.	
Function	Typically used to provide a linearized PV (in engineering units) for a sensor actuator, or process with nonlinear characteristics. The ACTLINSEG parameter is used to select the linearization segment tables to define the input-output relationship curves. This block can also be used to characterize functions of a single parameter, such as heat transfer vs. flow rate, or efficiency as a function of load. It is particularly useful when the relationship of the input to engineering units is empirically determined.	
Input	Two input values are required: • P1 must be fetched from another function block. • ACTLINSEG parameter value can be user-defined or fetched from another function block.	
Outputs	• PV, and PVSTS that displays the status of the PV. • Boolean flag (PVSTSFL.BAD) to indicate to other function blocks, that this block's PV status is bad.	
Platform Supported	The ENHGENLIN block is supported on the following platforms. • C300 (20 ms CEE and 50 ms CEE) • C200E • ACE • SIM-C200E • SIM-C300	
Parameters	ACTLINSEG COMMIT DESC DIFFVALUE ENBTUNE EUDESC IN[1..4][0..12] NAME NUMLINSEG NUMSEGS[1..4] NUMSEGS [INDEX] ORDERINCM OUT[1..4][0…12] P1 P1STS	PV PVFORMAT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN PVVALSTS RESTORE TEMPIN[1..4][0..12] TEMPOUT[1..4][0..12] VIEWLINSEG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ENHGENLIN Block. The ENHGENLIN block is typically applicable in the C200E and C300 (20ms CEE and 50ms CEE) controllers, and ACE controllers.	

3.1.7 FLOWCOMP (Flow Compensation) block

Description	Operates on uncompensated flow measurements of liquids, steam, gases or vapors. It computes a flow compensation factor based on variations in parameters like temperature, pressure, specific gravity, and molecular weight. The block derives a compensated flow value as its output.
Function	Offers you five different equations for calculating the flow compensation term (COMPTERM). There is one equation for liquids, one for steam, and three for gases and vapors. Each equation may require different inputs. For example, depending on which gases and vapors equation you choose, one requires temperature and pressure measurements, another requires temperature, pressure and specific gravity, and a third requires temperature, pressure and molecular weight.
Inputs	• The PV Equation Type (PVEQN) you select determines the number of inputs that the FLOWCOMP block requires. – Equation A requires 2 inputs – Equation B requires 3 inputs – Equation C requires 4 inputs – Equation D requires 4 inputs – Equation E requires 5 inputs • All inputs must be fetched from other function blocks
Outputs	This block produces the following outputs: • PV and its status, PVSTS

Parameters	BADCOMPTERM.FL	P
	BADCOMPTERM.PR	P0
	BADCOMPTERM.SV	PSTS
	CF1	PV
	CF2	PVCHAR
	COMPHILM	PVEQN
	COMPLOLM	PVFORMAT
	COMPTERM	PVSTS
	CPV	PVSTSFL.BAD
	DESC	PVSTSFL.MAN
	EUDESC	PVSTSFL.NORM
	F	PVSTSFL.UNCERTN
	FSTS	Q
	G	QSTS
	GSTS	RG
	HIALM.PR	RMW
	HIALM.SV	RP
	HIALM.TYPE	RQ
	INALM	RT
	MAXCYCLE	RX
	MW	T
	MWSTS	T0
	NUMPINPT	TSTS
	NAME	X
	ORDERINCM	XSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the FLOWCOMP Block.	

3.1.8 GENLIN (General Linearization) block

Description	Calculates an output value (PV) as a function of the input value (P1) based on a separate function that can be represented by 2 to 13 user-defined coordinates. (You specify the IN and OUT values of each coordinate to make a segment.) The input value (P1) is then compared with the input range of each segment and the output is set at the intersection of the input with the appropriate segment.
Function	Typically used to provide a linearized PV (in engineering units) for a sensor with nonlinear characteristics. Block can also be used to characterize functions of a single parameter, such as heat transfer versus flow rate, or efficiency as a function of load. It is particularly useful when the relationship of the input to engineering units is empirically determined.
Input	One input value (P1) is required: • P1 must be fetched from another function block. • Number of process input connections is 1.
Outputs	PV and its status, PVSTS, as well as a Boolean flag, PVSTSFL.BAD, to indicate to other function blocks, that this block's PV status is bad.

Segment Extension	The first and last segments are treated as if they are infinitely extended. This means, if P1 is less than IN[0] or greater than IN (NUMSEGS), PV is computed by assuming that the slope in the appropriate segment continues from the intersection point.	
Parameters	DESC EUDESC IN[0..12] NAME NUMSEGS ORDERINCM OUT[0..12] P1 P1STS	PV PVFORMAT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN PVVALSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the GENLIN Block.	

3.1.9 LEADLAG (Lead Lag) block

Description	May be configured to apply a lead-time and two lag-time compensation factors to a process input value.	
Function	Provides dynamic lead-lag compensation to the P1 input. It supports one lead compensation and two lag compensation factors. There is a time constant for each compensation factor. Specifying a zero value for any time constant will suppress the corresponding compensation.	
Input	One input value (P1) is required: • P1 must be fetched from another function block.	
Outputs	The following output is produced: • PV and its status, PVSTS and PVSTSFL	
Equations	This function block only supports one equation - a single input filtered with one lead compensation and two lag compensations. There is a time constant for each compensation factor. Specifying a zero value for any time constant will suppress the corresponding compensation.	
Parameters	CPV DPV DESC EUDESC INITREQ LAG1TIME LAG2TIME LEADTIME NAME ORDERINCM P1	P1STS PV PVFORMAT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN PVVALSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the LEADLAG Block.	

3.1.10 ROC (Rate of Change) block

Attention
The ROC block can only be used with C300, C200E, and ACE Controllers.

Description	It is desirable to prevent a random change in the output sequence. The Rate of Change block is a computational block used on the input side of Control blocks for limiting the input variable to the block (typically SP). In Power plants, speed control of HT motors requires control without exceeding the rate of change of the current. In some temperature control applications, dynamic profiling is needed based on the deviation of SP and the actual temperature.	
Function	• If the input variation is more than the rate trip limit in either direction, the rate of change of output is limited to the change specified by the rate trip limits. The output changes at the specified rate limits until the value is equal to the input variable. • PVROCBYPASSFL is provided in the function block to BYPASS the rate trip limit • If the rate limits are NaN, then limits are not applied and PV is set to P1. • This block provides a Bad PV alarm based on the status of the output • For an invalid input (=NaN), rate limiting is not done and the output is NaN. • With R410, ROC block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.	
Input	• P1 - Process Input 1. • PVROCPOSLM - Indicates a positive PV rate of change limit • PVROCNEGLM - Indicates a negative PV Rate of Change limit	
Outputs	• PVROCPOSFL - This flag turns ON when the rate limiting is in the positive direction. • PVROCNEGFL - This flag turns ON when the rate limiting is in the negative direction. • PV - Output of ROC. • BADPVFL - This flag is set when a bad input is received at the block.	
Parameters	P1STS	P1
	PV	PV
	PVFORMAT	PVEUHI
	PVSTS	PVEULO
	PVSTSFL.BAD	PVROCNEGFL
	PVSTSFL.MAN	PVROCBYPASSFL
	PVSTSFL.NORM	PVROCPOSLM
	PVSTSFL.UNCERTN	PVROCPOSFL
	PVVALSTS	PVROCNEGLM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ROC Block.	

3.1.11 SIGNALSEL (Signal Selector) block

Description	Lets you select one of up to six inputs using configured selection criteria, or allows you to average two or more of the inputs.

Function	The Signal Selector function block accepts as many as six input signals, and may be configured to do one of the following on these inputs: • Select the input with the minimum value. • Select the input with the maximum value. • Select the median input. • Calculate the average of the inputs. • Select an input based on the value of an external control signal; i.e., act as a multiplexor. With this option, the function block accepts two to six inputs plus a control signal. • Force the function block output to Bad. The SIGNALSEL block provides a number of configuration options, which makes it extremely flexible. Some examples of how it can be used are: • Select the middle of three inputs. If one input goes Bad, calculate the average of the remaining two; and, if two inputs go Bad, force the output to Bad. • Calculate the average of two inputs. If the difference between the inputs exceeds a certain value for more than “x” seconds, issue an alarm; and, if it exceeds another value for more than “y” seconds, force the output to Bad. • Select from two inputs, based on a Boolean value pushed from another function block. Also provide bumpless switching between the inputs. • With R410, SIGNALSEL block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband units for all the alarms.
Inputs	This function block accepts between two to six selectable inputs, P[1..6]. Minimum two inputs (P[1] and P[2]) are required. If the block acts as a multiplexor then additionally a multiplex-selector input (MUXSEL) is also required. All inputs shall be fetched from other input blocks The minimum number of inputs is two. If less than two inputs are connected a warning *“At least two inputs need to be connected”* shall be given during load, and activation of the block shall be prevented If the total number of valid inputs.(NUMINPTS) goes less than the value of the configurable parameter Minimum Valid Inputs (NMIN), the output of the block shall go bad. The NMIN parameter applies only to the following selection methods: MIN, MAX, MED, or AVG, and is not applicable if the selection method is MUX or Force selection is performed.

<table>
<tr><td>Input Ranges and Limits</td><td colspan="2">The function block always ignores Bad inputs (NaN). In addition, the user may choose to ignore the “n” highest (IGNORHI) or/and “m” lowest (IGNORLO) inputs. These values can be from Logic blocks and user programs may also store to it - hence, the number of ignored inputs may be dynamic.
• If all the inputs are ignored, output shall go Bad.
• If the total number of inputs to be ignored (n+m) is equal to or greater than the total number of connected inputs, a warning message “IGNORHI+IGNORLO should be less than the number of connected inputs” will be given during load and activation of the block will be prevented. During the running state, for the same condition, a non-critical error with the same error message is displayed and the previous value of IGNORHI or IGNORLO (whichever is causing the error) is retained
The user may also choose to ignore inputs that are outside user-specified ignore limits.
Ignore Limit Checking
Ignore Limit is the maximum allowable range between the lowest and highest input. Inputs that are outside this range (IGNORLM) for more than a specified time (IGNORTM) are ignored.
The SIGNALSEL block performs ignore limit checking as follows:
• It ignores Bad inputs (NaN), and the highest and lowest inputs (defined by IGNORHI and IGNORLO).
• If there are no remaining inputs, ignore limit checking is not done for the block.
• It calculates a “center value” from the inputs that remain:
– If the number of remaining inputs is odd, the “center value” = the median input.
– If the remaining inputs is even, “center value” = the average of the middle two inputs.
• It calculates a high and low ignore limit from specified limit (IGNORLM):
– High ignore limit = “center value” + IGNORLM / 2
– Low ignore limit = “center value” - IGNORLM / 2
• It compares each of the remaining inputs with the high and low ignore limits.
• If an input is outside the ignore limits for more than IGNORTM (the ignore time) seconds, the function block updates the appropriate parameters (the ignored input flags IGNORD, IGNORDFL[1…6] and the current number of valid inputs CURPINPT). As a result, the input will be ignored in future processing (i.e., Input Selection).
• The center value, high and low ignore limits shall be computed every cycle of execution of the block.
• Inputs that have been ignored on exceeding ignore limits, shall become valid again when their value returns back within the high and low ignore limits.
If there are only two remaining inputs, and the difference between them exceeds the ignore limit, the block's output (PV) is set to NaN.
IGNORHI, IGNORLO and ignore limit checking shall not be applicable for the MUX selection method.</td></tr>
<tr><td>Outputs</td><td colspan="2">This auxiliary PV block produces an output PV and its status, PVSTS
The output parameter SELINP denotes which input, if any has been selected as the output.
The Ignore Input feature produces the following output flags
• The flag IGNORD indicates if any of the inputs is ignored or not.
• The parameter IGNORDFL[1..6] provides individual flags for each input indicating if it was ignored</td></tr>
<tr><td rowspan="3">Equation Options</td><td colspan="2">The method for selecting inputs is determined by the configuration parameter SELMETHOD, whose values are tabulated below. Detailed operations of the selection methods are provided in the section “Selection methods” on page 51</td></tr>
<tr><td>Method</td><td>Processing</td></tr>
<tr><td>MIN</td><td>Select the input with the minimum value. Ignored inputs are excluded.</td></tr>
</table>

	MAX	Select the input with the maximum value. Ignored inputs are excluded.
	MED	Select the median input. Ignored inputs are excluded.
	AVG	Calculate the average of the inputs. Ignored inputs are excluded.
	MUX	Select an input based on the Multiplex value; i.e., act as a multiplexor. Inputs are not ignored.

Force-Select:

- The operator or a user program may override the selection method and “force select” a particular input using the FRCPERM, FRCREQ, and FRCSEL parameters.
- Force-select may override only the following selection methods: MIN, MAX, MED, or AVG and is not applicable if the selection method is MUX.
- If the force selected input is not connected, then the PV value goes Bad (NaN) and the respective unconnected input remains selected
- Ignore Inputs, Ignore limit checking, NMIN and deviation alarming are not applicable during force selection. Also the deviation alarm state should return to normal.

Bumpless Input Switching

The function block may be configured to provide bumpless switching between inputs. If so configured, the block will ramp to the new input value when any of the following occur:

- The selected input changes.
- The number of valid inputs changes.

Ramping rate (PVRATE) is specified in rate of change per minute. PV shall ramp at this rate to the new value. If the ramp rate is zero bumping would occur. Ramping can be disabled by setting ramp rate to NaN.

Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 BOOLMUX CURPINPT DESC DEVALM.ALL DEVALM.DB DEVALM.FL DEVALM.PR DEVALM.SV DEVALM.TM DEVALM.TMO DEVALM.TP DEVLM DEVTM EUDESC FRCPERM FRCREQ FRCSEL IGNORD IGNORDFL[1..6] IGNORDHI IGNORDLO	IGNORDHI IGNORDLO MEDOPT MUXSEL NAME NMIN NUMPINPT ORDERINCM P[1..6] PDESC[1..10] PSTS[1..6] PV PVFORMAT PVRATE PVSTS PVSTSFL.ALL PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN SELDESC SELIN SELMETHOD
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the SIGNALSEL Block.	

3.1.12 Selection methods

MIN

- The output (PV) gets the minimum value of all the valid (not ignored) inputs.
- The selected input shall be the input that has this minimum value.
- If two or more inputs have the minimum value then the selected input would be the input with the highest index. For instance, if P[2] and P[5] have the minimum value then the selected input would be P[5].

MAX

- PV gets the maximum value of all the valid inputs.
- The selected input shall be the input that has this maximum value.
- If two or more inputs have the maximum value then the selected input would be the input with the highest index. For instance, if P[2] and P[5] have the maximum value then the selected input would be P[5].

AVG

- PV shall be the average of only the valid inputs.
- The selected input shall be None because PV is a calculated value and not any input by itself.

MED

- All the valid inputs are arranged in ascending order and median value is taken as PV.
- If odd number of valid inputs is present then the middle value will be the PV and the selected input shall be the respective input.
- If even number of valid inputs is present then the PV shall be any one of the following depending on the parameter ‘Median Option for Middle Two Inputs (MEDOPT)’:
- If MEDOPT is MIN, then PV shall be the minimum of the middle two values and the respective input shall be selected input.
- If MEDOPT is MAX, then PV shall be the maximum of the middle two values and the respective input shall be selected input.
- If MEDOPT is AVG, then PV shall be the average of the middle two values and selected input shall be none because average is computed.

Attention

While arranging in ascending order, if two inputs have same value then the input that comes first in order 1 to 6 precedes the other.

MUX

- A Boolean flag BOOLMUX is employed to choose between Integer Mux selection and Boolean Mux selection. If the flag is set to On, Boolean selection will be performed, otherwise Integer selection will performed.
- In Integer Mux selection, a control signal MUXSEL (multiplex-selector) is required, which shall be user configurable or fetched from other function block, or user programs could also store to it.
- If the fetched or configured MUXSEL value goes invalid, such as greater than the number of process inputs, then the previous valid value of MUXSEL is retained and the respective input remains selected.
- If the fetched or configured MUXSEL is valid, but the input corresponding to MUXSEL is not connected, then the PV value goes bad (NaN) and the respective unconnected input remains selected.
- In Boolean Mux selection, the SELXFL[1..6] flags are scanned from 1 to 6 and the block selects an input whose corresponding SELXFL flag is first On.
- If the Boolean selected input is not connected, then the PV value goes bad (NaN) and SELIN will have the index of unconnected input.
- And, if none of the SELXFL flag is on (but only the BOOLMUX is on and SELMETHOD is Mux), then the PV value goes bad (NaN) and SELIN's value will have None value.
- Bad inputs may also be selected.
- Ignoring of Inputs and deviation alarming are not applicable for MUX. Also, the deviation alarm state should return to normal.
- PV gets the value of the selected input.
- If the value of the input denoted by the control signal is Bad, then the PV also goes Bad.

3.1.13 TOTALIZER block

Description	Periodically adds an input value (P1) to an accumulator value (PV); sets status flags to indicate when accumulator value is “near”, “nearer”, “nearest” the user specified target value.

Function	Typically used to accumulate flows. For situations where flow transmitter may not be precisely calibrated near zero-flow value, a zero-flow cutoff feature is provided such that when P1 is below the cutoff value it clamps to 0 (zero). Block also supports warm restart.		
Input	One input (P1) is required: • P1 is the value to be accumulated -- input value may be real, integer or Boolean, but is stored as a real number. • P1 must be fetched from another block. • Number of process input connections is 1.		
Outputs	The following outputs are produced: • Accumulated value (PV) and its status (PVSTS), as well as a Boolean flag, PVSTSFL.BAD, to indicate to other function blocks, that this block's PV status is bad. • Flags, indicating if accumulated value has reached user-specified target value or one of the accumulator deviation trip points (ACCTVFL and ACCDEV.FL(1-4]).		
Equations	You can configure PVEQN to specify how the block should handle bad input and warm restarts. Specific handling combinations for a given PVEQN selection are:		
	Equation	Bad Input Handling	Warm Restart Handling
	EqA	Use zero if input is bad.	Continue after input turns valid
	EqB	Use last good value if input is bad	Continue after input turns valid
	EqC	Stop if the input is bad and set PV to NaN	Continue after input turns valid
	EqD	Use zero if input is bad.	Stop after a warm restart
	EqE	Use last good value if input is bad	Stop after a warm restart
	EqF	Stop if the input is bad and set PV to NaN	Stop after a warm restart
Parameters	ACCDEV.FL[1..4] ACCDEV.TP[1..4] ACCTV ACCTVFL C1 CMDATTR COMMAND CUTOFF.LM DESC EUDESC HIALM LASTGOOD NAME OLDAV ORDERINCM P1 P1STS	PV PVEQN PVFORMAT PVSTS PVSTSFL.BAD PVSTSFL.NORM PVSTSFL.NORM PVSTSFL.UNCERTN PVVALSTS RESETFL RESETVAL STARTFL STATE STOPFL TIMEBASE	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the TOTALIZER Block.		

3.2 Device Control Block

Related topics

"DEVCTL (Device Control) block" on page 54

3.2.1 DEVCTL (Device Control) block

Description	Provides multi-input, multi-output function for interfacing to discrete devices such as motors, pumps, solenoid valves and motor-operated valves. The Device Control block contains built-in structures for handling interlocks and supports display of the interlock conditions in group, detail and graphic displays.
Function	Allows the manipulation of sets of digital outputs and interprets corresponding feedback of digital inputs represented by the state parameter PV (Current Feedback State). Operation consists of transmitting commands represented by state parameter OP (commanded output state), monitoring PV, and producing alarms based on various configurations, such as if PV has not achieved state commanded in OP. Provides safety interlocks, individual state interlocks, initialization manual, maintenance statistics, and batch level 1 drive functions.
Inputs	May have from 0 to 4 inputs (DI [1..4]); each input is a Boolean value that represents the state of other block output or a field DICHANNEL block.
Outputs	May have from 0 to 3 outputs. Each output can be Boolean (DO[1..3]) or pulsed (PO[1..3]). You can only connect a DO[1..3] or a PO[1..3] to any one output at a time. • You can connect the Boolean output DO[1..3] to a Boolean parameter in any other function block or to the DO.SO in the DOCHANNEL block. • You can only connect the pulsed output PO[1..3] to a DO.ONPULSE or DO.OFFPULSE in the DOCHANNEL block. • Note that you can only connect one Boolean (DO[1..3]) or one pulsed (PO[1..3]) output to any one DOCHANNEL block as a DO.SO or DO.ONPULSE or DO.OFFPULSE, respectively.
Alarms	An available set of PV state alarms may be configured to represent Bad PV or disagreements between the commanded output state (OP) and the feedback state (PV). A variety of override alarms are also available. Each of these alarms possesses all the standard attributes of system alarms.

Parameters	ASTEPID	MOMSTATE
	BADPVALM.FL	NAME
	BADPVALM.PR	NORMMODE
	BADPVALM.SV	NORMMODEATTR
	BYPASS	NULLPVFL
	BYPPERM	NUMDINPUTS
	CLROPREQFL	NUMDOUTS
	CMDDISALM.FL	NUMSIOVRD
	CMDDISALM.PR	NUMSTATES
	CMDDISALM.SV	NUMTRANS[0..2]
	CMDDISALM.TM[0..2]	OFFNRMALM.FL
	CMDFALALM.FL	OFFNRMALM.PR
	CMDFALALM.PR	OFFNRMALM.SV
	CMDFALALM.SV	OI[0..2]
	CMDFALALM.TM[0..2]	OIALM.FL[0..2]
	CONTROLREQ	OIALM.OPT[0..2]
	DESC	OIALM.PR[0..2]
	DI[1..4]	OIALM.SV[0..2]
	DIPVMAP[0..15]	OP
	DO[1..3]	OPCMD[0..2]
	EUDESC	OPDOMAP[0..3][1..3]
	GOP	OPFINAL
	GOPFINAL	OPREQ
	GOPREQ	OPTYPE
	GOPSCADA	ORDER
	GPV	ORDERINCM
	GPVAUTO	PI[0..2]
	HIALM.PR	PO[1..3]
	HIALM.SV	POCONNECTED[1..3]
	HIALM.TYPE	PULSEWIDTH[1..3]
	HOLDOPT	PV
	INALM	PVAUTO
	INBETFL	PVFL[0..2]
	INITCONNECTD[1..3]	PVSOURCE
	INITMAN	PVSRCOPT
	INITOPOPT	PVSTS
	INITREQ[0..2]	REDTAG
	LASTGOPREQ	RESETFL
	LASTOPREQ	RESTARTOPT
	LASTOPTYPE	SAFEOP
	LASTREQFL	SAFEREDTAG
	LASTSTEP	SEALOPT

	LOCALMAN	SI
	MAINTOPT	SIALM.FL
	MAXTIME[0..2]	SIALM.OPT
	MAXTRANS[0..2]	SIALM.PR
	MODE	SIALM.SV
	MODEATTR	STARTOPT
	MODEATTRFL.NORM	STATETEXT[0..6]
	MODEATTRFL.OPER	STATETIME[0..2]
	MODEATTRFL.PROG	STOPOPT
	MODETRACK	UNCMDALM.FL
		UNCMDALM.PR
		UNCMDALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the DEVCTL Block.	

3.3 Data Acquisition Block

Related topics

“DATAACQ (Data Acquisition) block” on page 57

3.3.1 DATAACQ (Data Acquisition) block

Description	Processes a specified process input value (P1) with or without filtering into an output value (PV).
Function	Normally configured to fetch process input from an AI device, controller or another function block; it performs the following major functions: • Brings input data and updates the input (P1) and its status P1STS. If input provides value only, P1STS is derived from the value. • PV characterization option lets you configure Linear or Square Root conversion on the P1 input, if required. • Low signal cut off function lets you configure a low cutoff value for P1 with Linear or Square Root PV characterization. • Performs filtering (P1FILTTIME) and clamping (P1CLAMPOPT) on P1 through parameters P1FILTTIME and P1CLAMPOPT, and stores the result in PVAUTO. • Generates alarm flags when PV exceeds any of a number of user-specified alarm trip points for more than a designated time interval. • PV source selection option (PVSOURCE) supports automatic, manual, and substitute. A PV source selection of manual means an operator can store a value to the output (PV). A selection of substitute means a user program can store a value to PV.
Input	Requires one process input value (P1) that must be fetched from another block. • Number of process input connections (NUMPINT) is 1. • P1STS provides the status of P1.

Input Ranges and Limits	• PVEUHI and PVEULO define the full range of P1 in engineering units.
	  – PVEUHI is 100% of full scale value.
	  – PVEULO is 0% of full scale value.
	• PVEXHILM and PVEXLOLM define the high and low limits of P1in engineering units.
	  – If P1 clamping is desired (P1CLAMPOPT = Enable), the block clamps the input within PVEXHILM and PVEXLOLM.

Output	Produces an output value (PV) and its status (PVSTS).

Alarm Processing	With R410, DATAACQ block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.
	Block may be configured to generate an alarm when PV exceeds one of various trip points (XXXX.TP parameters) for more than a specified time.
	• Parameters with the following suffixes also apply to alarm processing:
	  – XXXX.DB, XXXX.DBU (deadband, deadband units)
	  – XXXX.FL (alarm flag)
	  – XXXX.PR (priority)
	  – XXXX.SV (severity)
	  – XXXX.TP (trip point)
	  – XXXX.CT (alarm count)
	Where XXXX stands for one of the following:

	PVHIALM	PVLLALM	PVHISIGCHG
	PVHHALM	ROCPOSALM	PVLOSIGCHG
	PVLOALM	ROCNEGALM	Bad PV Alarm

Parameters	ALMDB ALMDBU ALMTM BADPVALM.FL BADPVALM.PR BADPVALM.SV BADPVALM.TM BADPVALM.TMO DESC EUDESC HIALM.PR HIALM.SV HIALM.TYPE INALM INSBLOCK[1..10] INSFAILFL INSFAIL.PR INSFAIL.SV LASTGOODPV LOCUTOFF NAME ORDERINCM NUMINSERT P1 P1CLAMPOPT P1EU P1FILTINIT P1FILTTIME P1STS PV PVAUTO PVAUTOSTS PVCHAR PVEUHI PVEULO PVEXHIFL PVEXHILM PVEXLOFL PVEXLOLM PVFORMAT PVHHALM.DB PVHHALM.DBU	PVHHALM.TM PVHHALM.TMO PVHHALM.TP PVHIALM.DB PVHIALM.DB PVHIALM.DBU PVHIALM.FL PVHIALM.PR PVHIALM.SV PVHIALM.TM PVHIALM.TMO PVHIALM.TP PVHISIGCHG.CT PVHISIGCHG.TP PVLLALM.DB PVLLALM.DBU PVLLALM.FL PVLLALM.PR PVLLALM.SV PVLLALM.TM PVLLALM.TMO PVLLALM.TP PVLOALM.DB PVLOALM.DBU PVLOALM.FL PVLOALM.PR PVLOALM.SV PVLOALM.TM PVLOALM.TMO PVLOALM.TP PVLOSIGCHG.CT PVLOSIGCHG.TP PVP PVSOURCE PVSRCOPT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN PVVALSTS

	PVHHALM.FL PVHHALM.PR PVHHALM.SV	ROCNEGALM.FL ROCNEGALM.PR ROCNEGALM.SV ROCNEGALM.TM ROCNEGALM.TMO ROCNEGALM.TP ROCPOSALM.FL ROCPOSALM.PR ROCPOSALM.SV ROCPOSALM.TM ROCPOSALM.TMO ROCPOSALM.TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the DATAACQ Block.	

3.4 IO Channel Blocks

Related topics

"AICHANNEL" on page 61
"AOCHANNEL" on page 61
"DICHANNEL" on page 62
"DOCHANNEL" on page 62
"PWMCHANNEL" on page 63
"SIFLAGARRCH" on page 64
"SINUMARRCH" on page 64
"SITEXTARRCH" on page 65

3.4.1 AICHANNEL

Description	Provides standard analog interface to control function blocks.	
Function	1. Brings PV data from an associated IOM block. 2. Assigns BAD status to PV parameter when appropriate.	
Inputs	Floating point value in engineering units.	
Outputs	Floating point value in engineering units.	
Parameters	BADCAL BADCODE CALBIAS CJOFFSET DEBUG FETCHMODE FREEZETIME IOCTYPE	IOMCONN OHMOFFSET ORDERINCM OVERRANGE PV PVRAW PVSTS PVVALSTS UNDERRANGE
Associated Block	Prior to loading, block must be "associated"with 1 channel of corresponding IOM block that interfaces with the physical AI hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.2 AOCHANNEL

Description	Provides a standard analog output signal for operating final control elements.
Function	• Brings OP data from connected blocks and conveys OP data to be stored in an associated IOM block. • Reverses OP direction if OPTDIR option is REVERSE. • Sets INITVAL parameter to appropriate value based on echo data. • Assigns safe value if STS parameter is BAD or UNCERTAIN. • Sets INITREQ to TRUE value if AOC or IOM block is inactive or a communications error occurs.
Inputs	Only one control block can interface to this block.

Outputs	Floating point value in engineering units.	
Parameters	BACKCALCOUT BADCAL BADCODE CALIBALL CALBIAS COMMFAILFL DEBUG FETCHMODE FREEZETIME INITREQ	INITVAL IOCSTATE IOCTYPE IOMCONN OP OPFINAL OPSOURCE OPTDIR ORDERINCM
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with physical AO hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.3 DICHANNEL

Description	Provides a standard digital interface to control blocks.	
Function	• Brings PV data from an associated IOM block. • Assigns Bad status to PV parameter when appropriate	
Inputs	Digital (PV) signals received from the field.	
Outputs	PV status value that can be used by other data points in system.	
Parameters	BADCODE BADPV COS DEBUG FETCHMODE FREEZETIME HWFAULT INBADOPT IOCSTATE	IOCTYPE IOMCONN NOFIELDPWR ORDERINCM PVFL PVSTS PVVAL PVVALSTS WIREOFF
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with physical Digital Input hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.4 DOCHANNEL

Description	Generates status output [0 or 1), pulsed output (ON or OFF) for specified pulse time based on origin of input and parameters.

Function	• Brings SO or PO from connected blocks and stores value in an associated IOM block. • Sets INITVAL parameter to appropriate value based on echo data. • Stops SO if INTREQ is TRUE • Sets INITREQ to TRUE value if DOC or IOM block is inactive or a communications error occurs. • You can configure PO to be Direct or Reverse by connecting ONPULSE or OFFPULSE pin.	
Inputs	Only one control block can interface to this block.	
Outputs	Digital (Boolean) value or pulsed (real) value.	
Parameters	BACKCALCOUT BADCODE COMMFAILFL DEBUG DOMSO DOTYPE FETCHMODE FREEZETIME INITREQ INITVAL LASTSERIAL	IOCSTATE IOCTYPE IOMCONN NOFIELDPWR NOLOAD OFFPULSE ONPULSE ORDERINCM SHORT SO SOSOURCE VERIFYLOST
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with physical DO hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.5 PWMCHANNEL

Description	Provides a pulse width modulated output signal for operating final control elements in combination with a DO Module.
Function	• Brings OP data from connected block and stores data in an associated IOM block. • Reverses OP direction if OPTDIR option is REVERSE. • Sends out a pulse based on the configured pulse width period (PWMPERIOD) with its duty cycle determined by the OP data. • Sets INITVAL parameter to appropriate value based on echo data. • Assigns safe value if status parameter is BAD or UNCERTAIN. • Sets INITREQ to TRUE (ON) value if CM containing PWMC block or IOM block is inactive or a communications error occurs. If communication fails, the pulse function terminates. So, be sure you select the proper SHED VALUE for the DOM channel to reflect the desired inactive digital state.
Inputs	OP value from another block. Typically, output in 0 to 100% from a PID block, which indicates the proportion of time period that the output will be turned on.
Outputs	Pulsed (real) value

Parameters	COMMFAILFL DOMSO FETCHMODE INITREQ INITREQLATCH INITVAL IOCNUMBER	NOFIELDPWR NOLOAD OP ORDERINCM PWMPERIOD SHORT VERIFYLOST
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with physical DO hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.6 SIFLAGARRCH

Description	Provides a read/write interface to a Boolean array of data from a serial device.	
Function	• Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block. • Supports up to 512 Boolean values(PVFL[1..512] from the device. • Provides access to the array of data by other blocks - one element at a time. • Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRORCODE). • Provides bad PV flag (BADPVFL) and initialization request flag (INITREQ) parameters to mirror the status of the ERRFL parameter - data is valid or invalid.	
Inputs	Boolean value from device or another block	
Outputs	Boolean value	
Parameters	AUXDATA[0..7] BADCODE BADPVFL DEVADDR ERRCODE ERRFL FETCHMODE INITREQ IOCNUMBER	IOCSTATE IOCTYPE IOMCONN NFLAG ORDERINCM PVFL[1..512] PVSTS STARTINDEX WRITEOPT
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding SIM block that interfaces with physical FTA A and FTA B hardware at execution runtime. Use channels 0-15 for FTA A and channels 16-31 for FTA B. For optimum performance, assign channels to SIM block for given FTA contiguously. For example, if you have four SIFLAGARRCH blocks to use with the FTA A, assign them to SIM block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.7 SINUMARRCH

Description	Provides a read/write interface to a Numeric array of data from a serial device.

Function	• Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block. • Provides Numeric values of the type 64-bit floating point, but data from the device can be of type 32-/64-bit floating point (Real: 4-byte), 32-bit integer (Integer: 2-byte), or Boolean (Byte: 1-byte). • Supports up to 64 Numeric values (PV [1..64] from the device. Since the maximum size of the interface to the device is 64 bytes, the number of Numerics (NNUMERIC) per data type is 0 to 16 for Real, 0 to 32 for Integer, or 0 to 64 for Byte type register in the device. • Provides access to the array of data by other blocks - one element at a time. • Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRORCODE). • Provides bad PV flag (BADPVFL) and initialization request flag (INITREQ) parameters to mirror the status of the ERRFL parameter - data is valid or invalid.	
Inputs	Up to 64 bytes of Real, Integer, or Byte type data from the device. (Block always provides Numeric values of 64-bit floating point type.)	
Outputs	See above.	
Parameters	AUXDATA[0..7] BADCODE BADPVFL DEVADDR ERRCODE ERRFL FETCHMODE INITREQ	IOCNUMBER IOCSTATE IOCTYPE IOMCONN NNUMERIC ORDERINCM PV[1..64] PVSTS[1..64] STARTINDEX WRITEOPT
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding SIM block that interfaces with physical FTA A and FTA B hardware at execution runtime. Use channels 0-15 for FTA A and channels 16-31 for FTA B. For optimum performance, assign channels to SIM block for given FTA contiguously. For example, if you have four SINUMARRCH blocks to use with the FTA A, assign them to SIM block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.4.8 SITEXTARRCH

Description	Provides a read/write interface to a Text (or String) array of data from a serial device.

<table>
<tr><td>Function</td><td colspan="2">

Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block.
Supports up to 8 Text values (STR[1..8]) from the device. Since the maximum size of the interface to the device is 64 bytes, the valid range of values depends on the combination of number of string values (NSTRING) and length of string values (STRLEN) as follows.

If NSTRING is 1 and STRLEN is 64, valid STR[1..8] range is 1.
If NSTRING is 2 and STRLEN is 32, valid STR[1..8] range is 1 to 2.
If NSTRING is 4 and STRLEN is 16, valid STR[1..8] range is 1 to 4.
If NSTRING is 8 and STRLEN is 8, valid STR[1..8] range is 1 to 8.

Provides access to the array of data by other blocks - one element at a time.
Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRORCODE).
Provides bad PV flag (BADPVFL) and initialization request flag (INITREQ) parameters to mirror the status of the ERRFL parameter - data is valid or invalid.

</td></tr>
<tr><td>Inputs</td><td colspan="2">Up to 8 string values depending on whether the length of the string is 8, 16, 32, or 64 characters.</td></tr>
<tr><td>Outputs</td><td colspan="2">See above.</td></tr>
<tr><td>Parameters</td><td>AUXDATA[0..7]
BADCODE
BADPVFL
DEVADDR
ERRCODE
ERRFL
FETCHMODE
INITREQ</td><td>IOCNUMBER
IOCSTATE
IOCTYPE
IOMCONN
NSTRING
ORDERINCM
STARTINDEX
STR[1..8]
STRLEN
WRITEOPT</td></tr>
<tr><td>Associated Block</td><td colspan="2">Prior to loading, block must be “associated” with 1 channel of corresponding SIM block that interfaces with physical FTA A and FTA B hardware at execution runtime. Use channels 0-15 for FTA A and channels 16-31 for FTA B. For optimum performance, assign channels to SIM block for given FTA contiguously. For example, if you have four SITEXTARRCH blocks to use with the FTA A, assign them to SIM block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.</td></tr>
</table>

3.5 Exchange Blocks (ControlNet Interoperability)

Related topics

"REQFLAGARRAY (Request Flag Array) block" on page 67
"REQNUMARRAY (Request Number Array) block" on page 67
"REQTEXTARRAY (Request Text Array) block" on page 68
"RSPFLAGARRAY (Response Flag Array) block" on page 69
"RSPNUMARRAY (Response Number Array) block" on page 69
"RSPTEXTARRAY (Response Text Array) block" on page 70

3.5.1 REQFLAGARRAY (Request Flag Array) block

Description	Provides storage for up to 512 Boolean output flags. The value can be accessed as a simple Boolean (Off or On) using the PVFL[n] or PVVALSTS[n] parameters. Where"n" is the number of the flag.	
Function	Used to define two separate states (Off/On) to indicate status of a particular input. Number of flag values (NFLAG) is user configurable. Current state of flags can be changed/read using flag value (PVFL[n] or PVVALSTS[n]) (Boolean).	
Inputs/Outputs	Boolean output flags (PVFL[0..511]) Boolean output flags (PVVALSTS[0..511])	
Parameters	COMMAND	ERRINFO
	DHCHANNEL	FILENUM
	DHDESTLINK	LASTRESPTM
	DHFL	NFLAG
	DHNODE	ORDERINCM
	DHSRCLINK	PATH
	DONEFL	PVFL[0..511]
	ERRCODE	PVVALSTS[0..511]
	ERRFL	READYFL
		SENDFL
Associated Block	"REQNUMARRAY (Request Number Array) block" on page 67 and "REQTEXTARRAY (Request Text Array) block" on page 68	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REQFLAGARRAY Block.	

3.5.2 REQNUMARRAY (Request Number Array) block

Description	Provides storage for up to 64 integers or floating point values that are accessible through the corresponding PV configuration parameter (PV[n]). Where"n" is the number of the numeric.
Function	Use outputs (PV[0..63]) as source parameters to provide predefined analog constants to other function blocks. A bad numeric output parameter typically has the value NaN (Not-a-Number). Number of Numeric Values (NNUMERIC) is user configurable.

Inputs/Outputs	Up to 64 outputs (PV[0..63]), depending on the number of numeric values (NNUMERIC) configured	
Parameters	COMMAND DHCHANNEL DHDESTLINK DHFL DHNODE DHSRCLINK DONEFL ERRCODE ERRFL ERRINFO FILENUM	LASTRESPTM NNUMERIC ORDERINCM PATH PV[0..63] PVSTS[0..63] PVVALSTS[0..63] READYFL SENDFL TGTDATATYPE
Associated Block	"REQFLAGARRAY (Request Flag Array) block" on page 67 and "REQTEXTARRAY (Request Text Array) block" on page 68	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REQNUMARRAY Block.	

3.5.3 REQTEXTARRAY (Request Text Array) block

Description	Provides storage for up to 64 ASCII characters that are accessible through the corresponding string configuration parameter (STR[n]). Where"n" is the number of the text string.	
Function	• Provides predefined text (STR[0..7]) strings to other blocks. • Number of string values (NSTRING) is user configurable. • The length of the text strings (STRLEN) is user configurable to 64 characters • Supports a maximum size of 64 two-byte characters.	
Inputs/Outputs	Up to 8 output strings (STR[0..7]), depending on the number of string (NSTRING) and length of string (STRLEN) values configured.	
Parameters	COMMAND DHCHANNEL DHDESTLINK DHFL DHNODE DHSRCLINK DONEFL ERRCODE ERRFL	ERRINFO FILENUM LASTRESPTM NSTRING ORDERINCM PATH READYFL SENDFL STR[0..7] STRLEN
Associated Block	"REQNUMARRAY (Request Number Array) block" on page 67 and "REQFLAGARRAY (Request Flag Array) block" on page 67.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REQTEXTARRAY Block.	

3.5.4 RSPFLAGARRAY (Response Flag Array) block

Description	The Flag Array Block is used to read or write an array of up to 512 BOOLEAN values.	
Function	The Response Array function blocks can be used to respond to requests from third-party devices using the PCCC or CIP protocols. These function blocks will be loaded to and run in the CPM, and will have the ability to have data values read and written by third-party devices.	
Inputs/Outputs	The Response Array Blocks are configured with the address used by the remote device to reference its data and the data size. The configuration information cannot be modified at run-time. If changes must be made to the configuration of a Response Array Block, these changes must be made in the Project Database and the block must be reloaded to the controller for the changes to take effect. These blocks are not internally triggered, data reads and writes occur as a result of external communications requests. They are also not Alarming Blocks and so cannot generate Alarms by themselves. No fragmentation or re-assembly is done either in the blocks themselves.	
Parameters	FILENUM NFLAG ORDERINCM	PVFL[0..511] PVVALSTS[0..511]
Associated Block	"RSPNUMARRAY (Response Number Array) block" on page 69 and "RSPTEXTARRAY (Response Text Array) block" on page 70.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RSPFLAGARRAY Block.	

3.5.5 RSPNUMARRAY (Response Number Array) block

Description	The Numeric Array Block is used to read or write an array of up to 64 integer or single precision float values.	
Function	The Response Array function blocks can be used to respond to requests from third-party devices using the PCCC or CIP protocols. These function blocks will be loaded to and run in the CPM, and will have the ability to have data values read and written by third-party devices.	
Inputs/Outputs	The Response Array Blocks are configured with the address used by the remote device to reference its data and the data size. The configuration information cannot be modified at run-time. If changes must be made to the configuration of a Response Array Block, these changes must be made in the Project Database and the block must be reloaded to the controller for the changes to take effect. These blocks are not internally triggered, data reads and writes occur as a result of external communications requests. They are also not Alarming Blocks and so cannot generate Alarms by themselves. No fragmentation or re-assembly is done either in the blocks themselves.	
Parameters	CIPNAME DATATYPE FILENUM NNUMERIC	ORDERINCM PV[0..63] PVSTS[0..63]
Associated Block	"RSPFLAGARRAY (Response Flag Array) block" on page 69 and "RSPTEXTARRAY (Response Text Array) block" on page 70.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RSPNUMARRAY Block.	

3.5.6 RSPTEXTARRAY (Response Text Array) block

Description	The Text Array Block is used to read or write an array of up to 64 ASCII characters.	
Function	The Response Array function blocks can be used to respond to requests from third-party devices using the PCCC or CIP protocols. These function blocks will be loaded to and run in the CPM, and will have the ability to have data values read and written by third-party devices.	
Inputs/Outputs	The Response Array Blocks are configured with the address used by the remote device to reference its data and the data size. The configuration information cannot be modified at run-time. If changes must be made to the configuration of a Response Array Block, these changes must be made in the Project Database and the block must be reloaded to the controller for the changes to take effect. These blocks are not internally triggered, data reads and writes occur as a result of external communications requests. They are also not Alarming Blocks and so cannot generate Alarms by themselves. No fragmentation or re-assembly is done either in the blocks themselves.	
Parameters	FILENUM NSTRING ORDERINCM	STR[0..7] STRLEN
Associated Block	"RSPFLAGARRAY (Response Flag Array) block" on page 69 and "RSPNUMARRAY (Response Number Array) block" on page 69.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RSPTEXTARRAY Block.	

3.6 HART DEVICE Block

Related topics

"HART DEVICE (Generic HART Device)" on page 71

3.6.1 HART DEVICE (Generic HART Device)

Description	Identifies the physical HART Device for the CPM to provide links to associated IOM.	
Function	Defines type of HART field device, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one point, as part of IOM configuration.	
Inputs	Real-time data transmission to or from configured IOC.	
Outputs	Real-time data transmission to or from physical device.	
Parameters	ASSOCIOMLOC ASSOCPROCDEF BINITIALIZE BLKASSOCSTATUS DATE DATEFORMAT DESCRIPTOR DEVICEIDNO DEVICELOC DEVICETYPE DEVREVNO DEVSPCBIT[0..135] DIGFV DIGFVDESC DIGFVUNITS DIGPV DIGPVDESC DIGPVUNITS DIGSV DIGSVDESC DIGSVUNITS DIGTV DIGTVDESC DIGTVUNITS FINALASSNO HARTCOMMCHNFAIL	HARTDEVSTATUS HARTFLAGS HARTREVNO HWREVNO IOMBLOCK IOMCHANNEL MANUFACTURER MESSAGE MODE NOREQUESTPREAMBLES PVRANGELOW PVRANGEHIGH PVRANGEUNITS SLOT0DDESC SLOT0UNITS SLOT0VALUE SLOT1DESC SLOT1UNITS SLOT1VALUE SLOT2DESC SLOT2UNITS SLOT2VALUE SLOT3DESC SLOT3UNITS SLOT3VALUE SWREVNO TAG

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.

3.7 Pulse Input Channel/Module Blocks

Related topics

"Pulse Input Channel with Fast Cutoff" on page 73
"Pulse Input Channel" on page 74
"Pulse Input Totalizer" on page 74
"Pulse Input Module Block TC-MDP081/TK-MDP081" on page 75
"Pulse Input Module Block CC-PPIX01" on page 76
"PI Channel" on page 77

3.7.1 Pulse Input Channel with Fast Cutoff

Description	Standard Interface to channels 6-7 of the Pulse Input IOM.	
Function	The three main functions of channels 6-7 are: • Pulse Count (PULSECOUNT) - both raw count and Engineering Unit converted values. • Frequency value (PV) in Engineering Units • Digital Output (fast cutoff)	
Inputs	If the Pulse Input Channel with Fast Cutoff is running it fetches AVRAW, AV, PV, TV and SO from its associated Pulse Input Module. If the associated IOM does not exist in the CPM, fail-safe values are set.	
Outputs	After Input Processing, the Pulse Input Channel with Fast Cutoff will send commands to the IOM function block. If multiple commands are sent to the device, the execution order in the Pulse Input firmware will be: 1. Reset Counter 2. Write Output Value 3. Write Target Value	
Parameters	AV AVRAW AVRAWSTS AVSTS BADCODE BADSO C1 C2 C3 DEBUG EDGEDETECT FETCHMODE FREEZETIME FREQPERIOD IOCNUMBER	IOCTYPE IOMCONN ORDERINCM PULSEMODE PV PVSTS PVVALSTS RESETFL SAFEOUTPUT SO SOCMDOFF SOCMDON TIMEBASE TV TVPROC VOLTAGE

Associated Block	"Pulse Input Module Block TC-MDP081/TK-MDP081" on page 75, "Pulse Input Channel" on page 74, and "Pulse Input Totalizer" on page 74.
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.

3.7.2 Pulse Input Channel

Description	Standard interface to channels 0-5 of the Pulse Input IOM function block.	
Function	The three main functions of channels 0-5 are: • Pulse count - both raw count and Engineering Unit converted values • Frequency value in Engineering Units • Pulse Length measurement in Engineering Units	
Inputs	If the PIC function block is running, it fetches AVRAW, AV, PV, PL and CHANSTS from its associated PIM function blocks. If the associated IOM does not exist in the CPM; fail-safe values are used.	
Outputs	If RESETFL is set, the PIC function block will pass this command to the associated PIM function block. RESETFL will then be reset regardless of whether the associated IOM exists.	
Parameters	AV	FREQPERIOD
	AVRAW	IOCTYPE
	AVRAWSTS	IOMCONN
	AVSTS	ORDERINCM
	BADCODE	PL
	C1	PLSTS
	C2	PULSEMODE
	C3	PV
	DEBUG	PVSTS
	EDGEDETECT	RESETFL
	FETCHMODE	TIMEBASE
	FREEZETIME	VOLTAGE
Associated Block	"Pulse Input Module Block TC-MDP081/TK-MDP081" on page 75, "Pulse Input Channel with Fast Cutoff" on page 73, and "Pulse Input Totalizer" on page 74. Prior to loading, block must be"associated" with 1 channel of corresponding PIM block that interfaces with the physical pulse input hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.7.3 Pulse Input Totalizer

Description	Periodically adds an input value (P1) to an accumulator value (PV). Sets status flags to indicate when accumulator value is"near", "nearer", "nearest" to the user-specified target value.
Function	Typically used to accumulate flows. For situations where flow transmitter may not be precisely calibrated near zero-flow value, a zero flow cutoff feature is provided such that when P1 is below the cutoff value it clamps to 0 (zero).
Inputs	One input (P1) is required: P1 is the value to be accumulated - input value must be an integer value. P1 must be fetched from another function block.

Outputs	The following outputs are produced: Accumulated value (PV) and its status (PVSTS), as well as a Boolean flag, PVSTSFL.BAD to indicate to other function blocks that this block's PV status is bad. Flags, indicating if accumulated value has reached user-specified target value or one of the accumulator deviation trip points (ACCTVFL and ACCDEV.FL[1..4])	
Equations	Parameter PVEQN may be configured to specify how the block should handle bad input and warm restarts. Specific handling combinations for a given PVEQN selection are listed in the Control Builder Components Theory document.	
Parameters	ACCDEV.FL[1..4]	PV
	ACCDEV.TP[1..4]	PVEQN
	ACCTV	PVFORMAT
	ACCTVFL	PVSTS
	C1	PVSTSFL.BAD
	C2	PVSTSFL.MAN
	CMDATTR	PVSTSFL.NORM
	COMMAND	PVSTSFL.UNCERTN
	CUTOFF.LM	PVVALSTS
	LASTGOOD	RESETFL
	OLDAV	RESETVAL
	ORDERINCM	STARTFL
	P1	STATE
	P1STS	STOPFL
Associated Block	“Pulse Input Module Block TC-MDP081/TK-MDP081” on page 75, “Pulse Input Channel” on page 74, and “Pulse Input Channel with Fast Cutoff” on page 73.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.7.4 Pulse Input Module Block TC-MDP081/TK-MDP081

Description	Defines the name/location and channel specifics for all Pulse Input Modules in the Control Processor.
Function	Supports the configuration of the TC-MDP081/TK-MDP081 Pulse Input module and acts as the interface between Pulse Input Channel blocks and controller's IO Manager.
Inputs	Every execution cycle the Pulse Input Module block gets the current assembly sent from the Pulse Input device using IO Manager services.
Outputs	Every 50 ms, the Pulse Input Module block sends assembly data to the Pulse Input device using IO Manager services.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO AV[0..7] AVRAW[0..7] BADSO[6..7] C1[0..7] C2[0..7] C3[0..5] CATNUMBER CEESTATE DESC DLCNBSLOT EDGEDETECT[0..7] ESTWEIGHT EXECSTATE EUDESC FREQPERIOD[0..7] INALM IOCTYPE IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV NUMCHANS NUMCONN	NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PL[0..5] PRODTYPE PULSEMODE[0..5] PV[0..7] RESETFL[0..7] SAFEOUTPUT[6..7] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SO[6..7] TIMEBASE[0..7] TV[6..7] TVPROC[6..7] TVRAW[6..7] ULCNBMAC VENDOR VOLTAGE[0..7]
Associated Block	"Pulse Input Totalizer" on page 74, "Pulse Input Channel" on page 74, and "Pulse Input Channel with Fast Cutoff" on page 73.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.7.5 Pulse Input Module Block CC-PPIX01

Description	Acts as an interface between the C300 Controller and pulsed output transducers such as tachometers, flow meters, and magnetic pickups.
Function	• Provides highly accurate frequency/period calculations of inputs for certain frequency ranges. • Supports pulse multiplexing that enables Prover pulses to be generated by copying the selected good pulses to the Prover pulse output. • Supports Dual Pulse Integrity in accordance with ISO6551:1996 Level A which is required to support interfacing of custody transfer meters with pulse outputs.
Inputs	Based on the configured module scan rate, the IOLINK collects all the process input data using the IOL Interface services.

Outputs	Data consumed by this module is sent through the IOL Interface as it is received.	
Parameters	ACTUALPROVERSIGNAL AVSTS AVRAWSTS BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHNLNAME CONFIGPROVERSIGNAL CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB CTRLCONFIRM DESC FWINVALIDA FWINVALIDB GROUP.NUMPARAMS HIST.NUMPARAMS IOLINK IOLINKCOLOR IOMBTREVA IOMCOMMAND IOMFWREVA IOMHWREVA IOMLHFSTA IOMNUM IOMOPERA IOMOPERB IOMPLD2REVA IOMSTATE	IOMSTSA IOMSTSB IOMTYPE IOPDESCA IOPLOCATION IOREDOPT MAINTAINONFAULT NUMCHANS NUMSIGS PARNERINCOMPATIBLEA PARNERINCOMPATIBLEB PLSTS PRIMARYSIG REASONSET REDDATAA REDDATTAB RESETFL RDNAUTOSYNC SCANCTRLLVL SCANGRPDTL SCANRATE SCANPNTDTL SCANASSOCDSP SECONDARYSIG SECSOGSECLVL SERIALNUMA TREND.NUMPARAMS TVRAW TYPEINVALIDA TYPEINVALIDB
Associated Block	“PI Channel” on page 77	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.7.6 PI Channel

Description	The PI channel block represents a single pulse input point on a Series C Pulse Input Module.
Function	• Pulse count - both raw count and Engineering Unit converted values. • Frequency calculation in Engineering Units. • Pulse Length measurement in Engineering Units. • Digital output (fast cutoff).

Inputs	• When the PI channel block is configured for pulse input type, it fetches AVRAW, AV, PV, PVSTS, and PL from its associated Pulse Input Module. • When the PI channel is configured for fast cutoff, it fetches AVRAW, AV, PV, PVSTS, TV, and SO from its associated Pulse Input Module. In both scenarios, if the associated Pulse Input Module does not exist in the C300 Controller, fail-safe values are set.	
Outputs	• When the PI channel is configured as pulse input type, if RESETFL is set, the PIC function block will pass this command to the associated PIM function block. RESETFL will then be reset regardless of whether the associated PIM exists or not. • When the PI channel is configured for fast cutoff, after input processing, the PI channel sends commands to the Pulse Input Module function block. If multiple commands are sent to the device, the execution order in the Pulse Input firmware is as follows: – Reset counter – Write output value – Write target value	
Parameters	ASSOCCHANNEL AV AVRAW BADAVRAW BADPVFL BADSO CHANNUM CONTAINEDIN C1 C2 C3 DEVICELOCATION EDGEDETECT ENPULSEWIDTHREJ FREQPERIOD INPUTSTREAM IOP	IOPTYPE PL PTEXECST PULSEINTYPE PULSEMODE PV PVSTS SO SAFEOUTPUT SOCMDOFF SOCMDON TV TVPROC VOLTAGE
Associated Block	“Pulse Input Module Block CC-PPIX01” on page 76	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.8 Logic Blocks

Related topics

"AND block" on page 80
"CHECKBAD block" on page 80
"CHECKBOOL block" on page 81
"CHGEXEC (Change Execution) block" on page 81
"CONTACTMON (Contact Monitoring) block" on page 82
"DELAY block" on page 83
"EQ (Equal) block" on page 83
"FTRIG (Falling-edge Trigger) block" on page 84
"GE (Greater than or Equal to) block" on page 84
"GT (Greater Than) block" on page 85
"LE (Less than or Equal to) block" on page 85
"LIMIT block" on page 86
"LT (Less Than) block" on page 86
"MAX block" on page 87
"MAXPULSE block" on page 87
"MIN block" on page 87
"MINPULSE block" on page 88
"MUX (Multiplexer) block" on page 88
"MUXREAL (Real Multiplexer) block" on page 89
"MVOTE (Majority Voting) block" on page 89
"NAND block" on page 89
"NE (Not Equal) block" on page 90
"nOON (n out of N voting) block" on page 91
"NOR block" on page 91
"NOT block" on page 92
"OFFDELAY block" on page 92
"ONDELAY block" on page 93
"OR block" on page 93
"PULSE block" on page 93
"QOR (Qualified OR) block" on page 94
"ROL (Rotate Output Left) block" on page 94
"ROR (Rotate Output Right) block" on page 95
"RS (Reset dominant SR-FLIP-FLOP) block" on page 95
"RTRIG (Rising edge Trigger) block" on page 95
"SEL (Binary Selection) block" on page 96
"SELREAL (Real Selection) block" on page 96
"SHL (Shift Output Left) block" on page 96
"SHR (Shift Output Right) block" on page 97
"SR (Set dominant SR-FLIP-FLOP) block" on page 97
"STARTSIGNAL block" on page 98
"TRIG (Rising or Falling edge Trigger) block" on page 98
"WATCHDOG block" on page 99
"XOR block" on page 99
"2OO3 (2 out of 3 voting) block" on page 100

3.8.1 AND block

Description	Provides an up to 8-input AND algorithm, meaning that it performs the Boolean operation of conjunction. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.		
Function	Turns the Boolean value output (OUT) ON only when all inputs (IN[1], IN[2], …, IN[8]) are ON. Therefore: • If all inputs (IN[1..8]) are ON, then: OUT = ON. • If any input (IN[x]) is OFF, then: OUT = OFF. If input is inverted, then: • Actual_IN[x] = NOT (IN[x]) • Else, Actual_IN[x] = IN[x] Where x equals any valid input.		
Truth Table	IN[1]	IN[2]	OUT
	OFF	OFF	OFF
	OFF	ON	OFF
	ON	OFF	OFF
	ON	ON	ON
Inputs	IN[1..8] = Boolean value		
Outputs	OUT = Boolean value controlled by the status of the input signals.		
Parameters	IN[1..8]		ORDERINCM
	INPTINVSTS[1..8]		OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the AND Block.		

3.8.2 CHECKBAD block

Description	Provides bad input handling for desired input.	
Function	Checks if input (IN) value equals NaN. • If IN = NaN • Then, OUT = ON • Else, OUT = OFF	
Inputs	IN = Real number	
Outputs	OUT = Boolean value	
Parameters	IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the CHECKBAD Block.	

3.8.3 CHECKBOOL block

Description	Evaluates the input connections and passes these input values through to its associated outputs based on specific configuration settings.	

Function	Determines the action to be taken in the event of an invalid input. If the value of INSTS[1..8] is kBadValSts, the value passed through the block, from IN[1..8] to OUT[1..8], will be modified based on the configuration of the BADINACT[1..8] parameter.
	Also, the Inactive Input Detection Threshold, (INACTINDETTM[n]) parameter is used, in conjunction with BADINACT[n] as the amount of time that must expire before the block determines if it should take the configured Bad Input Action. During this detection time, the inputs status must be continually INACTIVE in order for the action to be taken. When the input is INACTIVE for less than this time, no action is taken. If the input goes INACTIVE again, the time starts counting over. This time is configured in seconds and has a range of 0-8000 seconds.
	`If BADINACT is configured as OFF then OUT[1..8] is set equal to OFF`
	`If BADINACT is configured as ON then OUT[1..8] is set equal to ON`
	`If BADINACT is configured as HoldLast then OUT[1..8] is set equal to LASTIN[1..8]`	

Inputs	IN = Boolean value	
Outputs	OUT = Boolean value	
Parameters	IN[1..8]	BADINACT[1..8]
	INACTINDETTM[1..8]	BADINDETTM[1..8]
	INSTS[1..8]	BADINACTMINTM[1..8]
	INSRC[1..8]	OUT[1..8]
	LASTIN[1..8]	OUTSTS[1..8]
	LASTINSTS[1..8]	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the CHECKBOOL Block.	

3.8.4 CHGEXEC (Change Execution) block

Attention

Refer to the *Control Builder Components Theory* for the list of function blocks qualified to run under change driven execution.

Description	The CHGEXEC block helps in optimizing the performance of control modules (CM) used exclusively for logic computation. CHGEXEC is used to create CM logic strategies which execute most of their logic by exception, thereby reducing the average processing power consumed by the strategy. It runs within the CEE on the C300 controller, C200E controller, and ACE controller.

Function	The CHGEXEC block enables the design of change driven logic strategies within CMs. Some of the key functionalities of the CHGEXEC block are as follows: • Supports up to 32 change detected Boolean inputs. • Supports output of captured inputs to downstream blocks to prevent consumption of inconsistent data during CM execution. • Supports a cascade output that allows operation of multiple CHGEXEC instances to be coordinated. • Supports a slow, periodic, background execution for convenience in strategy design. • Supports engineer-only test options which allow strategy designers to force it to suspend background execution or to execute continuously in the absence of input changes.	
Inputs	DATA: Supports 32 change detected Boolean inputs.	
Outputs	• Supports output of captured inputs to downstream blocks to prevent consumption of inconsistent data during CM execution. • Supports a cascade output that allows operation of multiple CHGEXEC instances to be coordinated.	
Parameters	AUTOPERIOD AUTOPHASE BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHGINDEX	DATA[1..32] EXITOPT LASTDATA[1..32] NUMDATA TESTOPT TRIGGER
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Components Theory* for more information on the CHGEXEC Block.	

3.8.5 CONTACTMON (Contact Monitoring) block

> **Attention**
> The CONTACTMON block can only be used with C300, C200E, and ACE Controllers.

Description	The Contact Monitoring function block is used for limit switches with NO and NC Contact with the same activation mechanism. In most cases, the limit switch serves some critical applications. When both digital inputs are ON or OFF at the same time indicating that the switch is malfunctioning, an alarm needs to be generated to attract immediate attention to the problem. This functionality is achieved using the Contact Monitoring function block.
Function	• Enables alarm generation whenever the state of both inputs is same or different based on the normal state configuration. • Provides the input switch status as OUT1 and OUT2. • With R410, CONTACTMON block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.
Inputs	IN[1..2] - Input parameter corresponding to the NO and NC Contact of the same micro switch
Outputs	• PVFL - gives the negated XOR results of IN[1] and IN[2]. • OUT1 - Tracks IN[1] when both inputs of COC are not ON. • OUT2 - Tracks IN[2] when both inputs of COC are not ON.

Parameters	HIALM.PR HIALM.SV HIALM.TYPE IN[1,2] INALM NORMAL OFFNRMALM.FL OFFNRMALM.SV OFFNRMALM.PR	OFFNRMALM.TM OFFNRMALM.TMO PV PVFL STATE0 OUT1 OUT2 STATE1 STATETEXT[0..1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the CONTACTMON Block.	

3.8.6 DELAY block

Description	Provides the ability to delay the output (OUT) response to the given input (IN) by one sample time delay.	
Function	The OUT always follows the input (IN) action after one sample time delay.	
Inputs	IN = Boolean value	
Outputs	OUT = Boolean value	
Parameters	IN ORDERINCM	OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the DELAY Block.	

3.8.7 EQ (Equal) block

Description	Provides a 2-input Compare Equal (with deadband range) function, meaning that it compares two inputs for equality within a specified deadband range or, for single input, a designated trip point (TP) parameter.
Function	Turns the digital output (OUT) ON only when the two inputs (IN[1] and IN[2]) are considered equal within a specified deadband range or, for single input, a designated trip point (TP) parameter..
Inputs	IN[1..2] = real numbers • If only 1 input connection is configured, an input port is displayed for parameter TP and the value of TP is used instead of IN[2]. • If IN[1] and/or IN[2] are NaN (Not a Number), OUT = INBADOPT. • DEADBAND1, DEADBAND2 and TP have the same data types as the inputs. • DEADBAND1 and DEADBAND2 must satisfy this constraint: 0 <= DEADBAND1 <= DEADBAND2
Outputs	OUT = Boolean value controlled by the status of the input signals. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.

Parameters	DEADBAND1 DEADBAND2 IN[0..2] INBADOPT	NUMOFINPUTS ORDERINCM OUT TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the EQ Block.	

3.8.8 FTRIG (Falling-edge Trigger) block

Description	Falling-edge Trigger Block -- sets the output (OUT) to ON following the ON-to-OFF transition of the input and stays ON until the next execution cycle, at which time it returns to OFF.	
Function	Provides falling edge change detection, thereby turning the output ON if an ON-to-OFF transition is detected.	
Input	IN = Boolean value	
Output	OUT = Boolean value	
Parameters	IN	ORDERINCM OUT
Reference	Refer to the Control *Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the FTRIG Block.	

3.8.9 GE (Greater than or Equal to) block

Description	Provides a 2-input Compare Greater Than or Equal (with deadband) function, meaning it checks to see if one designated input (IN[1]) is greater than or equal to either a second input (IN[2]) or, for single input, a designated trip point parameter.	
Function	Turns the digital output (OUT) ON only when one designated input (IN[1]) is greater than or equal to a second input (IN[2]) or, for single input, a designated trip point parameter (TP) as follows: • If IN[1] >= IN[2], then: OUT = ON. • If IN[1] < (IN[2] - DEADBAND), then: OUT = OFF. • If (IN[2] - DEADBAND) < IN[1] < IN[2], then output is not changed.	
Inputs	IN[1..2] = Real numbers • If only one input connection is configured, an input port is displayed for parameter TP and the value of TP is used instead of IN[2]. • If IN[1] and/or IN[2] are NaN (Not a Number), OUT is set to INBADOPT. • DEADBAND and TP have the same data type as that of the inputs.	
Output	OUT = Boolean value controlled by the status of the input signals. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.	
Parameters	DEADBAND IN[0..2] INBADOPT NUMOFINPUTS	ORDERINCM OUT TP

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the GE Block.

3.8.10 GT (Greater Than) block

Description	Provides a 1- or 2-input Compare Greater Than (with deadband) function, meaning that it checks to see if one designated input (IN[1]) is greater than either a second input (IN[2]) or, for single input, a designated trip point parameter (TP).	
Function	Turns the digital output (OUT) ON only when one designated input (IN[1]) is greater than a second input (IN[2]) or, for single input, a designated trip point parameter (TP) as follows: • If IN[1] > IN[2], then: OUT = ON. • If IN[1] <= (IN[2] - DEADBAND), then: OUT = OFF. • If (IN[2] - DEADBAND) < IN[1] <= IN[2], then: OUT is not changed.	
Input	IN[1..2] = Real numbers • If only one input connection is configured, an input port is displayed for parameter TP and the value of TP is used instead of IN[2]. • If IN[1] and/or IN[2]are NaN (Not a Number), OUT is set to INBADOPT. • DEADBAND and TP have the same data type as that of the inputs.	
Outputs	OUT = Boolean value controlled by the status of the input signals. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.	
Parameters	DEADBAND IN[0..2] INBADOPT NUMOFINPUTS	ORDERINCM OUT TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the GT Block.	

3.8.11 LE (Less than or Equal to) block

Description	Provides a 2-input Compare Less Than or Equal (with deadband) function, meaning it checks to see if one designated input (IN[1]) is less than or equal to either a second input (IN[2]) or, for single input, a designated trip point parameter (TP).
Function	Turns the digital output (OUT) ON only when one designated input (IN[1]) is less than or equal to a second input (IN[2]) or, for single input, a designated trip point parameter (TP) as follows: • If IN[1] <= IN[2], then: OUT = ON. • If IN[1] > (IN[2] + DEADBAND), then: OUT = OFF. • If IN[2] < IN[1] <= (IN[2] + DEADBAND), then: output is not changed.
Inputs	IN[1..2] = Real numbers • If only one input connection is configured, an input port is displayed for parameter TP and the value of TP is used instead of IN[2]. • If IN[1] and/or IN[2] are NaN (Not a Number), OUT is set to INBADOPT. • DEADBAND and TP have the same data type as that of the inputs.

Outputs	OUT = Boolean value controlled by the status of the input signals. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.
Parameters	DEADBAND IN[0..2] INBADOPT NUMOFINPUTS ORDERINCM OUT TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LE Block

3.8.12 LIMIT block

Description	Provides a 3-input limit function, meaning that it provides an output that is maintained within a specified range as defined by user-specified minimum and maximum values.
Function	Provides an output that is maintained within a specified range as follows: • MIN <= OUT <= MAX • If IN = NaN, then, OUT = NaN
Inputs	IN = real number
Output	OUT = real number maintained within a specified range
Parameters	IN MAX MIN ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LIMIT Block..

3.8.13 LT (Less Than) block

Description	Provides a 1- or 2-input Compare Less Than (with deadband) function, meaning that it checks to see if one designated input (IN[1]) is less than either a second input (IN[2]) or, for single input, a designated trip point parameter (TP).
Function	Turns the digital output (OUT) ON only when one designated input (IN[1]) is less than a second input (IN[2]) or, for single input, a designated trip point parameter (TP) as follows: • If IN[1] < IN[2], then: OUT = ON. • If IN[1] >= (IN[2] + DEADBAND), then: OUT = OFF. • If IN[2] <= IN[1] < (IN[2] + DEADBAND), then: OUT is not changed.
Inputs	IN[1..2] = Real numbers • If only one input connection is configured, an input port is displayed for parameter TP and the value of TP is used instead of IN[2]. • If IN[1] and/or IN[2] are NaN (Not a Number), OUT is set to INBADOPT. • DEADBAND and TP have the same data type as that of the inputs.
Outputs	OUT = Boolean value controlled by the status of the input signals. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.

Parameters	DEADBAND IN[0..2] INBADOPT NUMOFINPUTS	ORDERINCM OUT TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LT Block.	

3.8.14 MAX block

Description	Provides an 8-input MAX function, meaning that it provides an output that is the maximum value of eight inputs.	
Function	Used to isolate the highest value of multiple input values and use it as a designated output value. This block ignores NaN inputs.	
Inputs	IN[1..8] = Real numbers	
Output	OUT = Real number	
Parameters	EUDESC HIALM IN[1..8] INPTINVSTS[1..8]	NUMOFINPUTS ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MAX Block.	

3.8.15 MAXPULSE block

Description	Provides a maximum time limit pulse output (OUT) each time the input (IN) transitions from OFF to ON. You specify the maximum output pulse width (PULSEWIDTH) in seconds through configuration.	
Function	Used to limit the output (OUT) pulse to a maximum width. • If the input (IN) pulse time is less than or equal to the specified PULSEWIDTH time, IN is assumed to equal one output (OUT) pulse. • If the IN pulse time is greater than the specified PULSEWIDTH time, OUT pulse terminates at end of specified PULSEWIDTH time.	
Inputs	IN = Boolean value	
Output	OUT = Boolean value	
Parameters	IN ORDERINCM	OUT PULSEWIDTH
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MAXPULSE Block.	

3.8.16 MIN block

Description	Provides an 8-input MIN function, meaning that it provides an output that is the minimum value of eight inputs.

Function	Used to isolate the lowest value of multiple input values and use it as a designated output value. This block ignores NaN inputs.	
Inputs	IN[..8] = Real numbers	
Output	OUT = Real number	
Parameters	IN[1..8] NUMOFINPUTS	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MIN Block.	

3.8.17 MINPULSE block

Description	Provides a minimum time limit pulse output (OUT) each time the input (IN) transitions from OFF to ON. You specify the minimum output pulse width (PULSEWIDTH) in seconds through configuration.	
Function	Used to define the minimum output (OUT) pulse width. • If the input (IN) pulse time is less than or equal to the specified PULSEWIDTH time, output (OUT) pulse width equals the specified PULSEWIDTH time. • If the IN pulse time is greater than the specified PULSEWIDTH time, OUT pulse width tracks IN pulse time, so OUT pulse exceeds specified PULSEWIDTH time.	
Inputs	IN = Boolean value	
Output	OUT = Boolean value	
Parameters	IN ORDERINCM	OUT PULSEWIDTH
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MINPULSE Block.	

3.8.18 MUX (Multiplexer) block

Description	Provides an up to 8-input Extensible Multiplexer algorithm, meaning that it selects 1 of “n” inputs depending on a separate input K.	
Function	Sets the actual output (OUT) to a particular input (IN[1], IN[2], …, IN[8]) depending on the value of a separate input K. Input K is clamped at 0 and 7.	
Truth Table	*K* 0 1 n-1	*OUT* IN1 IN2 INn
Inputs	IN[1..8] = Boolean value K = 8-bit unsigned integer.	
Output	OUT = Boolean value	
Parameters	IN[1..8] K	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MUX Block.	

3.8.19 MUXREAL (Real Multiplexer) block

Description	Provides an up to 8-input real Multiplexer algorithm, meaning that it selects 1 of“n” inputs depending on a separate input K.	
Function	Sets the actual output (OUT) to a particular input (IN[1], IN[2], …, IN[8]) depending on the value of a separate input K. Input K is clamped at 0 and 7.	
Truth Table	*K* 0 1 n-1	*OUT* IN1 IN2 INn
Inputs	IN[..8] = Real numbers K = 8-bit unsigned integer.	
Outputs	OUT = real number	
Parameters	IN[1..8] K	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MUXREAL Block.	

3.8.20 MVOTE (Majority Voting) block

Description	Provides an output (MAJ) value that equals the value of the majority of the inputs (IN[1..8]) and sets another output (DISCREP) to ON if not all inputs agree for a specified time (DELAY). You specify the time (DELAYTIME) in seconds through configuration. You must also specify the number of inputs (NUMOFINPUTS) through configuration.	
Function	Sets the MAJ output equal to the value of the majority of the inputs (IN[1..8]). Sets the DISCREP output to ON, if not all inputs agree during the specified time (DELAY). DELAY is a unit integer with time unit in seconds.	
Inputs	IN[1..8] = Boolean value.	
Outputs	MAJ, DISCREP =Boolean value	
Parameters	DELAYTIME DISCREP IN[1..8]	MAJ NUMOFINPUTS ORDERINCM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MVOTE Block.	

3.8.21 NAND block

Description	Provides an up to 8-input NAND algorithm, meaning that it performs an inverted AND function. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.
Function	Turns the digital output (OUT) OFF only when all inputs (IN[1], IN[2], …, IN[8]) are ON; therefore: • If all inputs are ON, then: OUT = OFF. • If any input is OFF, then: OUT = ON.

Truth Table	*IN[1]*	*IN[2]*	*OUT*
	OFF	OFF	ON
	OFF	ON	ON
	ON	OFF	ON
	ON	ON	OFF
Inputs	IN[1], IN[2], …, IN[8] = digital signals		
Output	OUT = digital signal controlled by status of the input signals.		
Parameters	IN[1..8] INPTINVSTS[1..8]	ORDERINCM OUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the NAND Block.		

3.8.22 NE (Not Equal) block

Description	Provides a 2-input Compare Not Equal (with deadband range) function, meaning that it checks to see if one designated input (IN[1]) is not equal to either a second input (IN[2]) or, for single input, a designated trip point parameter (TP).	
Function	Turns the digital output (OUT) ON only when the two inputs (IN[1] and IN[2]) are not considered equal within a specified deadband range. • If ABS (IN[1] -IN[2]) <= DEADBAND1, then: OUT = OFF. • Else, if ABS (IN[1] -IN[2]) > DEADBAND2, then: OUT = ON. • If IN[1] and/or IN[2] are NaN (Not a Number), OUT is set to INBADOPT. • DEADBAND1 and DEADBAND2 must satisfy the following constraint: 0<= DEADBAND1 <= DEADBAND2. • DEADBAND1, DEADBAND2, and TP = real numbers.	
Inputs	IN[1] and IN[2] = real numbers • If there is only one input, then IN[2] = TP.	
Output	OUT = Boolean value. Comparison blocks set their outputs to a configurable INBADOPT that defines the output fail-safe value when any input is NaN. This is required, since it is not specified whether the comparison is ordered or unordered.	
Parameters	DEADBAND1 DEADBAND2 IN[0..2] INBADOPT	NUMOFINPUTS ORDERINCM OUT TP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the NE Block.	

3.8.23 nOON (n out of N voting) block

Description	n (N)-out-of-N (IN) voting block; outputs are computed as follows: • VOTED output is set to ON if at least n (N) inputs are ON, otherwise it is set to OFF. • ORED output is set to ON if any input is ON, otherwise it is set to OFF. • ALARM output is a pulse output -- every time an input turns ON, a fixed pulse (of the pulsewidth specified by PULSEWIDTH parameter) is generated, provided the total number of inputs which are ON is less than n.	
Function	Provides VOTED, ORED and ALARM outputs in support of logical functions.	
Inputs	IN[1..20] = Boolean value N = 8-bit unsigned integer (range = 1-5) • There can be a maximum of 20 inputs (IN = 20)	
Outputs	• VOTED, ORED = Boolean state (ON or OFF) as determined by the inputs. • ALARM = pulse output, width specified by parameter PULSEWIDTH. PULSEWIDTH is a unit integer with time unit in seconds	
Parameters	ALARM IN[1..20] N	ORDERINCM ORED PULSEWIDTH VOTED
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the nOON Block.	

3.8.24 NOR block

Description	Provides an up to 8-input NOR algorithm, meaning that it performs an inverted OR function. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.		
Function	Turns the digital output (OUT) OFF if any one input (IN[1], IN[2], …, IN[8]) is ON; therefore: • If all inputs are OFF, then: OUT = ON. • If any one input is ON, then: OUT = OFF.		
Truth Table	*IN[1]*	*IN[2]*	*OUT*
	OFF	OFF	ON
	OFF	ON	OFF
	ON	OFF	OFF
	ON	ON	OFF
Inputs	IN[1..8] = Boolean values		
Outputs	OUT = Boolean value controlled by status of input signals		
Parameters	INPTINVSTS[1..8]	ORDERINCM OUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the NOR Block.		

3.8.25 NOT block

Description	Provides a NOT algorithm, meaning it performs an inversion function.	
Function	Reverses the state of a digital input (IN) such that the output (OUT) is the complement of the single input; therefore: • OUT = opposite of IN – If IN = ON, then: OUT = OFF. – If IN = OFF, then OUT = ON.	
Truth Table	*IN* OFF ON	*OUT* ON OFF
Input	IN = Boolean value	
Output	OUT = complement of input signal (Boolean)	
Parameters	EUDESC HIALM IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the NOT Block.	

3.8.26 OFFDELAY block

Description	When the input state changes from ON to OFF, an internal timer starts counting down the delay specified by DLYTIME. When it times out, the input is monitored again, and if it is still OFF, the output is set OFF, When the input state transitions too ON, the output is set to ON immediately and the timer is shut off.	
Function	Used to delay the input by a specified delay time after an ON/OFF device transitions from the ON state to the OFF state. • Delay time in seconds is specified by the DELAYTIME parameter.	
Inputs	IN = Boolean value • No delay is provided when the input goes from the OFF state back to the ON state.	
Outputs	OUT = Boolean value When the input transitions from the OFF state to the ON state, the output is set to ON immediately.	
Parameters	DELAYTIME IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the OFFDELAY Block.	

3.8.27 ONDELAY block

Description	When the input state changes from OFF to ON, an internal timer starts counting down the delay specified by DLYTIME. When it times out, the input is monitored again, and if it is still ON, the output is set ON, When the input state transitions to OFF, the output is set to OFF immediately and the timer is shut off.	
Function	Used to delay the input by a specified delay time after an ON/OFF device transitions from the OFF state to the ON state. • Delay time in seconds is specified by the DELAYTIME parameter.	
Input	IN = Boolean value • No delay is provided when the input goes from the ON state back to the OFF state.	
Output	OUT = Boolean value • When the input transitions from the ON state to the OFF state, the output is set to OFF immediately.	
Parameters	DELAYTIME IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ONDELAY Block.	

3.8.28 OR block

Description	Provides an up to 8-input OR algorithm, meaning that it performs the inclusive OR Boolean function. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.		
Function	Turns the digital output (OUT) ON if any one input (IN[1], IN[2], …, IN[8]) is ON; therefore: • If all inputs are OFF, then: OUT = OFF. • If any one input is ON, then: OUT = ON.		
Truth Table	*IN[1]*	*IN[2]*	*OUT*
	OFF	OFF	OFF
	ON	OFF	ON
	OFF	ON	ON
	ON	ON	ON
Inputs	IN[1..8] = Boolean value		
Output	OUT = Boolean value controlled by the status of input signals.		
Parameters	IN[1..8] INPTINVSTS[1..8]	ORDERINCM OUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the OR Block.		

3.8.29 PULSE block

Description	Provides a fixed pulse output (OUT) each time the input (IN) transitions from OFF to ON. You specify the fixed output pulse width (PULSEWIDTH) in seconds through configuration.

Function	Used to define the fixed output (OUT) pulse width. • If the input (IN) pulse time is less than or equal to the fixed PULSEWIDTH time, output (OUT) pulse width equals the fixed PULSEWIDTH time. • If the IN pulse time is greater than the fixed PULSEWIDTH time, OUT pulse width is restricted to the fixed PULSEWIDTH time. Another output pulse cannot be generated until the preceding pulse has completed.	
Inputs	IN = Boolean value	
Output	OUT = Boolean value	
Parameters	IN ORDERINCM	OUT PULSEWIDTH
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the PULSE Block.	

3.8.30 QOR (Qualified OR) block

Description	Qualified-OR provides an (N + 1)-input generic qualified-OR function, meaning that the output (OUT) is turned ON if a certain number (k) of total inputs (IN[n]) is ON. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.						
Function	Turns the output (OUT) ON if a specified number (K) of total inputs is ON.						
Truth Table	*IN[1]* ON	*IN[2]* ON	*IN[3]* OFF	*IN[4]* ON	*IN[5]* OFF	*K* 3	*OUT* ON
Inputs	IN[1..8] = Boolean value K = 1 to 8 (Integer)						
Outputs	OUT = Boolean value controlled by status of input signals.						
Parameters	IN[1..8] INPTINVSTS[1..8]	K ORDERINCM OUT					
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the QOR Block.						

3.8.31 ROL (Rotate Output Left) block

Description	Provides a 16-bit integer output (OUT) that is rotated to the left by the number of bits (N) specified from the 16-bit integer input (IN). You specify the number of bits through configuration.	
Function	Used to shift out bits in the output (OUT) by rotating the bits in the input (IN) left by the number of bits (N) specified. • OUT = IN left rotated by N bits, circular.	
Inputs	IN = 16-bit integer only	
Output	OUT = 16-bit integer	
Parameters	IN N	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ROL Block.	

3.8.32 ROR (Rotate Output Right) block

Description	Provides a 16-bit integer output (OUT) that is rotated to the right by the number of bits (N) specified from the 16-bit integer input (IN). You specify the number of bits through configuration.	
Function	Used to shift out bits in the output (OUT) by rotating the bits in the input (IN) right by the number of bits (N) specified. • OUT = IN right rotated by N bits, circular.	
Inputs	IN = 16-bit integer only	
Output	OUT = 16-bit integer	
Parameters	IN N	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ROR Block.	

3.8.33 RS (Reset dominant SR-FLIP-FLOP) block

Description	Provides a bistable Reset Dominant flip-flop as defined in the IEC DIS 1131-3 standard.		
Function	Specifies the output (Q) of the flip-flop as a function of the input S (Set), the input R (Reset), and the last state of Q.		
Truth Table	*S*	*R*	*Q*
	0 (OFF)	0 (OFF)	No Change
	0 (OFF)	1 (ON)	0 (OFF)
	1 (ON)	0 (OFF)	1 (ON)
	1 (ON)	1 (ON)	0 (OFF)
Inputs	S and R = Boolean value		
Output	Q = Boolean value controlled by the status of the input signals.		
Parameters	ORDERINCM Q	R S	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the RS Block.		

3.8.34 RTRIG (Rising edge Trigger) block

Description	Rising-edge Trigger sets the output (OUT) to ON following the OFF-to-ON transition of the input (IN) and stays at ON until the next execution cycle, at which time it returns to OFF.	
Function	Provides rising edge change detection, thereby turning the output ON if an OFF-to-ON transition is detected.	
Input	IN = Boolean value	
Output	OUT = Boolean value	
Parameters	IN	ORDERINCM OUT

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the RTRIG Block.

3.8.35 SEL (Binary Selection) block

Description	Provides a 3-input selector function, meaning it selects 1 of 2 inputs (IN[1] or IN[2]) depending on the separate input G.			
Function	Sets the actual output (OUT) equal to the value of 1 of 2 inputs (IN[1] or IN[2]), depending on the value of a separate input (G).			
Truth Table	*IN[1]*	*IN[2]*	*G*	*OUT*
	IN[1]	IN[2]	OFF	IN[1]
	IN[1]	IN[2]	ON	IN[2]
Inputs	IN[1..2] =Boolean value G = Boolean value			
Output	OUT =Boolean value depending on the values of IN[1] and IN[2].			
Parameters	G IN[0..2]		INPTINVSTS[1..2] ORDERINCM OUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SEL Block.			

3.8.36 SELREAL (Real Selection) block

Description	Provides a 3-input selector function, meaning it selects 1 of 2 inputs (IN[1] or IN[2]) depending on the separate input (G).			
Function	Sets the actual output (OUT) equal to the value of 1 of 2 inputs (IN[1] or IN[2]), depending on the value of a separate input (G).			
Truth Table	*IN[1]*	*IN[2]*	*G*	*OUT*
	IN[1]	IN[2]	OFF	IN[1]
	IN[1]	IN[2]	ON	IN[2]
Inputs	IN_1 and IN_2 = real numbers G = Boolean value			
Output	OUT = Real number			
Parameters	G IN[0..2]		ORDERINCM OUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SELREAL Block.			

3.8.37 SHL (Shift Output Left) block

Description	Provides a 16-bit integer output (OUT) that is shifted to the left by the number of bits (N) specified from the 16-bit integer input (IN). You specify the number of bits (N) through configuration.

Function	Used to shift out bits in the output (OUT) by shifting the bits in the input (IN) left by the number of bits (N) specified. • OUT = IN left shifted by N bits, zero filled on right.	
Inputs	IN = 16-bit integer only	
Output	OUT = 16-bit integer	
Parameters	IN N	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SHL Block.	

3.8.38 SHR (Shift Output Right) block

Description	Provides a 16-bit integer output (OUT) that is shifted to the right by the number of bits (N) specified from the 16-bit integer input (IN). You specify the number of bits through configuration.	
Function	Used to shift out bits in the output (OUT) by shifting the bits in the input (IN) right by the number of bits (N) specified. • OUT = IN right shifted by N bits, zero filled on left.	
Inputs	IN = 16-bit integer only	
Output	OUT = 16-bit integer	
Parameters	IN N	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SHR Block.	

3.8.39 SR (Set dominant SR-FLIP-FLOP) block

Description	Provides a bistable Set Dominant flip-flop as defined in the IEC DIS 1131-3 standard.		
Function	Specifies the output (Q) of the flip-flop as a function of the input S (set), the input R (Reset), and the last state of Q.		
Truth Table	*S*	*R*	*Q*
	0 (OFF)	0 (OFF)	No Change
	0 (OFF)	1 (ON)	0 (OFF)
	1 (ON)	0 (OFF)	1 (ON)
	1 (ON)	1 (ON)	1 (ON)
Inputs	S and R = Boolean values		
Outputs	Q = Boolean value controlled by the status of the input signals.		
Parameters	ORDERINCM PVERSION	Q R S	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SR Block.		

3.8.40 STARTSIGNAL block

Description	Supports handling of restarts within Control Modules (CM). Can be used within any CM to provide better control over how the module initializes in response to events such as Cold or Warm restart.	
Function	Supports an enumeration-valued summary parameter named RESTART. The normal value for the RESTART parameter is NONE. Following a transition, it shows a value other than NONE until the end of the first block execution. The possible enumeration values for RESTART are as follows: NONE (0) CMLOAD (1) CMACTIVE (3) CEECOLD (4) CEEWARM (5) CEESWITCH (6)	
Inputs	Boolean value	
Outputs	RESTART = Enumerated value	
Parameters	ANYRESTARTFL CEECOLDFL CEESWITCHFL CEEWARMFL	CMACTIVEFL CMLOADFL ORDERINCM RESTART
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the STARTSIGNAL Block.	

3.8.41 TRIG (Rising or Falling edge Trigger) block

Description	Sets the output (OUT) to ON following the OFF-to-ON or ON-to-OFF transition of the input (IN) and stays at ON until the next execution cycle, at which time it returns to OFF.	
Function	Provides edge change detection, thereby turning the output ON if an OFF-to-ON or ON-to-OFF transition is detected. This block assumes that the input is starting at its OFF stage the first time it is activated.	
Inputs	IN = Boolean value	
Outputs	OUT = Boolean value	
Parameters	IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the TRIG Block.	

3.8.42 WATCHDOG block

Description	Monitors other system functions or remote devices and sets the output (OUT) to ON if the monitored function or device fails to communicate. Example: Applications like TotalPlant Batch uses the Watchdog item to determine the communication between TotalPlant Batch server (TPB) and Experion controllers. **Note** TotalPlant Batch (TPB) is a legacy batch product that is not available for new sale. However, Honeywell continues to support the existing TotalPlant Batch (TPB) customer. New batch customers are recommended to buy the Experion Batch Manager (EBM). Watchdog also provides a Watchdog protocol in Experion, which is simple set/reset protocol. The application such as TPB server sets the Watch dog by writing one to the watchdog item. The Experion controller accessed through the data server is referred as TPB Phase Data Interface on Experion. It is expected to reset the watchdog to zero when a non-zero value is detected. Therefore, when you reset the watch dog values, TPB server can detect the presence of a healthy controller. However, the controller can detect the presence of a healthy TPB server by periodic setting of the watchdog. If the controller determines that the TPB server has not written to the watchdog for a long period of time, then it assumes that the TPB server is no longer available and responds by holding all active phases within. If the TPB server detects that the controller does not reset the watchdog, it assumes that communication to all tags defined in the data server connection is not reliable. Batches using any of these tags is in response when the communication problems that occur.	
Function	Used to monitor other system functions or remote devices. • Monitored function or device must set IN parameter to ON within a specified time interval (DELAYTIME), otherwise it is assumed to have failed and output (OUT) is set to ON. The DELAYTIME is an integer with unit time in seconds. • If output (OUT) is ON, it is reset to OFF as soon as IN is set to ON.	
Input	IN = Boolean value (ON/OFF)	
Output	OUT = Boolean value (ON/OFF)	
Parameters	DELAYTIME IN	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the WATCHDOG Block.	

3.8.43 XOR block

Description	Provides an up to 8-input XOR algorithm, meaning it performs the exclusive OR function. Each input (IN[1], IN[2], …, IN[8]) has the capability of being optionally inverted, if required.		
Function	Turns output (OUT) ON only if an odd number of inputs are ON; otherwise, OUT is OFF.		
Truth Table	*IN[1]*	*IN[2]*	*OUT*
	OFF	OFF	OFF
	ON	OFF	ON
	OFF	ON	ON
	ON	ON	OFF
Inputs	IN[1..8] = Boolean value		

Outputs	OUT = Boolean value controlled by the status of input signals.	
Parameters	IN[1..8] INPTINVSTS[1..8]	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory*for more information on the XOR Block.	

3.8.44 2OO3 (2 out of 3 voting) block

Description	2-out-of-3 Voting block - outputs (DISCREP and MAJ) are determined as follows: • DISCREP = NOT (IN[1] = IN[2] = IN[3]) for duration >= DELAY • MAJ = value held by the majority of the inputs.	
Function	Sets the output (DISCREP) to ON if NOT all inputs agree for a specified time duration (DELAY); otherwise, it is set to OFF.	
Inputs	IN[1..3] = Boolean values	
Outputs	DISCREP & MAJ = Boolean values	
Parameters	DELAYTIME DISCREP	MAJ ORDERINCM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the 2003 Block.	

3.9 Math Blocks

Related topics

3.9.1 ABS block

Description	Provides the Absolute Value function.	
Function	At runtime, the output (OUT) becomes the absolute value of the user connected input (IN[1]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = Absolute value of IN1	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ABS Block.	

3.9.2 ADD block

Description	Provides the N-input add function.

Function	Sums the value of the number of inputs (IN[1..8]) configured by the user as the output (OUT). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..8] = Input connection value	
Output	OUT = Sum of (IN_1, $1N_2$, ..IN_n	
Parameters	IN[1..8] NAME NUMOFINPUTS	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ADD Block.	

3.9.3 DIV block

Description	Provides the divide function.	
Function	At runtime, the output (OUT) becomes the quotient of input 1 (IN[1]) divided by input 2 (IN[2]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..2] = Input connection value	
Output	OUT = Quotient of IN_1 divided by IN_2.	
Parameters	IN[1..2] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the DIV Block.	

3.9.4 EXP block

Description	Provides the exponent function.
Function	At runtime, the output (OUT) becomes the exponent of the user connected input (IN[1]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.
Input	IN[1] = Input connection value

Output	OUT = e to the power of $IN_{1.}$	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the EXP Block.	

3.9.5 LN block

Description	Provides the natural logarithm function.	
Function	At runtime, the output (OUT) becomes the natural logarithm of the user connected input (IN[1], log to the base of e). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = Natural logarithm of IN_1 (log to the base of e)	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LN Block.	

3.9.6 LOG block

Description	Provides the logarithm function.	
Function	At runtime, the output (OUT) becomes the base 10 logarithm of the user connected input (IN[1]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = Base 10 logarithm of $IN_{1.}$	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LOG Block.	

3.9.7 MOD block

Description	Provides the modulo function.	
Function	At runtime, the user connected inputs (IN[1], IN[2]) are truncated to integer and the output (OUT) becomes the Remainder of IN[1] divided by IN[2]. Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..2] = Input connection value	
Output	OUT = Remainder of IN1 divided by IN2 OUT = NaN for divide by zero, NaN, or infinity value inputs	
Parameters	IN[1..2] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MOD Block.	

3.9.8 MUL block

Description	Provides the N-input multiply function.	
Function	At runtime, the output (OUT) becomes the Product of the user connected inputs (IN[1..8]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..8] = Input connection value	
Output	OUT = Product of (IN_1, $1N_2$, ..IN_n	
Parameters	IN[1..8] NAME NUMOFINPUTS	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MUL Block.	

3.9.9 NEG block

Description	Provides the negative function.

Function	At runtime, the output (OUT) becomes the negative (-) equivalent of the user connected input (IN[1]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = - (IN_1)	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the NEG Block.	

3.9.10 POW block

Description	Provides the raise to power function.	
Function	At runtime, the output (OUT) becomes the user connected input 1 (IN[1]) raised to the power the user connected input 2 (IN[2]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..2] = Input connection value	
Output	OUT = IN_1 raised to the power of IN_2.	
Parameters	IN[1..2] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the POW Block.	

3.9.11 ROUND block

Description	Provides the round up function.
Function	At runtime, the output (OUT) becomes the user connected input (IN[1]) rounded up to the nearest integer value. Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.

Input	IN[1] = Input connection value	
Output	OUT = REAL number of (IN_1 rounded up to the nearest integer number)	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the ROUND Block.	

3.9.12 SQRT block

Description	Provides the square root function.	
Function	At runtime, the output (OUT) becomes the square root of the user connected input (IN[1]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = Square root of IN_1	
Parameters	IN[1] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SQRT Block.	

3.9.13 SUB block

Description	Provides the subtract function.	
Function	At runtime, the output (OUT) becomes the user connected input 1 (IN[1]) minus the user connected input 2 (IN[2]). Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1..2] = Input connection value	
Output	OUT = IN_2 minus IN_2	
Parameters	IN[1..2] NAME	ORDERINCM OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SUB Block.	

3.9.14 TRUNC block

Description	Provides the truncate function.	
Function	At runtime, the output (OUT) becomes the user connected input (IN[1]) rounded down to the nearest integer value. Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for Reals). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN[1] = Input connection value	
Output	OUT = REAL number of (IN_1 rounded down to the nearest integer number)	
Parameters	IN[1]	ORDERINCM
	NAME	OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the TRUNC Block.	

3.9.15 ROLLAVG block

Description	Provides the rolling average function.	
Function	At runtime, the output (OUT) becomes the quotient of sum of all good samples (IN) divided by number of good samples. Math blocks execute a self describing and expression-less math function based on the user's configuration within a control strategy in Control Builder with minimum and more efficient memory usage. When a block cannot fetch a value from the input connection, the value defaults to a failsafe value (NaN for FLOAT64s). Math functions use existing capabilities to process infinity. For example, divide by zero. If bad input handling is desired, use the Logic CHECKBAD block on the desired input.	
Input	IN = Input connection value (FLOAT64 double-precision floating-point number).	
Output	OUT = Quotient of sum of all good values in buffer (IN) divided by number of good values in buffer (FLOAT64 double-precision floating-point number).	
Parameters	ROLLAVGBAD	ROLLMULTIPLE
	ROLLAVGOK	ROLLAVGSZ
	ROLLBFBASE	ROLLAVGRST
	ROLLBUF	IN
	ROLLFRBASE	ORDERINCM
	ROLLFREQ	OUT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the RollAvg Block.	

3.10 Power Generation Blocks

Related topics

"GRPCAPRBK (Group Capability and Runback) block" on page 108
"HTMOTOR (HT Motor Drive Control) block" on page 109
"LEVELCOMP (Drum Level Computation) block" on page 113
"LTMOTOR (LT Motor Drive Control) block" on page 114
"MAINIBV (Main IBV Logic) block" on page 116
"SOLENOID (Solenoid Valve Drive Control) block" on page 117
"VALVEDAMPER (Valve/Damper Drive Control) block" on page 120

3.10.1 GRPCAPRBK (Group Capability and Runback) block

Attention
The GRPCAPRBK block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	A Coordinated Master Control (CMC) scheme includes a Unit Load Demand Scheme that uses unit capability and runback rate to ensure that the unit load increase or decrease is executed within the limits that are safe for the operation of a Boiler- Turbine-Generator (BTG) unit.The unit load demand scheme is a part of CMC scheme that receives the load demand from load dispatcher and checks the demand signal against the allowable operating limits for the unit and the minimum load desired. During online operation, the demand can also be Runback in the event of loss of auxiliary equipment such as losing one forced draft fan, which limits the load carrying capability of the unit. All the auxiliary equipment like ID Fan, FD Fan, PA Fan, primary/secondary Air Pre Heaters, BCW Pump, Turbine, CEP Pump, CW Pump, Mills, and so on are monitored continuously for unit capability because each group of equipment is compared with the Load SP and the minimum is selected as the SP to the Unit Capability ramp generator. When one of the equipment trips, the SP to the Unit capability ramp generator comes down correspondingly. The output of the Ramp Generator is brought down by runback action. The runback rate depends upon the equipment that has tripped. The Group Capability and Runback function block provides the Group Capability and Group Runback Rate of the configured number of equipment. The block accepts configurable number of equipments' status inputs and desired unit load set point.
Function	• Provides the capability for a group of similar equipment depending upon the equipment status. • Provides the Group Capability and Group Runback Rate for the configured number of equipments. • Accepts configurable number of equipments' input status and desired unit load set point. • Provides a configurable parameter NUMBEROFEQP which represents the number of equipments used for input connection to the block. • Enables the user to configure equipment OFF state Capability Value (OFFCAP) and ON state Capability Value (ONCAP) for individual equipment. • Generates a Safe output flag when the load setpoint input status bad or all equipement on/off staus are bad. • Generates alarm for Run back Active when out capability is less than unit load set point
Inputs	• LOADSP - Load Set point. The input LOADSP of this block can be from any Regulatory Control block like AUTOMAN, SWITCH or RATIOBIAS. • DI[1..10]- DI of this block can be from any block with digital output.

Outputs	• OUTCAP - Output capability of a group of equipment • RUNBKACTFL - Runback control flag • ROCLM - Current ROC Limit • SAFEOPTRIGFL- Safe output triggered	
Parameters	CAPVALOPT CAPSAFEVAL DISTS[1..10] GCBLOCKSINCM HIALM.PR HIALM.SV HIALM.TYPE INALM LOADSPSTS LOADSP NUMBEROFEQP OFFCAP[1..10] ONCAP[1..10] OUTCAP	RBROCLM ROCLM ROCLMOPT ROCSAFELM RUNBKACTALM.FL RUNBKACTALM.PR RUNBKACTALM.SV RUNBKACTFL SAFEOPALM.FL SAFEOPALM.PR SAFEOPALM.SV SAFEOPTRIGFL
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the GRPCAPRBK Block.	

3.10.2 HTMOTOR (HT Motor Drive Control) block

Attention
The HTMOTOR block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	The HTMOTOR Control block is derived from the Device Control block in Experion (DEVCTL) customized to meet the HT Motor Drive control requirements found in power plants. The HTMOTOR block is designed to accept inputs and interlocks pertaining to a conventional HTMOTOR drive and its switchgear and is capable of controlling the drive through outputs governed by predetermined logic.
Function	Processes inputs typically from a HT Switch Gear, Local and Unit Control Panel, equipment protection related feedback, process related interlocks and generates output based on predetermined logic. Supports control of HT Motor drives from the Unit or Local Control Panel, in SEQ (through SCM) or AUTO mode or from Operator Station or a combination of all these options. Generates an alarm based on active states, output command and equipment or process interlocks tripping. Built-in First Up Logic to determine the cause of trip and built-in interlocks with vibration, winding and bearing temperature. Optional feedback and trip time recording. Maintenance statistics similar to those in the Device Control block. Preserves Device Control functionality.

Inputs	• **Process Feedback (from HT Swgr)**: Motor RUN/STOP (MTR/MTS), MOTOR Trip (MTT), Lockup Relay Reset (LRR), Breaker in Remote (BKR), Breaker in Service or Test (BKS or BKT). • **Other Inputs** : Local/Remote Selection Switch,LOCALRUN/STOP, AUTORUN/STOP,SEQRUN/STOP, UCPRELease/OPEN/CLOSE, Console Run, Console Stop, Permissives (PI[0,1]), STOP/RUN Process Interlocks(OI[0,1]),Bearing Temp Trip,Winding Temp Trip,Vibration Trip.
Outputs	Command RUN/STOP (Latched) and PORUN/STOP (Pulsed), Track Run/Stop/Trip feedback time.

Parameters	AUTOSTART AUTOSTOP AUTOSTOPACTED AUTONOCMD ASTEPID BACKCALCIN BADPVALM.SV BADPVALM.FL BADPVALM.PR BRNGTEMTRIP BYPASS BKTBYPASS BKS BKT BKR BRNGTEMPALM.PR BRNGTEMPTRIPACTED BRNGTEMPALM.FL BRNGTEMPALM.OPT BRNGTEMPALM.SV CMDDISALM.TM CMDDISALM.SV CMDDISALM.PR CMDDISALM.FL CMDFALALM.PR CMDFALALM.TM CMDFALALM.FL CLROPREQFL CMDRUN CMDSTOP CONSSTARTRDY DOVALSTS FIRSTUPACTED GOPACTED GPVAUTO GOPSCADA GPV GOPFINAL GOP HIALM.SV HIALM.PR HIALM.TYPE	MTT MTR MTS MTTACTED NORMMODE NORMMODEATTR NULLPVFL NUMALLTRANS NUMSIOVRD NUMTRANS OI[0..1] OIALM[0..2].FL OIALM[0..2].PR OIALM[0..2].OPT OIALM[0..2].SV OIACTED OP OPACTED OPFINAL OPREQ OPTYPE OFFNRMALM.OPT OFFNRMALM.FL OFFNRMALM.PR OFFNRMALM.SV PI[0..1] PVAUTO PVFL/ PVSOURCE PVSRCOPT PORUN POSTOP POSTOPCONNECTED PORUNCONNECTED REDTAG REMOTESWITCH RESETFIRSTUP RESTARTOPT RESETFL RUNPULSEWIDTH RUNTIME SAFEREDTAG

	HOLDOPT	SEALOPT
	INALM	SEQSTOP
	INITCONNECTD	SEQSTOPACTED
	INITMAN	SI
	INITOPOPT	SIALM.OPT
	INITOPAFTLD	SEQSTART
	INITREQ	SIALM.FL
	LASTREQFL	SIALM.SV
	LASTOPTYPE	SIALM.PR
	LASTOPREQ	SIACTED
	LASTSTEP	STARTOPT
	LASTGOPREQ	STOPOPT
	LOCALMAN	STOPTIME
	LOCALSWITCH	STOPPULSEWIDTH
	LOCALSTART	TRIPTIME
	LOCALSTOPACTED	TRKNUMTRANS
	LOCALSTOP	TRKSTATETIME
	LRR	TRKSIOVRD
	LRRACTED	UCPREL
	LRRALM.FL	UCPSTART/
	LRRALM.PR	UCPSTOP
	LRRALM.SV	UCPSTOPACTED
	LRRALM.OPT	UNCMDALM.FL
	MODE	UNCMDALM.PR
	MODETRACK	UNCMDALM.SV
	MODEATTR	VIBRTRIP
	MODEATTRCHGFL	VIBTRIPACTED
	MODEATTRFL.PROG	VIBRALM.FL
	MODEATTRFL.OPER	VIBRALM.PR
	MODEATTRFL.NORM	VIBRALM.SV
	MOTORTRIPALM.FL	VIBRALM.OPT
	MOTORTRIPALM.PR	WDGTEMPTRIP
	MOTORTRIPALM.SV	WDGTEMPTRIPACTED
	MOTORTRIPALM.OPT	WDGTEMPALM.FL
		WDGTEMPALM.PR
		WDGTEMPALM.SV
		WDGTEMPALM.OPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the HTMOTOR Block.	

3.10.3 LEVELCOMP (Drum Level Computation) block

Attention
The LEVELCOMP block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	In power plants, the Drum Level is measured indirectly in terms of differential pressure (DP), and the level is computed based on the measured DP. DP to Level computation techniques are employed in Level measurements in Boiler Drums, Deaerators, and HP Heaters, where the process fluid is at high pressure and temperature. In such cases, the Level component is measured in terms of DP across the measuring setup. It is also assumed that the vessel remains in a saturated state of the process fluid. The pressure inside the container is measured and the density of Steam and Water are derived from this pressure input using an algorithm. Based on Density and DP, the level is computed using the equation of continuity. This new block computes Drum level from the measured DP and other field specific constants.	
Function	• The Drum Level Computation function block computes the drum level from the measured DP, Pressure and other field specific constants. • The block has an algorithm for generating steam and water density from the given Pressure input as long as the pressure input is good. • The Level Status (PV) becomes bad when one of the input's status becomes bad and PV is not be computed. • Enables the user to select ENGUNIT for DP, Pressure, WETLEGTEMP, and drum level.	
Inputs	• DP - Differential Pressure of the Drum • PRESSURE - Pressure Input	
Outputs	• PV - Drum Level in Engineering Units • PVP - Drum level in % • DENSTEAM - Density of steam in drum • DENWATER - Density of water in drum • DENWATERREF - Density of WETLEG water	
Parameters	BADPVALM.PR	INALM
	BADPVALM.SV	PRESSURE
	BADPVALM.FL	PV
	DP	PVEUHI
	DPSLOPE	PVEULO
	DPBIAS	PRSLOPE
	DENWATERREF	PRBIAS
	DPENGUNIT	PVP
	DENWATER	PVSTS
	DENSTEAM	PRENGUNIT
	NUMBEROFEQP	PVENGUNIT
	HIALM.PR	STNDPIPELEN
	HIALM.SV	TMPENGUNIT
	HIALM.TYPE	WETLEGTEMP
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LEVELCOMP Block.	

3.10.4 LTMOTOR (LT Motor Drive Control) block

Attention
The LTMOTOR block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	The LTMOTOR Control FB is derived from the Device Control block in Experion (DEVCTL) customized to meet the LT Motor Drive control requirements found in power plants. The LTMOTOR FB is designed to accept inputs and interlocks pertaining to a conventional LTMOTOR drive's MCC and is capable of controlling the drive through outputs governed by predetermined logic.
Function	• Process inputs typical from a MCC, Local and Unit Control Panel, Process related interlocks and generate output based on predetermined logic. • Supports control of LT Motor drives from the Unit or Local Control Panel, in SEQ (through SCM) or AUTO mode or from Operator Station or any combination of all these options. • Generates an alarm based on active states, output command and process interlocks tripping. • Optional feedback and trip time recording. • Maintenance Statistics similar to those in the Device Control block. • Preserves Device Control functionality.
Inputs	• **Process Feedback (from MCC):** Motor RUN/STOP (MTR/MTS) and MOTOR Trip (MTT). • **Other Input:** Local/Remote Selection Switch, LOCALRUN/STOP, AUTORUN/STOP, SEQRUN/STOP, UCPRELease/OPEN/CLOSE, STOP/RUN Permissives (PO [0,1]), STOP/RUN Interlocks(OI[0,1]) and Track Run/Stop/Trip feedback time.
Outputs	Command RUN/STOP (Latched) and PORUN/STOP (Pulsed).

Parameters	ASTEPID	MOTORTRIPALM.SV
	AUTOSTART	MTR
	AUTOSTOP	MTS
	BACKCALCIN	MTT
	BADPVALM.FL	NORMMODE
	BADPVALM.PR	NORMMODEATTR
	BADPVALM.SV	NULLPVFL
	BYPASS	NUMALLTRANS
	BYPERM	NUMSIOVRD
	CLROPREQFL	NUMTRANS
	CMDDISALM.FL	OFFNRMALM.FL
	CMDDISALM.PR	OFFNRMALM.OPT
	CMDDISALM.SV	OFFNRMALM.PR
	CMDDISALM.TM	OFFNRMALM.SV
	CMDFALALM.FL	OI[0..1]
	CMDFALALM.PR	OIALM[0..2].FL
	CMDFALALM.SV	OIALM[0..2].OPT
	CMDFALALM.TM	OIALM[0..2].PR
	CMDRUN	OIALM[0..2].SV
	CMDSTOP	OP
	CONSSTARTRDY	OPFINAL
	DOVALSTS	OPREQ
	GOP	OPTYPE
	GOPFINAL	PI[0..1]
	GOPREQ	PORUN
	GOPSCADA	POSTOP
	GPV	POSTOPCONNECTED
	GPVAUTO	PV
	HIALM.PR	PVAUTO
	HIALM.SV	PVFL/
	HIALM.TYPE	PVSOURCE
	HOLDOPT	PVSRCOPT
	INALM	REDTAG
	INITCONNECTD	REMOTESWITCH
	INITMAN	RESTARTOPT
	INITOPAFTLD	RESETFL
	INITOPT	RUNPULSEWIDTH
	INITREQ	RUNTIME
	LASTGOPREQ	SEQSTART
	LASTOPREQ	SEQSTOP
	LASTOPTYPE	SI
	LASTREQFL	SIALM.FL

	LASTSTEP	SIALM.SV
	LOCALMAN	SIALM.OPT
	LOCALSTART	SIALM.PR
	LOCALSTOP	STARTOPT
	LOCALSWITCH	STOPOPT
	MODE	STOPPULSEWIDTH
	MODEATTR	STOPTIME
	MODEATTRCHGFL	TRIPTIME
	MODEATTRFL.NORM	TRKNUMTRANS
	MODEATTRFL.OPER	TRKSIOVRD
	MODEATTRFL.PROG	TRKSTATETIME
	MODETRACK	UCPREL
	MOTORTRIPALM.FL	UCPSTART
	MOTORTRIPALM.OPT	UCPSTOP
	MOTORTRIPALM.PR;	UNCMDALM.FL
		UNCMDALM.PR
		UNCMDALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the LTMOTOR Block.	

3.10.5 MAINIBV (Main IBV Logic) block

Attention

The MAINIBV block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	Main IBV logic is required for huge valves which hav a small bypass valve connected across the main valve port. The main valve opens only when the bypass valve opens fully and a sufficient time is allowed to equalize the pressure across the main valve ports. The Main IBV block receives a command from the operator, processes the commands and schedules it to the IBV and Main valve with a predetermined logic built into it. In the Main IBV logic function block, there is a user configurable DELAY parameter for providing the time delay before releasing an open command to the Main valve from the moment the IBV open feedback is sensed by the block. The feedback can be open limit switch of IBV. The bypass valve in not kept open if the main valve open command fails. A user configurable timer is provided to close the IBV in such eventualities.
Function	• The Main IBV block receives a command from the OPER or PROG, processes it and schedules the command to the IBV and Main valve with a predetermined logic built into it. • An open sequence failure alarm is triggered in case an OPEN command fails to open the Main valve. • A close sequence failure alarm is triggered in case a CLOSE command failed to close Main valve. • A BADPV alarm is triggered in case IBV open feedback or Main close feedback is in BAD state. • The Main IBV commands the Drive Control blocks for further operation of the Main and IBV valve.

Inputs	• OPENSEQ - Open Sequence command from PROGRAM to the valve system. • CLOSESEQ - Close Sequence command from PROGRAM to the valve system. • IBVOPNFDBK - IBV open feed back switch • MAINCLOSEFDBK - Main valve close feedback switch.	
Outputs	• OPENIBV\CLOSEIBV - Open\Close Command to IBV drive control. • OPENMAIN\CLOSEMAIN - OPEN\CLOSE command to Main Valve drive control.	
Parameters	BADPVALM.FL BADPVALM.PR BADPVALM.SV CLOSEIBV CLOSEMAIN CLROPREQFL CLOSESEQ CLOSESEQALM.FL CLOSESEQALM.PR CLOSESEQALM.SV DELAY HIALM.PR HIALM.SV HIALM.TYPE IBVOPNFDBK IBVOPNFDBKSTS IBVVLVFDBKTO INALM MAINCLOSEFDBK	MAINCLOSEFDBKSTS MAINVLVFDBKTO MODE MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG NORMMODE NORMMODEATTR OP OPENIBV OPENMAIN OPENSEQ OPENSEQALM.FL OPENSEQALM.PR OPENSEQALM.SV SEQSTATUS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the MAINIBV Block.	

3.10.6 SOLENOID (Solenoid Valve Drive Control) block

Attention

The SOLENOID block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	The Solenoid Valve Control FB is derived from the Device Control block in Experion (DEVCTL) customized to meet the Solenoid Valve control requirements commonly found in a power plant. The Solenoid FB is designed to accept inputs and interlocks pertaining to a conventional solenoid and is capable of controlling the valve through output governed by predetermined logic.
Function	• Processes feedback from the field, Local and Unit Control Panel, process related interlocks and generates output based on predetermined logic. • Supports control of Solenoid valve from the Unit or Local Control Panel, in SEQ (through SCM) or AUTO mode or from Operator Station or a combination of all these options. • Generates an alarm based on active states, output command and interlocks tripping. • Provides latched OPEN commands to the MCC/solenoid and the Outputs are reset only on issuance of a CLOSE command. • Preserves Device Control functionality.

Inputs	• **Process Feedback:** Limit Switch OPEN/CLOSE. • **Other Inputs:** Local/Remote Selection Switch,LOCALOPEN/CLOSE, AUTOOPEN/CLOSE, SEQOPEN/CLOSE, UCPRELease/OPEN/CLOSE, Close/Open Permissives (PO[0,1]) and Close/Open Interlocks(OI[0,1]).
Outputs	CMPOPEN(latched), OP, PV, NUMTRANS[0,1], NUMSIOVRD

Parameters	ASTEPID	MODEATTR
	AUTOCLOSE	MODEATTRCHGFL
	AUTOOPEN	MODEATTRFL.NORM
	BACKCALCIN	MODEATTRFL.OPER
	BADPVALM.FL	MODEATTRFL.PROG
	BADPVALM.PR	MODETRACK
	BADPVALM.SV	NORMMODE
	BYPASS	NORMMODEATTR
	BYPERM	NULLPVFL
	CLROPREQFL	NUMALLTRANS
	CMDDISALM.FL	NUMSIOVRD
	CMDDISALM.PR	NUMTRANS
	CMDDISALM.SV	OFFNRMALM.FL
	CMDDISALM.TM	OFFNRMALM.OPT
	CMDFALALM.FL	OFFNRMALM.PR
	CMDFALALM.PR	OFFNRMALM.SV
	CMDFALALM.SV	OI[0..1]
	CMDFALALM.TM	OIALM[0..2].FL
	CMDOPEN	OIALM[0..2].OPT
	CMDSTOP	OIALM[0..2].PR
	CONSOPENRDY	OIALM[0..2].SV
	DOVALSTS	OP
	GOP	OPFINAL
	GOPFINAL	OPREQ
	GOPREQ	OPTYPE
	GOPSCADA	PI[0..1]
	GPV	PV
	GPVAUTO	PVAUTO
	HIALM.PR	PVFL/
	HIALM.SV	PVSOURCE
	HIALM.TYPE	PVSRCOPT
	HOLDOPT	REDTAG
	INALM	REMOTESWITCH
	INBETFL	RESTARTOPT
	INITCONNECTD	RESETFL
	INITMAN	SAFEREDTAG
	INITOPAFTLD	SEALOPT
	INITOPT	SI
	INITREQ	SIALM.FL
	LASTGOPREQ	SIALM.SV
	LASTOPREQ	SIALM.OPT
	LASTOPTYPE	SIALM.PR

	LASTREQFL	STARTOPT
	LASTSTEP	STOPOPT
	LOCALCLOSE	TRKNUMTRANS
	LOCALMAN	TRKSIOVRD
	LOCALOPEN	UCPCLOSE
	LOCALSWITCH	UCPOPEN
	LTC	UCPREL
	LTO	UNCMDALM.FL
	MODE	UNCMDALM.PR
		UNCMDALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the SOLENOID Block.	

3.10.7 VALVEDAMPER (Valve/Damper Drive Control) block

Attention
The VALVEDAMPER block can only be used with C300 Controllers. It is not supported for use on C200 Controllers.

Description	The Valve/Damper Control FB is derived from the Device Control block in Experion (DEVCTL) and customized to meet the motor operated Valve or Damper control requirements of Power Plants. The ValveDamper FB is designed to accept inputs and interlocks pertaining to a conventional valve and is capable of controlling the valve through outputs governed by predetermined logic.
Function	• Processes inputs from MCC, process feedback, Local and Unit Control Panel (UCP), process related interlocks and generate output based on predetermined logic. • Supports control of Valve/Damper from the Unit or Local Control Panel, in SEQ (through SCM) or AUTO mode or from Operator Station or a combination of all these options. • Generates an alarm based on active states, output command and interlocks tripping. • Provides optional torque switch based command cut-off for tight shut off of valves and standard limit switch based command cut-off for normal valves. • Optional feedback (OPEN and CLOSE feedback) time recording. • Maintenance Statistics similar to those in the Device Control block. • It withdraws the command when feedback is achieved to command initiate. • Preserves Device Control functionality.
Inputs	• **Process Feedback:** Limit Switch OPEN/CLOSE, Torque Switch OPEN/CLOSE, MOTOR Trip from switch gear and Winding temperature switch. • **Other Inputs:** Local/Remote Selection Switch,LOCALOPEN/CLOSE, AUTOOPEN/CLOSE, AUTONOCMD, SEQOPEN/CLOSE, SEQNOCMD, UCPREL/OPEN/CLOSE,Close/Open Permissives (PO[0..2]), Close/Open Interlocks(OI[0..1]),Torque switch enabled (for Tight Shut-Off),Torque Switch for Protection and Track Open/Close feedback time.
Outputs	POOPEN, POCLOSE, CMDOPEN, CMDCLOSE, OP, PV, NUMTRANS[0,1,2], NUMALLTRANS, NUMSIOVRD, OPENFEEDBKTIME, CLOSEFEEDBKTIME

Parameters	ASTEPID	NUMTRANS
	AUTOCLOSE	OFFNRMALM.FL
	AUTONOCMD	OFFNRMALM.OPT
	AUTOOPEN	OFFNRMALM.PR
	BACKCALCIN	OFFNRMALM.SV
	BADPVALM.FL	OI[0..1]
	BADPVALM.PR	OIALM[0..2].FL
	BADPVALM.SV	OIALM[0..2].OPT
	BYPASS	OIALM[0..2].PR
	BYPERM	OIALM[0..2].SV
	CLOSEFEEDBKTIME	OP
	CLOSEPULSEWIDTH	OPENFEEDBKTIME
	CLROPREQFL	OPENPULSEWIDTH
	CMDCLOSE	OPFINAL
	CMDDISALM.FL	OPREQ
	CMDDISALM.PR	OPTYPE
	CMDDISALM.SV	PI[0..1]
	CMDDISALM.TM	POCLOSE
	CMDFALALM.FL	POCLOSECONNECTED
	CMDFALALM.PR	POOPEN
	CMDFALALM.SV	POOPENCONNECTED
	CMDFALALM.TM	PV
	CMDOPEN	PVAUTO
	CONSOPENRDY	PVFL/
	DOVALSTS	PVSOURCE
	GOP	PVSRCOPT
	GOPFINAL	REDTAG
	GOPREQ	REMOTESWITCH
	GOPSCADA	RESTARTOPT
	GPV	RESETFL
	GPVAUTO	SAFEREDTAG
	HIALM.PR	SEALOPT
	HIALM.SV	SEQCLOSE
	HIALM.TYPE	SEQNOCMD
	HOLDOPT	SEQOPEN
	INALM	SI
	INBETFL	SIALM.FL
	INITCONNECTD	SIALM.SV
	INITMAN	SIALM.OPT
	INITOPAFTLD	SIALM.PR
	INITOPOPT	STARTOPT
	INITREQ	STOPOPT

	LASTGOPREQ LASTOPREQ LASTOPTYPE LASTREQFL LASTSTEP LOCALCLOSE LOCALMAN LOCALOPEN LOCALSWITCH LTC LTO MODE MODEATTR MODEATTRCHGFL MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODETRACK MOTORTRIPALM.FL MOTORTRIPALM.OPT MOTORTRIPALM.PR MOTORTRIPALM.SV NORMMODE NORMMODEATTR NULLPVFL NUMALLTRANS NUMSIOVRD	TRKCLOSEFEEDBKTIME TRKNUMTRANS TRKOPENFEEDBKTIME TRKSIOVRD TSC TSC TSCALM.FL TSCALM.OPT TSCALM.PR TSCALM.SV TSENABLED TSFORPROT TSO TSOALM.FL TSOALM.OPT TSOALM.PR TSOALM.SV UCPCLOSE UCPOPEN UCPREL UNCMDALM.FL UNCMDALM.PR UNCMDALM.SV WTS WTSALM.FL WTSALM.OPT WTSALM.PR WTSALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the VALVEDAMPER Block.	

3.11 Regulatory Control Blocks

Related topics

“AUTOMAN (Auto Manual) block ” on page 123
“ENHREGCALC (Enhanced Regulatory Control Calculator) block” on page 127
“FANOUT block” on page 133
“OVRDSEL(Override Selector) block” on page 136
“PID block” on page 140
“PIDER block” on page 145
“PID-PL block” on page 150
“PIDFF (PID Feedforward) block” on page 155
“POSPROP (Position Proportional) block” on page 161
“PULSECOUNT block” on page 166
“PULSELENGTH block” on page 167
“RAMPSOAK block” on page 169
“RATIOBIAS block” on page 173
“RATIOCTL block” on page 178
“REEOUT (Remote EEOUT) block” on page 182
“REGCALC (Regulatory Control Calculator) block” on page 183
“REGSUMMER (Regulatory Control Summer)” on page 189
“REMCAS block” on page 194
“SWITCH block” on page 198

3.11.1 AUTOMAN (Auto Manual) block

Description	Applies a user-specified gain and bias as well as a calculated bias (OPBIAS.FLOAT) to the output. The user-specified values can be fixed or external. A fixed value is stored manually or by a program, and an external value is brought from another function block.
Function	Provides control initialization and override feedback processing. Typically used either: • in cascade control strategy where an upstream block may not accept an initialization request from its secondary, • between FANOUT block and a final control element to provide“bumpless” output on return to cascade. With R410, AUTOMAN block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband units for all the alarms.

Timeout Monitoring	In cascade mode, performs timeout monitoring on the input (X1). If the X1 value is not updated within a predefined time, this block invokes the following timeout processing. • Sets the "input timeout" flag (TMOUTFL). • Sets the input value to Bad (NaN). • Requests the X1 primary to initialize. Note that this block does not support mode shedding on timeout and therefore the TMOUTMODE parameter is not applicable to this block. But the mode shedding of this block occurs indirectly depending on the BADCTLOPT parameter value as timeout processing, setting the input value to Bad (NaN). • When BADCTLOPT = No_Shed, there is no mode shedding on timeout. • When BADCTLOPT = SHEDHOLD/SHEDHIGH/SHEDLOW/SHEDSAFE, the mode sheds to manual. Note that time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Inputs	• X1 = initializable input which must come from another function block; an operator cannot set it. • XEUHI and XEULO define the full range of X1: – XEUHI is the value that represents 100% of full scale. – XEULO is the value that represents 0% of full scale.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it brings the secondary's input range through the BACKCALC and sets its CV range to that. If it has no secondary, CVEUHI and CVEULO track its own input range (XEUHI and XEULO). • OPHILM and OPLOLM define the normal high and low limits for OP as a percent of CV range; these are user-specified values. OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another function block or user program attempts to store an OP value that exceeds them. However, an operator may store an OP value that is outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range. These are user-specified values. Operator is prevented from storing an OP that exceeds these limits.

Parameters	ALMDB ALMDBU ALMTM ARWNETIN ARWNET[1..8] ARWOP ARWOPIN ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB COMPUTEARW CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DESC ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC FBORSTS GAINHILM GAINLOLM HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL	NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OPTYPE ORDERINCM OUTIND OUTTYPE PRIM.[1..8].INITIALIZABLE

	INALM	PRIMDATA.[1..8].HISELECT
	INITMAN	PRIMDATA.[1..8].ORFBSTS
	INITREQ[1..8]	PRIMDATA.[1..8].ORFBVAL
	INITVAL[1..8]	PRIMDATA.[1..8].OROFFSET
	INSBLOCK[1..10]	PRIMDATA.[1..8].PROPOVRD
	INSFAILALM.FL	REDTAG
	INSFAILALM.PR	RESTARTOPT
	INSFAILALM.SV	SAFEOP
	INSFAILFL	SECDATAIN.ARWSTS
	K	SECDATAIN.EUHI
	LASTMODEREQ	SECDATAIN.EULO
	LASTOPREQ	SECDATAIN.HISELECT
	LASTOPTYPE	SECDATAIN.INITSTS
	LASTREQFL	SECDATAIN.INITVAL
	LASTSTEP	SECDATAIN.LOCALMAN
	MODE	SECDATAIN.ORFBSTS
	MODEAPPL[1..4]	SECDATAIN.ORFBVAL
	MODEATTR	SECDATAIN.OROFFSET
	MODEATTRFL.NORM	SECDATAIN.PROPOVRD
	MODEATTRFL.OPER	SECINITOPT[1..8]
	MODEATTRFL.PROG	SIALM.FL
	MODECHANGE	SIALM.OPT
	MODEFL.AUTO	SIALM.PR
	MODEFL.BCAS	SIALM.SV
	MODEFL.CAS	SIFL
	MODEFL.MAN	SIOPT
	MODEFL.NORM	STARTOPT
	MODEPERM	STARTRATE
	MODEREQ	STARTVAL
	MODETRACK	STOPOPT
	NAME	STOPRATE
	NORMMODE	STOPVAL
	NORMMODEATTR	TMOUTFL
	NUMINSERT	TMOUTTIME
	NUMPRI	UNCMDCHGALM.FL
		UNCMDCHGALM.OPT
		UNCMDCHGALM.PR
		UNCMDCHGALM.SV
		X1
		X1P
		X1STS

		XEUHI XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the AUTOMAN Block.	

3.11.2 ENHREGCALC (Enhanced Regulatory Control Calculator) block

Description	• Lets you write up to eight expressions for creating custom algorithms for Calculated Variable (CV) calculations. • Provides an interface to windup, initialization and override feedback processing, so you can add user-defined control blocks to your control strategies. • The ENHREGCALC block provides the following enhancements over the REGCALC block. – Expands existing arrayed input parameters XSTS, XCONN and X. – These arrayed parameters are added to correspond to each of the ten inputs. Input Description Enable/Disable Switch XSUB Substitute Parameter – An initializable Set Point (SP) input parameter with limit checking and SP ramping is added. Also, the X[1] input is converted to a general purpose input – Mode can be placed in Automatic so operator or program can supply SP. – This block uses memory based on the number of expressions configured, pcode size of each expression and the number of references in the expression.
Function	• Each expression can contain any valid combination of inputs, operators and functions; and may perform arithmetic or logic operations. • You can write expressions for calculating CV under normal, initialization and override feedback conditions. Or, you can write expressions which produce initialization and override feedback values for this block and its primaries. • You can assign the result of an expression or an input to any assignable output that produces the same outputs as every other regulatory control block. You can assign the same input to multiple outputs. • With R410, ENHREGCALC block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, this block performs timeout monitoring on SP. If the SP input value is not updated within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL) • Holds SP at its last good value. • Requests the SP primary to initialize. • Sheds to a user-specified timeout mode (MODE = TMOUTMODE). The ENHREGCALC block sets its cascade request flag (CASREQFL), if SP times out and sheds to AUTOmatic mode. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.

Control Initialization	Block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. However, you can disable the SECINITOPT so the block ignores initialization requests from the secondary. If the secondary is requesting initialization, block: • Initializes its output: – CV = CVINIT (assignable output) • Builds an initialization request for the designated primaries, using INITREQ and INITVAL (both assignable outputs).
Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector (OVRDSEL) block, it receives override feedback data. The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy. The offset flag only applies to PID type function blocks. However, you can disable the SECINITOPT so the block ignores override requests from the secondary. When override status changes from selected to unselected, this block: • Initializes its output: – CV = CVORFB (assignable output) • Computes a feedback value for SP input: – feedback value for SP = ORFBVAL (assignable output) – feedback status for SP = ORFBSTS (assignable output) If ORFBVAL and ORFBSTS are not assigned and this block has a secondary, the ORFBVAL and ORFBSTS received from the secondary are used to compute ORFBVAL for the primary.
Inputs	The following inputs are optional and they only accept real data types. • SP - An initializable input. If Mode is CAScade, SP is pulled from another function block. If Mode is AUTO, it may be stored by the operator or a user program. • X[1] through X[10] general purpose inputs. • XWHIFL - An external windup high flag. • XWLOFL - An external windup low flag.
Input Ranges	SPEUHI and SPEULO define the full range of SP input in engineering units. This block applies no range checking, since it assumes that SP is within SPEUHI and SPEULO. If this function is required, you must write an expression for it. • SPEUHI represents the 100% of full scale value. • SPEULO represents the 0% of full scale value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.

Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it uses the secondary's input range through BACKCALC to set its CV range. If it does not have a secondary, you must define the range through CVEUHI and CVEULO. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of the CV range (user-specified values). – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store OP value exceeding them. – Operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range (user-specified values). – Operator is prevented from storing an OP value that exceeds these limits.
Assignable Outputs	You can assign expression results and/or inputs to the following outputs. • CV - This block's CV under normal operating conditions. • CVINIT - This block's CV during initialization. • CVORFB - This block's CV during override (in unselected path). • INITREQ - Initialization request flag, to be provided to the primary. • INITVAL - Initialization value, to be provided to the primary. • ORFBVAL - Override feedback value, to be provided to the primary. • ORFBSTS - Override feedback status, to be provided to the primary.
Operators and Functions	"AUXCALC (Auxiliary Calculation) block" on page 34 lists the expression operators and functions supported by this block for reference.
Parameter Identification	You must specify a parameter by its full tag name. For example, "CM25.PumpASelect.PVFL", or "CM57.PID100.MODE". In effect, tag names allow expressions to have an unlimited number of inputs and work with any data type. However, do not use more than six parameter references in an expression. The expression syntax has been expanded. Delimiters (') can be used in an expression containing an external reference component. The format for the delimiter usage is as follows: • TagName.'text'

Expression Rules	• Must include full tag.parameter name for X inputs in the expression and enclose identification number in brackets instead of parenthesizes. For example, CM151.REGCALC BLOCK.X[1] * CM151.REGCALC BLOCK.X[2] is valid. • Expressions cannot contain an assignment operation (a colon and equal sign with the current syntax) For example, “CM1.PID1.MODE:=X[1]” is invalid. Each expression produces a single value (arithmetic or logical which is automatically stored in a “C” parameter. For example, if you write four expressions, the result of the first expression is stored in C[1], the result of the second is stored in C[2], etc. You can use these results, by name, in succeeding expressions. In this example, you could use C[1] as an input to expressions 2, 3, and 4. • You can mix and nest all operators and functions (including conditional assignments) in any order as long as value types match or can be converted. • You can use blanks between operators and parameter names, but they are not required. • You can use all data types in expressions, including enumerations. They are all treated as numeric types. • You must configure calculator expressions contiguously (without breaks) in the arrays. • A short description can be provided for the expressions using the expression descriptor parameter (EXPRDESC[1..8]). The results of the expressions, which use the CONST[1…8] parameters, are affected if you change the values of these parameters on the Constants tab. • With R410, non-CEE controllers such as PMD and Safety Manager, and Experion server points such as TPS and SCADA, can be configured in the Expressions. • With R410, when you write the expressions using the TPS point's parameter references, ensure that the TPS reference parameter is configured using the parentheses “()”to specify array index. However, when you write the expressions using the other non-CEE points you can use the brackets “[].”

Parameters	ALMDB	NUMSEC
	ALMDBU	OP
	ALMTM	OPBIAS
	ARWNETIN	OPBIAS.FIX
	ARWNET[1..8]	OPBIAS.FLOAT
	ARWOP	OPBIAS.RATE
	ARWOPIN	OPEU
	ASTEPID	OPEXHIFL
	BADCTLALM.FL	OPEXHILM
	BADCTLALM.PR	OPEXLOFL
	BADCTLALM.SV	OPEXLOLM
	BADCTLALM.TM	OPHIALM.DB
	BADCTLALM.TMO	OPHIALM.DBU
	BADCTLFL	OPHIALM.FL
	BADCTLOPT	OPHIALM.PR
	BADOCOPT	OPHIALM.SV
	BADOCOPTENB	OPHIALM.TM
	C[1..8]	OPHIALM.TMO
	CASREQFL	OPHIALM.TP
	COMPUTEARW	OPHIFL
	CONFIGCODE	OPHILM
	CONFIGDESC	OPLOALM.DB
	CONFIGSTS	OPLOALM.DBU
	CONST[1…8]	OPLOALM.FL
	CONSTACCLOCK	OPLOALM.PR
	CONSTENABLE	OPLOALM.SV
	CSTS[1..8]	OPLOALM.TM
	CTLINIT	OPLOALM.TMO
	CTLSTATE	OPLOALM.TP
	CV	OPLOFL
	CVEUHI	OPLOLM
	CVEULO	OPMINCHG
	CVINIT	OPREQ
	CVINITSRC	OPROCLM
	CVORFB	OPROCNEGFL
	CVORFBSRC	OPROCPOSFL
	CVSRC	OPTOL
	CVTYPE	OPTYPE
	DESC	ORDERINCM
	ESWENB	ORFBSTSSRC
		ORFBVALSRC

	ESWFL.AUTO	OUTIND
	ESWFL.BCAS	PUSHSP
	ESWFL.CAS	REDTAG
	ESWFL.MAN	RESTARTOPT
	ESWPERM	SAFEOP
	EUDESC	SECDATAIN.ARWSTS
	EXECCODE[1..8]	SECDATAIN.EUHI
	EXECDESC[1..8]	SECDATAIN.EULO
	EXECSTS[1..8]	SECDATAIN.HISELECT
	EXPR[1..8]	SECDATAIN.INITSTS
	EXPRDESC[1…8]	SECDATAIN.INITVAL
	FBORSTS	SECDATAIN.LOCALMAN
	GAINHILM	SECDATAIN.ORFBSTS
	GAINLOLM	SECDATAIN.ORFBVAL
	HIALM.PR	SECDATAIN.OROFFSET
	HIALM.SV	SECDATAIN.PROPOVRD
	HIALM.TYPE	SECINITOPT[1..8]
	HOLDOPT	SIALM.FL
	HOLDRATE	SIALM.OPT
	HOLDVAL	SIALM.PR
	INALM	SIALM.SV
	INITMAN	SIFL
	INITREQ[1..8]	SIOPT
	INITREQSRC	SP
	INITVAL[1..8]	SPEUHI
	INITVALSRC	SPEULO
	INSBLOCK[1..10]	SPFORMAT
	INSFAILALM.FL	SPHIFL
	INSFAILALM.PR	SPHILM
	INSFAILALM.SV	SPLOFL
	INSFAILFL	SPLOLM
	K	SPP
	LASTMODEREQ	SPRATEREQ
	LASTOPREQ	SPREQ
	LASTOPTYPE	SPTOL
	LASTREQFL	SPTV
	LASTSPREQ	SPTVNORMRATE
	LASTSPTVREQ	SPTVOPT
	LASTSTEP	SPTVP
	MODE	SPTVRATE
	MODEAPPL[1..4]	SPTVREQ
	MODEATTR	SPTVSTATE

	MODEATTRFL.NORM	SPTVTIME
	MODEATTRFL.OPER	STARTOPT
	MODEATTRFL.PROG	STARTRATE
	MODECHANGE	STARTVAL
	MODEFL.AUTO	STOPOPT
	MODEFL.BCAS	STOPRATE
	MODEFL.CAS	STOPVAL
	MODEFL.MAN	TMOUTFL
	MODEFL.NORM	TMOUTMODE
	MODEPERM	TMOUTTIME
	MODEREQ	UNCMDCHGALM.FL
	MODETRACK	UNCMDCHGALM.OPT
	NAME	UNCMDCHGALM.PR
	NORMMODE	UNCMDCHGALM.SV
	NORMMODEATTR	X[1..10]
	NUMINSERT	XB[1..10]
	NUMPRI	XDESC[1..10]
		XENABLE[1..10]
		XK[1..10]
		XKB[1..10]
		XSTS[1..10]
		XSUB[1..10]
		XWHIFL
		XWLOFL
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ENHREGCALC Block.	

3.11.3 FANOUT block

Description	Uses one input and provides up to eight initializable outputs. It may also have up to eight secondaries, since there is one secondary per initializable output. You may specify a separate gain, bias, and rate for each output. Each specified value can be fixed or external. A fixed value is stored manually or by a program, and an external value is brought from another function block. This block calculates a separate floating bias for each output following an initialization or mode change. This provides a “bumpless” transition for each output.
Function	Provides a “bumpless” output for each of up to 8 outputs following initialization or mode changes. With R410, FANOUT block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.

<table>
<tr><td>Timeout Monitoring</td><td>In cascade mode, performs timeout monitoring on X1. If the X1 value is not updated within a predefined time, this block invokes the following timeout processing.
• Sets the “input timeout” flag (TMOUTFL).
• Sets the input value to Bad (NaN).
• Requests the X1 primary to initialize (through BACKCALCOUT).
This block does not support mode shedding on timeout.
Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.</td></tr>
<tr><td>Input</td><td>• X1 = initializable input which must come from another function block; an operator cannot set it.
• XEUHI and XEULO define the full range of X1:
– XEUHI represents the 100% of full scale value.
– XEULO represents the 0% of full scale value.</td></tr>
<tr><td>Outputs</td><td>May have up to 8 initializable outputs as follows:
• OP[1..8] = Calculated output in percent.
• OPEU[1..8] = Calculated output in engineering units.
Note that the default OP[1], [2] connection pins are exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX[1..8]/OPEUX[1..8]) connection when required.</td></tr>
<tr><td>Output Ranges</td><td>CVEUHI[1..8] and CVEULO[1..8] define the full range of CV[1..8] in engineering units -- block has separate output range for each output based on the input range of each secondary.
• OPHILM and OPLOLM define the normal high and low limits for OP as a percent of the CV range; these are user-specified values -- the same limits apply to all outputs. An operator may store an OP value that is outside these limits.
• OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range; these are user-specified values -- the same limits apply to all outputs and operator is prevented from storing an OP that exceeds these limits.</td></tr>
<tr><td>Windup Processing</td><td>• If all secondaries are in high windup, block propagates a high windup status to its primary (ARWNET[1..8] = Hi)
• If all secondaries are in low windup, block propagates a low windup status to its primary (ARWNET[1..8] = Lo)
• If at least one secondary has a normal windup status or is in high windup and another is in low, block propagates a normal windup status to its primary.
If the gain is reversed for one of the outputs, then high windup on that output is the same as low windup on the others.</td></tr>
</table>

Parameters	ARWMULTOP[1..8] ARWNET[1..8] ARWNETIN ARWOP ARWOPIN ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB CASREQFL COMPUTEARW CTLINIT CTLSTATE CV[1..8] CVEUHI[1..8] CVEULO[1..8] CVTYPE DESC ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC FBORSTS GAINHILM GAINLOLM HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM INITMAN	MODETRACK NAME NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP[1..8] OPBIAS[1..8] OPBIAS[1..8].FIX OPBIAS[1..8].FLOAT OPBIAS[1..8].RATE OPEU[1..8] OPEXHIFL[1..8] OPEXHILM OPEXLOFL[1..8] OPEXLOLM OPHIFL[1..8] OPHILM OPLOFL[1..8] OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL[1..8] OPROCPOSFL[1..8] OPTYPE ORDERINCM OUTIND OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS PRIMDATA.[1..8].ORFBVAL PRIMDATA.[1..8].OROFFSET PRIMDATA.[1..8].PROPOVRD REDTAG RESTARTOPT SAFEOP SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO

	INITREQ[1..8]	SECDATAIN.HISELECT
	INITVAL[1..8]	SECDATAIN.INITSTS
	INSBLOCK[1..10]	SECDATAIN.INITVAL
	INSFAILALM.FL	SECDATAIN.LOCALMAN
	INSFAILALM.PR	SECDATAIN.ORFBSTS
	INSFAILALM.SV	SECDATAIN.ORFBVAL
	INSFAILFL	SECDATAIN.OROFFSET
	K[1..8]	SECDATAIN.PROPOVRD
	LASTMODEREQ	SECINITOPT[1..8]
	LASTOPREQ	SIALM.FL
	LASTOPTYPE	SIALM.OPT
	LASTREQFL	SIALM.PR
	LASTSTEP	SIALM.SV
	MODE	SIFL
	MODEAPPL[1..4]	SIOPT
	MODEATTR	SPHILM
	MODEATTRFL.NORM	STARTOPT
	MODEATTRFL.OPER	STARTRATE
	MODEATTRFL.PROG	STARTVAL
	MODECHANGE	STOPOPT
	MODEFL.AUTO	STOPRATE
	MODEFL.BCAS	STOPVAL
	MODEFL.CAS	TMOUTFL
	MODEFL.MAN	TMOUTMODE
	MODEFL.NORM	TMOUTTIME
	MODEPERM	UNCMDCHGALM.FL
	MODEREQ	UNCMDCHGALM.OPT
		UNCMDCHGALM.PR
		UNCMDCHGALM.SV
		X1
		X1P
		X1STS
		XEUHI
		XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the FANOUT Block.	

3.11.4 OVRDSEL(Override Selector) block

Description	Provides override feedback data to every block in an upstream cascade control strategy. Also provides bypass processing, control initialization, and override feedback propagation.

Function	Accepts up to four inputs (primaries) and selects the one with the highest or lowest value. With R410, OVRDSEL block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on all inputs X[1] through X[4] that are not bypassed. If an input value is not updated within a predefined time, this block invokes the following timeout processing. • Sets the “input timeout” flag (TMOUTFL). • Sets the input value to Bad (NaN). • Requests the input's primary to initialize. This block does not support mode shedding on timeout. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Inputs	Accepts up to 4 inputs -- X[1] through X[4]. • At least 2 inputs (X1 and X2) are required, others are optional. • You can configure a 15-character description for each input. • The inputs must come from other function blocks; an operator cannot store to them.
Input Ranges	• XEUHI and XEULO define the full range of inputs. – XEUHI represents the 100% of full scale value. – XEULO represents the 0% of full scale value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it brings the secondary'ls input range through the BACKCALC and sets its CV range to that. If it has no secondary, CVEUHI and CVEULO track its own input range (XEUHI and XEULO). • OPHILM and OPLOLM (user-specified values) define normal high and low limits for OP as a percent of the CV range. – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store an OP value exceeding them. – Operator may store an OP value outside these limits. • OPEXHILM and OPEXLOLM (user-specified values) define the extended high and low limits for OP as a percent of the CV range. – Operator is prevented from storing an OP value that exceeds these limits.

Parameters	ALMDB	OPBIAS
	ALMDBU	OPBIAS.FIX
	ALMTM	OPBIAS.FLOAT
	ARWNET[1..8]	OPBIAS.RATE
	ARWOP	OPEU
	ASTEPID	OPEXHIFL
	BADCTLALM.FL	OPEXHILM
	BADCTLALM.PR	OPEXLOFL
	BADCTLALM.SV	OPEXLOLM
	BADCTLALM.TM	OPHIALM.DB
	BADCTLALM.TMO	OPHIALM.DBU
	BADCTLFL	OPHIALM.FL
	BADCTLOPT	OPHIALM.PR
	BADOCOPT	OPHIALM.SV
	BADOCOPTENB	OPHIALM.TM
	BADINPTOPT[1..4]	OPHIALM.TMO
	CONTROLREQ	OPHIALM.TP
	COMPUTEARW	OPHIFL
	CTLEQN	OPHILM
	CTLINIT	OPLOALM.DB
	CTLSTATE	OPLOALM.DBU
	CV	OPLOALM.FL
	CVEUHI	OPLOALM.PR
	CVEULO	OPLOALM.SV
	CVTYPE	OPLOALM.TM
	DESC	OPLOALM.TMO
	ESWENB	OPLOALM.TP
	ESWFL.AUTO	OPLOFL
	ESWFL.BCAS	OPLOLM
	ESWFL.CAS	OPMINCHG
	ESWFL.MAN	OPREQ
	ESWPERM	OPROCLM
	EUDESC	OPROCNEGFL
	FBORSTS	OPROCPOSFL
	HIALM.PR	OPTYPE
	HIALM.SV	ORBYPASSFL[1..4]
	HIALM.TYPE	ORBYPPERM
	HOLDOPT	ORDERINCM
	HOLDRATE	OROFFSET
	HOLDVAL	OROPT
	INALM	OUTIND
	INITMAN	OUTTYPE

	INITREQ[1..8]	PRIM.[1..8].INITIALIZABLE
	INITVAL[1..8]	PRIMDATA.[1..8].HISELECT
	INSBLOCK[1..10]	PRIMDATA.[1..8].ORFBSTS
	INSFAILALM.FL	PRIMDATA.[1..8].ORFBVAL
	INSFAILALM.PR	PRIMDATA.[1..8].OROFFSET
	INSFAILALM.SV	PRIMDATA.[1..8].PROPOVRD
	INSFAILFL	REDTAG
	LASTMODEREQ	RESTARTOPT
	LASTOPREQ	SAFEOP
	LASTOPTYPE	SECDATAIN.ARWSTS
	LASTREQFL	SECDATAIN.EUHI
	LASTSTEP	SECDATAIN.EULO
	MODE	SECDATAIN.HISELECT
	MODEAPPL[1..4]	SECDATAIN.INITSTS
	MODEATTR	SECDATAIN.INITVAL
	MODEATTRFL.NORM	SECDATAIN.LOCALMAN
	MODEATTRFL.OPER	SECDATAIN.ORFBSTS
	MODEATTRFL.PROG	SECDATAIN.ORFBVAL
	MODECHANGE	SECDATAIN.OROFFSET
	MODEFL.AUTO	SECDATAIN.PROPOVRD
	MODEFL.BCAS	SECINITOPT[1..8]
	MODEFL.CAS	SELXDESC
	MODEFL.MAN	SELXFL[1..4]
	MODEFL.NORM	SELXINP
	MODEPERM	SIALM.FL
	MODEREQ	SIALM.OPT
	MODETRACK	SIALM.PR
	NAME	SIALM.SV
	NORMMODE	SIFL
	NORMMODEATTR	SIOPT
	NUMINSERT	STARTOPT
	NUMPRI	STARTRATE
	NUMSEC	STARTVAL
	OP	STOPOPT
		STOPRATE
		STOPVAL
		TMOUTFL
		TMOUTTIME
		UNCMDCHGALM.FL
		UNCMDCHGALM.OPT
		UNCMDCHGALM.PR
		UNCMDCHGALM.SV

		X[1..4] XEUHI XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the OVRDSEL Block.	

3.11.5 PID block

Description	Operates as a proportional-integral-derivative (PID) controller and supports the Ideal form of calculating the PID terms.
Function	Accepts 2 analog inputs -- process variable (PV) and set point (SP); produces output calculated to reduce the difference between PV and SP. Provides anti-windup protection, control initialization and override feedback processing. With R410, PID block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on SP. If a good SP value is not received within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL). • Holds the SP value at its last good value. • Changes the mode to a user-specified TMOUTMODE. • Requests the input's primary to initialize. If SP times out and the block sheds to Auto mode, block sets the Cascade Request flag (CASREQFL). Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Inputs	Required number of inputs is determined by this block's mode: • If Mode = Cascade, 2 inputs are required - PV and SP. Both must be pulled from other function blocks. • If Mode = Auto or Man, only PV is required. PV must be pulled from another function block; the user cannot store to it. • SP contains set point value in engineering units; SPP contains value in percent. – If Mode = Auto, operator or user program may store to either SP or SPP. • SP is an initializable input; PV is non-initializable.
Input Ranges and Limits	• PVEUHI and PVEULO define full range of PV in engineering units. They also define the engineering unit range of SP, since PV and SP are assumed to have the same range. – PVEUHI represents the 100% of full scale value. – PVEULO represents the 0% of full scale value. • SPHILM and SPLOLM define set point operating limits in engineering units. – Prevents operator from storing SP value outside limits; if primary or user program attempts to store value outside limits, block clamps it to appropriate limit and sets primary's windup status.

Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges and Limits	CVEUHI and CVEULO define full range of CV in engineering units. If this block has a secondary, it brings the secondary's input range through the BACKCALC and sets its CV range to that. If it has no secondary, you must specify CVEUHI and CVEULO range. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of CV range (user-specified values). – OP clamps to limits if calculated CV exceeds them, or another block or user program attempts to store OP value exceeding them; operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define extended high and low limits for OP as percent of CV range (user-entered values). – Prevents operator from storing OP value that exceeds these limits.
Equation Options	• Equation A - Proportional, Integral, and Derivative on error. • Equation B -- Proportional and Integral on error and Derivative on changes in PV. • Equation C -- Integral on error and Proportional and Derivative on changes in PV. • Equation D -- Integral only. • Equation E -- Proportional only; this equation supports the following two options that affect CV: – Output bias processing which adds fixed and floating bias to unbiased CV. – Reverse-control action causes the sign of the unbiased CV to be reversed.
Gain Options	If equation A, B, or C is selected, any of the following gain options may be chosen: • Linear Gain -- provides proportional control action that is equal to a constant (K) times the error. • Gap Gain -- used to reduce sensitivity of control system when PV is in user-specified band (gap) around set point. • Nonlinear Gain -- control action is proportional to square of error, rather than error itself. • External Gain -- gain (K) is modified by input value that can come from the process, another block or user program.
Direct or Reverse Control	• Direct action -- increase in error increases output (OP). • Reverse action -- increase in error decreases output (OP).

Parameters	ADVDEVALM.DB ADVDEVALM.DBU ADVDEVALM.FL ADVDEVALM.PR ADVDEVALM.SV ADVDEVALM.TM ADVDEVALM.TMO ADVDEVALM.TP ADVDEVOPT ADVSP ADVSPP ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB CASREQFL COMPUTEARW CTLACTN CTLEQN CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DELCV DESC DEV DEVHIALM.DB DEVHIALM.DBU	OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OPTYPE ORDERINCM OUTIND OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS PRIMDATA.[1..8].ORFBVAL

	DEVHIALM.FL	PRIMDATA.[1..8].OROFFSET
	DEVHIALM.PR	PRIMDATA.[1..8].PROPOVRD
	DEVHIALM.SV	PUSHSP
	DEVHIALM.TM	PV
	DEVHIALM.TMO	PVEUHI
	DEVHIALM.TP	PVEULO
	DEVLOALM.DB	PVFORMAT
	DEVLOALM.DBU	PVMANOPT
	DEVLOALM.FL	PVP
	DEVLOALM.PR	PVSTS
	DEVLOALM.SV	PVSTSFL.BAD
	DEVLOALM.TM	PVSTSFL.MAN
	DEVLOALM.TMO	PVSTSFL.NORM
	DEVLOALM.TP	PVSTSFL.UNCER
	EQNEUNITSOPT	PVTRAKOPT
	ESWENB	PVTRAKOPTAI
	ESWFL.AUTO	REDTAG
	ESWFL.BCAS	RESTARTOPT
	ESWFL.CAS	SAFEOP
	ESWFL.MAN	SECDATAIN.ARWSTS
	ESWPERM	SECDATAIN.EUHI
	EUDESC	SECDATAIN.EULO
	FBORSTS	SECDATAIN.HISELECT
	GAINHILM	SECDATAIN.INITSTS
	GAINLOLM	SECDATAIN.INITVAL
	GAINOPT	SECDATAIN.LOCALMAN
	GAPHILM	SECDATAIN.ORFBSTS
	GAPLOLM	SECDATAIN.ORFBVAL
	HIALM.PR	SECDATAIN.OROFFSET
	HIALM.SV	SECDATAIN.PROPOVRD
	HIALM.TYPE	SECINITOPT[1..8]
	HOLDOPT	SIALM.FL
	HOLDRATE	SIALM.OPT
	HOLDVAL	SIALM.PR
	INALM	SIALM.SV
	INITMAN	SIFL
	INITREQ[1..8]	SIOPT
	INITVAL[1..8]	SP
	INSBLOCK[1..10]	SPEUHI
	INSFAILALM.FL	SPEULO
	INSFAILALM.PR	SPFORMAT
	INSFAILALM.SV	SPHIFL

	INSFAILFL K] KLIN KMODIFEXT KMODIFGAP KMODIFNL LASTGOODPV LASTMODEREQ LASTOPREQ LASTOPTYPE LASTRATEREQ LASTREQFL LASTSPREQ LASTSPTVREQ LASTSTEP LEGACYGAP MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ MODETRACK NAME NLFORM NLGAIN NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP	SPHILM SPLOFL SPLOLM SPP SPRATEREQ SPREQ SPTV SPTVDEVFL SPTVDEVMAX SPTVNORMRATE SPTVOPT SPTVP SPTVRATE SPTVREQ SPTVSTATE SPTVTIME STARTOPT STARTRATE STARTVAL STOPOPT STOPRATE STOPVAL T1 T1HILM T1LOLM T2 T2HILM T2LOLM TMOUTFL TMOUTMODE TMOUTTIME UNCMDCHGALM.FL UNCMDCHGALM.OPT UNCMDCHGALM.PR UNCMDCHGALM.SV

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the PID Block.

3.11.6 PIDER block

Description	Operates as a proportional-integral-derivative (PID) controller that accepts a reset feedback signal, a tracking value, and a tracking control switch. It supports the same Ideal form of calculating the PID terms as the PID block. It also prevents windup when the secondary does not propagate windup status or control initialization data back to the primary of a remote (foreign) controller.
Function	It accepts five analog inputs - a process variable (PV), a set point (SP), a reset feedback value (RFB), a tracking value (TRFB), and a tracking control switch (S1). The difference between PV and SP is the error and this block calculates a control output (OP) that should drive the error to zero. With R410, PIDER block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on SP. If a good SP value is not received within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL). • Holds the SP value at its last good value. • Changes the mode to a user-specified TMOUTMODE. • Requests the input's primary to initialize. If SP times out and the block sheds to Auto mode, block sets the Cascade Request flag (CASREQFL). Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Inputs	Required number of inputs is determined by this block's mode: • If Mode is CAScade, five inputs are required - PV, SP, RFB, TRFB and S1. • If Mode is AUTOmatic or MANual, PV, RFB, TRFB and S1 are required. • SP contains set point value in engineering units; SPP contains value in percent. – If Mode = Auto, operator or user program may store to either SP or SPP. • SP is the only initializable input; other inputs are non-initializable. • RFB and TRFB must be pulled from another block, you cannot store to them. The RFB input is optional. If the PIDER block is used for external tracking features only, the RFB input is not required. • S1 can be triggered by another function block or set by a user-written program.
Input Ranges and Limits	• PVEUHI and PVEULO define full range of PV in engineering units. They also define the engineering unit range of SP, since PV and SP are assumed to have the same range. – PVEUHI represents the 100% of full scale value. – PVEULO represents the 0% of full scale value. • SPHILM and SPLOLM define set point operating limits in engineering units. – Prevents operator from storing SP value outside limits; if primary or user program attempts to store value outside limits, block clamps it to appropriate limit and sets primary's windup status. • The RFB and TRFB values typically come from a remote controller. The PIDER block applies no range check for these parameters. • The S1 input is a Boolean flag and the values are only On and Off.

Outputs	The block does not support output initialization, and therefore cannot have a secondary. Initialization only occurs when the tracking control switch (S1) is On. Block has following outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges and Limits	CVEUHI and CVEULO must be specified by the user. The specified values must match the engineering units (EU) range of the RFB and TRFB signals, which are the range of the remote (foreign) controller or secondary. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of CV range (user-specified values). – OP clamps to limits if calculated CV exceeds them, or another block or user program attempts to store OP value exceeding them; operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define extended high and low limits for OP as percent of CV range (user-entered values). – Prevents operator from storing OP value that exceeds these limits. • MAXRFBDEV is the maximum deviation allowed between CV and RFB, in percent. It is used to provide windup protection for OP. – If the scaled, integrated deviation of CV from RFB exceeds MAXRFBDEV in the positive direction, the PIDER block sets the output windup status (ARWOP) to High, which will prevent CV from going higher. If the deviation exceeds MAXRFBDEV in the negative direction, it sets ARWOP to Low, which will prevent CV from going lower. This occurs only if the tracking control switch (S1) is Off.
Equation Options	• Equation A - Proportional, Integral, and Derivative on error. • Equation B -- Proportional and Integral on error and Derivative on changes in PV. • Equation C -- Integral on error and Proportional and Derivative on changes in PV. • Equation D -- Integral only. • In addtion to the PID equations above, the PIDER block supports equation variations depending on the status of the tracking switch (S1). – Output bias processing which adds fixed and floating bias to unbiased CV. – Reverse-control action causes the sign of the unbiased CV to be reversed.
Gain Options	If equation A, B, or C is selected, any of the following gain options may be chosen: • Linear Gain -- provides proportional control action that is equal to a constant (K) times the error. • Gap Gain -- used to reduce sensitivity of control system when PV is in user-specified band (gap) around set point. • Nonlinear Gain -- control action is proportional to square of error, rather than error itself. • External Gain -- gain (K) is modified by input value that can come from the process, another block or user program.
Direct or Reverse Control	• Direct action -- increase in error increases output (OP). • Reverse action -- increase in error decreases output (OP).

Parameters	ADVDEVALM.DB ADVDEVALM.DBU ADVDEVALM.FL ADVDEVALM.PR ADVDEVALM.SV ADVDEVALM.TM ADVDEVALM.TMO ADVDEVALM.TP ADVDEVOPT ADVSP ADVSPP ALMDB ALMDBU ALMTM ARWNET[1..8] ARWNETIN[1..8] ARWOP ARWOPIN ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT CASREQFL COMPUTEARW CTLACTN CTLEQN CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DELCV DESC DEV DEVHIALM.DB DEVHIALM.DBU DEVHIALM.FL	NLGAIN NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OPTOL

	DEVHIALM.PR DEVHIALM.SV DEVHIALM.TM DEVHIALM.TMO DEVHIALM.TP DEVLOALM.DB DEVLOALM.DBU DEVLOALM.FL DEVLOALM.PR DEVLOALM.SV DEVLOALM.TM DEVLOALM.TMO DEVLOALM.TP ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC FBORSTS GAINHILM GAINLOLM GAINOPT GAPHILM GAPLOLM HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM INITMAN INITREQ[1..8] INITVAL[1..8] INSBLOCK[1..10] INSFAILALM.FL INSFAILALM.PR INSFAILALM.SV INSFAILFL K	OPTYPE ORDERINCM OUTIND OUTTYPE PUSHSP PV PVEUHI PVEULO PVFORMAT PVMANOPT PVP PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCER PVTRAKOPT PVTRAKOPTAI REDTAG RESTARTOPT RFB S1 SAFEOP SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO SECDATAIN.HISELECT SECDATAIN.INITSTS SECDATAIN.INITVAL SECDATAIN.LOCALMAN SECDATAIN.ORFBSTS SECDATAIN.ORFBVAL SECDATAIN.OROFFSET SECDATAIN.PROPOVRD SECINITOPT[1..8] SIALM.FL SIALM.OPT SIALM.PR SIALM.SV SIFL SIOPT SP

	K1	SPEUHI
	KLIN	SPEULO
	KMODIFEXT	SPFORMAT
	KMODIFGAP	SPHIFL
	KMODIFNL	SPHILM
	LASTGOODPV	SPLOFL
	LASTMODEREQ	SPLOLM
	LASTOPREQ	SPP
	LASTOPTYPE	SPRATEREQ
	LASTRATEREQ	SPREQ
	LASTREQFL	SPTOL
	LASTSPREQ	SPTV
	LASTSPTVREQ	SPTVDEVFL
	LASTSTEP	SPTVDEVMAX
	LEGACYGAP	SPTVNORMRATE
	MAXRFBDEV	SPTVOPT
	MODE	SPTVP
	MODEAPPL[1..4]	SPTVRATE
	MODEATTR	SPTVREQ
	MODEATTRFL.NORM	SPTVSTATE
	MODEATTRFL.OPER	SPTVTIME
	MODEATTRFL.PROG	STARTOPT
	MODECHANGE	STARTRATE
	MODEFL.AUTO	STARTVAL
	MODEFL.BCAS	STOPOPT
	MODEFL.CAS	STOPRATE
	MODEFL.MAN	STOPVAL
	MODEFL.NORM	T1
	MODEPERM	T1HILM
	MODEREQ	T1LOLM
	MODETRACK	T2
	NAME	T2HILM
	NLFORM	T2LOLM
		TMOUTFL
		TMOUTMODE
		TMOUTTIME
		TRFB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the PIDER Block.ck.	

3.11.7 PID-PL block

Description	A regulatory control block combining the functionality of PID controller with a robust, model-based, predictive controller and optimizer, Profit Loop PKS. In many cases, Profit Loop PKS provides superior control and should be used in place of the standard PID algorithms. See the *Control Builder Components Theory, Regulatory Control* section, *PID-PL (Profit Loop PKS) Block* for a discussion of benefits.
Function	Belongs to a class of controllers known as “model predictive control.” These controllers rely on a dynamic model to predict future movement in the process variable. If the predicted PV does not meet the control objectives (maintain at current setpoint), control action is taken to realign the PV with its objectives. In contrast, a PID controller uses past and current error trajectories to restore the PV to its SP within one control move, regardless of the long-term consequences of the move. With R410, PID-PL block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms. See the *Control Builder Components Theory, Regulatory Control* section, *PID-PL (Profit Loop PKS) Block* for a discussion of benefits.
Timeout Monitoring	The PID-PL block monitors for communication timeouts between primary and secondary controllers of a cascade pair. This block uses the same methodology as the PID block. See *Timeout Monitoring* for PID block for implementation details.
Inputs	The PID-PL block requires two inputs: PV and SP. See *Required Inputs* for PID block for details on these inputs. For range control, the PID-PL block requires two additional inputs: SPHI and SPLO. These parameters cannot be initialized. They can be pulled from another block, set through operator entry, or stored by a user program.
Input Ranges and Limits	For PID or Profit Loop PKS setpoint control, the PID range limits apply. See *Input ranges and limits* for PID block for details. For range control, • PVEUHI and PVEULO define the engineering unit range for MODELPV, SP, SPHI, and SPLO. • SPHILM and SPLOLM define the operating limits in engineering units for SPHI, and SPLO. If a value is entered outside these limits but within the engineering unit range, this block clamps the value at its appropriate limit. Otherwise, an out of bounds error is generated. • SPHI and SPLO define the operating limits in engineering units for SP. The operator is prevented from storing a setpoint value that is outside these limits. If the primary or a user program attempts to store a value outside of the limits, this block clamps it to the appropriate limit and sets the primary's windup status.
Outputs	The PID-PL block supports a single initializable output. Like PID, this calculated output can be either in percent, OP, or in engineering units, OPEU. See *Initializable outputs* for PID block for more details.
Output Ranges and Limits	The output range of a PID-PL block is identical to the PID block. See *Output ranges and limits* for PID block for details.
Equation Options	The PID-PL block supports the standard five PID equations (EqA through EqE) as well as the Profit Loop equation, PROFITLOOP. For details on the PROFITLOOP equation type, see the *Control Builder Components Theory, Regulatory Control* section, *PID-PL (Profit Loop PKS) Block* for a discussion of benefits. For details on the PID equations, see *PID equations*.

Gain Options	The gain options only apply for applicable PID equations A, B, or C. The PROFITLOOP equation does not support the gain options. See *Gain options* for use with PID equations.
Direct or Reverse Control	A PID-PL block may be configured for direct-control action or reverse-control action, effectively changing the sign of the controller gain. See *Direct or reverse control* for PID block for implementation details. With Profit Loop PKS, the sign of the controller gain is determined by the sign of the process model gain. To prevent mismatch between the model gain and control action, changes to CTLACTN are not allowed. Furthermore, when the model gain changes (including initial loading of the function block), CTLACTN is set to match the new gain direction. • A positive process model gain leads to reverse control action. • A negative process model gain leads to direct control action.
Parameters	All PID parameters apply to the PID-PL block. They are listed here. Parameters unique to the PID-PL block are presented in bold.

	ADVDEVALM.DB	OPEXHIFL
	ADVDEVALM.DBU	OPEXHILM
	ADVDEVALM.FL	OPEXLOFL
	ADVDEVALM.PR	OPEXLOLM
	ADVDEVALM.SV	OPHIACTIVE
	ADVDEVALM.TM	OPHIALM.DB
	ADVDEVALM.TMO	OPHIALM.DBU
	ADVDEVALM.TP	OPHIALM.FL
	ADVDEVOPT	OPHIALM.PR
	ADVSP	OPHIALM.SV
	ADVSPP	OPHIALM.TM
	ALMDB	OPHIALM.TMO
	ALMDBU	OPHIALM.TP
	ALMTM	OPHIFL
	ARWNET[1..8]	OPHILM
	ARWOP	OPHILMOPT
	ARWRATIO	OPHIOPTOFFSET
	ASTEPID	OPLOACTIVE
	BADCTLALM.FL	OPLOALM.DB
	BADCTLALM.PR	OPLOALM.DBU
	BADCTLALM.SV	OPLOALM.FL
	BADCTLFL	OPLOALM.PR
	BADCTLOPT	OPLOALM.SV
	BADOCOPT	OPLOALM.TM
	BADOCOPTENB	OPLOALM.TMO
	BADPVALM.FL	OPLOALM.TP
	BADPVALM.PR	OPLOFL
	BADPVALM.SV	OPLOLM
	BADCTLALM.TM	OPLOLMOPT
	BADCTLALM.TMO	OPLOOPTOFFSET
	BADPVALM.TP	OPMINCHG
	CALIBRATION	OPREQ
	CASREQFL	OPROCLM
	CLOSEDLOOPRESP	OPROCNEGFL
	COMPUTEARW	OPROCPOSFL
	CTLACTN	OPTMODE
	CTLEQN	OPTSPEED
	CTLINIT	OPTYPE
	CTLSTATE	ORDERINCM
	CTRLMODE	OUTIND
	CV	OUTTYPE
	CVEUHI	PREDPVHIALM.FL

	CVEULO	PREDPVHIALM.PR
	CVTYPE	PREDPVHIALM.SV
	D[1..5]	PREDPVHIALM.TP
	DELCV	PREDPVLOALM.FL
	DESC	PREDPVLOALM.PR
	DEV	PREDPVLOALM.SV
	DEVHIALM.DB	PREDPVHIALM.TM
	DEVHIALM.DBU	PREDPVHIALM.TMO
	DEVHIALM.FL	PREDPVLOALM.TP
	DEVHIALM.PR	PRFRATIO
	DEVHIALM.SV	PRIM.[1..8].INITIALIZABLE
	DEVHIALM.TM	PRIMDATA.[1..8].HISELECT
	DEVHIALM.TMO	PRIMDATA.[1..8].ORFBSTS
	DEVHIALM.TP	PRIMDATA.[1..8].ORFBVAL
	DEVLOALM.DB	PRIMDATA.[1..8].OROFFSET
	DEVLOALM.DBU	PRIMDATA.[1..8].PROPOVRD
	DEVLOALM.FL	PROCDEADTIME
	DEVLOALM.PR	PROCDEADTIMEACT
	DEVLOALM.SV	PROCDEADTIMEBIAS
	DEVLOALM.TM	PROCGAINACT
	DEVLOALM.TMO	PROCGAINEU
	DEVLOALM.TP	PROCGAINMULT
	ESWENB	PROCGAINPCT
	ESWFL.AUTO	PROCGAINUNITS
	ESWFL.BCAS	PUSHSP
	ESWFL.CAS	PV
	ESWFL.MAN	PVASYNCOPT
	ESWPERM	PVEUHI
	EUDESC	PVEULO
	FBORSTS	PVFORMAT
	GAINHILM	PVMANOPT
	GAINLOLM	PVP
	GAINOPT	PVSTS
	GAPHILM	PVSTSFL.BAD
	GAPLOLM	PVSTSFL.MAN
	HIALM.PR	PVSTSFL.NORM
	HIALM.SV	PVSTSFL.UNCER
	HIALM.TYPE	PVTRAKOPT
	HOLDOPT	PVTRAKOPTAI
	HOLDRATE	REDTAG
	HOLDVAL	RESET
	INALM	RESETMODEL

	INITMAN	RESTARTOPT
	INITREQ[1..8]	SAFEOP
	INITVAL[1..8]	SECDATAIN.ARWSTS
	INSBLOCK[1..10]	SECDATAIN.EUHI
	INSFAILALM.FL	SECDATAIN.EULO
	INSFAILALM.PR	SECDATAIN.HISELECT
	INSFAILALM.SV	SECDATAIN.INITSTS
	INSFAILFL	SECDATAIN.INITVAL
	K]	SECDATAIN.LOCALMAN
	KLIN	SECDATAIN.ORFBSTS
	KMODIFEXT	SECDATAIN.ORFBVAL
	KMODIFGAP	SECDATAIN.OROFFSET
	KMODIFNL	SECDATAIN.PROPOVRD
	LASTGOODPV	SECINITOPT[1..8]
	LASTMODEREQ	SIALM.FL
	LASTOPREQ	SIALM.OPT
	LASTOPTYPE	SIALM.PR
	LASTRATEREQ	SIALM.SV
	LASTREQFL	SIFL
	LASTSPREQ	SIOPT
	LASTSPTVREQ	SP
	LASTSTEP	SPEUHI
	LEGACYGAP	SPEULO
	MODE	SPFORMAT
	MODEAPPL[1..4]	SPHI
	MODEATTR	SPHIACTIVE
	MODEATTRFL.NORM	SPHIFL
	MODEATTRFL.OPER	SPHILM
	MODEATTRFL.PROG	SPHILMOPT
	MODECHANGE	SPHIOPTOFFSET
	MODEFL.AUTO	SPHIRAMPRATE
	MODEFL.BCAS	SPLO
	MODEFL.CAS	SPLOACTIVE
	MODEFL.MAN	SPLOFL
	MODEFL.NORM	SPLOLM
	MODELPV	SPLOLMOPT
	MODEPERM	SPLOOPTOFFSET
	MODEREQ	SPLORAMPRATE
	MODETRACK	SPP
	N[1..5]	SPRATEREQ
	NEWSAMPLE	SPREQ
	NLFORM	SPTV

	NLGAIN	SPTVDEVFL
	NORMMODE	SPTVDEVMAX
	NORMMODEATTR	SPTVNORMRATE
	NUMCOEFDEN	SPTVOPT
	NUMCOEFNUM	SPTVP
	NUMINSERT	SPTVRATE
	NUMPRI	SPTVREQ
	NUMSEC	SPTVSTATE
	OP	SPTVTIME
	OPBIAS	STARTOPT
	OPBIAS.FIX	STARTRATE
	OPBIAS.FLOAT	STARTVAL
	OPBIAS.RATE	STEADYSTATEOP
	OPENLOOPRESP	STEADYSTATEOPEU
	OPEU	STEADYSTATEPV
	OPEUMOVE	STOPOPT
		STOPRATE
		STOPVAL
		T1
		T1HILM
		T1LOLM
		T2
		T2HILM
		T2LOLM
		TMOUTFL
		TMOUTMODE
		TMOUTTIME
		UNCMDCHGALM.FL
		UNCMDCHGALM.OPT
		UNCMDCHGALM.PR
		UNCMDCHGALM.SV
		UPDATEMODEL
		VALVETRAVELRDCT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the PID-PL Block.	

3.11.8 PIDFF (PID Feedforward) block

Description	The PIDFF block is like the PID block but it accepts a feedforward signal as an additional input. You can configure the PIDFF block so the feedforward signal is added to or multiplied by the normal PID algorithm's incremental output to meet your particular control requirements.

Function	• The multiplicative feedforward action is typically used to compensate for variations in process gain that are caused by changes in the throughput. It is usually used with a lead/lag relay to provide dynamic feedforward control for a given application. For example, if the feed rate is doubled in a heating application, twice the amount of fuel might be required, which is equivalent to doubling the process gain. • Includes the feedforward signal (FF) in the calculation of the PID's incremental output before the full value output is accumulated. • With R410, PIDFF block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on SP. If a good SP value is not received within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL). • Holds the SP value at its last good value. • Changes the mode to a user-specified TMOUTMODE. • Requests the input's primary to initialize. If SP times out and the block sheds to Auto mode, block sets its Cascade Request flag (CASREQFL). Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Inputs	• Requires both PV and FF inputs to provide its feedforward function. The PV and FF inputs must be pulled from other blocks; you cannot store to them. The feedforward signal may come from a field device (via an IA channel block) or an Auxiliary function block. Field inputs are typically subjected to deadtime or lead-lag compensation before being connected to the FF input of this block, which may be provided by the Deadtime or Lead-Lag Auxiliary function blocks. • The SP input is not required, since it does not have to be pulled from another function block. – If Mode is CAScade and the SP is pulled from another function block, it receives its value from an upstream primary and it is an initializable input. – If Mode is CAScade and the SP is **not** connected to another function block, the value of the SP is frozen at the last acquired value. – If Mode is AUTOmatic, the SP value may be stored by the operator or a user program. • SP is an initializable input; PV and FF are non-initializable.
Input Ranges and Limits	• PVEUHI and PVEULO define full range of PV in engineering units. They also define the engineering unit range of SP, since PV and SP are assumed to have the same range. – PVEUHI represents the 100% of full scale value. – PVEULO represents the 0% of full scale value. • SPHILM and SPLOLM define set point operating limits in engineering units. – Prevents operator from storing SP value outside limits; if primary or user program attempts to store value outside limits, block clamps it to appropriate limit and sets primary's windup status.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.

<table>
<tr><td>Output Ranges and Limits</td><td>CVEUHI and CVEULO define full range of CV in engineering units. If this block has a secondary, it brings the secondary's input range through the BACKCALC and sets its CV range to that. If it has no secondary, you must specify CVEUHI and CVEULO range.
OPHILM and OPLOLM define normal high and low limits for OP as a percent of CV range (user-specified values).
OP clamps to limits if calculated CV exceeds them, or another block or user program attempts to store OP value exceeding them; operator may store OP value outside these limits.
OPEXHILM and OPEXLOLM define extended high and low limits for OP as percent of CV range (user-entered values).
Prevents operator from storing OP value that exceeds these limits.</td></tr>
<tr><td>Equation Options</td><td>Equation A - Proportional, Integral, and Derivative on error.
Equation B -- Proportional and Integral on error and Derivative on changes in PV.
Equation C -- Integral on error and Proportional and Derivative on changes in PV.
Equation D -- Integral only.
Equation E -- Proportional only; this equation supports the following two options that affect CV:
Output bias processing which adds fixed and floating bias to unbiased CV.
Reverse-control action causes the sign of the unbiased CV to be reversed.</td></tr>
<tr><td>Gain Options</td><td>If equation A, B, or C is selected, any of the following gain options may be chosen:
Linear Gain -- provides proportional control action that is equal to a constant (K) times the error.
Gap Gain -- used to reduce sensitivity of control system when PV is in user-specified band (gap) around set point.
Nonlinear Gain -- control action is proportional to square of error, rather than error itself.
External Gain -- gain (K) is modified by input value that can come from the process, another block or user program.</td></tr>
<tr><td>Direct or Reverse Control</td><td>Direct action -- increase in error increases output (OP).
Reverse action -- increase in error decreases output (OP).</td></tr>
</table>

Parameters	ADVDEVALM.DB ADVDEVALM.DBU ADVDEVALM.FL ADVDEVALM.PR ADVDEVALM.SV ADVDEVALM.TM ADVDEVALM.TMO ADVDEVALM.TP ADVDEVOPT ADVSP ADVSPP ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB BFF CASREQFL COMPUTEARW CTLACTN CTLEQN CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DELCV DESC DEV DEVHIALM.DB	NLFORM NLGAIN NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL

	DEVHIALM.DBU	OPTYPE
	DEVHIALM.FL	ORDERINCM
	DEVHIALM.PR	OUTIND
	DEVHIALM.SV	OUTTYPE
	DEVHIALM.TM	PRIM.[1..8].INITIALIZABLE
	DEVHIALM.TMO	PRIMDATA.[1..8].HISELECT
	DEVHIALM.TP	PRIMDATA.[1..8].ORFBSTS
	DEVLOALM.DB	PRIMDATA.[1..8].ORFBVAL
	DEVLOALM.DBU	PRIMDATA.[1..8].OROFFSET
	DEVLOALM.FL	PRIMDATA.[1..8].PROPOVRD
	DEVLOALM.PR	PUSHSP
	DEVLOALM.SV	PV
	DEVLOALM.TM	PVEUHI
	DEVLOALM.TMO	PVEULO
	DEVLOALM.TP	PVFORMAT
	ESWENB	PVMANOPT
	ESWFL.AUTO	PVP
	ESWFL.BCAS	PVSTS
	ESWFL.CAS	PVSTSFL.BAD
	ESWFL.MAN	PVSTSFL.MAN
	ESWPERM	PVSTSFL.NORM
	EUDESC	PVSTSFL.UNCER
	FBORSTS	PVTRAKOPT
	FF	PVTRAKOPTAI
	FFOPT	REDTAG
	FFSTS	RESTARTOPT
	GAINHILM	SAFEOP
	GAINLOLM	SECDATAIN.ARWSTS
	GAINOPT	SECDATAIN.EUHI
	GAPHILM	SECDATAIN.EULO
	GAPLOLM	SECDATAIN.HISELECT
	HIALM.PR	SECDATAIN.INITSTS
	HIALM.SV	SECDATAIN.INITVAL
	HIALM.TYPE	SECDATAIN.LOCALMAN
	HOLDOPT	SECDATAIN.ORFBSTS
	HOLDRATE	SECDATAIN.ORFBVAL
	HOLDVAL	SECDATAIN.OROFFSET
	INALM	SECDATAIN.PROPOVRD
	INITMAN	SECINITOPT[1..8]
	INITREQ[1..8]	SIALM.FL
	INITVAL[1..8]	SIALM.OPT
	INSBLOCK[1..10]	SIALM.PR

	INSFAILALM.FL	SIALM.SV
	INSFAILALM.PR	SIFL
	INSFAILALM.SV	SIOPT
	INSFAILFL	SP
	K	SPEUHI
	KFF	SPEULO
	KLIN	SPFORMAT
	KMODIFEXT	SPHIFL
	KMODIFGAP	SPHILM
	KMODIFNL	SPLOFL
	LASTGOODPV	SPLOLM
	LASTMODEREQ	SPP
	LASTOPREQ	SPRATEREQ
	LASTOPTYPE	SPREQ
	LASTRATEREQ	SPTV
	LASTREQFL	SPTVDEVFL
	LASTSPREQ	SPTVDEVMAX
	LASTSPTVREQ	SPTVNORMRATE
	LASTSTEP	SPTVOPT
	LEGACYGAP	SPTVP
	MODE	SPTVRATE
	MODEAPPL[1..4]	SPTVREQ
	MODEATTR	SPTVSTATE
	MODEATTRFL.NORM	SPTVTIME
	MODEATTRFL.OPER	STARTOPT
	MODEATTRFL.PROG	STARTRATE
	MODECHANGE	STARTVAL
	MODEFL.AUTO	STOPOPT
	MODEFL.BCAS	STOPRATE
	MODEFL.CAS	STOPVAL
	MODEFL.MAN	T1
	MODEFL.NORM	T1HILM
	MODEPERM	T1LOLM
	MODEREQ	T2
	MODETRACK	T2HILM
	NAME	T2LOLM
		TMOUTFL
		TMOUTMODE
		TMOUTTIME
		UNCMDCHGALM.FL
		UNCMDCHGALM.OPT

		UNCMDCHGALM.PR UNCMDCHGALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the PIDFF (PID Feedforward) Block.	

3.11.9 POSPROP (Position Proportional) block

Description	Used to pulse two digital output devices (one for raise pulses and another for lower pulses) to drive a process variable (PV) toward its set point (SP). The only valid output destinations are to Digital Output Channel blocks or the Pulse Count and Pulse Length blocks.
Function	• Typically used to step a valve open or closed, raise or lower a rotary device, or move the plates of a pulp mill refiner together or apart. • Compares the error signal (PV - SP) with an error deadband for the raise and lower directions at an interval based on the configurable cycle time parameter (CYCLETIME). You can also configure the raise and lower deadband values that are denoted as the parameters ERRORDBR and ERRORDBL, respectively. • Generates a raise pulse, when the PV is less than the SP minus the raise error deadband (ERRORDBR); or a lower pulse, when the PV is greater than the SP plus the lower error deadband (ERRORDBL) to reduce the error. • The pulse duration determines the magnitude of a pulse - the longer the duration, the bigger the pulse. The POSPROP block will not issue a raise or lower pulse that is longer than the configured cycle time (CYCLETIME) or the respective maximum pulse time parameter MAXPULSER or MAXPULSEL, whichever is smaller. The block uses the following values in its pulse duration calculation. – Error signal (PV - SP) – Raise or lower gain setting (KR or KL) – Raise or lower pulse stroke rate (RAISERATE or LOWERRATE) – Additional raise or lower pulse time (RAISEDEADTM or LOWERDEADTM) based on stiction compensation (STICTIONR or STICTIONL), when a motor starts up; or backlash compensation (BACKLASHR or BACKLASHL), when a motor changes direction. – Minimum raise or lower pulse time (MINPULSER or MINPULSEL) • With R410, POSPROP block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on SP. If a good SP value is not received within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL) • Holds the SP value at its last good value. • Changes the mode to a user-specified TMOUTMODE. • Requests the input's primary to initialize. If SP times out and the block sheds to Auto mode, block sets its Cascade Request flag (CASREQFL). Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.

<table>
<tr><td>Inputs</td><td>The required number of inputs is determined by the mode of the POSPROP block.
• If Mode is CAScade, two inputs are required - PV and SP.
• If Mode is AUTOmatic or MANual, only PV is required.
– SP is an initializable input; PV is non-initializable.
– PV must be pulled from another block; you cannot store to it - typically it is connected to the output of an auxiliary or data acquisition (DATAACQ) block.
– If Mode is CAScade, SP is pulled from another block; if Mode is AUTOmatic, it may be stored by the operator.
– The POSPROP block may have one primary or none, depending on whether SP is configured or not; there is one primary per initializable input.
The optional raise and lower flag inputs (RAISELMFL and LOWERLMFL) may be set externally to inhibit raise and lower pulses, respectively. These optional inputs can be pulled from other function blocks.</td></tr>
<tr><td>Input Ranges and Limits</td><td>• PVEUHI and PVEULO define full range of PV in engineering units. They also define the engineering unit range of SP, since PV and SP are assumed to have the same range.
– PVEUHI represents the 100% of full scale value.
– PVEULO represents the 0% of full scale value.
• SPHILM and SPLOLM define set point operating limits in engineering units.
Prevents operator from storing SP value outside limits; if primary or user program attempts to store value outside limits, block clamps it to appropriate limit and sets primary's windup status.</td></tr>
<tr><td>Outputs</td><td>The POSPROP block has the following initializable outputs:
• RAISETIME = Raise pulse duration.
• LOWERTIME = Lower pulse duration.
• PULSETIME = Pulse duration.</td></tr>
<tr><td>Output Ranges and Limits</td><td>The POSPROP block uses the maximum and minimum pulse parameters to define pulse duration ranges and limits.
• MAXPULSER and MAXPULSEL define the maximum pulse time in the Raise and Lower directions, respectively. The POSPROP block will not issue a Raise/Lower pulse with a duration that exceeds these values. If the output and CYCLETIME are greater than MAXPULSER/MAXPULSEL, the output is clamped to MAXPULSER/MAXPULSEL.
• MINPULSER and MINPULSEL define the minimum pulse time in the Raise and Lower directions, respectively. The POSPROP block will not issue a Raise/Lower pulse with a duration that is less than these values. If the output is less than MINPULSER/MINPULSEL, the output retains its old value.
(Note that the POSPROP block does not use these common regulatory control block range and limit parameters: CVEUHI, CVEULO, OPHILM, OPLOLM, OPEXHILM, and OPEXLOLM.)</td></tr>
<tr><td>Equation Options</td><td>The POSPROP block generates Raise and Lower pulses at a rate specified by the configurable cycle time (CYCLETIME) parameter. It calculates the pulse duration at the beginning of each cycle depending on whether:
• The PVP is greater than (SPP - ERRORDBR) and the Raise limit flag (RAISELMFL) is OFF, then issue a Raise pulse.
• The PVP is less than (SPP + ERRORDBL) and the Lower limit flag (LOWERLMFL) is OFF, then issue a Lower pulse.
The PULSETIME output is set to either the RAISETIME or -LOWERTIME, when either RAISETIME or LOWERTIME is non-zero.</td></tr>
</table>

Control Initialization	The POSPROP block accepts initialization information from its three initializable outputs: RAISETIME, LOWERTIME, and PULSETIME. If any output requests initialization, the POSPROP block sets its INITMAN parameter to ON. When no output requests initialization, the POSPROP block sets its INITMAN parameter to OFF. When cycling resumes after initialization, the Raise and Lower outputs are both set to OFF (or their normal states) and the cycle time is restarted.
Override Feedback Processing	The POSPROP block does not propagate override feedback data. It ignores any override feedback requests.

Parameters	ADVDEVALM.DB ADVDEVALM.DBU ADVDEVALM.FL ADVDEVALM.PR ADVDEVALM.SV ADVDEVALM.TM ADVDEVALM.TMO ADVDEVALM.TP ADVDEVOPT ADVSP ADVSPP ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BACKLASHL BACKLASHR BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT CASREQFL COMPUTEARW CTLINIT CVTYPE CYCLETIME DESC DEV DEVHIALM.DB DEVHIALM.DBU DEVHIALM.FL DEVHIALM.PR DEVHIALM.SV DEVHIALM.TM DEVHIALM.TMO DEVHIALM.TP	MINPULSER MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ MODETRACK NAME NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OPREQ OPTYPE ORDERINCM OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS PRIMDATA.[1..8].ORFBVAL PRIMDATA.[1..8].OROFFSET PRIMDATA.[1..8].PROPOVRD PULSECMD PULSECMDTEXT[0..2] PULSETIME PV PVEUHI PVEULO PVFORMAT PVMANOPT PVP PVSTS

	DEVLOALM.DB DEVLOALM.DBU DEVLOALM.FL DEVLOALM.PR DEVLOALM.SV DEVLOALM.TM DEVLOALM.TMO DEVLOALM.TP ERRORDBL ERRORDBR ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC EXTRAPULSE EXTRAPULSETM FBORSTS HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM INITMAN INITREQ[1..8] INITVAL[1..8] INSBLOCK[1..10] INSFAILALM.FL INSFAILALM.PR INSFAILALM.SV INSFAILFL KL KR LASTGOODPV LASTMODEREQ LASTOPREQ LASTOPTYPE LASTRATEREQ	PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN RAISEDEADTM RAISEDESC RAISELMFL RAISERATE RAISETIME REDTAG RESTARTOPT SAFEOPCMD SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO SECDATAIN.HISELECT SECDATAIN.INITSTS SECDATAIN.INITVAL SECDATAIN.LOCALMAN SECDATAIN.ORFBSTS SECDATAIN.ORFBVAL SECDATAIN.OROFFSET SECDATAIN.PROPOVRD SECINITOPT[1..8] SIALM.FL SIALM.OPT SIALM.PR SIALM.SV SIFL SIOPT SP SPEUHI SPEULO SPFORMAT SPHIFL SPHILM SPLOFL SPLOLM SPP SPRATEREQ SPREQ SPTV

	LASTREQFL	SPTVDEVFL
	LASTSPREQ	SPTVDEVMAX
	LASTSPTVREQ	SPTVNORMRATE
	LASTSTEP	SPTVOPT
	LOWERDEADTM	SPTVP
	LOWERDESC	SPTVRATE
	LOWERLMFL	SPTVREQ
	LOWERRATE	SPTVSTATE
	LOWERTIME	SPTVTIME
	MANPULSECMD	STARTOPT
	MANPULSETIME	STARTRATE
	MAXPULSEL	STARTVAL
	MAXPULSER	STICTIONL
	MINPULSEL	STICTIONR
		STOPOPT
		STOPRATE
		STOPVAL
		TMOUTFL
		TMOUTMODE
		TMOUTTIME
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the POSPROP (Position Proportional) Block.	

3.11.10 PULSECOUNT block

Description	The PULSECOUNT block generates pulses according to its pulse count control algorithm. The pulsed outputs are usually fed to Digital Output Channel blocks. Dual Pulse Train: A control algorithm turns on either a “raise” channel or a “lower” channel after every execution of this algorithm. The output is modulated with a 50% duty-cycle pulse train. The on-duration (or pulse length) is configured for the channel and is indicted with a tuning parameter. The calculated on-duration will be in 10 msec increments. Single Pulse Train: A single output channel is used to indicate the direction (raise or lower) of the actuator. A second output channel is used to deliver a 50% duty cycle pulse train. The on-duration (or pulse length) is configured for the channel and is indicted with a tuning parameter. The calculated on-duration will be in 10 msec increments.
Function	• Typically used in conjunction with a POSPROP block to step a valve open or closed, raise or lower a rotary device, or move the plates of a pulp mill refiner together or apart. • The POSPROP block feeds the PULSETIME input parameter to the PULSECOUNT block. This parameter is an internal structure that contains the pulse width specification (in seconds). It also contains a Serial Number that changes every time there is a new pulse width value. The PULSECOUNT block checks for a change in the Serial Number before reacting to the pulse width specification.

Inputs	• Requires a pulse time (PULSETIME) input from another block. A POSPROP block usually supplies this. • The POPERIOD input is user configurable in seconds. • The PDELAYDIRCHG input is user configurable in seconds. • The optional LOCALMAN input should come from another block in a logic strategy where an ON condition means that the CEE is not controlling the output of the device.	
Outputs	The PULSECOUNT block has the following initializable outputs: • PORAISE = Pulse output for Raise pulses. These pulses are generated if the pulse width specified by the PULSETIME input is positive. • POLOWER = Pulse output for Lower pulses. These pulses are generated if the pulse width specified by the PULSETIME input is negative. • PO = Pulse output for both Raise and Lower pulses. These pulses are generated as a logical OR between the PORAISE and POLOWER pulses. • PODIR = Direction for PO. This output is OFF for a Lower pulse and is ON for a Raise pulse.	
Parameters	COMPUTEARW	PODIR
	CVTYPE	POLOWER
	INITMAN	POPERIOD
	INITREQ	PORAISE
	INSBLOCK[1..10]	PRIM.[1..8].INITIALIZABLE
	INSFAILALM.FL	PRIMDATA.[1..8].HISELECT
	INSFAILALM.PR	PRIMDATA.[1..8].ORFBSTS
	INSFAILALM.SV	PRIMDATA.[1..8].ORFBVAL
	INSFAILFL	PRIMDATA.[1..8].OROFFSET
	LOCALMAN	PRIMDATA.[1..8].PROPOVRD
	MODECHANGE	PULSETIME
	NAME	SECDATAIN.HISELECT
	NUMINSERT	SECDATAIN.ORFBSTS
	ORDERINCM	SECDATAIN.ORFBVAL
	OUTTYPE	SECDATAIN.OROFFSET
	PDELAYDIRCHG	
	PO	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the PULSECOUNT Block.	

3.11.11 PULSELENGTH block

Description	Generates pulse trains according to its pulse length control algorithm. The pulsed outputs are usually fed to Digital Output Channel blocks. Dual Pulse Length: A control algorithm turns on either a “raise” channel or a “lower” channel after every execution of this algorithm. The selected output stays on for a time period that is calculated by the control algorithm. The calculated on-duration will be in 10 msec increments. Single Pulse Length: A single output channel is used to indicate the direction (raise or lower) of the actuator. A second output channel is used to indicate the calculated on-duration (or length) of the pulse. The calculated on-duration will be in 10 msec increments.

<table>
<tr><td>Function</td><td colspan="2">

Typically used in conjunction with a POSPROP block to step a valve open or closed, raise or lower a rotary device, or move the plates of a pulp mill refiner together or apart.
The POSPROP block feeds the PULSETIME input parameter to the PULSELENGTH block. This parameter is an internal structure that contains the pulse width specification (in seconds). It also contains a Serial Number that changes every time there is a new pulse width value. The PULSELENGTH block checks for a change in the Serial Number before reacting to the pulse width specification.
</td></tr>
<tr><td>Inputs</td><td colspan="2">

Requires a pulse time (PULSETIME) input from another block. A POSPROP block usually supplies this.
The PDELAYDIRCHG input is user configurable in seconds.
The optional LOCALMAN input should come from another block in a logic strategy where an ON condition means that the CEE is not controlling the output of the device.
</td></tr>
<tr><td>Outputs</td><td colspan="2">The PULSELENGTH block has the following initializable outputs:

PORAISE = Pulse output for Raise pulses. These pulses are generated if the pulse width specified by the PULSETIME input is positive.
POLOWER = Pulse output for Lower pulses. These pulses are generated if the pulse width specified by the PULSETIME input is negative.
PO = Pulse output for both Raise and Lower pulses. These pulses are generated as a logical OR between the PORAISE and POLOWER pulses.
PODIR = Direction for PO. This output is OFF for a Lower pulse and is ON for a Raise pulse.
</td></tr>
<tr><td>Parameters</td><td>COMPUTEARW
CTLSTATE
CVTYPE
INITMAN
INITREQ
INSBLOCK[1..10]
INSFAILALM.FL
INSFAILALM.PR
INSFAILALM.SV
INSFAILFL
LOCALMAN
MODECHANGE
NAME
NUMINSERT
ORDERINCM
OUTTYPE
PDELAYDIRCHG</td><td>PO
PODIR
POLOWER
PORAISE
PRIM.[1..8].INITIALIZABLE
PRIMDATA.[1..8].HISELECT
PRIMDATA.[1..8].ORFBSTS
PRIMDATA.[1..8].ORFBVAL
PRIMDATA.[1..8].OROFFSET
PRIMDATA.[1..8].PROPOVRD
PULSETIME
SECDATAIN.HISELECT
SECDATAIN.ORFBSTS
SECDATAIN.ORFBVAL
SECDATAIN.OROFFSET</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.

Refer to the Control Builder Component Theory for more information on the PULSELENGTH Block.</td></tr>
</table>

3.11.12 RAMPSOAK block

Description	Provides an output that tracks a user configured set point versus time profile. The block supports up to 99 separate profiles with up to 50 user configured ramp and soak segment pairs per profile, for a total of 60 segments (where each segment is one ramp or one soak) Each ramp/soak pair is defined by a soak value (i.e., the target value for the ramp segment), a ramp rate and a soak time. This lets you implement a set point program control function by driving the set point of another regulatory control function block.
Function	This function is also known as a “set point programmer” because the output follows a sequence of user-programmed functions, and is typically used as the set point of a PID. Typically used for automatic temperature cycling in furnaces and ovens. It can also be used for automatic startup of units and for simple batch-sequence control where the batch sequence is part of a process that is otherwise a continuous process. This block monitors an input value (typically the PV of the PID), and guarantees that its output will not deviate from the input by more than some user-specified limits. This function block may be configured to execute a profile once and stop; repeat continuously the same profile; or execute the next profile in order after completion of the current profile. With R410, RAMSOAK block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Inputs	Only requires a PV input for the guaranteed ramp option.
Input Ranges and Limits	PVEUHI and PVEULO define full range of PV in engineering units. The default range is 0 to 100. • PVEUHI represents the 100% of full scale value. • PVEULO represents the 0% of full scale value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges and Limits	CVEUHI and CVEULO define full range of CV in engineering units. If this block has a secondary, it brings the secondary's input range through the BACKCALC and sets its CV range to that. If it has no secondary, you must specify CVEUHI and CVEULO range. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of CV range (user-specified values). – OP clamps to limits if calculated CV exceeds them, or another block or user program attempts to store OP value exceeding them; operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define extended high and low limits for OP as percent of CV range (user-entered values). Prevents operator from storing OP value that exceeds these limits. that the RAMPSOAK block does not apply a floating bias to the output.
Guaranteed Ramp Rate	If you configure a maximum ramp deviation (MAXRAMPDEV[n]) value for a given profile, the RAMPSOAK block makes sure that the calculated output (CV) value does not deviate from the input (PV) by more than the configured deviation value
Guaranteed Soak Time	If you configure the maximum high soak deviation (MAXHISOAKDEV[n]) and/or the maximum low soak deviation (MAXLOSOAKDEV[n]) value, the RAMPSOAK block makes sure the calculated output (CV) value is at the proper value before it starts the soak timer.

Event Timers	You can configure up to 16 event flags (EVENTFL[n,e]) to provide Boolean outputs for a specified time during a given ramp or soak segment in a given profile. This means you can have up to 16 events per profile or a total of 160 events in 10 profiles.

Parameters	ACTRAMPRATE[1..NUMPROFILES][1..NUMRAMPSOAK] ACTSOAKTIME[1..NUMPROFILES][1..NUMRAMPSOAK] ACTSOAKVAL[1..NUMPROFILES][1..NUMRAMPSOAK] ACTSTARTOP[1..NUMPROFILES] ACTSTARTSEG[1..NUMPROFILES] ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADOCOPT BADOCOPTENB CASREQFL COMPUTEARW CTLINIT CTLSTATE CURPROFILEID CURSEGID CURSEGTYP CV CVEUHI CVEULO CVTYPE CYCLEOPT[1..NUMPROFILES] DESC DEVHIALM.DB DEVHIALM.DBU DEVHIALM.FL DEVHIALM.PR DEVHIALM.SV DEVHIALM.TM DEVHIALM.TMO DEVHIALM.TP DEVLOALM.DB DEVLOALM.DBU DEVLOALM.FL DEVLOALM.PR DEVLOALM.SV DEVLOALM.TM	NAME NETELAPSEDTM NORMMODE NORMMODEATTR NUMEVENTS[1..NUMPROFILES] NUMINSERT NUMPRI NUMPROFILES NUMRAMPSOAK[1..NUMPROFILES] NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ

	DEVLOALM.TMO	OPROCLM
	DEVLOALM.TP	OPROCNEGFL
	ESWENB	OPROCPOSFL
	ESWFL.AUTO	OPTYPE
	ESWFL.BCAS	ORDERINCM
	ESWFL.CAS	OUTIND
	ESWFL.MAN	OUTTYPE
	ESWPERM	PRIM.[1..8].INITIALIZABLE
	EUDESC	PRIMDATA.[1..8].HISELECT
	EVENTBGNTIME[1..NUMPROFILES] [1..NUMEVENTS]	PRIMDATA.[1..8].ORFBSTS
		PRIMDATA.[1..8].ORFBVAL
	EVENTENDTIME[1..NUMPROFILES] [1..NUMEVENTS]	PRIMDATA.[1..8].OROFFSET
		PRIMDATA.[1..8].PROPOVRD
	EVENTFL[1..NUMEVENTS]	PROFILEDESC[1..NUMPROFILES]
	EVENTSEGID[1..NUMPROFILES] [1..NUMEVENTS]	
	FBORSTS	PROFRESET
	HIALM.PR	PV
	HIALM.SV	PVEUHI
	HIALM.TYPE	PVEULO
	HOLDCMD	PVSTS
	HOLDOPT	RAMPRATE[1..NUMPROFILES] [1..NUMRAMPSOAK]
	HOLDRATE	
	HOLDVAL	REDTAG
	INALM	REMSOAKTIME
	INITMAN	RESETTIMR
	INITREQ[1..8]	RESTARTOPT
	INITVAL[1..8]	SAFEOP
	INSBLOCK[1..10]	SECDATAIN.ARWSTS
	INSFAILALM.FL	SECDATAIN.EUHI
	INSFAILALM.PR	SECDATAIN.EULO
	INSFAILALM.SV	SECDATAIN.HISELECT
	INSFAILFL	SECDATAIN.INITSTS
	LASTMODEREQ	SECDATAIN.INITVAL
	LASTOPREQ	SECDATAIN.LOCALMAN
	LASTOPTYPE	SECDATAIN.ORFBSTS
	LASTREQFL	SECDATAIN.ORFBVAL
	LASTSTEP	SECDATAIN.OROFFSET
	MAXHISOAKDEV[1..NUMPROFILES]	SECDATAIN.PROPOVRD
	MAXLOSOAKDEV[1..NUMPROFILES]	SECINITOPT[1..8]
	MAXRAMPDEV[1..NUMPROFILES]	SIALM.FL
	MAXSOAKVAL[1..NUMPROFILES]	SIALM.OPT
	MINSOAKVAL[1..NUMPROFILES]	SIALM.PR

	MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ MODETRACK	SIALM.SV SIFL SIOPT SOAKTIME[1..NUMPROFILES][1..NUMRAMPSOAK] SOAKVAL[1..NUMPROFILES][1..NUMRAMPSOAK] STARTOP[1..NUMPROFILES] STARTOPT STARTRATE STARTSEG[1..NUMPROFILES] STARTVAL STOPOPT STOPRATE STOPVAL TMOUTFL TMOUTMODE TMOUTTIME TOTALTIME[1..50] TOTELAPSEDTM UNCMDCHGALM.FL UNCMDCHGALM.OPT UNCMDCHGALM.PR UNCMDCHGALM.SV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RAMPSOAK Block.	

3.11.13 RATIOBIAS block

Description	Accepts a ratio value input (RT) and an input value (X1) to provide a calculated output based on the ratio of the input variables plus a fixed and/or a floating bias. The input value must come from another function block. In the Cascade mode, the ratio input value must come from another function block; but, in the Automatic (Auto) Mode, an operator or user program can set the ratio value.
Function	Lets you implement a form of ratio control by using this block between two PID blocks. In this case, the output from one PID block is used as the X1 input to the RATIOBIAS block and the output from the RATIOBIAS block is used as the SP input to the second PID block. With R410, RATIOBIAS block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.

Timeout Monitoring	In cascade mode, this block performs timeout monitoring on both inputs (X1 and RT). If either input value is not updated within a predefined time, this block invokes the following timeout processing.
	• If RT times out, block
	  – Sets the input timeout flag (TMOUTFL).
	  – Holds RT at its last good value.
	  – Sheds to the configured timeout mode (TMOUTMODE).
	  – Requests the RT primary to initialize.
	• If X1 times out, block
	  – Sets the X1 value to NaN. This causes CV to go to NaN, which results in the initialization of the RT and X1 primaries.
	If RT times out and the block sheds to Auto mode, block sets the Cascade Request flag (CASREQFL). When CASREQFL is set, it means the block is waiting to return to the cascade mode, and will do so as soon as it gets a good X1 value. This is true only, if the original mode was Cascade and the TMOUTMODE is Auto . If you change the mode, this clears the CASREQFL and disables the return to cascade operation.
	Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.

Control Initialization	Block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. However, you can disable the SECINITOPT so the block ignores initialization requests from the secondary.
	If the secondary is requesting initialization, block:
	• Initializes its output:
	  – CV = initialization value from the secondary,
	• Calculates an initialization value for the X1 and RT primaries.
	  – INITVAL[1] = CV - OPBIAS.FIX / RT
	  – INITVAL[2] = CV - OPBIAS.FIX / INITVAL[1]
	• Requests both primaries to initialize:
	  – INITREQ[1] = ON
	  – INITREQ[2] = ON

Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector (OVRDSEL) block, it receives override feedback data. The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy. The offset flag only applies to PID type function blocks. However, you can disable the SECINITOPT so the block ignores override requests from the secondary.
	When override status changes from selected to unselected, this block:
	• Computes a feedback value for X1 and RT primaries:
	  – feedback value for X1 = ORFBVAL - OPBIAS.FIX -OPBIAS.FLOAT / RT
	  – feedback value for RT = ORFBVAL - OPBIAS.FIX - OPBIAS.FLOAT / X1 override feedback value

Inputs	Required number of inputs is determined by this block's mode: • If Mode = Cascade, 2 inputs are required - X1 and RT. Both must come from other function blocks. • If Mode = Auto or Man, only X1 is required. X1 must come from another function block; an operator cannot set it. • Both X1 and RT are initializable inputs. So, this block may have one or two primaries, depending upon whether RT input is used or not. • If mode = Auto, an operator or user program can set the RT value.

Input Ranges	• XEUHI and XEULO define the full range of X1 inputs in engineering units. This block applies no range checking, since it assumes that X1 is within XEUHI and XEULO. – XEUHI represents the 100% of full scale value. – XEULO represents the 0% of full scale value. • RTHILM and RTLOLM define the ratio limits for RT inputs in engineering units. An operator is prevented from setting an RT value that is outside these limits. If the RT value from a function block or user program is outside these limits, this block clamps the value to the appropriate limit and sets RT primary windup status. – RTHILM represents high ratio limit value. – RTLOLM represents low ratio limit value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it uses the secondary's input range through BACKCALC to set its CV range. If it does not have a secondary, its CV range tracks its own input range (XEUHI and XEULO). • OPHILM and OPLOLM define normal high and low limits for OP as a percent of the CV range (user-specified values). – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store OP value exceeding them. – Operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range (user-specified values). – Operator is prevented from storing an OP value that exceeds these limits.

Parameters	ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB BSHILM BSLOLM CASREQFL COMPUTEARW CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DESC ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC FBORSTS HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM	OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OPTYPE ORDERINCM OUTIND OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS

	INITMAN	PRIMDATA.[1..8].ORFBVAL
	INITREQ[1..8]	PRIMDATA.[1..8].OROFFSET
	INITVAL[1..8]	PRIMDATA.[1..8].PROPOVRD
	INSBLOCK[1..10]	RBOPTION
	INSFAILALM.FL	REDTAG
	INSFAILALM.PR	RESTARTOPT
	INSFAILALM.SV	RT
	INSFAILFL	RTHIFL
	LASTMODEREQ	RTHILM
	LASTOPREQ	RTLOFL
	LASTOPTYPE	RTLOLM
	LASTREQFL	SAFEOP
	LASTSTEP	SECDATAIN.ARWSTS
	MODE	SECDATAIN.EUHI
	MODEAPPL[1..4]	SECDATAIN.EULO
	MODEATTR	SECDATAIN.HISELECT
	MODEATTRFL.NORM	SECDATAIN.INITSTS
	MODEATTRFL.OPER	SECDATAIN.INITVAL
	MODEATTRFL.PROG	SECDATAIN.LOCALMAN
	MODECHANGE	SECDATAIN.ORFBSTS
	MODEFL.AUTO	SECDATAIN.ORFBVAL
	MODEFL.BCAS	SECDATAIN.OROFFSET
	MODEFL.CAS	SECDATAIN.PROPOVRD
	MODEFL.MAN	SECINITOPT[1..8]
	MODEFL.NORM	SIALM.FL
	MODEPERM	SIALM.OPT
	MODEREQ	SIALM.PR
	MODETRACK	SIALM.SV
	NAME	SIFL
	NORMMODE	SIOPT
	NORMMODEATTR	STARTOPT
	NUMINSERT	STARTRATE
	NUMPRI	STARTVAL
	NUMSEC	STOPOPT
		STOPRATE
		STOPVAL
		TMOUTFL
		TMOUTMODE
		TMOUTTIME
		UNCMDCHGALM.FL
		UNCMDCHGALM.OPT

		UNCMDCHGALM.PR UNCMDCHGALM.SV X1 XEUHI XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RATIOBIAS Block.	

3.11.14 RATIOCTL block

Description	Accepts the actual value of the controlled flow (X1), the actual value of the uncontrolled flow (X2) and the target ratio between the flows (SP), and calculates the target value of the controlled flow (OP) and the actual ratio between the flows (PV) as outputs.
Function	Provides four user-selectable methods for calculating the ratio between the flows (PV). The target value for the controlled flow (OP) is calculated according to the selected method for calculating PV. RATIOCTL block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	If mode is CAScade, the block performs time-out monitoring of the initializable input, SP. - if good SP value is not received within a predefined time (TMOUTTIME), the block invokes timeout processing as noted below. If MODE is Cascade and SP times-out, the RATIOCTL block does the following: • Sets the “input timeout” flag (TMOUTFL) • Holds SP at its last good value • Changes the mode to a user-specified “timeout mode” (MODE = TMOUTMODE) • Requests the SP primary to initialize (via BACKCALCOUT) If SP times-out and the block sheds to Auto mode, it sets the Cascade Request flag (CASREQFL). When CASREQFL is set, it means the block is waiting to return to the Cascade mode, and will do so as soon as it fetches a good SP value. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Control Initialization	The RATIOCTL block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. (Note that SECINITOPT may be used to ignore initialization requests from the secondary.) If the secondary is requesting initialization, the RATIOCTL block: • initializes its output: – CV = initialization value from the secondary Builds an initialization request for its primary based on CTLEQN selected.

Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector block, it will receive override feedback data when any of the following occur. • the block's windup state changes • the block is requested to do a oneshot initialization • the block's override status changes The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy (as determined by the Selector block). The offset flag only applies to PID-type function blocks. When the override status changes from selected to unselected, this block does the following: • Does *not* initialize its CV • Computes a feedback value for the SP primary depending on the CTLEQN selected.
Inputs	• A RATIOCTL block requires these three inputs: – X1 - the actual value of the controlled flow. – X2 - the actual value of the uncontrolled flow – SP - the target ratio between the controlled and uncontrolled flows. • The SP is an initializable input. This means the block can have one primary depending upon whether the SP input is configured or not. There is one primary for each initializable input. • The X1and X2 inputs must come from other function blocks. You cannot store to them. • If Mode is Cascade, SP is pulled from another function block. If Mode is Automatic, it may be stored by the operator or a user program.
Input Ranges	• You must specify X1 and X2 engineering unit range, XEUHI and XEULO. – XEUHI and XEULO define the full range of the X inputs in engineering units. XEUHI represents the 100% of full scale value. XEULO represents the 0% of full scale value. • This block assumes X inputs are within XEUHI and XEULO - it applies no range check • You must specify SPHILM and SPLOLM to define the set point limits, expressed as a ratio. The operator is prevented from storing a set point value that is outside these limits. If the primary or a user program attempts to store a value outside the limits, this block will clamp it to the appropriate limit and set the input windup status.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it fetches the secondary's input range through BACKCALC and sets its CV range to that. If it has no secondary, CVEUHI and CVEULO must be specified by the user. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of the CV range (user-specified values). – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store OP value exceeding them. – Operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range (user-specified values). – Operator is prevented from storing an OP value that exceeds these limits.

Parameters	ADVDEVALM.DB ADVDEVALM.DBU ADVDEVALM.FL ADVDEVALM.PR ADVDEVALM.SV ADVDEVALM.TM ADVDEVALM.TMO ADVDEVALM.TP ADVDEVOPT ADVSP ADVSPP ALMDB ALMDBU ALMTM ARWNET[1..8] ARWNETIN[1..8] ARWOP ARWOPIN ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB CASREQFL COMPUTEARW CTLEQN CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DELCV DESC DEV	OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OPTOL OPTYPE ORDERINCM OUTIND PUSHSP PV

	DEVHIALM.DB	PVSTS
	DEVHIALM.DBU	REDTAG
	DEVHIALM.FL	RESTARTOPT
	DEVHIALM.PR	SAFEOP
	DEVHIALM.SV	SECDATAIN.ARWSTS
	DEVHIALM.TM	SECDATAIN.EUHI
	DEVHIALM.TMO	SECDATAIN.EULO
	DEVHIALM.TP	SECDATAIN.HISELECT
	DEVLOALM.DB	SECDATAIN.INITSTS
	DEVLOALM.DBU	SECDATAIN.INITVAL
	DEVLOALM.FL	SECDATAIN.LOCALMAN
	DEVLOALM.PR	SECDATAIN.ORFBSTS
	DEVLOALM.SV	SECDATAIN.ORFBVAL
	DEVLOALM.TM	SECDATAIN.OROFFSET
	DEVLOALM.TMO	SECDATAIN.PROPOVRD
	DEVLOALM.TP	SECINITOPT[1..8]
	ESWENB	SIALM.FL
	ESWFL.AUTO	SIALM.OPT
	ESWFL.BCAS	SIALM.PR
	ESWFL.CAS	SIALM.SV
	ESWFL.MAN	SIFL
	ESWPERM	SIOPT
	EUDESC	SP
	FBORSTS	SPEUHI
	GAINHILM	SPEULO
	GAINLOLM	SPFORMAT
	HIALM.PR	SPHIFL
	HIALM.SV	SPHILM
	HIALM.TYPE	SPLOFL
	HOLDOPT	SPLOLM
	HOLDRATE	SPP
	HOLDVAL	SPRATEREQ
	INALM	SPREQ
	INITMAN	SPTV
	INITREQ[1..8]	SPTVDEVFL
	INITVAL[1..8]	SPTVDEVMAX
	INSBLOCK[1..10]	SPTVNORMRATE
	INSFAILALM.FL	SPTVOPT
	INSFAILALM.PR	SPTVP
	INSFAILALM.SV	SPTVRATE
	INSFAILFL	SPTVREQ
		SPTVSTATE

	K1 K2 LASTMODEREQ LASTOPREQ LASTOPTYPE LASTRATEREQ LASTREQFL LASTSPREQ LASTSPTVREQ LASTSTEP MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ MODETRACK NAME NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC	SPTVTIME STARTOPT STARTRATE STARTVAL STOPOPT STOPRATE STOPVAL TMOUTFL TMOUTMODE TMOUTTIME UNCMDCHGALM.FL UNCMDCHGALM.OPT UNCMDCHGALM.PR UNCMDCHGALM.SV X1 X1BIAS X1KB X1STS X2 X2BIAS X2KB X2STS XEUHI XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the RATIOCTL Block.	

3.11.15 REEOUT (Remote EEOUT) block

Description	Supports regulatory cascades between regulatory control function blocks included in an ACE supervisory controller control strategy and regulatory control points included in an ACE supervisory controller strategy contained in another Experion cluster.
Function	The REEOUT block supports inter-cluster ACE to ACE regulatory cascades by connecting to the Inter Cluster Gateway block in the secondary cluster using an OPC Gateway in the primary cluster.

Inputs/Outputs	• SPPIN: SP value in percent, derived from a regulatory control point in the FB's cluster • SPPOUT: SP output to a regulatory control point in the secondary cluster, passed to the Inter Cluster Gateway in the secondary cluster using an OPC Gateway in the primary cluster. Regulatory points in the secondary cluster must be configured to allow their SP to be pushed from the Inter Cluster Gateway resident in that cluster.	
Parameters	BACKCALCOUT BCOUT.ARWSTS BCOUT.EUHI BCOUT.EULO BCOUT.HISELECT BCOUT.INITREQ BCOUT.INITSTS BCOUT.INITVAL BCOUT.LOCALMAN BCOUT.ONESHOT BCOUT.ORFBSTS BCOUT.ORFBVAL BCOUT.OROFFSET BCOUT.PROPOVRD BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 NAME ORDERINCM	SECDATAIN SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO SECDATAIN.HISELECT SECDATAIN.INITREQ SECDATAIN.INITSTS SECDATAIN.INITVAL SECDATAIN.LOCALMAN SECDATAIN.ONESHOT SECDATAIN.ORFBSTS SECDATAIN.ORFBVAL SECDATAIN.OROFFSET SECDATAIN.PROPOVRD SPOUT SPOUTSTS SPPIN USERSYMNAME
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory*for more information on the REEOUT Block.	

3.11.16 REGCALC (Regulatory Control Calculator) block

Description	• Lets you write up to eight expressions for creating custom algorithms for Calculated Variable (CV) calculations. • Provides an interface to windup, initialization and override feedback processing, so you can add user-defined control blocks to your control strategies.
Function	• Each expression can contain any valid combination of inputs, operators and functions; and may perform arithmetic or logic operations. • You can write expressions for calculating CV under normal, initialization and override feedback conditions. Or, you can write expressions which produce initialization and override feedback values for this block and its primaries. • You can assign the result of an expression or an input to any assignable output that produces the same outputs as every other regulatory control block. You can assign the same input to multiple outputs. • With R410, REGCALC block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.

Timeout Monitoring	In cascade mode, this block performs timeout monitoring on X[1]. If the X[1] input value is not updated within a predefined time, this block invokes the following timeout processing. • Sets the input timeout flag (TMOUTFL) • Sets the input value to Bad (NaN). • Requests the X1 primary to initialize. This block does not support mode shedding on timeout. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Control Initialization	Block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. However, you can disable the SECINITOPT so the block ignores initialization requests from the secondary. If the secondary is requesting initialization, block: • Initializes its output: – CV = CVINIT (assignable output) • Builds an initialization request for the designated primaries, using INITREQ and INITVAL (both assignable outputs).
Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector (OVRDSEL) block, it receives override feedback data. The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy. The offset flag only applies to PID type function blocks. However, you can disable the SECINITOPT so the block ignores override requests from the secondary. When override status changes from selected to unselected, this block: • Initializes its output: – CV = CVORFB (assignable output) • Computes a feedback value for X1 input: – feedback value for X1 = ORFBVAL (assignable output) – feedback status for X1 = ORFBSTS (assignable output) If ORFBVAL and ORFBSTS are not assigned and this block has a secondary, the ORFBVAL and ORFBSTS received from the secondary are used to compute ORFBVAL for the primary.
Inputs	The REGCALC block can function without any inputs. The following inputs are optional and they only accept real data types. • X[1] - An initializable input that must come from another block, an operator can not set it. • X[2] through X[6] general purpose inputs. • XWHIFL - An external windup high flag. • XWLOFL - An external windup low flag.
Input Ranges	XEUHI and XEULO define the full range of X[1] input in engineering units. This block applies no range checking, since it assumes that X1 is within XEUHI and XEULO. If this function is required, you must write an expression for it. • XEUHI represents the 100% of full scale value. • XEULO represents the 0% of full scale value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.

Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. If this block has a secondary, it uses the secondary's input range through BACKCALC to set its CV range. If it does not have a secondary, you must define the range through CVEUHI and CVEULO. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of the CV range (user-specified values). – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store OP value exceeding them. – Operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range (user-specified values). – Operator is prevented from storing an OP value that exceeds these limits.
Assignable Outputs	You can assign expression results and/or inputs to the following outputs. • CV - This block's CV under normal operating conditions. • CVINIT - This block's CV during initialization. • CVORFB - This block's CV during override (in unselected path). • INITREQ - Initialization request flag, to be provided to the primary. • INITVAL - Initialization value, to be provided to the primary. • ORFBVAL - Override feedback value, to be provided to the primary. • ORFBSTS - Override feedback status, to be provided to the primary.
Operators and Functions	Table 3 lists the expression operators and functions supported by this block for reference.
Parameter Identification	You must specify a parameter by its full tag name. For example, “CM25.PumpASelect.PVFL”, or “CM57.PID100.MODE”. In effect, tag names allow expressions to have an unlimited number of inputs and work with any data type. However, do not use more than six parameter references in an expression. The expression syntax has been expanded. Delimiters (') can be used in an expression containing an external reference component. The format for the delimiter usage is as follows: • TagName.‘text’

Expression Rules	• Must include full tag.parameter name for X inputs in the expression and enclose identification number in brackets instead of parenthesizes. For example, CM151.REGCALC BLOCK.X[1] * CM151.REGCALC BLOCK.X[2] is valid. • Expressions cannot contain an assignment operation (a colon and equal sign with the current syntax) For example, “CM1.PID1.MODE:=X[1]“ is invalid. Each expression produces a single value (arithmetic or logical which is automatically stored in a “C” parameter. For example, if you write four expressions, the result of the first expression is stored in C[1], the result of the second is stored in C[2], etc. You can use these results, by name, in succeeding expressions. In this example, you could use C[1] as an input to expressions 2, 3, and 4. • You can mix and nest all operators and functions (including conditional assignments) in any order as long as value types match or can be converted. • You can use blanks between operators and parameter names, but they are not required. • You can use all data types in expressions, including enumerations. They are all treated as numeric types. • You must configure calculator expressions contiguously (without breaks) in the arrays. • A short description can be provided for the expressions using the expression descriptor parameter (EXPRDESC[1..8]). The results of the expressions, which use the CONST[1…8] parameters, are affected if you change the values of these parameters on the Constants tab. • With R410, non-CEE controllers such as PMD and Safety Manager, and Experion server points such as TPS and SCADA, can be configured in the Expressions. • With R410, when you write the expressions using the TPS point's parameter references, ensure that the TPS reference parameter is configured using the parentheses “()”to specify array index. However, when you write the expressions using the other non-CEE points you can use the brackets “[].”

Parameters	ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB C[1..8] CASREQFL COMPUTEARW CONFIGCODE CONFIGDESC CONFIGSTS CONST[1…8] CONSTACCLOCK CONSTENABLE CSTS[1..8] CTLINIT CTLSTATE CV CVEUHI CVEULO CVINIT CVINITSRC CVORFB CVORFBSRC CVSRC CVTYPE DESC ESWENB ESWFL.AUTO ESWFL.BCAS	MODETRACK NAME NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL

	ESWFL.CAS ESWFL.MAN ESWPERM EUDESC EXECCODE[1..8] EXECDESC[1..8] EXECSTS[1..8] EXPR[1..8] EXPRDESC[1…8] FBORSTS GAINHILM GAINLOLM HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM INITMAN INITREQ[1..8] INITREQSRC INITVAL[1..8] INITVALSRC INSBLOCK[1..10] INSFAILALM.FL INSFAILALM.PR INSFAILALM.SV INSFAILFL K LASTMODEREQ LASTOPREQ LASTOPTYPE LASTREQFL LASTSTEP MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE	OPTYPE ORDERINCM ORFBSTSSRC ORFBVALSRC OUTIND OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS PRIMDATA.[1..8].ORFBVAL PRIMDATA.[1..8].OROFFSET PRIMDATA.[1..8].PROPOVRD REDTAG RESTARTOPT SAFEOP SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO SECDATAIN.HISELECT SECDATAIN.INITSTS SECDATAIN.INITVAL SECDATAIN.LOCALMAN SECDATAIN.ORFBSTS SECDATAIN.ORFBVAL SECDATAIN.OROFFSET SECDATAIN.PROPOVRD SECINITOPT[1..8] SIALM.FL SIALM.OPT SIALM.PR SIALM.SV SIFL SIOPT STARTOPT STARTRATE STARTVAL STOPOPT STOPRATE STOPVAL TMOUTFL TMOUTMODE TMOUTTIME

	MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ	UNCMDCHGALM.FL UNCMDCHGALM.OPT UNCMDCHGALM.PR UNCMDCHGALM.SV X[1..6] XB[1..6] XEUHI XEULO XK[1..6] XKB[1..6] XSTS[1..6] XWHIFL XWLOFL
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REGCALC Block.	

3.11.17 REGSUMMER (Regulatory Control Summer)

Description	Lets you calculate an output value which is the sum of up to four input values.
Function	The RegSummer algorithm calculates an output value which is the sum of up to four inputs. Each of the inputs may be individually scaled. In addition, the output may be scaled by an overall gain, and an overall bias may be added to the result. With R410, REGSUMMER block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	If MODE = Cascade, RegSummer performs timeout monitoring on the initializable input X(1). If X(1) is not updated within a predefined time, the block invokes timeout processing. For RegSummer in case of X1 timeout, X2 to X4 still fetch the values from the upstream blocks. The timeout time (in seconds) is specified by TMOUTTIME. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Timeout Processing	If MODE is Cascade and X(1) times out, this block does the following: • Sets the “input timeout” flag (TMOUTFL) • Sets the input value to Bad (NaN) • Requests the X(1) primary to initialize This block does not support mode shedding on timeout.

Inputs	The RegSummer block accepts up to four inputs -- X(1) through X(4). X(1) is an initializable input; all others are non-initializable. This X[1] input can be connected to non-initializable inputs also. In this case there is no primary for this block. The inputs must be pulled from other function blocks; the user cannot store to them. This block has one primary. (There is one primary per initializable input.) X[1] input connection is mandatory. If X[1] is not connected and the block is loaded an error will be raised during load time saying “At least input one needs to be connected“ NUMXINPT represents the number of input connections that have been made to this block
Input Ranges **and Limits**	The user must specify an X-input engineering unit range, XEUHI and XEULO. XEUHI and XEULO define the full range of the inputs. XEUHI is the value that represents 100% of full scale, and XEULO is the value that represents 0%. XEUHI and XEULO apply to all of the X-inputs. This block assumes all of the X-inputs are within XEUHI and XEULO; it applies no range-checks.
Outputs	The RegSummer block has the following initializable outputs: OP - Calculated output, in percent. OPEU - Calculated output, in engineering units. The user may create a connection to OP or OPEU, but not both. Therefore, this block may have only one secondary. If the user does not create a connection to OP or OPEU, then the block does not have a secondary. Alternately, if the user connects OP or OPEU to a non-initializable input, then this block does not have a secondary
Output Ranges	CVEUHI and CVEULO define the full range of CV, in engineering units. If this block has a secondary, it fetches the secondary's input range via BACKCALC and sets its CV range to that. If it has no secondary, CVEUHI and CVEULO track the X-input range (XEUHI and XEULO). Note: This block fetches the secondary's input range regardless of SECINITOPT (i.e., regardless of whether the secondary's initialization and override data will be used) OPHILM and OPLOLM define the normal high and low limits for OP, as a percent of the CV range. These are user-specified values.OP will be clamped to these limits if the algorithm's calculated result (CV) exceeds them, or another function block or user program attempts to store an OP value that exceeds them. However, the operator may store an OP value that is outside these limits. OPEXHILM and OPEXLOLM define the extended high and low limits for OP, as a percent of the CV range. These are user-specified values. The operator is prevented from storing an OP value that exceeds these limits. OPTOL allow the user to configure a tolerance limit for the manually entered OP. If the difference betweenthe new OP value and the current OP value is greater than OPTOL then confirmation is required from the user to store the new value.

Equation Options	CV is calculated as follows: For 2 to 4 inputs: CV = K * [XK(1) * X(1) + XK(2) * X(2) + XK(3) * X(3) + XK(4) * X(4)] + OPBIAS For one input: CV = K * X1 + B where: CV = Current full value of the output of this algorithm in engineering units K = Overall gain for CV XK(1..4) = Individual gain for each input OPBIAS = total output bias (i.e., OPBIAS.FIX + OPBIAS.FLOAT) X(1..4) = Current full values of each X-input in use.

Parameters	ALMDB	OPHIALM.DB
	ALMDBU	OPHIALM.DBU
	ALMTM	OPHIALM.FL
	ARWNET[1..8]	OPHIALM.PR
	ARWNETIN[1..8]	OPHIALM.SV
	ARWOP	OPHIALM.TM
	ARWOPIN	OPHIALM.TMO
	ASTEPID	OPHIALM.TP
	BADCTLALM.FL	OPHIFL
	BADCTLALM.PR	OPHILM
	BADCTLALM.SV	OPLOALM.DB
	BADCTLALM.TM	OPLOALM.DBU
	BADCTLALM.TMO	OPLOALM.FL
	BADCTLFL	OPLOALM.PR
	BADCTLOPT	OPLOALM.SV
	BADOCOPT	OPLOALM.TM
	BADOCOPTENB	OPLOALM.TMO
	BLCKCOMMENT1	OPLOALM.TP
	BLCKCOMMENT2	OPLOFL
	BLCKCOMMENT3	OPLOLM
	BLCKCOMMENT4	OPMINCHG
	COMPUTEARW	OPREQ
	CTLINIT	OPROCLM
	CTLSTATE	OPROCNEGFL
	CV	OPROCPOSFL
	CVEUHI	OPTOL
	CVEULO	OPTYPE
	CVTYPE	ORDERINCM
	DESC	OUTIND
	ESWENB	OUTTYPE
	ESWFL.AUTO	PRIMDATA.[1..8].ARWSTS
	ESWFL.BCAS	PRIMDATA.[1..8]. EUHI
	ESWFL.CAS	PRIMDATA.[1..8]. EULO
	ESWFL.MAN	PRIMDATA.[1..8].HISELECT
	ESWPERM	PRIMDATA.[1..8].INITSTS
	EUDESC	PRIMDATA.[1..8].INITVAL
	FBORSTS	PRIMDATA.[1..8].LOCALMAN
	GAINHILM	PRIMDATA.[1..8].ORFBSTS
	GAINLOLM	PRIMDATA.[1..8].ORFBVAL
	HIALM.PR	PRIMDATA.[1..8].OROFFSET
	HIALM.SV	PRIMDATA.[1..8].PROPOVRD
		REDTAG

	HIALM.TYPE	RESTARTOPT
	HOLDOPT	SAFEOP
	HOLDRATE	SECDATAIN.ARWSTS
	HOLDVAL	SECDATAIN.EUHI
	INALM	SECDATAIN.EULO
	INITMAN	SECDATAIN.HISELECT
	INITREQ[1..8]	SECDATAIN.INITSTS
	INITVAL[1..8]	SECDATAIN.INITVAL
	INSBLOCK[1..10]	SECDATAIN.LOCALMAN
	INSERTSTS[1..10]	SECDATAIN.ORFBSTS
	INSFAILFL	SECDATAIN.ORFBVAL
	INSTYPE[1..10]	SECDATAIN.OROFFSET
	K	SECDATAIN.PROPOVRD
	LASTMODEREQ	SECINITOPT[1..8]
	LASTOPREQ	SIALM.FL
	LASTOPTYPE	SIALM.OPT
	LASTREQFL	SIALM.PR
	LASTSTEP	SIALM.SV
	MODE	SIFL
	MODEAPPL[1..4]	SIOPT
	MODEATTR	STARTOPT
	MODEATTRFL.NORM	STARTRATE
	MODEATTRFL.OPER	STARTVAL
	MODEATTRFL.PROG	STOPOPT
	MODECHANGE	STOPRATE
	MODEFL.AUTO	STOPVAL
	MODEFL.BCAS	TMOUTFL
	MODEFL.CAS	TMOUTMODE
	MODEFL.MAN	TMOUTTIME
	MODEFL.NORM	UNCMDCHGALM.FL
	MODEPERM	UNCMDCHGALM.OPT
	MODEREQ	UNCMDCHGALM.PR
	MODETRACK	UNCMDCHGALM.SV
	NAME	X[1..4]
	NORMMODE	XDESC[1..4]
	NORMMODEATTR	XEUHI
	NUMINSERT	XEULO
	NUMPRI	XK[1..4]
	NUMSEC	XSTS[1..6]
	OP	
	OPBIAS	
	OPBIAS.FIX	

	OPBIAS.FLOAT OPBIAS.RATE OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REGSUMMER Block..	

3.11.18 REMCAS block

Description	Receives two inputs (X1 and X2), - X1 comes from a remote cascade source and X2 comes from a backup cascade - performs timeout monitoring on both inputs, and normally operates in Cascade mode.
Function	Provides automatic switching between a remote and backup cascade - typically used with PID block that normally gets its set point from a remote source, but sheds to a local source if there is a communications failure. With R410, REMCAS block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade or backup cascade mode, performs timeout monitoring on both inputs (X1 and X2). If either input value is not updated within a predefined time, this block invokes the following timeout processing. (Note that this block does **not** support the timeout shed mode parameter TMOUTMODE.) • If X1 times out, but X2 is good, block – Sets the input timeout flag (TMOUTFL). – Sets MODE to backup cascade. – Sets the currently selected input (SELXINP) to X2. – Requests the X1 primary to initialize. • If X2 times out, but X1 is good, block – Requests the X2 primary to initialize. Since mode is cascade and X1 is already the currently selected input. • If both inputs timeout, block – Sets CV to NaN, which forces a “Bad Control” condition. The user specifies what actions to take on Bad Control through the BADCTLOPT. – Sets the currently selected input (SELXINP) to None. – Requests both primaries to initialize. If X1 times out and the block sheds to Backup Cascade mode, block sets the Cascade Request flag (CASREQFL). When CASREQFL is set, it means the block is waiting to return to the cascade mode, and will do so as soon as it brings a good X1 value. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.

Control Initialization	Block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. However, SECINITOPT[1..8] may be used to ignore initialization requests from this secondary. If the secondary is requesting initialization, block: • Initializes its output: – CV = initialization value from the secondary • Builds an initialization request for X1 primary as: – INITREQ[1] = ON – INITVAL[[1] = CV - OPBIAS.FIX • Builds an initialization request for X2 primary as: – INITREQ[2] = ON – INITVAL[2] = CV - OPBIAS.FIX
Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector block, it receives override feedback data. The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy. The offset flag only applies to PID type function blocks. However, SECINITOPT[1..8] may be used to ignore override requests from the secondary. When override status changes from selected to unselected, this block: • Computes a feedback value for the selected primary. – The selected primary feedback value = BACKCALCOUT.ORFBVAL - OPBIAS.FIX - OPBIAS.FLOAT. – The non-selected primary is propagated with“non-connected” status. The Selected input of the REMCAS block gets the propagated ORFBSTS status of either ‘Selected or Not-Selected’ from the Override Selector secondary while the unselected primary of the REMCAS block always gets non-connected status for Override Feedback status by the REMCAS block, regardless of whether TRACKING is On or Off.
Inputs	• X1 = initializable input from a remote source. • X2 = initializable input from backup cascade. • You can configure a description of up to 15 characters for each input.
Input Ranges	• XEUHI and XEULO define the full range of inputs. – XEUHI represents the 100% of full scale value. – XEULO represents the 0% of full scale value.
Outputs	Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	CVEUHI and CVEULO define the full range of CV in engineering units. • OPHILM and OPLOLM define normal high and low limits for OP as a percent of the CV range (user-specified values). – OP clamps to these limits if algorithm's calculated result (CV) exceeds them or another block or user program attempts to store OP value exceeding them. – Operator may store OP value outside these limits. • OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range (user-specified values). – Operator is prevented from storing an OP value that exceeds these limits.

Parameters	ALMDB	OPBIAS
	ALMDBU	OPBIAS.FIX
	ALMTM	OPBIAS.FLOAT
	ARWNET[1..8]	OPBIAS.RATE
	ARWOP	OPEU
	ASTEPID	OPEXHIFL
	BADCTLALM.FL	OPEXHILM
	BADCTLALM.PR	OPEXLOFL
	BADCTLALM.SV	OPEXLOLM
	BADCTLALM.TM	OPHIALM.DB
	BADCTLALM.TMO	OPHIALM.DBU
	BADCTLFL	OPHIALM.FL
	BADCTLOPT	OPHIALM.PR
	BADOCOPT	OPHIALM.SV
	BADOCOPTENB	OPHIALM.TM
	CASREQFL	OPHIALM.TMO
	COMPUTEARW	OPHIALM.TP
	CTLINIT	OPHIFL
	CTLSTATE	OPHILM
	CV	OPLOALM.DB
	CVEUHI	OPLOALM.DBU
	CVEULO	OPLOALM.FL
	CVTYPE	OPLOALM.PR
	DESC	OPLOALM.SV
	ESWENB	OPLOALM.TM
	ESWFL.AUTO	OPLOALM.TMO
	ESWFL.BCAS	OPLOALM.TP
	ESWFL.CAS	OPLOFL
	ESWFL.MAN	OPLOLM
	ESWPERM	OPMINCHG
	EUDESC	OPREQ
	FBORSTS	OPROCLM
	HIALM.PR	OPROCNEGFL
	HIALM.SV	OPROCPOSFL
	HIALM.TYPE	OPTYPE
	HOLDOPT	ORDERINCM
	HOLDRATE	OUTIND
	HOLDVAL	OUTTYPE
	INALM	PRIM.[1..8].INITIALIZABLE
	INITMAN	PRIMDATA.[1..8].HISELECT
	INITREQ[1..8]	PRIMDATA.[1..8].ORFBSTS
	INITVAL[1..8]	PRIMDATA.[1..8].ORFBVAL

	INSBLOCK[1..10]	PRIMDATA.[1..8].OROFFSET
	INSFAILALM.FL	PRIMDATA.[1..8].PROPOVRD
	INSFAILALM.PR	REDTAG
	INSFAILALM.SV	RESTARTOPT
	INSFAILFL	SAFEOP
	LASTMODEREQ	SECDATAIN.ARWSTS
	LASTOPREQ	SECDATAIN.EUHI
	LASTOPTYPE	SECDATAIN.EULO
	LASTREQFL	SECDATAIN.HISELECT
	LASTSTEP	SECDATAIN.INITSTS
	MODE	SECDATAIN.INITVAL
	MODEAPPL[1..4]	SECDATAIN.LOCALMAN
	MODEATTR	SECDATAIN.ORFBSTS
	MODEATTRFL.NORM	SECDATAIN.ORFBVAL
	MODEATTRFL.OPER	SECDATAIN.OROFFSET
	MODEATTRFL.PROG	SECDATAIN.PROPOVRD
	MODECHANGE	SECINITOPT[1..8]
	MODEFL.AUTO	SELXDESC
	MODEFL.BCAS	SELXINP
	MODEFL.CAS	SIALM.FL
	MODEFL.MAN	SIALM.OPT
	MODEFL.NORM	SIALM.PR
	MODEPERM	SIALM.SV
	MODEREQ	SIFL
	MODETRACK	SIOPT
	NAME	STARTOPT
	NORMMODE	STARTRATE
	NORMMODEATTR	STARTVAL
	NUMINSERT	STOPOPT
	NUMPRI	STOPRATE
	NUMSEC	STOPVAL
	OP	TMOUTFL
		TMOUTTIME
		TRACKING
		UNCMDCHGALM.FL
		UNCMDCHGALM.OPT
		UNCMDCHGALM.PR
		UNCMDCHGALM.SV
		X1
		X2
		XDESC[1..2]
		XEUHI

		XEULO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the REMCAS Block.	

3.11.19 SWITCH block

Description	Accepts up to 8 initializable inputs (that is, primaries) and operates as a single-pole, 8-position rotary switch. • An Operator, user program or another block may change switch position.
Function	Typically used to assign different primary to a secondary; allows user to select one from as many as 8 inputs and outputs the selected value. With R410, SWITCH block allows you to configure individual values for the deadband, deadband unit, on-delay time, and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. However, you can configure only identical deadband unit for all the alarms.
Timeout Monitoring	In cascade mode, performs timeout monitoring on all inputs X[1] through X[8]. If an input value is not updated within a predefined time, this block invokes the following timeout processing. • Sets the “input timeout” flag (TMOUTFL). • Sets the input value to Bad (NaN). • Requests the input's primary to initialize. This block does not support mode shedding on timeout. Time-out monitoring is enabled by setting TMOUTTIME to a non-zero value, and disabled by setting it to zero.
Control Initialization	Block brings initialization requests from its secondary through BACKCALC. In addition, the secondary may propagate oneshot initialization requests to this block. However, SECINITOPT[1..8] may be used to ignore initialization requests from this secondary. If the secondary is requesting initialization, block: • Initializes its output: – CV = initialization value from the secondary • Builds an initialization request for selected primary as: – INITREQ(s) = ON – INITVAL(s) = CV - OPBIAS.FIX • If TRACKING is ON, block also builds an initialization request for the non-selected primaries as: – INITREQ(n) = ON – INITVAL(n) = CV - OPBIAS.FIX

Override Feedback Processing	If this block is in a cascade strategy with a downstream Override Selector block, it receives override feedback data. The data consists of an override status, override feedback value and an override offset flag. The status indicates if this block is in the selected or unselected strategy. The offset flag only applies to PID type function blocks. However, SECINITOPT[1..8] may be used to ignore override requests from the secondary. When override status changes from selected to unselected, this block: • Computes a feedback value for the selected primary. – The selected primary feedback value = BACKCALCOUT.ORFBVAL - OPBIAS.FIX - OPBIAS.FLOAT – The non-selected primaries are propagated with “not selected” status. The Selected input of the SWITCH block gets the propagated ORFBSTS status of either ‘Selected or Not-Selected’ from the Override Selector secondary while the unselected primary of the SWITCH always gets non-connected status for Override Feedback status by the Switch block, regardless of whether TRACKING is On or Off. If this block and a primary are on the same node, this block propagates the override data to the primary. If a primary is on a different node, this block stores the data in the BACKCALC packet for that primary, which the primary brings on its next execution.
Inputs	Accepts up to 8 initializable inputs -- X[1] through X[8]. • Inputs must be pulled from other blocks (cannot be stored). • You can configure a description of up to 15 characters for each input. • This block may have two to eight primaries, depending on the number of inputs that are configured. (There is one primary per initializable input.)
Input Ranges and Limits	User must specify an X-input engineering unit range XEUHIandXEULO which defines the full range of inputs (for all X-inputs). • XEUHI represents the 100% of full scale value. • XEULO represents the 0% of full scale value. Block provides its input range (XEUHI/XEULO) to the primaries through BACKCALC. The primaries use this for their output range (CVEUHI/CVEULO).
Outputs	Block has the following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units. – User may specify a fixed bias to be added to the output. – Block calculates floating bias to provide bumpless transition after input switching, initialization or mode change. Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.
Output Ranges	• CVEUHI and CVEULO define full range of CV in engineering units. If block has no secondary, CVEUHI and CVEULO track the “X” input range (XEUHI and XEULO). • OPHILM and OPLOLM define normal high and low limits for OP as a percent of CV range (user-specified values). • OPEXHILM and OPEXLOLM define extended high and low limits for OP as a percent of CV range (user-specified). – Prevents operator from storing an OP that exceeds these limits.

Parameters	ALMDB ALMDBU ALMTM ARWNET[1..8] ARWOP ASTEPID BADCTLALM.FL BADCTLALM.PR BADCTLALM.SV BADCTLALM.TM BADCTLALM.TMO BADCTLFL BADCTLOPT BADOCOPT BADOCOPTENB BADINPTOPT[1..8] COMPUTEARW CTLEQN CTLINIT CTLSTATE CV CVEUHI CVEULO CVTYPE DESC ESWENB ESWFL.AUTO ESWFL.BCAS ESWFL.CAS ESWFL.MAN ESWPERM EUDESC FBORSTS HIALM.PR HIALM.SV HIALM.TYPE HOLDOPT HOLDRATE HOLDVAL INALM INITMAN	OPEU OPEXHIFL OPEXHILM OPEXLOFL OPEXLOLM OPHIALM.DB OPHIALM.DBU OPHIALM.FL OPHIALM.PR OPHIALM.SV OPHIALM.TM OPHIALM.TMO OPHIALM.TP OPHIFL OPHILM OPLOALM.DB OPLOALM.DBU OPLOALM.FL OPLOALM.PR OPLOALM.SV OPLOALM.TM OPLOALM.TMO OPLOALM.TP OPLOFL OPLOLM OPMINCHG OPREQ OPROCLM OPROCNEGFL OPROCPOSFL OUTIND OPTYPE ORDERINCM OUTTYPE PRIM.[1..8].INITIALIZABLE PRIMDATA.[1..8].HISELECT PRIMDATA.[1..8].ORFBSTS PRIMDATA.[1..8].ORFBVAL PRIMDATA.[1..8].OROFFSET PRIMDATA.[1..8].PROPOVRD REDTAG RESTARTOPT

	INITREQ[1..8] INITVAL[1..8] INSBLOCK[1..10] INSFAILALM.FL INSFAILALM.PR INSFAILALM.SV INSFAILFL LASTMODEREQ LASTOPREQ LASTOPTYPE LASTREQFL LASTSTEP MODE MODEAPPL[1..4] MODEATTR MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODECHANGE MODEFL.AUTO MODEFL.BCAS MODEFL.CAS MODEFL.MAN MODEFL.NORM MODEPERM MODEREQ MODETRACK NAME NORMMODE NORMMODEATTR NUMINSERT NUMPRI NUMSEC OP OPBIAS OPBIAS.FIX OPBIAS.FLOAT OPBIAS.RATE	SAFEOP SECDATAIN.ARWSTS SECDATAIN.EUHI SECDATAIN.EULO SECDATAIN.HISELECT SECDATAIN.INITSTS SECDATAIN.INITVAL SECDATAIN.LOCALMAN SECDATAIN.ORFBSTS SECDATAIN.ORFBVAL SECDATAIN.OROFFSET SECDATAIN.PROPOVRD SECINITOPT[1..8] SELXDESC SELXFL[1..8] SELXINP SIALM.FL SIALM.OPT SIALM.PR SIALM.SV SIFL SIOPT STARTOPT STARTRATE STARTVAL STOPOPT STOPRATE STOPVAL TMOUTFL TMOUTTIME TRACKING UNCMDCHGALM.FL UNCMDCHGALM.OPT UNCMDCHGALM.PR UNCMDCHGALM.SV X[1..8] XDESC[1..8] XEUHI XEULO

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the SWITCH Block.

3.12 Sequential Control Module Blocks

Related topics

"HANDLER Block" on page 203
"STEP Block" on page 204
"SYNC Block" on page 205
"TRANSITION Block" on page 206

3.12.1 HANDLER Block

Description	SCM HANDLER blocks are execution modules that group STEP and TRANSITION blocks. • Multiple Handler blocks may be contained within an SCM block, each modeled as a set of STEP and TRANSITION blocks, based on the following categories: – Edit Handler – Main Handler – Check Handler – Interrupt Handler – Restart Handler – Hold Handler – Stop Handler – Abort Handler • Choices of which HANDLER block of each category to invoke are manifested through a HANDLER block selection list on the SCM block. • A HANDLER block is invoked when – its invoke conditions, modeled in its Invoke TRANSITION block, are met – when the SCM block is commanded to invoke the Handler (for example, the STOP command causes the STOP Handler to execute).	
Function	Used to describe, group, and categorize sequential control behavior.	
Parameters	lt;CONFIGCODE CONFIGDESC CONFIGSTS DESC EXECCODE EXECDESC EXECSTS HANDLER	INVOKT.HANDLE NAME NUM ORDERINCM PROCESSED STATE TYPE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Sequential Control Module User Guide* for more information on the HANDLER Block.	

3.12.2 STEP Block

Description	An SCM block which defines specific output actions. • A specified output action usually generates a request to a control device to do something (for example, open a valve, start a pump, set furnace temperature). • The source value of each output can be an expression (thereby enabling calculations in each output).
Function	Organizes the output expressions of an SCM HANDLER block at a specific stage of the HANDLER's execution thread.
Outputs	• Up to 16 outputs may be defined per SCM STEP block. • The STEP block whose outputs are active is called the Active Step.
Expressions	You enter desired output expressions into the Step output configuration form. You compose an output expression to include a target store destination for a source value and a source expression that generates the value to be stored. (For example, cm1.pid1.sp := cm2.pid2.op + 50.0.) Source expressions can evaluate to a Boolean value using a combination of arithmetic and logical operators, to an arithmetic value using arithmetic operators, or may simply specify any scalar value (Floating Point, Boolean, Enumeration) for comparison in a logical expression or as a value to be stored to the target store destination. Parameters of other blocks can be referenced as long as the block is already defined in the system database. Note that : • String data types are supported. • Enumerations and Boolean are supported, but values must be entered as integers. For example: – cm1.flag1.pvfl := 1 (PVFL is turned ON) • With R410, non-CEE controllers such as PMD and Safety Manager, and Experion server points such as TPS and SCADA, can be configured in the Expressions. • With R410, when you write the expressions using the TPS point's parameter references, ensure that the TPS reference parameter is configured using the parentheses “()”to specify array index. However, when you write the expressions using the other non-CEE points you can use the brackets “[].”
Operators and Functions	Table 3 lists the expression operators and functions supported by this block for reference.

Parameters	ACTIVEFL ACTVTNTIME CONFIGCODE CONFIGDESC CONFIGSTS DESC EUDESC EXECCODE EXECDESC EXECSTS HANDLER IC.BYPPERM IC.BYPREQ IC.CONFIGCODE IC.CONFIGDESC IC.CONFIGSTS IC.DESC IC.EXECCODE IC.EXECDESC IC.EXECSTS IC.EXPR IC.FL IC.INVOKFL IC.OPT ID MAXTIME MAXTIMEFL MINTIME NAME NEXTCOMP[1..10] NEXTHANDLE[1..10] NEXTNUMBER NUM	NUMOUTPUTS OP[1..16].CONFIGCODE OP[1..16].CONFIGDESC OP[1..16].CONFIGSTS OP[1..16].DELAYTIME OP[1..16].DELAYTIMEREM OP[1..16].DESC OP[1..16].EXECCODE OP[1..16].EXECDESC OP[1..16].EXECSTS OP[1..16].SRCEXPR OP[1..16].STATE OP[1..16].TYPE ORDERINCM PROCESSED SC.BYPPERM SC.BYPREQ SC.CONFIGCODE SC.CONFIGDESC SC.CONFIGSTS SC.DESC SC.EXECCODE SC.EXECDESC SC.EXECSTS SC.EXPR SC.FL SC.INVOKFL SC.OPT STATE TIME UPDRESOPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Sequential Control Module User Guide* for more information on the STEP Block.	

3.12.3 SYNC Block

Description	The SYNC block lets you configure SCMs to have steps and transitions executing in parallel. The sync block will synchronize the start and finish of a parallel section.

Function	Waits until all preceding steps and transitions have completed before moving on the next threads. You can use SYNC blocks to be sure, that at a certain point in time, all previous threads are synchronized before proceeding to the next set of steps and transitions. You can also use them to start the next set of parallel steps and transitions.	
Inputs/Outputs	Sync blocks can have any combination of up to fifteen inputs and fifteen outputs (either step or transition blocks).	
Parameters	CONFIGCODE CONFIGDESC CONFIGSTS DESC EPREV EUDESC EXECCODE EXECDESC EXECSTS HANDLER	NEXTCOMP[1..15] NEXTHANDLE[1..10] NEXTNUMBER NUMORDERINCM PROCESSED STATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Sequential Control Module User Guide* for more information on the STEP Block.	

3.12.4 TRANSITION Block

Description	An SCM block that defines specific input conditions for a Handler. • Input conditions and conjoining TRANSITION blocks define a distinct process state that must be achieved in order for the output actions specified by the next STEP block to be performed. • The input conditions grouped into a TRANSITION block are the condition expressions that direct sequential execution flow. Nesting of Transitions may be required when a single Transition cannot accommodate all inputs in required in a logical expression. Transition is considered Free Standing when the input pin, DESC, and the Output pin, NEXTCOMP, are not connected to any other EBM component Block. For more information on Nesting Transition, see Sequential Control User's Guide.
Function	Defines the distinct process state that must be achieved in order to allow the SCM HANDLER to advance to the control step (that is, the STEP block) so that it can perform the output actions specified.
Input Conditions	• A maximum of 10 standard input conditions are supported per SCM TRANSITION block. • The Invoke TRANSITION block in the MAIN HANDLER of the SCM block provides the Start Conditions for the SCM. • Logic gates may be AND, OR, NAND, NOR, NOT, XOR, CONNECT, NONE, OFF, or ON. – XOR must have two inputs. – CONNECT and NOT have only one input -- the output is the same as the input and the output is the logical negation of the input, respectively. – NONE, ON, and OFF have no inputs.

<table>
<tr><td>Expressions</td><td colspan="2">You enter desired condition expressions into the transition condition configuration form. Condition expressions can evaluate to a Boolean value using a combination of arithmetic and logical operators (for example, cm1.pid1.sp + cm2.pid2.op >= 50.0). Parameters of other blocks can be referenced as long as the block is already defined in the system database. Note that:
• String data types are not supported.
• Enumerations and Boolean are supported, but values must be entered as integers. For example:
– cm2.pid1.mode = 2 (Mode is compared to Cascade)</td></tr>
<tr><td>Operators and Functions</td><td colspan="2">Table 3 lists the expression operators and functions supported by this block for reference.</td></tr>
<tr><td>Parameters</td><td>C[1..10].BYPPERM
C[1..10].BYPREQ
C[1..10].CONFIGCODE
C[1..10].CONFIGDESC
C[1..10].CONFIGSTS
C[1..10].DESC
C[1..10].EXECCODE
C[1..10].EXECDESC
C[1..10].EXECSTS
C[1..10].EXPR
C[1..10].FL
C[1..10].GATEASGN
C[1..10].INVOKFL
C1
CONFIGCODE
CONFIGDESC</td><td>CONFIGSTS
DESC
EUDESC
EXECCODE
EXECDESC
EXECSTS
G[1..4].ALGID
G[1..4].FIRSTCOND
G[1..4].NUMINPTS
G[1..4].SO
HANDLER
NEXTCOMP
NEXTHANDLE
NAME
NUM
NUMCONDS
ORDERINCM
PROCESSED
SO
STATE</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.
Refer to the Sequential Control Module User Guide for more information on the TRANSITION Block.</td></tr>
</table>

3.13 System Blocks

Related topics

“CONTROL MODULE block (Continuous Control)” on page 208
“SEQUENTIAL CONTROL MODULE block (Sequential Control)” on page 209
“Recipe Control Module (RCM) Block” on page 211
“Unit Control Module (UCM) Block” on page 211
“Master Recipe (MR) Block” on page 211
“Proxy Master Recipe (MR) Block” on page 211
“Proxy FTEB Block” on page 212
“Proxy Node” on page 212
“Proxy Sequential Control Module (SCM) Block” on page 212
“Proxy Recipe Control Module (RCM) Block” on page 212

3.13.1 CONTROL MODULE block (Continuous Control)

Description	One of two system container blocks supported by CEE. It holds continuous and discrete function blocks.
Function	Configurable building block for defining control strategies. Lets you encapsulate strategies according to function. It provides these basic services for configured blocks: • Serves as the unit of load for continuous and discrete control strategies. • Transfers data between passive parameters that have no associated active connector. • Executes component function blocks in an established order, which is configurable or arbitrarily determined by the CM. • Provides independent tag names component blocks their parameters. • Serves the execution master for continuous and discrete control strategies.
vInputs	Input parameters for component blocks that connect to other CMs and SCMs.
Outputs	Output parameters for component blocks that connect to other CMs and SCMs.

Parameters	ALIASOPT	ORDERINLINK
	ALMENBSTATE	PHASE
	BPS	PRIMARYSIG
	BPSDELAY	QUALSTATE
	BPSDELAYREM	REASONSET
	CBBLOCKPROP	PREVLOADSTAT
	CEESTATE	SCALEPERIOD
	CONTCUTOUT	SCANASSOCDSP
	CTRLCONFIRM	SCANCTRLLVL
	DESC	SCANGRPDTL
	ESTWEIGHT	SCANPNTDTL
	EUDESC	SCMASTEP
	EXECSTATE	SCMID
	FFPERIOD	SCMNAME
	PERIOD	SCMOPT
	PERIODSEC	SCMSTATE
	INALM	SCRIPTOR
	INSERTINDEX	SECONDARYSIG
	IOSCHEDOPT	SECSIGSECLVL
	JOURNALONLY	STALECOUNT
	KEYWORD	UNITTEXT
	LOADSTATE	VERSION
	LOGICINITOPT	
	NAME	
	NUMSIGS	
	ORDERINCEE	
	Starting in Experion R311.1, the LOGICINITOPT parameter is added to the Control Module's Main configuration form to govern how outputs of the Logic blocks FTRIG, MAXPULSE, MINPULSE, MVOTE, nooN, OFFDELAY, ONDELAY, PULSE, RTRIG, TRIG, and 2oo3 react when they go through state transitions of activate, cold start, warm start, or RAM Retention Restart (RRR). Refer to the given Logic block description in the *Control Builder Components Theory* for more information related to a given Logic block.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the Control Module Block.	

3.13.2 SEQUENTIAL CONTROL MODULE block (Sequential Control)

Description	A system container block that consists of sequences of STEP and TRANSITION blocks grouped by specific HANDLER blocks. • The SCM block may only contain its own components (that is, HANDLER, STEP and TRANSITION blocks); it cannot contain other basic blocks such as PID or logic blocks.
Function	Used to organize normal- and exception-based sequential control logic.

Parameters	ABORTALM.FL	INVFRMHNDTYP[1..8]
	ABORTALM.PR	INVFROMSTEP[1..8][1..15]
	ABORTALM.SV	INVFROMSTEPN[1..8][1..15]
	ABORTLOCK	INVREASON[1..8]
	ACTIVEHANDLR[1..8]	INVTHREAD[1..8]
	ACTIVELOC.HANDLER	INVTIME[1..8]
	ACTIVELOC.HANDLERN	KEYWORD
	ACTIVELOC.HNDTYPE	LOADSTATE
	ACTIVELOC.STEP[1..15]	MODE
	ACTIVELOC.STEPN[1..15]	MODEATTR
	ACTIVELOC.TIME[1..15]	NAME
	ALIASBLKTYP[ ]	NORMMODE
	ALIASOPT	NORMMODEATTR
	ALIASPRMTYP[ ]	NUMALIASES
	ALMENBSTATE	NUMHISTPARMS
	AUXCMD	NUMINSTANCES
	AUXOPT	NUMRECPARMS
	AUXREQ	NUMTHREADS
	AUXREQDATA[1..5]	NUMTRANS
	AUXSTS	ORDERINCEE
	AUXUNIT	ORDERINCM
	CEESTATE	PAUSEFL
	CMDEXEC	PERIOD
	COMMAND	PHASE
	CONFIGCODE	PREVLOADSTAT
	CONFIGDESC	RECDESC[1..50]
	CONFIGSTS	RECMATCODE[1..50]
	CONTROLLOCK	RECSCALE[1..50]
	DESC	RECTARGET[1..50]
	ENBHANDLER[1..8]	RECTARGETMAX[1..50]
	ESTWEIGHT	RECTARGETMIN[1..50]
	EUDESC	RESADDR[1..10]
	EVALTRANS[1..10][1..10]	RESADDRFUTRN[1..10]
	EVALTRANSN[1..10][1..10]	RESADDRFUTUR[1..10]
	EXCMODEOPT	RESADDRN[1..10]
	EXECCODE	RSTPROCESSED
	EXECDESC	SCANASSOCDSP
	EXECSTATE	SCANCTRLLVL
	EXECSTS	SCANGRPDTL
	FAILALM.FL	SCANPNTDTL
	FAILALM.PR	SELHANDLER[1..8]
	FAILALM.SV	SELHANDLERN[1..8]

	HIALM	SSTEPLOCK
	HIALM.PR	STATE
	HIALM.SV	STEPALM.FL
	HIALM.TYPE	STEPALM.PR
	HISTDESC	STEPALM.SV
	HISTTYPE[1..50]	STOPALM.FL
	HISTVALUE[1..50]	STOPALM.PR
	HOLDALM.FL	STOPALM.SV
	HOLDALM.PR	TARGETSTEP[1..10]
	HOLDALM.SV	TIME[1..8]
	INALM	UNITTEXT
	INSERTINDEX	VERSION
	INSTSELECT	
	INVCOND[1..8]	
	INVFRMHNDLER[1..8]	
	INVFRMHNDLRN[1..8]	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Sequental Control Module User Guide* for more information on the Sequential Control Module Block.	

3.13.3 Recipe Control Module (RCM) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.4 Unit Control Module (UCM) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.5 Master Recipe (MR) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.6 Proxy Master Recipe (MR) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*

- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.7 Proxy FTEB Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.8 Proxy Node

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.9 Proxy Sequential Control Module (SCM) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.13.10 Proxy Recipe Control Module (RCM) Block

Refer to the following documents for EBM-related documentation.

- *Batch Overview and Planning Guide*
- *Batch Implementation Guide*
- *Sequential Control User's Guide*

3.14 Thermodynamic Utility Function Block

Related topics

"STEAMPROP (Steam Property) block" on page 213

3.14.1 STEAMPROP (Steam Property) block

<table>
<tr><td>Description</td><td>In steam generators and utilities, thermodynamic efficiency is calculated to optimize/maximize the efficiency of the plant operation. To calculate the thermodynamic efficiency, you need the thermodynamic properties of steam/water such as entropy, enthalpy, specific volume, and density for a given pressure, temperature and the state of matter. The STEAMPROP function block accepts temperature/pressure as inputs from measurement and provides entropy/enthalpy, and so on, as outputs. These values can be used for efficiency calculation and optimal operation. reduction in efficiency can additionally indicate the health degradation of the thermodynamic system or plant equipment. This block supports the following units of measuring system.
• SI in kJ/kg degree K
• Metric in kcal/kg degree C
• English in Btu/lb degree F
You can use any one of them while configuring the inputs as applicable. The detailed display of this block includes graphical representation of steam tables. For more information about the graphical representation of the STEAMPROP block, refer to the STEAMPROP block detail displays topic in the Control Building User's Guide.</td></tr>
<tr><td>Function</td><td>Steam Property function block computes the thermodynamic quantities of water and steam. The thermodynamic quantities of water and steam can be one of the following:
• Enthalpy (H)
• Entropy (S)
• Pressure (P)
• Temperature (T)
• Specific volume (V)
• Steam quality/dryness fraction (X)
This function block uses the equations quoted in Industrial Formulation 1997 (IF-97) released by the International Association for the Properties of Water and Steam (IAPWS). For more information about Industrial Formulation 1997 (IF-97), refer to the http://www.iapws.org/.
It uses different equations to compute the required output. For example, in case of water, if the pressure and the temperature are the inputs to the block then the following outputs can be computed.
• Enthalpy (H)
• Entropy (S)
• Specific volume (V)</td></tr>
</table>

Inputs	Pressure (P) Temperature (T) Entropy (S) Steam quality/dryness fraction (X) For detailed information about the supported input types, refer to the *Control Builder Components Theory*.	
Outputs	Specific volume (V) Enthalpy (H) Entropy (S) Temperature (T) For detailed information about the supported output types, refer to the *Control Builder Components Theory*.	
Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 DESC EUDESC ENGUNITSYSTEM ENTHALPY ENTHALPYENGUNIT ENTHALPYIO ENTROPY ENTROPYENGUNIT ENTROPYIO	PRESSURE PRESSUREENGUNIT PRESSUREIO SPECIFICVOLUME SPECIFICVOLUMEENGUNIT SPECIFICVOLUMEIO STEAM_WATER STEAMTYPE STEAMQUALITY STEAMQUALITYIO TEMPERATURE TEMPERATUREIO TEMPERATUREENGUNIT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the STEAMPROP Block.	

3.15 Universal Control Network Interface (UCNIF) Block

Related topics

"UCNOUT block" on page 215
"EUCNOUT block" on page 216

3.15.1 UCNOUT block

Description	Supports Setpoint Control (SPC), Direct Digital Control (DDC), Remote Setpoint Control (RSP) and Direct Digital Control with Remote Setpoint (DDCRSP) remote cascade types between the regulatory control function blocks included in an ACE supervisory controller control strategy and the regulatory control points included in a Process Manager controller. **Note** The UCNOUT block requires an OPC gateway or a TPN Server to communicate with the Process Manager controller.	
Function	• Provides configurable connections and compatible data mapping between controllers. • Translates secondary data (SECDATA) from Process Manager regulatory control points to ACE controller compatible back calculation (BACKCALC) data. • Participates in Remote Cascade Request protocol for Process Manager regulatory control point MODE changes. • Forwards inputs from primary regulatory control blocks in ACE supervisory controller to Process Manager regulatory control point.	
Inputs/Outputs	The remote cascade type (REMCASTYPE) selection determines which UCNOUT block inputs/ outputs to use through the Configure Block form in Control Builder.	
Parameters	BACKCALCOUT BCOUT.ARWSTS BCOUT.EUHI BCOUT.EULO BCOUT.INITREQ BCOUT.INITVAL BCOUT.ONESHOT BCOUT.ORFBSTS BCOUT.ORFBVAL BCOUT.OROFFSET CASSTS MODEOUT NAME OPIN OPOUT ORDERINCM REMCASTYPE RSPPIN	SECDATAIN SECDATAIN.ARWSTS SECDATAIN.CASREQ SECDATAIN.CASSHED SECDATAIN.EULO SECDATAIN.EUSPAN100 SECDATAIN.INITREQ SECDATAIN.INITVAL SECDATAIN.SECTYPE SPOUT SPPIN

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the UCNOUT Block.

3.15.2 EUCNOUT block

Description	Supports Setpoint Control (SPC), Direct Digital Control (DDC), Remote Setpoint Control (RSP) and Direct Digital Control with Remote Setpoint (DDCRSP) remote cascade types between the regulatory control function blocks included in an ACE/C300 supervisory controller control strategy and the regulatory control points included in the Enhanced High-Performance Process Manager (EHPM) Controller. **Note** The EUCNOUT block does not require an OPC gateway or a TPN server to communicate with the EHPM.	
Function	• Provides configurable connections and compatible data mapping between controllers. • Translates secondary data (SECDATA) from the EHPM regulatory control points to ACE/C300 controller compatible back calculation (BACKCALC) data. • Participates in Remote Cascade Request protocol for the EHPM regulatory control point MODE changes. • Forwards inputs from primary regulatory control blocks in ACE/C300 supervisory controller to the EHPM regulatory control point.	
Inputs/Outputs	The remote cascade type (REMCASTYPE) selection determines which EUCNOUT block inputs/outputs to be used through the Configure Block form in Control Builder.	
Parameters	BACKCALCOUT	ORDERINCM
	BCOUT.ARWSTS	REMCASTYPE
	BCOUT.EUHI	RSPPIN
	BCOUT.EULO	SECDATAIN
	BCOUT.INITREQ	SECDATAIN.ARWSTS
	BCOUT.INITVAL	SECDATAIN.CASREQ
	BCOUT.ONESHOT	SECDATAIN.CASSHED
	BCOUT.ORFBSTS	SECDATAIN.EULO
	BCOUT.ORFBVAL	SECDATAIN.EUSPAN100
	BCOUT.OROFFSET	SECDATAIN.INITREQ
	CASSTS	SECDATAIN.INITVAL
	MODEOUT	SECDATAIN.SECTYPE
	NAME	SPOUT
	OPIN	SPPIN
	OPOUT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the EUCNOUT block.	

3.16 Hiway Interface (HIWAYIF) Blocks

Related topics

“HIWAYOUT block” on page 217

3.16.1 HIWAYOUT block

Description	Supports Setpoint Control (SPC) and Direct Digital Control (DDC), remote cascade types between the regulatory control function blocks included in an ACE supervisory controller control strategy and the Data Hiway regulatory control points. It participates in Remote Cascade Request protocol for Data Hiway point mode changes.	
Function	• Provides configurable connections and compatible data mapping between controllers. • Translates secondary data (SECDATA) from Data Hiway regulatory control points to ACE controller compatible back calculation (BACKCALC) data. • Participates in Remote Cascade Request protocol for Data Hiway regulatory control point MODE changes. • Forwards inputs it receives from the primary of a regulatory control function block in the ACE controller to a Data Hiway regulatory control point.	
Inputs/Outputs	The remote cascade type (REMCASTYPE) selection determines which HIWAYOUT block inputs/outputs to use through the Block Properties form in Control Builder.	
Parameters	BACKCALCOUT	SECDATAIN
	BCOUT.ARWSTS	SECDATAIN.ARWSTS
	BCOUT.EUHI	SECDATAIN.CASREQ
	BCOUT.EULO	SECDATAIN.CASSHED
	BCOUT.INITREQ	SECDATAIN.EULO
	BCOUT.INITVAL	SECDATAIN.EUSPAN100
	BCOUT.ONESHOT	SECDATAIN.INITREQ
	BCOUT.ORFBSTS	SECDATAIN.INITVAL
	BCOUT.ORFBVAL	SECDATAIN.SECTYPE
	BCOUT.OROFFSET	SIMMODE
	CASSTS	SPOUT
	MODEOUT	SPPIN
	NAME	
	OPIN	
	OPOUT	
	ORDERINCM	
	REMCASTYPE	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the Control *Builder Component Theory* for more information on the HIWAYOUT Block.	

3.17 Hiway Responder Block (HRB)

Hiway Responder Blocks are special purpose communication blocks that support transfer of Hiway messages between the C300 Controller and the LCN EHB.

HRB blocks are automatically instantiated in Control Modules when Hiway slot emulations are created by the HSE Creator Tool. They are then imported into Experion engineering repository database using Control Builder along with their parent Control Module.

The HRB blocks are classified into two major categories.

- Algorithm Hiway Responder Block: Acts as an interface between the CEE native blocks and the EHB. HRB receives data from the EHB and transfers it to the CEE native blocks for processing. They also read the processed data from the CEE and update the HG point.
  - Regulatory Algorithm Hiway Responder Block (HRBALG_REG)
  - Analog Input Algorithm Hiway Responder Block (HRBALG_AI)
  - Digital Input Hiway Responder Block (HRBALG_REG)
- Basic Controller Box Hiway Responder Block (HRBBOX_CB) and Extended Controller Box Hiway Responder Block (HRBBOX_EC): Box HRB represents the HG box in Experion.

Attention

- HRBs are available in the Control Builder library, in the folder HIWAYRB.
- Most HRB parameters are automatically configured. They are initially defined by the HSE Creator Tool and should not be modified thereafter unless it is necessary to change the configuration of the emulation. If that is done, the change should be injected into the HG point and points must be re-translated using the HSE Creator Tool. Leaving HRB parameters in a consistent state is the responsibility of the user as this guideline is not enforced by the system.
- Users should not modify parameters of native CEE blocks within the emulation from the Experion view.

Related topics

"Basic Controller Hiway Box Hiway Responder Block (HRBBOX_CB)" on page 218
"Extended Controller Hiway Box Hiway Responder Block (HRBBOX_EC)" on page 219
"Analog Input Algorithm Hiway Responder Block (HRBALG_AI)" on page 219
"Digital Input Algorithm Hiway Responder Block (HRBALG_DI)" on page 220
"Regulatory Algorithm Hiway Responder Block (HRBALG_REG)" on page 220

3.17.1 Basic Controller Hiway Box Hiway Responder Block (HRBBOX_CB)

The Basic Controller (CB) Hiway Box HRB is intended to execute by itself within a Control Module. The CM so formed acts as the Hiway Box Slot Emulation (HBSE) of the CB. This HRB performs the following two functions.

- Make available for LCN access those Hiway parameters which characterize the CB box as a whole such as the CB Status Word, CB the Alarm Word and the CB Sample Time Counter.
- Serve as a conduit for data communication between the EHB and the algorithm HRBs assigned to the box emulation.

The following parameters are available in the HRBBOX_CB block.

- BOXNUM
- BOXTYPE
- CARDTYPE[1]
- EHBADDED
- EHBADDEDIP

- EHBTHIS
- EHBTHISIP
- EHBRMIP1
- EHBRMIP2
- HWYNUMTHIS
- HYNUMADDED

For more information about the parameters, refer to the *Control Builder Parameter Reference*.

3.17.2 Extended Controller Hiway Box Hiway Responder Block (HRBBOX_EC)

The Extended Controller (EC) Hiway Box HRB is intended to execute by itself within a Control Module. The CM so formed acts as the Hiway Box Slot Emulation (HBSE) of the EC. This HRB performs the following two functions.

- Make available for LCN access those Hiway parameters which characterize the EC box.
- Serve as a conduit for data communication between the EHB and the algorithm HRBs assigned to the box emulation.

The following parameters are available in the HRBBOX_EC block.

- BOXNUM
- BOXTYPE
- EHBADDED
- EHBADDEDIP
- EHBTHIS
- EHBTHISIP
- EHBRMIP1
- EHBRMIP2
- HWYNUMTHIS
- HYNUMADDED
- TOGDESC
- TOGINTERVALS
- TOGINTSEL

For more information about the parameters, refer to the *Control Builder Parameter Reference*.

3.17.3 Analog Input Algorithm Hiway Responder Block (HRBALG_AI)

The Analog Input (AI) HRB is an algorithm HRB block which presents data of the CB AI emulation. It transfers data between the native DATAACQ block and the Hiway Slot Memory (HSM).

The following parameters are available in the HRBALG_AI block.

- ALGOREGSTATE
- ALGOPROCSTATE
- BOXNUM
- BOXTYPE
- BOXCON
- DATAACQBLOCKID1
- HSWDESC
- HSWVALUE

- PATHBRKSTS
- SLOTNUM
- SLOTTYPE

For more information about the parameters, refer to the *Control Builder Parameter Reference*.

3.17.4 Digital Input Algorithm Hiway Responder Block (HRBALG_DI)

The Digital Input (DI) HRB is an algorithm HRB block, which presents data of the EC DI emulation. The EC DI HRB algorithm block acts as an interface between the native Experion blocks and EHB. The native function blocks include, DI, DEVCTL, Logic OR, TRIG block, and the FLAG block.

The following parameters are available in the HRBALG_DI block.

- ALGOREGSTATE
- ALGOPROCSTATE
- BOXCON
- BOXNUM
- BOXTYPE
- DEVCTL
- NUMINPTS
- INPTDIR
- PATHBRKSTS
- SLOTNUM
- SLOTTYPE

For more information about the parameters, refer to the *Control Builder Parameter Reference*.

3.17.5 Regulatory Algorithm Hiway Responder Block (HRBALG_REG)

The Regulatory Hiway Responder Block communicates with its associated Box HRB to connect the HG point on the LCN with native blocks in a Control Module assigned to a CEEC300. It is specifically designed to present data of regulatory CEE blocks to the HG regulatory point.

The following parameters are available in the Regulatory block:

- ALGOREGSTATE
- ALGOPROCSTATE
- ALGOTYPE
- AUTOMAN
- AUTOMAN2
- BOOLEANA, BOOLEANB, BOOLEANC
- BOXNUM
- BOXTYPE
- BOXCON
- DATAACQ1
- DATAACQ2
- DATAACQ3
- DEADTIME
- ENHREGCALC
- HSWDESC
- HSWVALUE

- INT32A, INT32B, INT32C
- LEADLAG
- MODEEM
- MODEPOLICING
- MPVEM
- NUMERIC1, NUMERIC2
- OPEM
- OPTOAOCONN
- OVRDSEL2
- PTHBRKSTS
- PID
- PVEM
- RAMPSOAK
- RATIOEM
- REALA, REALB, REALC
- REGCALC
- REMCAS
- SLOTNUM
- SLOTTYPE
- STARTOPSEL
- SPEM

For more information about the parameters on the Native Block Reference tab, refer to *Control Builder Parameter Reference*.

For more information about the parameters, refer to the *Control Builder Parameter Reference*.

3.18 Utility Blocks

Related topics

"ALMWINDOW (Alarm Window - Alarm Annunciator) block" on page 222
"ANNPANEL (Annunciator Panel - Alarm Annunciator) block" on page 223
"DIGACQ (Digital Acquisition) block" on page 224
"EXECTIMER" on page 225
"FIRSTOUT (First Out Detection) block" on page 226
"FLAG block" on page 227
"FLAGARRAY block" on page 228
"MESSAGE block" on page 228
"NUMERIC block" on page 230
"NUMERICARRAY block" on page 230
"PUSH block" on page 230
"TEXTARRAY block" on page 231
"TEXTCOMMENT (Text Comment) block" on page 232
"TIMER block" on page 232
"TYPECONVERT block" on page 233

3.18.1 ALMWINDOW (Alarm Window - Alarm Annunciator) block

Attention
The ALMWINDOW block can only be used with C300, C200E, and ACE controllers.

Description	The Alarm Window (ALMWINDOW) function block accepts boolean inputs (1 to 16) and performs the configured sequence. It provides one Alarm output (ALMOUT) and group status output (FLSHSTAT). (The FLASHSTAT is further connected by the user to the Annunciator Panel function block during configuration) The alarm annunciator is implemented as two blocks, one encapsulating the function of individual alarm group, and one to control the lamp test, acknowledge, and reset functions.
Function	• The Alarm Window function block accepts boolean inputs (1 to 16) and performs the configured sequence. It provides one alarm output (ALMOUT) and group status output (FLSHSTAT). • It accepts multiple inputs (max 16) and provides system alarm in case of abnormal input. • ALMWINDOW block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.
Inputs	• ALMIN [1..16] - Boolean inputs whose transitions need to be monitored. The block supports 16 inputs. By default, only four inputs are exposed. • The NUMIN input parameter decides the number of alarm inputs that can be connected to the block.
Outputs	• FLSHSTAT - It takes the states, FASTFLASH, SLOWFLASH, LAMPSTEADY, and LAMPOFF. • ALMOUT - OR of all inputs the Alarm WIndow.

Parameters	ALMIN[1..16] ALMOUT ALMSEQ FLSHSTAT HIALM.PR HIALM.SV HIALM.TYPE	INALM NAME NUMIN OFFNRMALM.FL OFFNRMALM.PR OFFNRMALM.SV OFFNRMALM.TM OFFNRMALM.TMO
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ALMWINDOW Block.	

3.18.2 ANNPANEL (Annunciator Panel - Alarm Annunciator) block

Attention

The ANNPANEL block can only be used with C300, C200E, and ACE controllers.

Description	This Annunciator Panel function block accepts FLSHSTAT from the Alarm Window function block and provides Lamp output for the annunciation windows with synchronized lamp flash sequence and hooter annunciation. This block also accepts the TEST input which forces the entire Lamp out to glow steady. This block establishes a hidden connection with the Alarm window function block to pass the RESET and ACK parameter values.	
Function	• Accepts multiple window block output (max 32) and provides lamp and hooter outputs. • The FLSHSTAT can take several states including FASTFLASH, SLOWFLASH, LAMPSTEADY, and LAMPOFF. • Enables an input to be wired as the operator acknowledge button. • Enables an input to be wired as the operator RESET button. • Enables an input to be wired as the operator lamp test button. • Description is fetched from the preceding Alarm window block. Be sure the Control Module containing the ANNPANEL block is configured for an Execution Period of 100 milliseconds or faster. The flashing rate of the annunciator panel only works as expected when the block is placed in a 100 millisecond or faster CM.	
Inputs	• FLSHSTAT[1..32] - It can take several states including FASTFLASH, SLOWFLASH, LAMPSTEADY, and LAMPOFF • LAMP TEST • RESET • ACK	
Outputs	• LAMPOUT[1..32] - It can take several states including FASTFLASH, SLOWFLASH, LAMPSTEADY, and LAMPOFF. • OUTHORN1 - It turns ON if any of FLSHSTAT is in flastflashing mode • OUTHORN2 - It turns ON if any of FLASSTAT is in slowflashing mode.	
Parameters	ACK ALMWINTXT[1..32] FLSHSTAT[1..32] FSTFLSHSPD LAMPOUT[1..32] LAMPTEST	NUMANNWIN OUTHORN1 OUTHORN2 RESET SLWFLSHSPD

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the ANNPANEL Block.

3.18.3 DIGACQ (Digital Acquisition) block

Attention
The DIGACQ block can only be used with C300, C200E, and ACE controllers.

<table>
<tr><td>Description</td><td colspan="3">The Digital Acquisition function block uses a combination of a DICHANNEL and SEL/FLAG where PVSOURCE is defined by the operator. The Digital Acquisition block recieves input from the DI Channel block. This block is independent of Channel type that feeds the block.</td></tr>
<tr><td>Function</td><td colspan="3">• Enables the user to specify the source of the process variable - AUTO, SUB or MAN.
– AUTO: Value is taken from the Switch
– SUB: Value is taken from the PROGRAM or other CM
– MAN: input is Operator specified
• Enables manual force OPEN or CLOSE of the field digital input by operators during maintenance of field switches.
• Supports alarm generation, when the current process variable state is different from the configured NORMAL state.
• With R410, DIGACQ block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time.</td></tr>
<tr><td>Inputs</td><td colspan="3">• IN - Input parameter
• PVFL (In case PVSOURCE is SUB)</td></tr>
<tr><td rowspan="7">Outputs</td><td colspan="3">• PV - Currently selected input based on the PVSOURCE selection
• PVFL - Actual State Flag
• INVPVFL - Inverted State Flag
• Depending on the value of PVSRCOPT and PVSOURCE, the output is set to one of the following input values:</td></tr>
<tr><td>PVSRCOPT</td><td>PVSOURCE</td><td>PV</td></tr>
<tr><td>Only Auto</td><td>AUTO</td><td>Value of PVAUTOFL</td></tr>
<tr><td>ALL</td><td>AUTO</td><td>Value of PVAUTOFL</td></tr>
<tr><td>ALL</td><td>MAN</td><td>Value of PVFL</td></tr>
<tr><td>ALL</td><td>SUB</td><td>Value of PVFL</td></tr>
<tr><td colspan="3">• The PVFL and PV parameters are always matched. When the PVSOURCE is changed to MAN, the value of PVFL/PV is retained at the last value. This value can be changed as needed.
• INVPVFL is a read-only value which is inverse of the PVFL value.</td></tr>
</table>

Parameters	BADPVALM.FL BADPVALM.PR BADPVALM.SV BADPVALM.TM BADPVALM.TMO DABLOCKSINCM HIALM.PR HIALM.SV HIALM.TYPE IN INALM INVPVFL NAME NORMAL OFFNRMALM.FL OFFNRMALM.PR	OFFNRMALM.SV OFFNRMALM.TM OFFNRMALM.TMO PV PVAUTOFL PVFL PVSOURCE PVSRCOPT PVSTS PVSTSFL.BAD PVSTSFL.MAN PVSTSFL.NORM PVSTSFL.UNCERTN STATE0 STATE1 STATETEXT[0..1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the DIGACQ Block.	

3.18.4 EXECTIMER

Attention

The EXECTIMER Block can only be used with C300, C300 20MS, C200E, ACE, SIMC300, SIMC200E, and SIMACE. It is not supported for use on C200 controller.

Description	EXECTIMER is a CEE utility block used to measure execution timing of other CEE blocks.	
Function	EXECTIMER is used by creating two instances. One instance marks the beginning of a time interval, that is, the “BEGTIME” instance. The other instance marks the end of a time interval, that is, the “ENDTIME” instance. The output parameter BEGTIME.TIMEOUT is then connected to the input parameter ENDTIME.TIMEIN. With this configuration, any module, block, group of modules or group of blocks which execute between the two EXECTIMER instances is included in the time measurement.	
Inputs	The input parameter is ENDTIME.TIMEIN. There is no input for the BEGTIME instance that marks the beginning of a time interval.	
Outputs	The output parameter is BEGTIME.TIMEOUT. There is no output for the ENDTIME instance that marks the end of a time interval.	
Parameter	DTANORM DTAOFFSET DTASCALE DTIME DTIMEAVG DTIMEAVGCOMP DTIMEMAX DTIMEMIN DTIMESTD	DTIMESTDPRC ENABLE MAXMINRATIO REJFACTOR RESET TAU TIMEIN TIMEOUT

Reference	Refer to the *Control Builder Parameter Reference,* for definitions of each parameter. Refer to the *Control Builder Component Theory,* for more information on the EXECTIMER Block.

3.18.5 FIRSTOUT (First Out Detection) block

Attention
The FIRSTOUT block can only be used with C300, C200E, and ACE controllers.

Description	A First Out logic enables you to identify the digital input signal that was first to transition from its normal state, amongst a set of digital inputs connected to an equipment or a device. Usually, this block is associated with critical equipment. An equipment's or a drive's protection interlocks and stop commands are connected as input to the First up block. When an input signal transitions from its configured NORMAL state, the output flag of the First Out logic is raised. In addition, the input responsible for the First Out flag is recorded. All the logic processing is performed during runtime processing of the block. The recording is locked until a reset is applied to the block after all inputs are back to Normal state.
Function	• The block provides the First Out function. A First Out logic enables you to identify which digital input signal was first to transition from NORMAL state, amongst a set of digital inputs connected to the block. • The set of digital inputs connected to the block is scanned in ascending order and once a transition (from NORMAL state) is detected, the First Out is flagged and further scanning is stopped for rest of the cycles until a RESET. In a scenario where more than one input transitions in a single cycle, say 2 and 8, from the NORMAL state, the FIRSTOUTACTED flag is set. This leads to INPUTACTED [2] and INPUTACTED [8] to turn ON. The FIRSTOUTINPUT takes the value of “Multiple”. • The block provides an output which is an OR of all NORMAL state inputs and it goes high if any input goes to ABNORMAL state. It resets when all inputs come back to NORMAL state. • Enables you to reset the First Out flag using a raising edge pulse input only when all inputs come back to NORMAL state. • Provides an alarm once a First Out is detected. If a single input transitions from NORMAL state, the input that caused the alarm is identified and its description (INDESC[*]) is used for alarm. In case of multiple input transitions in a single cycle, the alarm description is as defined in the MULTIINPTDESC (Multiple Input description field).
Inputs	• IN [1..24] - Boolean inputs whose transitions need to be monitored. The block supports 24 inputs. By default, only eight inputs are exposed. • You can view the configured normal state of input at the faceplate of the function block without opening the configuration page. The hollow diamond symbol indicates that the input pin on the faceplate is configured with the normal input state as “True.” In the Monitoring view, the hollow symbol is indicated in the color configured under the **Pins and Wires** tab of the **System preferences**. • RESET - This parameter is used to reset the First Out recordings. • TRAN+SMON (Transition monitoring) - This parameteris enabled by a user with Engineer access. If FIRSTOUT has already acted, TRANSMON cannot be enabled. However, if TRANSMON is enabled before FIRSTOUT acted, it will continue to monitor upto 64,534 cycles and capture the list of inputs that became ABNORMAL in each cycle.

Outputs	• FIRSTOUTACTED - This flag is set when there is an input transition from its configured normal state. • INPUTACTED[1..24] - Indicates whether the corresponding input has transitioned from NORMAL state. • FIRSTOUTINPUT - This is an enumeration that indicates which input triggered First Out. • OREDOUT - It is an OR of all NORMAL state inputs and it goes high if any input goes to ABNORMAL state. It resets when all inputs come back to NORMAL state.	
Parameters	FIRSTOUTACTED FIRSTOUTALM.FL FIRSTOUTALM.PR FIRSTOUTALM.SV FIRSTOUTINPUT HIALM.PR HIALM.SV HIALM.TYPE IN[*] INDESC[*]	FIRSTOUTACTED FIRSTOUTALM.FL FIRSTOUTALM.PR FIRSTOUTALM.SV FIRSTOUTINPUT HIALM.PR HIALM.SV HIALM.TYPE IN[*] INDESC[*]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the FIRSTOUT Block.	

3.18.6 FLAG block

Description	Provides storage for a single two-state value which can be accessed as a simple Boolean (OFF or ON) value using the PVFL parameter, or as one of two user-configured state values (for example, Running and Stopped) via the PV parameter.
Function	• Used to define two separate states (for example, Running/Stopped, Off/On) to indicate status of a particular input. • There are 2 user-configurable state descriptors, STATETEXT[0] and STATETEXT[1] that are used to describe STATE0 and STATE1 respectively. • Current state of flag can be changed/read using PVFL (Boolean) or using PV (either STATETEXT[0] or STATETEXT[1]). • Block also supports: – configurable access lock which determines who can write a value to the block (such as operator, engineer, or other function block). – an off-normal alarm whereby one of the flag's states is configured as the normal state; whenever the flag changes state, the off-normal alarm is generated. – an off-normal alert whereby one of the flag's states is configured as the normal state; whenever the flag modifies state, the off-normal alert is generated. • With R410, FLAG block allows you to configure individual values for the on-delay time and off-delay time for the individual alarms. A new parameter (xxxxALM.TMO) is introduced for individual alarms to configure the off-delay time. Note: The ALTENBOPT parameter can be used to enable the block to generate an off-normal alarm or an off-normal alert whenever the flag's state differs from the configured normal state. • When ALTENBOPT is set to ON, the FLAG block can be configured to generate an alert on an off-normal condition. • When ALTENBOPT is set to OFF, the FLAG block can be configured to generate an alarm on an off-normal condition.
Inputs/Outputs	PVFL is the only exposed block connection, but you can expose other block parameters through the Configure Block form in Control Builder.

Parameters	ACCLOCK ALTENBOPT DESC EUDESC HIALM.PR HIALM.SV HIALM.TYPE INALM INALT NAME NORMAL OFFNRMALM.FL	OFFNRMALM.PR OFFNRMALM.SV OFFNRMALM.TM OFFNRMALM.TMO ORDERINCM PV PVFL STATE0 STATE1 STATETEXT[0..1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the FLAG Block.	

3.18.7 FLAGARRAY block

Description	Provides storage for up to 1000 2-state values. The value can be accessed as a simple Boolean (Off or On) using the PVFL[n] parameter. Where "n" is the number of the flag.	
Function	• Used to define two separate states (Off/On) to indicate status of a particular input. • Number of flag values (NFLAG) is user configurable. Current state of flags can be changed/read using flag value (PVFL[n]) (Boolean).	
Inputs/Outputs	Boolean output flags (PVFL[1..1000])	
Parameters	ACCLOCK NAME NFLAG	PVFL[1..1000]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the FLAGARRAY Block.	

3.18.8 MESSAGE block

Description	Provides up to 16 user configurable messages (MESSAGE[n]) that can be triggered by a client of the block. Where "n" is the number of the message. A client can be the output from a Step block in a Sequential Control Module (SCM).

<table>
<tr><td>Function</td><td colspan="2">

- Each message type can be configured (MSGTYPE[0..15]) to be either:
  - Information,
  - Confirmable,
  - Single Signature*, or
  - Double Signature*

* You must have the Electronic Signature system **license** to use Single Signature and Double Signature message types.

- When a client triggers a given send flag (SENDFL [n]) input, the corresponding message (MESSAGE[n]) is sent to the Message and the Event Summary displays in the Station application.
- For information) only type (INFO) messages, the client trigger sets the corresponding SENDFL[n] to True. Since the SENDFL[n] is a pulse trigger, it is automatically set to False during the next execution cycle. This means the MESSAGE block is ready to send the same message again in the next cycle.
- For confirmation type (CONFIRM) messages, the client trigger pulses the corresponding SENDFL[n] to send the MESSAGE[n] to the Server. The client of the MESSAGE block checks for the confirmed parameter (CONFIRMED[n]) to be set to True. The CONFIRMED[n] parameter indicates whether the MESSAGE block has received a confirmation.
- For single signature type (SINGLESIGNATURE) messages, the client trigger pulses the corresponding SENDFL[n] to send the MESSAGE[n] to the Server. Once a user acknowledges the message twice to confirm it through the Message Summary Display in Station, a Single Signature user interface appears for the user to record an electronic signature. The MEANINGPRI[n] parameter provides an indication for the meaning of the primary signature. Once the message is acknowledged and signature is obtained, the Message Summary Display sends a confirmation to the MESSAGE block that turns on the CONFIRMED[n] parameter to show that the message has been confirmed.
- For double signature type (DOUBLESIGNATURE) messages, the client trigger pulses the corresponding SENDFL[n] to send the MESSAGE[n] to the Server. Once a user acknowledges the message twice to confirm it through the Message Summary Display in Station, a Single Signature and Double Signature user interface appear for the user to record the required electronic signatures. The MEANINGPRI[n] and MEANINGSEC[n] parameters provide indications for the meaning of the primary and secondary signatures, respectively. Once the message is acknowledged and signatures are obtained, the Message Summary Display sends a confirmation to the MESSAGE block that turns on the CONFIRMED[n] parameter to show that the message has been confirmed. In addition, the MINLVLSECSIG[n] parameter lets users define the minimum security level required for a secondary signature.
- If the Message block is connected to a block whose output is a Boolean and the Boolean becomes True, the Message block will generate the message every time the Message block executes until the Boolean output turns False. If the desired behavior is to generate a one-time message, then the message block should be driven by a Pulse Output.

</td></tr>
<tr><td>Inputs/Outputs</td><td colspan="2">Up to 16 inputs (SENDFLAG[0..15]) and 16 outputs (CONFIRMED[0..15]), depending on the message types configured.</td></tr>
<tr><td>Parameters</td><td>CONFIRM[0..15]
CONFIRMED[0..15]
DESC
EUDESC
MEANINGPRI[0..15]</td><td>MEANINGSEC[0..15]
MESSAGE[0..15]
MINLVLSECSIG[0..15]
MSGTYPE[0..15]
NAME
ORDERINCM
SENDFL[0..15]</td></tr>
</table>

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the MESSAGE Block.	

3.18.9 NUMERIC block

Description	Provides storage for a floating point value that is accessible via the PV configuration parameter.	
Function	• Used to store up to 8 bytes of a floating point value within defined upper and lower limits for use in a control strategy. • Configurable high and low limits are also provided. • Also supports a configurable access lock which determines who can write a value to the block (such as operator, engineer, or another function block.).	
Inputs/Outputs	PV is only exposed block connection, but you can expose other block parameters through the Configure Block form in Control Builder.	
Parameters	ACCLOCK ORDERINCM PV	PVFORMAT PVHILM PVLOLM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the NUMERIC Block.	

3.18.10 NUMERICARRAY block

Description	Provides storage for up to 200 floating point values that are accessible through the corresponding PV configuration parameter (PV[n]). Where“n” is the number of the numeric.	
Function	Use outputs (PV[1..200]) as source parameters to provide predefined analog constants to other function blocks. A bad numeric output parameter typically has the value NaN (Not-a-Number). Number of Numeric Values (NNUMERIC) is user configurable.	
Inputs/Outputs	Up to 200 outputs (PV[1..200]), depending on the number of numeric values (NNUMERIC) configured	
Parameters	ACCLOCK NAME NNUMERIC	PV[1..200] PVFORMAT
Reference	R Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the NUMERICARRAY Block.	

3.18.11 PUSH block

Description	The PUSH function block provides store/push capability of different data types to the output destination. The block fetches input value and stores in this cycle to a destination parameter. The output is stored only if the output store enable flag (STOREENB) is true. When STOREENB is true, and store on change (STORONCHGENB) is true, the output is stored upon change of CM or CEE state, or when the input changes.

<table>
<tr><td>Function</td><td colspan="2">The PUSH function block supports all the data types supported by the TypeConvert function block.

Once the first store is successful, the block has features to store only on change for outputs of BOOLEAN and FLOAT64 data type. This feature is provided to prevent the unnecessary traffic on the network if the stores are peer-to-peer.

The function block performs the type conversion and clamping before it pushes the value to the destination. The destinations are configured either by using parameter connector edit box using point picker in the control builder or by using wire connections.</td></tr>
<tr><td>Inputs/Outputs</td><td colspan="2">If the PUSH block is able to fetch the value, then it is pushed to the destination.

There are two statuses provided; EXECSTS and STORESTS. These will give the user indication of the status of fetching input and status of output store.

If STORONCHGENB is set to ON, the input values are stored to the destination ONLY when the input is changed. In the case of floating point number, the value is considered changed only if the absolute value of the difference between the new value and the last value is larger than the configured INPUTCHGDB.

BOOLVALUEOFF, BOOLVALUEON and ENUMTOBOOLMAP[0..63] behave the same as in type convert block.</td></tr>
<tr><td>Parameters</td><td>BOOLVALUEOFF
BOOLVALUEON
ENUMBOOLMAP[0..63]
EXECSTS
IN.BOOLEAN
IN.ENUM
IN.FLOAT32
IN.FLOAT64
IN.INT8
IN.INT16
IN.INT32
IN.SDENUM
IN.UINT16
IN.UINT32
INPUTCHGDB
NAME
LASTSTORESTS
LASTSTORESTSN
ORDERINCM</td><td>OUT.BOOLEAN
OUT.ENUM
OUT.FLOAT32
OUT.FLOAT64
OUT.INT8
OUT.INT16
OUT.INT32
OUT.SDENUM
OUT.UINT16
OUT.UINT32
PVFL
SDENUMTEXT
STATE0
STATE1
STATETEXT[0..1]
STOREENB
STORESTS
STORONCHGENB
THRESHOLD
TRUNCATEOPT</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.

Refer to the Control Builder Component Theory for more information on the PUSH Block.</td></tr>
</table>

3.18.12 TEXTARRAY block

Description	Provides storage for up to 120 text strings that are accessible through the corresponding string configuration parameter (STR[n]). Where “n” is the number of the text string.

<table>
<tr><td>Function</td><td colspan="2">

Provides predefined text (STR[1..120]) strings to other blocks.
Number of string values (NSTRING) is user configurable.
The length of the text strings (STRLEN) is user configurable as 8, 16, 32, or 64 characters
Supports a maximum size of 960 two-byte characters. The maximum data combinations that you can configure through NSTRING and STRLEN values are as follows.

If NSTRING is 15 and STRLEN is 64, then the STR[n] range is 1 to 15.
If NSTRING is 30 and STRLEN is 32, then the STR[n] range is 1 to 30.
If NSTRING is 60 and STRLEN is 16, then the STR[n] range is 1 to 60.
If NSTRING is 120 and STRLEN is 8, then the STR[n] range is 1 to 120.

</td></tr>
<tr><td>Inputs/Outputs</td><td colspan="2">Up to 120 output strings (STR[1..120]), depending on the number of string (NSTRING) and length of string (STRLEN) values configured.</td></tr>
<tr><td>Parameters</td><td>ACCLOCK
NAME
NSTRING</td><td>ORDERINCM
STR[1..120]
STRLEN</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.
Refer to the Control Builder Component Theory for more information on the TEXTARRAY Block.</td></tr>
</table>

3.18.13 TEXTCOMMENT (Text Comment) block

<table>
<tr><td>Description</td><td colspan="2">The Text Comment function block can be used to insert comments into charts. The Text Comment block does not contain any logic and does not participate in control execution.
The Text Comment block can be used with the container modules such as CM, SCM, RCM, and UCM. The Text Comment block can be imported/exported like other function blocks. There is no restriction on the number of Text Comment blocks that can be inserted into a chart. The Text Comment block can be loaded while active without inactivating the strategy or setting the CEE to IDLE.</td></tr>
<tr><td>Function</td><td colspan="2">The Text Comment block can be used only to insert comments into a chart.</td></tr>
<tr><td>Inputs/Outputs</td><td colspan="2">The Text Comment block does not contain any logic and also does not have any pin connections.</td></tr>
<tr><td>Parameters</td><td>ORDERINCM
TEXTCOMMENT
BKGCOLOR
FONTSTYLE</td><td>TEXTCOLOR
TEXTSIZE</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.
Refer to the Control Builder Component Theory for more information on the TEXTCOMMENT block.</td></tr>
</table>

3.18.14 TIMER block

Description	Provides capability to time process events or create known delays.
Function	Used to keep track of elapsed time during a process and provides indication when elapsed time reaches predefined limit.

Inputs/Outputs	• SO is only exposed block connection, but you can expose other block parameters through the Configure Block form in Control Builder. • Commands are sent to timer in one of two ways: – By operator using COMMAND parameter. – Through connections to parameters STARTFL, STOPFL, RESETFL, and RESTARTFL.	
Parameters	COMMAND NAME ORDERINCM PV RESETFL RESTARTFL RV	SO SP STARTFL STATE STOPFL TIMEBASE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the TIMER Block.	

3.18.15 TYPECONVERT block

Description	Provides the ability to convert one data type to another for connecting parameters of different data types.
Function	• Supports data type conversions for all combinations among the following major data types. – Boolean – Integer (unsigned/signed 8/16/32-bit integers) – Real (32-bit and 64-bit IEEE floating point numbers) – Enumeration – If data type conversion is not necessary, then none will be done. • Used to connect one input parameter to one or many output parameters with different data types. The TYPCONVERT block reads the input value and only provides the converted output when the block connected to its output runs.
Inputs/Outputs	Up to nine inputs and nine outputs. The pins for the four most common inputs (IN.BOOLEAN, IN.INT32, IN.FLOAT64, IN.ENUM) and outputs (OUT.BOOLEAN, OUT.INT32, OUT.FLOAT64, OUT.ENUM) are exposed by default.

Parameters	BOOLVALUEOFF	ORDERINCM
	BOOLVALUEON	OUT.BOOLEAN
	ENUMBOOLMAP[0..63]	OUT.ENUM
	EXECSTS	OUT.FLOAT32
	IN.BOOLEAN	OUT.FLOAT64
	IN.ENUM	OUT.INT8
	IN.FLOAT32	OUT.INT16
	IN.FLOAT64	OUT.INT32
	IN.INT8	OUT.SDENUM
	IN.INT16	OUT.UINT8
	IN.INT32	OUT.UINT16
	IN.SDENUM	OUT.UINT32
	IN.UINT8	SDENUMTEXT
	IN.UINT16	THRESHOLD
	IN.UINT32	TRUNCATEOPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Component Theory* for more information on the TYPECONVERT Block.	

3.19 EtherNet/IP channel blocks

Related topics

"EtherNet/IP IO channel blocks" on page 235
"EtherNet/IP drive and relay channel blocks" on page 237

3.19.1 EtherNet/IP IO channel blocks

Related topics

"AICHANNEL" on page 235
"AOCHANNEL" on page 235
"DICHANNEL" on page 236
"DOCHANNEL" on page 236

3.19.1.1 AICHANNEL

Description	Provides standard analog interface to control function blocks.	
Function	• Brings PV data from an associated IOM block. • Assigns BAD status to PV parameter when appropriate.	
Inputs	Input data received from the field.	
Outputs	Floating point value in engineering units.	
Parameters	BADPVFL CHANNUM CONTAINEDIN FETCHMODE IOP ORDERINCM PV PVRAW	PVSTS SIMMODE SIMVALUE
Associated Block	Prior to loading, block must be "associated" with 1 channel of corresponding IOM block that interfaces with the physical AI hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.1.2 AOCHANNEL

Description	Provides a standard analog output signal for operating final control elements.
Function	• Brings OP data from connected blocks and conveys OP data to be stored in an associated IOM block. • Sets INITVAL parameter to appropriate value based on echo data. • Sets INITREQ to TRUE value if AOC or IOM block is inactive or a communications error occurs. • Reverses OP direction if OPTDIR option is REVERSE.
Inputs	Only one control block can interface to this block.

Outputs	Floating point value in engineering units.	
Parameters	BACKCALOUT BLKASSIGNSUPPORTED BLOCKLOGO BLKSTIDX BLKTYPE CHANNUM COMMFAILFL CONTAINEDIN INITREQ	INITREQLATCH INITVAL IOCTYPE IOP OP OPFINAL OPSOURCE OPTDIR SIMVALUE
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with the physical AO hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.1.3 DICHANNEL

Description	Provides a standard digital interface to control blocks.	
Function	• Brings PV data from an associated IOM block. • Assigns BAD status to PV parameter when appropriate.	
Inputs	Digital (PV) signals received from the field.	
Outputs	PV and PV status value that can be used by other data points in system.	
Parameters	BADPVFL CHANNUM CONTAINEDIN FETCHMODE INBADOPT IOP ORDERINCM PVFL	SIMMODE SIMVALUE
Associated Block	Prior to loading, the block must be associated with 1 channel of the corresponding IOM block that interfaces with physical Digital Input hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.1.4 DOCHANNEL

Description	Generates status output [0 or 1), based on origin of input and parameters.
Function	Brings SO from connected blocks and stores value in an associated IOM block. Sets INITVAL parameter to appropriate value based on echo data. Stops SO if INTREQ is TRUE Sets INITREQ to TRUE value if DOC or IOM block is inactive or a communications error occurs.
Inputs	Only one control block can interface to this block.
Outputs	Digital (Boolean) value

Parameters	INITVAL INITREQ BACKCALOUT SO SOSOURCE INITREQLATCH BLKTYPE	CONTAINEDIN BLKSTIDX IOCTYPE CHANNUM IOP BLKASSIGNSUPPORTED BLOCKLOGO
Associated Block	Prior to loading, the block must be “associated” with 1 channel of corresponding IOM block that interfaces with the physical DO hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.2 EtherNet/IP drive and relay channel blocks

Related topics

“PowerFlex Drive input channel” on page 237
“PowerFlex Drive output channel” on page 238
“E3 and E3 plus IN channels” on page 239
“E3 and E3 plus OUT channels” on page 240

3.19.2.1 PowerFlex Drive input channel

Description	Provides standard analog interface and Drive-specific parameters to control function blocks.
Function	• Brings PV data from an associated Drive block. • Brings Drive specific data from an associated Drive block. • Assigns BAD status to PV and also sets BADDATA parameter (used for Drive-specific parameters) when appropriate.
Inputs	Input data received from the field.
Outputs	Floating point value in engineering units for PV. Other Drive-specific parameters.

Parameters	ACCELERATING ACTDIRECT ACTIVE ALARM ATHOME ATLIMIT ATSPEED ATZEROSPEED BADDATA BADPVFL BUSFREQREG CHANNUM CLRFLTOWNER CMDDIRECT CONTAINEDIN CURLIMIT DBACTIVE DCBRAKE DECELERATING EDITLOCK ELAPSEDKWH ELAPSEDMWH ELSRUNTIME ENABLED FAULTAMPS FAULTED FAULTFREQ FETCHMODE	FLTBUSV IOP JOGGING JOGOWNER LASTFLTCODE LASTSTARTSRC LASTSTOPSRC MANUAL MOTOROVRLD OPCURRENT ORDERINCM POSMODE PV PVFL PVRAW PVSTS REGEN RUNNING RUNREADY SIMMODE SIMVALUE SPDREFSRC SPEEDMODE SPREFID STARTOWNER STOPOWNER STOPPING STRTINH TRQMODE
Associated Block	Prior to loading, the block must be associated with 1 channel of the corresponding IOM block that interfaces with the physical Drive hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.2.2 PowerFlex Drive output channel

Description	Provides a standard analog output signal and Drive-specific commands for operating final control elements.
Function	• Brings OP data and Drive specific commands from connected blocks and conveys the data to be stored in an associated Drive block. • Sets INITVAL parameter to an appropriate value, based on echo data. • Assigns safe OP value if status is BAD or UNCERTAIN. • Sets INITREQ to TRUE value if AOC or IOM block is inactive or a communications error occurs and also sets BADDATA parameter (used for Drive-specific commands) when appropriate.

Inputs	Inputs from blocks connected to the channel.	
Outputs	Floating point value in engineering units for PV. Other Drive-specific parameters.	
Parameters	ACCELTIME	JOGADTIME
	ACCELTIME1	JOGSPD1
	ACCELTIME2	JOGSPD2
	BADDATA	MANUALCMD
	CHANNUM	MAXFWDSPD
	CLEARFAULT	MAXREVSPD
	CMDDIRECTION	NORMALSTOP
	COASTSTOP	OP
	COMMFAILFL	OPCURRENT
	CONTAINEDIN	OPFINAL
	CURRLIMSTOP	ORDERINCM
	DECELTIME	OVERSPDLIMIT
	DECELTIME1	REFSELECTOR
	DECELTIME2	RUN
	EDITLOCK	SIMMODE
	FETCHMODE	SPDREFASTPT
	INITREQ	SPDREFBSTPT
	INITREQLATCH	SPDREFSCALE
	INITVAL	START
	IOP	STOPMODEA
	JOG1	STOPMODEB
	JOG2	ZEROSPDLIMIT
Associated Block	Prior to loading, the block must be associated with 1 channel of the corresponding IOM block that interfaces with the physical Drive hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.2.3 E3 and E3 plus IN channels

Description	Provides standard digital interface and Relay specific parameters to control function blocks.
Function	• Brings digital input data from an associated Relay block. • Brings Relay-specific data from an associated Relay block. • Sets BADDATA parameter (used for Device specific parameters) when appropriate.
Inputs	Input data received from the field.
Outputs	Digital Inputs value. Other Relay-specific parameters.

Parameters	**Parameters for E3** • E3 Device Status • E3 Data Status • E3_0_L1_CURR • E3_0_L2_CURR • E3_0_L3_CURR • DeviceNet Scanner Status • E3 Advance Data Status • CHANNUM • CONTAINEDIN • EDITLOCK • IOP • ORDERINCM • SIMMODE • EDITLOCK • FETCHMODE • SIMVALUE	**Parameters for E3 plus** • E3 Device Status • E3 Data Status • E3_0_L1_CURR • E3_0_L2_CURR • E3_0_L3_CURR • DeviceNet Scanner Status • E3 Advance Data Status • CHANNUM • CONTAINEDIN • EDITLOCK • IOP • ORDERINCM • SIMMODE • EDITLOCK • FETCHMODE • BADDATA • SIMVALUE
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding IOM block that interfaces with the physical relay hardware module at execution runtime.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

3.19.2.4 E3 and E3 plus OUT channels

Description	Provides a standard digital output signal and Relay specific commands for operating final control elements.	
Function	• Brings Digital Output data and relay-specific commands from connected blocks and conveys the data to be stored in an associated Relay block. • Sets INITVAL parameter to appropriate value based on echo data. • Sets BADDATA parameter (used for Device specific commands) when appropriate.	
Inputs	Inputs from blocks connected to the channel.	
Outputs	Digital Output value. Other Drive specific commands.	
Parameters	**Parameters for E3** CHANNUM CONTAINEDIN EDITLOCK SIMMODE FETCHMODE IOP ORDERINCM BADDATA SIMVALUE	**Parameters for E3 plus** CHANNUM CONTAINEDIN EDITLOCK SIMMODE FETCHMODE IOP ORDERINCM BADDATA SIMVALUE
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding relay block that interfaces with the physical relay hardware module at execution runtime.	

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.

3.20 IOREFERENCES Blocks

Related topics

"AIREF block" on page 242
"AOREF block" on page 243
"DIREF block" on page 244
"DOREF block" on page 244

3.20.1 AIREF block

<table>
<tr><td>Description</td><td colspan="2">Allows you to
• associate a configured analog input channel or a pulse input channel in a Control Module, or
• reference a parameter value of any block.</td></tr>
<tr><td rowspan="4">Function</td><td colspan="2">The PV parameter represents a value of its reference (REF). The AIREF block makes the PV available for connections to other blocks. The origin of PV is listed in the following table based on the REFTYPE parameter configuration.</td></tr>
<tr><td>If REFTYPE is configured as ...</td><td>Then, the PV origin is...</td></tr>
<tr><td>PMIO or SERIES_C_IO</td><td>PV value of the channel.</td></tr>
<tr><td>PARAMETER</td><td>values that are accessible in the Control Builder. The values can be
• Point.Parameters, or
• Point.Block.Parameter</td></tr>
<tr><td>Parameters</td><td colspan="2">CHANTYPE
NAME
REF
REFTYPE</td></tr>
<tr><td>Configuration Rules</td><td colspan="2">Reference type (REFTYPE) configuration rules are follows:
• If the REFTYPE is set to SERIES_C_IO, then the Reference block references a Series C I/O channel.
• If the REFTYPE is set to PMIO, then the Reference block references a PM I/O channel.
• If the REFTYPE is set to PARAMETER, then the Reference block references a parameter and works as a parameter connector.
• If the REFTYPE is NONE, then the Reference block is identical to any other basic block and does not have any special configuration consideration.
Reference (REF) must be selected based on the REFTYPE configuration.
For more information about the REFTYPE parameter, see Control Builder Components Theory.</td></tr>
<tr><td>Associated Block</td><td colspan="2">Prior to loading the CM containing the AIREF block, REF value must be associated with one channel block of corresponding IOM block that interfaces with the physical AI hardware module at runtime when the REFTYPE is set to Series C I/O or PM I/O.
Prior to loading the CM containing the AIREF block, REF value must be associated with valid parameter value when the REFTYPE is set to "PARAMETER."</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.</td></tr>
</table>

3.20.2 AOREF block

<table>
<tr><td>Description</td><td>Allows you to
associate a configured analog output channel in a Control Module, or
reference a parameter value of any block.</td></tr>
<tr><td>Function</td><td>The OP parameter represents a value of its reference (REF). The AOREF block makes the OP available for connections to other blocks. The AOREF block cannot accept the OP values from different controllers. The origin of OP is listed in the following table based on the REFTYPE parameter configuration.
<table>
<tr><th>If REFTYPE is configured as ...</th><th>Then, the OP origin is...</th></tr>
<tr><td>PMIO or SERIES_C_IO</td><td>OP value of the channel.</td></tr>
<tr><td>PARAMETER</td><td>values that are accessible in the Control Builder. The values can be
Point.Parameters, or
Point.Block.Parameter
Attention
The OP values are accepted only from the PUSH blocks or PUSH connections.</td></tr>
</table></td></tr>
<tr><td>Parameters</td><td>CHANTYPE
NAME
REF
REFTYPE</td></tr>
<tr><td>Configuration Rules</td><td>Reference type (REFTYPE) configuration rules are follows:
If the REFTYPE is set to SERIES_C_IO, then the Reference block references a Series C I/O channel.
If the REFTYPE is set to PMIO, then the Reference block references a PM I/O channel.
If the REFTYPE is set to PARAMETER, then the Reference block references a parameter instead of a Series C I/O or PM I/O channel.
Attention
When you set the REFTYPE as PARAMETER, you can use only PUSH parameters for the OP parameter of AOREF blocks.
If the REFTYPE is NONE, then the Reference block is identical to any other basic block and does not have any special configuration consideration.
Reference (REF) must be selected based on the REFTYPE configuration.
For more information about the REFTYPE parameter, see Control Builder Components Theory.</td></tr>
<tr><td>Associated Block</td><td>Prior to loading the CM containing the AOREF block, REF value must be associated with one channel block of corresponding IOM block that interfaces with the physical AO hardware module at runtime when the REFTYPE is set to Series C I/O or PM I/O.
Prior to loading the CM containing the AOREF block, REF value must be associated with valid parameter value when the REFTYPE is set to “PARAMETER.”</td></tr>
<tr><td>Reference</td><td>Refer to the Control Builder Parameter Reference for definitions of each parameter.</td></tr>
</table>

3.20.3 DIREF block

<table>
<tr><td>Description</td><td colspan="2">Allows you to
associate a configured digital input channel in a Control Module, or
reference a parameter value of any block.</td></tr>
<tr><td rowspan="4">Function</td><td colspan="2">The PV parameter represents a value of its reference (REF). The AIREF block makes the PV available for connections to other blocks. The origin of PV is listed in the following table based on the REFTYPE parameter configuration.</td></tr>
<tr><td>If REFTYPE is configured as ...</td><td>Then, the PV origin is...</td></tr>
<tr><td>PMIO or SERIES_C_IO</td><td>PV value of the channel.</td></tr>
<tr><td>PARAMETER</td><td>values that are accessible in the Control Builder. The values can be
Point.Parameters, or
Point.Block.Parameter</td></tr>
<tr><td>Parameters</td><td colspan="2">CHANTYPE
NAME
REF
REFTYPE</td></tr>
<tr><td>Configuration Rules</td><td colspan="2">Reference type (REFTYPE) configuration rules are follows:
If the REFTYPE is set to SERIES_C_IO, then the Reference block references a Series C I/O channel.
If the REFTYPE is set to PMIO, then the Reference block references a PM I/O channel.
If the REFTYPE is set to PARAMETER, then the Reference block references a parameter instead of a Series C I/O or PM I/O channel.
If the REFTYPE is NONE, then the Reference block is identical to any other basic block and does not have any special configuration consideration.
Reference (REF) must be selected based on the REFTYPE configuration.
For more information about the REFTYPE parameter, see Control Builder Components Theory.</td></tr>
<tr><td>Associated Block</td><td colspan="2">Prior to loading the CM containing the DIREF block, REF value must be associated with one channel block of corresponding IOM block that interfaces with the physical DI hardware module at runtime when the REFTYPE is set to Series C I/O or PM I/O.
Prior to loading the CM containing the DIREF block, REF value must be associated with valid parameter value when the REFTYPE is set to “PARAMETER.”</td></tr>
<tr><td>Reference</td><td colspan="2">Refer to the Control Builder Parameter Reference for definitions of each parameter.</td></tr>
</table>

3.20.4 DOREF block

<table>
<tr><td>Description</td><td>Allows you to
associate a configured digital output channel in a Control Module, or
reference a parameter value of any block.</td></tr>
<tr><td>Function</td><td>The SO parameter represents a value of its reference (REF). The DOREF block makes the SO available for connections to other blocks. The DOREF block cannot accept the SO values from different controllers. The origin of SO is listed in the following table based on the REFTYPE parameter configuration.</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><th>If REFTYPE is configured as ...</th><th>Then, the SO origin is...</th></tr>
<tr><td>PMIO or SERIES_C_IO</td><td>SO value of the channel.</td></tr>
<tr><td>PARAMETER</td><td>values that are accessible in the Control Builder. The values can be
• Point.Parameters, or
• Point.Block.Parameter

Attention
The SO values are accepted only from the PUSH blocks or PUSH connections.</td></tr>
</table>
</td></tr>
<tr><td>Parameters</td><td>CHANTYPE
NAME
REF
REFTYPE</td></tr>
<tr><td>Configuration Rules</td><td>Reference type (REFTYPE) configuration rules are follows:
• If the REFTYPE is set to SERIES_C_IO, then the Reference block references a Series C I/O channel.
• If the REFTYPE is set to PMIO, then the Reference block references a PM I/O channel.
• If the REFTYPE is set to PARAMETER, then the Reference block references a parameter instead of a Series C I/O or PM I/O channel.

Attention
When you set the REFTYPE as PARAMETER, you can use only PUSH parameters for the SO parameter of DOREF blocks.

• If the REFTYPE is NONE, then the Reference block is identical to any other basic block and does not have any special configuration consideration.

Reference (REF) must be selected based on the REFTYPE configuration.

For more information about the REFTYPE parameter, see Control Builder Components Theory.</td></tr>
<tr><td>Associated Block</td><td>Prior to loading the CM containing the DOREF block, REF value must be associated with one channel block of corresponding IOM block that interfaces with the physical DO hardware module at runtime when the REFTYPE is set to Series C I/O or PM I/O.

Prior to loading the CM containing the DOREF block, REF value must be associated with valid parameter value when the REFTYPE is set to “PARAMETER.”</td></tr>
<tr><td>Reference</td><td>Refer to the Control Builder Parameter Reference for definitions of each parameter.</td></tr>
</table>

4 Reference Data for Physical Equipment Block Types

This section provides detailed reference data for each physical equipment block type that is part of the hardware relation category for the Control Builder. The reference data is organized by the major block type and then alphabetically within the IOM function block by the given module's model number and input or output type.

The reference data for each block covers these topics as applicable:

- description
- function
- inputs and input ranges
- outputs and output ranges
- parameters

Related topics

“Control Processor Module Block” on page 248
“Control Execution Environment Block” on page 256
“Input/Output Link Interface Module Block” on page 269
“Input/Output Link Block” on page 270
“OLE for Process Control Server Block” on page 278
“Inter Cluster Gateway Block” on page 279
“Redundancy Module Block” on page 282
“Fault Tolerant Ethernet Bridge Module Block” on page 284
“Input Type I/O Module Blocks” on page 287
“Output Type I/O Module Blocks” on page 300
“Serial Interface Module (SIM) I/O Module Block” on page 312
“Process Manager Input/Output (PMIO) Blocks” on page 313
“Series C Input/Output (I/O) Blocks ” on page 332
“Series C IEC 61850 Interface Module (850M) blocks” on page 341
“Speed Protection Module (SPM)” on page 342
“Servo Valve Positioner (SVP) Module ” on page 350
“Universal Input/Output (UIO) Module” on page 357
“Peer Control Data Interface (PCDI) Blocks” on page 361
“PROFIBUS Gateway Module (PGM) Blocks” on page 369
“Foundation FieldBus Interface Module (FIM) Blocks” on page 371
“Enhanced High-Performance Process Manager (EHPM) Block” on page 372
“EtherNet/IP blocks” on page 374

This topic provides reference information about EtherNet/IP I/O devices, PowerFlex drives, E3, and E3 plus relays.

“ControlLogix and UDT blocks” on page 386

4.1 Control Processor Module Block

Related topics

"CPM Block (C200 Controller)" on page 248
"C200E Controller" on page 250
"C300 Block" on page 252
"Application Control Environment (ACE) Block" on page 255

4.1.1 CPM Block (C200 Controller)

Description	Identifies the primary and secondary Control Processor Modules (CPM) and associated CEE to implement the control strategy built in the Control Builder application. This block's parameters characterize the redundant CPM as a whole. This block always runs at an execution period of 2 seconds. It is redundancy compliant.
Function	Supports C200 Controller Redundant Chassis Pair hardware configurations. Publishes parameters describing the status and configuration of the CPM. Processes the computation of statistical parameters and notification reporting. Serves as a faceplate for any parameters whose scope corresponds to that of the entire CPM. Secondary waits to take control if the"Primary" fails. The address of the Secondary chassis equals the address of the Primary chassis plus one.
Inputs	Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ALMENBSTATE	NUMFREEDESC
	BATTERYNOTOK	NUMREGDESC
	BLCKCOMMENT1	NUMUSEDBLKS
	BLCKCOMMENT2	NUMUSEDDESC
	BLCKCOMMENT3	PCMCOMMAND
	BLCKCOMMENT4	PCMSTATE
	CCLCNT	RAMSCRUBERRS
	CCLINFO	RAMSWEEPERR
	CCLLOADSTAT	RDNCAPABILITY
	CCLNAME	RDNCHASSISID
	CEECOMMAND	RDNCMPT
	CEESTATE	RDNDELAYAVG
	CPMCOMMAND[0..numChans-1]	RDNDELAYMAX
	CPMSTATE	RDNLOS
	CPUFREEAVG	RDNSYNCSTATE
	CPUFREEMIN	RDNXFERAVG
	CURTIME	RDNXFERMAX
	DAY	SCANASSOCDSP
	DAYLIGHTTIME	SCANCTRLLVL
	DESC	SCANGRPDTL
	DIRECTSYNC	SCANPNTDTL
	DRIVERNAME	SECMODNAME
	ENBMEMALMFL	SECNAMESTRING
	EUDESC	SECOND
	FREEMEM	SECTMPNAME
	FREEMEMINK	SLOTNUMBER
	GROUP.NUMPARAMS	STATSRESET
	HIST.NUMPARAMS	TASKSTACKHILM
	HOUR	TASKSTACKSIZE
	IMAGEVER	TASKSTACKUSED
	INALM	TIMEZONE
	INDRCTSYNC	TOTALMEM
	INDSYNCCMD	TOTALMEMINK
	LASTOPMNAME	TREND.NUMPARAMS
	MAXFREEBLKSZ	ULCNBMAC
	MAXFREEINK	USEDMEM
	MINUTE	USEDMEMINK
	MODISREDUN	USESIM
	MONTH	WEEKDAY
	MULREDUNSTAT	WEEKDAYFMT
	NAME	YEAR
	NETWORKTYPE	YEARFMT

	NTOTMEMDESC NUMEXTBLKS NUMFREEBLKS	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.1.2 C200E Controller

Description	Identifies the primary and secondary C200E controllers and associated CEE to implement the control strategy built in the Control Builder application. This block's parameters characterize the redundant C200E as a whole. This block always runs at an execution period of 2 seconds. It is redundancy compliant.
Function	Supports C200E Controller Redundant Chassis Pair hardware configurations. Publishes parameters describing the status and configuration of the C200E. Processes the computation of statistical parameters and notification reporting. Serves as a faceplate for any parameters whose scope corresponds to that of the entire C200E. Secondary waits to take control if the “Primary” fails. The address of the Secondary chassis equals the address of the Primary chassis plus one. Supports Controller Redundancy with switchover time < 500 mSec. Supports Exchange Block connections to Rockwell PLC devices also using the CIP protocol and PCCC protocol. Supports Experion Batch Manager (EBM) functionality for the Sequential Control Module (SCM), Recipe Control Module (RCM), Unit Control Module (UCM) and Phase block. Supports Whole Array Transfer. Supports 200 maximum instantiated CDB types. SIM-C200E does not have a separate module function block. It can be created in a similar way as the C200E and enabling the simulation environment in the Simulation tab.
Inputs	Integrated Control Protocol (ICP) communication
Outputs	Refer to the description.

Parameters	ALMENBSTATE	NUMFREEDESC
	BATTERYNOTOK	NUMREGDESC
	BLCKCOMMENT1	NUMUSEDBLKS
	BLCKCOMMENT2	NUMUSEDDESC
	BLCKCOMMENT3	PCMCOMMAND
	BLCKCOMMENT4	PCMSTATE
	BOOTIMAGEVER	PROCESS_ID
	CCLINFO	RAMSCRUBERRS
	CCLNAME	RAMSWEEPERR
	CEECOMMAND	RDNCAPABILITY
	CEESTATE	RDNCHASSISID
	CNISLOTNUM	RDNCMPT
	CNETCONNECT	RDNCTLABILITY
	CPMCOMMAND[0..numChans-1]	RDNDELAYAVG
	CPMSTATE	RDNDELAYMAX
	CURTIME	RDNINHIBITSYNC
	DAY	RDNHISTTIME
	DESC	RDNHISTSTATE
	DIRECTSYNC	RDNHISTREASON
	DRIVERNAME	RDNISTIMEMAX
	ENBMEMALMFL	RDNSOTIMEMAX
	EUDESC	RDNLOS
	FREEMEM	RDNROLESTATE
	FREEMEMINK	RDNOPMFRZTIME
	FWREVISION	RDNSYNCSTATE
	GROUP.NUMPARAMS	RDNSYNCPROG
	HIST.NUMPARAMS	RDNXFERAVG
	HOSTIPPRI	RDNXFERMAX
	HOSTNAMEPRI	SCANASSOCDSP
	HOUR	SCANCTRLLVL
	HWREVMAJ	SCANGRPDTL
	HWREVMIN	SCANPNTDTL
	IMAGEVER	SECMODNAME
	INALM	SECNAMESTRING
	INDRCTSYNC	SECOND
	INDSYNCCMD	SERIALNUM
	LASTOPMNAME	SECTMPNAME
	MAXFREEBLKSZ	SIMTARGET
	MAXFREEINK	SIM_C200MACID
	MINUTE	SIMCOMMAND
	MODISREDUN	SYNCTIMEBEG
	MODTYPE	SYNCTIMEEND

	MONTH	SLOTNUMBER
	MULREDUNSTAT	TASKSTACKHILM
	NAME	TASKSTACKSIZE
	NETWORKTYPE	TASKSTACKUSED
	NTOTMEMDESC	TOTALMEM
	NUMEXTBLKS	TOTALMEMINK
	NUMFREEBLKS	TREND.NUMPARAMS
		ULCNBMAC
		USEDMEM
		USEDMEMINK
		USESIM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.1.3 C300 Block

Description	Identifies the primary and secondary C300 Controllers and associated CEE to implement the control strategy built in the Control Builder application. This block's parameters characterize the redundant C300 as a whole. This block always runs at an execution period of 2 seconds. It is redundancy compliant.
Function	Supports C300 Controller Redundant Partner hardware configurations. Publishes parameters describing the status and configuration of the C300. Processes the computation of statistical parameters and notification reporting. Serves as a faceplate for any parameters whose scope corresponds to that of the entire C300. Secondary waits to take control if the “Primary” fails. The address of the Secondary controller equals the address of the Primary plus one. With R410, you can configure native peer-to-peer communication between the CEE points and non-CEE points such as PMD, Safety Manager points and Experion server points (SCADA, TPS, OPC Advanced, DSA). The peer-to-peer communication between CEE and the Experion server points are licensed using "Experion server Peer Responder" feature.
Inputs	Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ALMENBSTATE	MAXNODEID
	ALTSYNCCMD	MINUTE
	AUXDESC	MODIFIEDBY
	BACKUPRAMSCRUBERRS	MODISREDUN
	BACKUPRAMSWEEPERR	MONTH
	BADIPCSUM	NAME
	BADUDPCSUM	NONFTEMARTADDRCOUNT
	BATTERYNOTOK	NONFTEMARTAVGDEPTH
	BECMPRICMD	NONFTEMARTCOLLCOUNT
	BLCKCOMMENT1	NONFTEMARTMAXDEPTH
	BLCKCOMMENT2	NUMACEINCON
	BLCKCOMMENT3	NUMACEOUTCON
	BLCKCOMMENT4	NUMCPMINCON
	BOOTIMAGEVER	NUMCPMOUTCON
	CONTCUTOUT	NUMFIMINCON
	C300COMMAND	NUMFIMOUTCON
	C300MODTYPE	NUMFTENODES
	C300STATE	NUMIOLMINCON
	CMAXTEMP	NUMIOLMOUTCN
	CMINTEMP	NUMSIGS
	CPUFREEAVG	NUMSIOLMINCN
	CPUFREEMIN	NUMSIOLMOUCN
	CPUFREERESET	NUMC3INCON
	CREATEDBY	NUMC3OUTCON
	CTEMP	NUMEFIMINCON
	CTRLCONFIRM	NUMEFIMOUTCON
	CURDEVICEIDX	PRIMARYSIG
	CURTIME	QUALSTATE
	DATECREATED	RAMSCRUBERRS
	DAY	RAMSWEEPERR
	DAYLIGHTTIME	RDNAUTOSYNC
	DEBUGFLOAT	RDNCAPABILTY
	DEBUGINT	RDNCMPT
	DESC	RDNDELAYAVG
	DEVFLAG	RDNDELAYMAX
	DEVICEIDX	RDNHISTREASON
	DIRECTSYNC	RDNHISTSTATE
	DSBLSYNCCMD	RDNLOS
	ENBLSTBYCMD	RDNROLESTATE
	ENBLSYNCCMD	RDNSYNCPROG
	ESIG	RDNSYNCSTATE
	FTEMARTADDRCOUNT	RDNXFERAVG

	FTEMARTAVGDEPTH	RDNXFERMAX
	FTEMARTCOLLCOUNT	REASONSET
	FTEMARTMAXDEPTH	SCANAREA
	FWREVISION	SCANASSOCDSP
	GOBACKREL	SCANCTRLLVL
	GROUP	SCANEUHI
	GROUP	SCANEULO
	GROUP.NUMPARAMS	SCANGRPDTL
	GROUP.PARAM	SCANPNTDTL
	GROUP.POSITION	SCRIPTOR
	HIST	SECMODNAME
	HIST.EXTD	SECNAME
	HIST.FAST	SECNAMESTRG
	HIST.GATEPARAM	SECOND
	HIST.GATESTRING	SECONDARYSIG
	HIST.GATEVALUE	SECTMPNAME
	HIST.NUMPARAMS	SERIALNUM
	HIST.PARAM	SIMDEVICEIDX
	HIST.STD	SIMENABLE
	HOUR	SIMIPADDRESS
	HWREVMJ	STATSRESET
	HWREVMN	SWITCHCMD
	ICMPINDESTUNREACHS	SYNCTIMEBEG
	ICMPINECHOREPS	SYNCTIMEEND
	ICMPINECHOS	TASKSTACKHILM
	ICMPINERRORS	TASKSTACKSIZE
	ICMPINMSGS	TASKSTACKUSED
	ICMPOUTDESTUNREACHS	TCPACTIVEOPENS
	ICMPOUTECHOREPS	TCPATTEMPTFAILS
	ICMPOUTECHOS	TCPCURRESTAB
	ICMPOUTERRORS	TCPESTABRESETS
	ICMPOUTMSGS	TCPINERRS
	ICONSTATE	TCPINSEGS
	IMAGEVER	TCPOUTRESETS
	INALM	TCPOUTSEGS
	INDRCTSYNC	TCPPASSIVEOPENS
	INDSYNCCMD	TCPRETRANSSEGS
	INITCOMPLETE	TIMELASTSKEW
	INTERLANFAILED	TIMELASTSYNC
	IPADDRESS	TIMESOURCE
	IPFRAGCREATES	TIMESYNCSTAT
	IPFRAGFAILS	TIMEZONE

	IPFRAGOKS IPINADDRERRORS IPINDELIVERS IPINDISCARDS IPINHDRERRORS IPINRECEIVES IPINUNKNOWNPORTS IPOUTDISCARDS IPOUTNOROUTES IPOUTREQUESTS IPREASSEMFAILS IPREASSEMOKS IPREASSEMREQS IPROUTINGDISCARDS JOURNALONLY LANAFAILED LANBFAILED LASTOPMNAME MACADDRA MACADDRB MACADDRR MAXFTENODES	TMBTCPAVGXMITMSGPS TMBTCPMAXXMITMSGPS TMBTCPAVGRCVMSGPS TMBTCPMAXRCVMSGPS TNUMPMDINCON TNUMPMDOUTCON TNUMQCSINCON TNUMQCSOUTCON TNUMSCADAINCON TNUMSMINCON TREND TREND.NUMBER TREND.NUMPARAMS TREND.PARAM TREND.POSITION UDPINDGRAMS UDPINERRORS UDPLISTENERS UDPNOPORTS UDPOUTDGRAMS VERSION VERSIONDATE VERSIONNUM WDTMAXREFRESH WDTMINREFRESH WEEKDAY WEEKDAYFMT YEAR YEARFMT XOVERFAILED
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.1.4 Application Control Environment (ACE) Block

Refer to the *Application Control Environment (ACE) User's Guide* for more information about this block.

4.2 Control Execution Environment Block

Related topics

"CEEC200 Block" on page 256
"CEEC200E Block" on page 258
"CEEC300 Block" on page 261
"C300 - 20mS CEE" on page 265
"CEEACE Block" on page 268

4.2.1 CEEC200 Block

Description	Provides control functionality for associated Control Processor Module block. This block's parameters characterize the CEE within the CPM. In the future, multiple CEEs may be assigned to a single CPM. This block always runs at an execution period of 2 seconds. There are two versions of the CEE available, the standard version CEE-50ms, and the fast version CEE-5ms.
Function	Publishes parameters describing the status and configuration of the CEE. Processes the computation of statistical parameters and notification reporting. Runs on the CPM hardware platform. In the future, CEE will run on other platforms as well. Serves as a faceplate for any parameters whose scope corresponds to that of the CEE rather than the CPM as a whole. Supports configurable subscription rate for peer-to-peer communications. Supports peer-to-peer communications among CEEs assigned to CPMs located in the same management domain. Supports configurable subscription rate and store response time for specific peer environment. Sequential Control Module function blocks are supported. Special care should be taken in configuring the SCMs in 5 msec CEE.
Inputs	Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ALMENBSTATE	IPEERCONNERRCODE
	BASEPERIOD	IPEERCONNERRINFO
	BLCKCOMMENT1	IPEERCONNSTS
	BLCKCOMMENT2	IPEERNAME
	BLCKCOMMENT3	IPEERPATH
	BLCKCOMMENT4	LSCYCLEOVRN[0…40]
	BLKTYPCOUNT	MAXBLKTYPES
	BLKTYPDESC	NUMACEOUTCON
	BLKTYPHELPTXT	NUMBLKTYPES
	BLKTYPLIB	NUMCCLRQU
	BLKTYPSIZE	NUMCPMINCON
	CDISPAVGPPS	NUMCPMOUTCON
	CDISPAVGPPSCONN	NUMEXCRQUAVG
	CDISPAVGSPSCONN	NUMEXCRQUMAX
	CDISPMAXPPS	NUMEXCRSPAVG
	CDISPMAXPPSCONN	NUMEXCRSPMAX
	CDISPMAXSPSCONN	NUMFIMINCON
	CEECOMMAND	NUMFIMOUTCON
	CEESTATE	NUMIOLMINCON
	CPEERAVGPPS	NUMIOLMOUTCN
	CPEERAVGPPSCONN	NUMNTFRQUAVG
	CPEERAVGSPSCONN	NUMNTFRQUMAX
	CPEERMAXPPS	NUMPARRSPAVG
	CPEERMAXPPSCONN	NUMPARRSPMAX
	CPEERMAXSPSCONN	NUMPEERENV
	CPUCYCLEAVG[0..39]	NUMSCEINCON
	CPUCYCLEMAX[0..39]	NUMSCEOUTCON
	CPUFREEAVG	NUMSIOLMINCN
	CPUFREEMIN	NUMSIOLMOUCN
	CRCYCLEOVRN[0..40]	PEERENV
	DESC	PEERGETAVG[1..30]
	EUDESC	PEERSTRAVG[1..30]
	EXTGETRQUAVG	PEERSTRRESP
	EXTGETRQUMAX	PEERSUBSCPER
	EXTSTRRQUAVG	RDISPDEGIMRCONN[1… RDISPCONNMAX]
	EXTSTRRQUMAX	RPEERNAME[1…RPEERCONNMAX]
	GROUP.NUMPARAMS	SCANASSOCDSP
	HIST.NUMPARAMS	SCANCTRLLVL
	INALM	SCANGRPDTL
	NAME	SCANPNTDTL
	NUMACCRQUAVG	STATSRESET
	NUMACCRQUMAX	STRRESP

	NUMACEINCON	SUBSCPERIOD TREND.NUMPARAMS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.2.2 CEEC200E Block

Description	Provides control functionality for associated C200E block. This block's parameters characterize the CEE within the C200E controller. The base execution cycle for CEEC200E block 50 mSec. The execution periods for function blocks loaded to the CEE controller may be set to any of the same values supported by these execution cycles.
Function	Publishes parameters describing the status and configuration of the CEE. Processes the computation of statistical parameters and notification reporting. Runs on the C200E hardware platform. Serves as a faceplate for any parameters whose scope corresponds to that of the CEE rather than the C200E as a whole. Supports configurable subscription rate for peer-to-peer communication. Supports peer-to-peer communication among CEEs assigned to controllers located in the same management domain. Supports peer connections to other controllers (C200, C200E, C300, and ACE Nodes) through FTEB. Supports configurable subscription rate and store response time for specific peer environment. Sequential Control Module function blocks are supported. With R410, C200E-CEE supports native peer-to-peer communication with non-CEE controllers such as Safety Manager and PMD points, and Experion server points such as SCADA and TPS points.The peer-to-peer communication between CEE and the Experion server points are licensed using "Experion server Peer Responder" feature.
Inputs	Integrated Control Protocol (ICP) communications
Outputs	Refer to the description.

Parameters	ALMENBSTATE	IPEERCONNERRCODE
	BASEPERIOD	IPEERCONNERRINFO
	BLCKCOMMENT1	IPEERCONNSTS
	BLCKCOMMENT2	IPEERNAME
	BLCKCOMMENT3	IPEERPATH
	BLCKCOMMENT4	LSCYCLEOVRN[0…40]
	BLKTYPCOUNT	MAXBLKTYPES
	BLKTYPDESC	NUMACEOUTCON
	BLKTYPHELPTXT	NUMBEVENTSAV
	BLKTYPLIB	NUMBEVENTSMAX
	BLKTYPSIZE	NUMBLKTYPES
	CCLCNT	NUMCCLRQU
	CCLLOADSTAT	NUMCPMINCON
	CDISPAVGPPS	NUMCPMOUTCON
	CDISPAVGPPSCONN	NUMEXCRQUAVG
	CDISPAVGSPSCONN	NUMEXCRQUMAX
	CDISPMAXPPS	NUMEXCRSPAVG
	CDISPMAXPPSCONN	NUMEXCRSPMAX
	CDISPMAXSPSCONN	NUMFIMINCON
	CEECOMMAND	NUMFIMOUTCON
	CEESTATE	NUMIOLMINCON
	CPEERAVGPPS	NUMIOLMOUTCN
	CPEERAVGPPSCONN	NUMNTFRQUAVG
	CPEERAVGSPSCONN	NUMNTFRQUMAX
	CPEERMAXPPS	NUMPARRSPAVG
	CPEERMAXPPSCONN	NUMPARRSPMAX
	CPEERMAXSPSCONN	NUMPEERENV
	CPUCYCLEAVG[0..39]	NUMPMDINCON
	CPUCYCLEMAX[0..39]	NUMPMDOUTCON
	CPUFREEAVG	NUMQCSINCON
	CPUFREEMIN	NUMQCSOUTCON
	CRCYCLEOVRN[0..40]	NUMSCADAINCON
	DESC	NUMSCEINCON
	DAYLIGHTTIME	NUMSCEOUTCON
	EUDESC	NUMSIOLMINCN
	EXTGETRQUAVG	NUMSIOLMOUCN
	EXTGETRQUMAX	NUMSMINCON
	EXTSTRRQUAVG	PEERENV
	EXTSTRRQUMAX	PEERGETAVG[1..30]
	GROUP.NUMPARAMS	PEERSTRAVG[1..30]
	HIST.NUMPARAMS	
	INALM	

	NAME NUMACCRQUAVG NUMACCRQUMAX NUMACEINCON	PEERSTRRESP PEERSUBSCPER RDISPDEGIMRCONN[1…RDISPCONNMAX] RPEERNAME[1…RPEERCONNMAX] SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL STATSRESET STRRESP SUBSCPERIOD TREND.NUMPARAMS TIMEZONE TNUMACEINCON TNUMC3INCON TNUMCPMINCON TNUMSCFIMINCON TNUMSCEINCON TNUMLIOMINCON TNUMACEOUTCON TNUMC3OUTCON TNUMCPMOUTCON TNUMSCFIMOUTCON TNUMSCEOUTCON TNUMLIOMOUTCON TNUMNTFRQUAVG TNUMNTFRQUMAX TINUMINMSGAVGPS TINUMINMSGMAXPS TINUMOUTMSGAVGPS TINUMOUTMSGMAXPS TRNUMINMSGAVGPS TRNUMINMSGMAXPS TRNUMOUTMSGAVGPS TRNUMOUTMSGMAXPS YEARFMT WEEKDAYFMT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.2.3 CEEC300 Block

Description	Provides control functionality for associated C300 block. This block's parameters characterize the CEE within the C300 controller. In the future, multiple CEEs may be assigned to a single C300. Execution periods for this block may be 50 mSec, 100 mSec, 200 mSec, 500 mSec, 1 sec, or 2 sec. The execution cycle for CEEC300 block is 50ms.
Function	Publishes parameters describing the status and configuration of the CEE. Processes the computation of statistical parameters and notification reporting. Runs on the C300 hardware platform. In the future, CEE will run on other platforms as well. Serves as a faceplate for any parameters whose scope corresponds to that of the CEE rather than the C300 as a whole. Supports configurable subscription rate for peer-to-peer communications. Supports peer-to-peer communications among CEEs assigned to controllers located in the same management domain. Supports configurable subscription rate and store response time for specific peer environment. Sequential Control Module function blocks are supported. With R410, C300-CEE supports native peer-to-peer communication with non-CEE controllers such as Safety Manager and PMD points, and Experion server points such as SCADA and TPS points.The peer-to-peer communication between CEE and the Experion server points are licensed using "Experion server Peer Responder" feature.
Inputs	Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ALMENBSTATE	NUMBLKTYPES
	AUXDESC	NUMCCLRQU
	BASEPERIOD	NUMCYCLE
	BLCKCOMMENT1	NUMEXCRQUAVG
	BLCKCOMMENT2	NUMEXCRQUMAX
	BLCKCOMMENT3	NUMEXCRSPAVG
	BLCKCOMMENT4	NUMEXCRSPMAX
	BLKTYPCOUNT	NUMEXTBLKS
	BLKTYPDESC	NUMFREEBLKS
	BLKTYPHELPTXT	NUMFREEDESC
	BLKTYPLIB	NUMMBTCPXMITMSGAVG
	BLKTYPSIZE	NUMMBTCPXMITMSGMAX
	CDISPAVGLPS	NUMMBTCPXMITBYTEAVG
	CDISPAVGLPSCONN	NUMMBTCPXMITBYTEMAX
	CDISPAVGPPS	NUMMBTCPRCVMSGAVG
	CDISPAVGPPSCONN	NUMMBTCPRCVMSGMAX
	CDISPAVGSPSCONN	NUMMBTCPRCVBYTEAVG
	CDISPMAXLPS	NUMMBTCPRCVBYTEMAX
	CDISPMAXLPSCONN	MBTCPINVALIDRCVMSGCOUNT
	CDISPMAXPPS	MBTCPFREEBUFCOUNT
	CDISPMAXPPSCONN	NUMNTFRQUAVG
	CDISPMAXSPSCONN	NUMNTFRQUMAX
	CEECOMMAND	NUMPARRSPAVG
	CEESTATE	NUMPARRSPMAX
	CONTCUTOUT	NUMPEERENV
	CPEERAVGLPS	NUMREGDESC
	CPEERAVGLPSCONN	NUMSIGS
	CPEERAVGPPS	NUMUSEDBLKS
	CPEERAVGPPSCONN	NUMUSEDDESC
	CPEERAVGSPSCONN	PEERENV
	CPEERMAXLPS	PEERGETAVG
	CPEERMAXLPSCONN	PEERSTRAVG
	CPEERMAXPPS	PEERSTRRESP
	CPEERMAXPPSCONN	PEERSUBSCPER
	CPEERMAXSPSCONN	PRIMARYSIG
	CPUCYCLEAVG[0..39]	QUALSTATE
	CPUCYCLEMAX	RACYCIMRAVAIL
	CRCYCLEOVRN	RACYCIMRMAX
	CREATEDBY	RACYCIMRUSED
	CTRLCONFIRM	RACYCLGREQAVAIL
	DATECREATED	RACYCLGREQMAX
	DESC	RACYCLRQUSED

	ENBMEMALMFL	RACYCSMREQAVAIL
	ESIG	RACYCSMREQMAX
	EXTGETRQUAVG	RACYCSRQUSED
	EXTGETRQUMAX	RCYCIMRAVAIL
	EXTSTRRQUAVG	RCYCIMRMAX
	EXTSTRRQUMAX	RCYCIMRUSED
	FRC	RCYCLGREQAVAIL
	FREEMEM	RCYCLGREQMAX
	FREEMEMINK	RCYCLRQUSED
	GROUP	RCYCSMREQAVAIL
	GROUP.NUMBER	RCYCSMREQMAX
	GROUP.NUMPARAMS	RCYCSRQUSED
	GROUP.PARAM	RDEGRADALM
	GROUP.POSITION	RDISPAVGLPS
	HIST	RDISPAVGLPSCONN
	HIST.EXTD	RDISPAVGPPS
	HIST.FAST	RDISPAVGPPSCONN
	HIST.GATEPARAM	RDISPCONN
	HIST.GATESTRING	RDISPCONNMAX
	HIST.GATEVALUE	RDISPDEGIMRCONN
	HIST.NUMPARAMS	RDISPMAXLPS
	HIST.PARAM	RDISPMAXLPSCONN
	HIST.STD	RDISPMAXPPS
	IACYCIMRAVAIL	RDISPMAXPPSCONN
	IACYCIMRMAX	REASONSET
	IACYCIMRUSED	RPEERAVGLPS
	IACYCIMRUSEDCONN	RPEERAVGLPSCONN
	IACYCLGREQAVAIL	RPEERAVGPPS
	IACYCLGREQMAX	RPEERAVGPPSCONN
	IACYCLGREQUSED	RPEERCONN
	IACYCLGREQUSEDCONN	RPEERCONNMAX
	IACYCSMREQAVAIL	RPEERDEGIMRCONN
	IACYCSMREQMAX	RPEERMAXLPS
	IACYCSMREQUSED	RPEERMAXLPSCONN
	IACYCSMREQUSEDCONN	RPEERMAXPPS
	ICYCIMRAVAIL	RPEERMAXPPSCONN
	ICYCIMRMAX	RPEERNAME
	ICYCIMRUSED	RPEERSUBPER
	ICYCIMRUSEDCONN	RTOTACYCIMR
	ICYCLGREQAVAIL	RTOTACYCLGREQ
	ICYCLGREQMAX	RTOTACYCSMREQ
	ICYCLGREQUSED	RTOTCYCIMR

	ICYCLGREQUSEDCONN	RTOTCYCLGREQ
	ICYCSMREQAVAIL	RTOTCYCSMREQ
	ICYCSMREQMAX	SCANAREA
	ICYCSMREQUSED	SCANASSOCDSP
	ICYCSMREQUSEDCONN	SCANCTRLLVL
	INALM	SCANEUHI
	IPEERAVGLPS	SCANEULO
	IPEERAVGPPS	SCANGRPDTL
	IPEERCONN	SCANPNTDTL
	IPEERCONNMAX	SCRIPTOR
	IPEERMAXLPS	SECONDARYSIG
	IPEERMAXPPS	SECSIGSECLVL
	IPEERNAME	SIMCOMMAND
	JOURNALONLY	SIMSTATE
	LSCYCLEOVRN[0…40]	SCANAREA
	MAXBLKTYPES	SCANASSOCDSP
	MAXFREEBLKSZ	SCANCTRLLVL
	MAXFREEINK	SCANEUHI
	MODIFIEDBY	SCANEULO
	NEXTPHASE	SCANGRPDTL
	NOTIFINHIBIT	SCANPNTDTL
	NTOTMEMDESC	STATSRESET
	NUMACCRQUAVG	TASKSTACKHILM
	NUMACCRQUMAX	TASKSTACKSIZE
	NUMASSIGNFBS	TASKSTACKUSED
		TNUMQCSINCON
		TNUMQCSOUTCON
		TNUMSCADAINCON
		TNUMSMINCON
		TREND.PARAM
		TREND.POSITION
		TOTALMEM
		TOTALMEMINK
		USEDMEM
		USEDMEMINK
		VERSION
		VERSIONDATE
		VERSIONNUM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.2.4 C300 - 20mS CEE

Attention

- A separate license is required to use the C300 - 20mS CEE controller. For details on the license, refer to Control Hardware Planning Guide.
- C300 - 20mS CEE controller does not support native peer-to-peer communication with non-CEE controllers such as Safety Manager and PMD points, and Experion server points such as SCADA and TPS points.

Description	Provides control functionality for associated C300 block. This block's parameters characterize the CEE within the C300 - 20mS CEE controller. The base execution period for C300 - 20mS CEE block is 20ms. Control Module assigned to this block can be configured with the execution period of 20mSec, 40 mSec, 80 mSec, 200 mSec, 400 mSec or 800 mSec.
Function	Publishes parameters describing the status and configuration of the CEE. Processes the computation of statistical parameters and notification reporting. Runs on the C300 hardware platform. In the future, CEE will run on other platforms as well. Serves as a faceplate for any parameters whose scope corresponds to that of the CEE rather than the C300 as a whole. Supports configurable subscription rate for peer-to-peer communications. Supports peer-to-peer communications among CEEs assigned to controllers located in the same management domain. Supports configurable subscription rate and store response time for specific peer environment. Supports Speed Protection Module (SPM) and Servo Valve Positioner Module (SVPM).
Inputs	
Outputs	

Parameters	ALMENBSTATE	NUMEXCRSPAVG
	AUXDESC	NUMEXCRSPMAX
	BASEPERIOD	NUMEXTBLKS
	BLCKCOMMENT1	NUMFREEBLKS
	BLCKCOMMENT2	NUMFREEDESC
	BLCKCOMMENT3	NUMMBTCPXMITMSGAVG
	BLCKCOMMENT4	NUMMBTCPXMITMSGMAX
	BLKTYPCOUNT	NUMMBTCPXMITBYTEAVG
	BLKTYPDESC	NUMMBTCPXMITBYTEMAX
	BLKTYPHELPTXT	NUMMBTCPRCVMSGAVG
	BLKTYPLIB	NUMMBTCPRCVMSGMAX
	BLKTYPSIZE	NUMMBTCPRCVBYTEAVG
	CDISPAVGLPS	NUMMBTCPRCVBYTEMAX
	CDISPAVGLPSCONN	MBTCPINVALIDRCVMSGCOUNT
	CDISPAVGPPS	MBTCPFREEBUFCOUNT
	CDISPAVGPPSCONN	NUMNTFRQUAVG
	CDISPAVGSPSCONN	NUMNTFRQUMAX
	CDISPMAXLPS	NUMPARRSPAVG
	CDISPMAXLPSCONN	NUMPARRSPMAX
	CDISPMAXPPS	NUMPEERENV
	CDISPMAXPPSCONN	NUMREGDESC
	CDISPMAXSPSCONN	NUMSIGS
	CEECOMMAND	NUMUSEDBLKS
	CEESTATE	NUMUSEDDESC
	CONTCUTOUT	PEERENV
	CPEERAVGLPS	PEERGETAVG
	CPEERAVGLPSCONN	PEERSTRAVG
	CPEERAVGPPS	PEERSTRRESP
	CPEERAVGPPSCONN	PEERSUBSCPER
	CPEERAVGSPSCONN	PRIMARYSIG
	CPEERMAXLPS	QUALSTATE
	CPEERMAXLPSCONN	RACYCIMRAVAIL
	CPEERMAXPPS	RACYCIMRMAX
	CPEERMAXPPSCONN	RACYCIMRUSED
	CPEERMAXSPSCONN	RACYCLGREQAVAIL
	CPUCYCLEAVG[0..39]	RACYCLGREQMAX
	CPUCYCLEMAX	RACYCLRQUSED
	CRCYCLEOVRN	RACYCSMREQAVAIL
	CREATEDBY	RACYCSMREQMAX
	CTRLCONFIRM	RACYCSRQUSED
	DATECREATED	RCYCIMRAVAIL
	DESC	RCYCIMRMAX

	ENBMEMALMFL	RCYCIMRUSED
	ESIG	RCYCLGREQAVAIL
	EXTGETRQUAVG	RCYCLGREQMAX
	EXTGETRQUMAX	RCYCLRQUSED
	EXTSTRRQUAVG	RCYCSMREQAVAIL
	EXTSTRRQUMAX	RCYCSMREQMAX
	FRC	RCYCSRQUSED
	FREEMEM	RDEGRADALM
	FREEMEMINK	RDISPAVGLPS
	GROUP	RDISPAVGLPSCONN
	GROUP.NUMBER	RDISPAVGPPS
	GROUP.NUMPARAMS	RDISPAVGPPSCONN
	GROUP.PARAM	RDISPCONN
	GROUP.POSITION	RDISPCONNMAX
	HIST	RDISPDEGIMRCONN
	HIST.EXTD	RDISPMAXLPS
	HIST.FAST	RDISPMAXLPSCONN
	HIST.GATEPARAM	RDISPMAXPPS
	HIST.GATESTRING	RDISPMAXPPSCONN
	HIST.GATEVALUE	REASONSET
	HIST.NUMPARAMS	RPEERAVGLPS
	HIST.PARAM	RPEERAVGLPSCONN
	HIST.STD	RPEERAVGPPS
	IACYCIMRAVAIL	RPEERAVGPPSCONN
	IACYCIMRMAX	RPEERCONN
	IACYCIMRUSED	RPEERCONNMAX
	IACYCIMRUSEDCONN	RPEERDEGIMRCONN
	IACYCLGREQAVAIL	RPEERMAXLPS
	IACYCLGREQMAX	RPEERMAXLPSCONN
	IACYCLGREQUSED	RPEERMAXPPS
	IACYCLGREQUSEDCONN	RPEERMAXPPSCONN
	IACYCSMREQAVAIL	RPEERNAME
	IACYCSMREQMAX	RPEERSUBPER
	IACYCSMREQUSED	RTOTACYCIMR
	IACYCSMREQUSEDCONN	RTOTACYCLGREQ
	ICYCIMRAVAIL	RTOTACYCSMREQ
	ICYCIMRMAX	RTOTCYCIMR
	ICYCIMRUSED	RTOTCYCLGREQ
	ICYCIMRUSEDCONN	RTOTCYCSMREQ
	ICYCLGREQAVAIL	SCANAREA
	ICYCLGREQMAX	SCANASSOCDSP
	ICYCLGREQUSED	SCANCTRLLVL

	ICYCLGREQUSEDCONN	SCANEUHI
	ICYCSMREQAVAIL	SCANEULO
	ICYCSMREQMAX	SCANGRPDTL
	ICYCSMREQUSED	SCANPNTDTL
	ICYCSMREQUSEDCONN	SCRIPTOR
	INALM	SECONDARYSIG
	IPEERAVGLPS	SECSIGSECLVL
	IPEERAVGPPS	SIMCOMMAND
	IPEERCONN	SIMSTATE
	IPEERCONNMAX	SCANAREA
	IPEERMAXLPS	SCANASSOCDSP
	IPEERMAXPPS	SCANCTRLLVL
	IPEERNAME	SCANEUHI
	JOURNALONLY	SCANEULO
	LSCYCLEOVRN[0…40]	SCANGRPDTL
	MAXBLKTYPES	SCANPNTDTL
	MAXFREEBLKSZ	STATSRESET
	MAXFREEINK	TASKSTACKHILM
	MODIFIEDBY	TASKSTACKSIZE
	NEXTPHASE	TASKSTACKUSED
	NOTIFINHIBIT	TREND.PARAM
	NTOTMEMDESC	TREND.POSITION
	NUMACCRQUAVG	TOTALMEM
	NUMACCRQUMAX	TOTALMEMINK
	NUMASSIGNFBS	USEDMEM
	NUMBLKTYPES	USEDMEMINK
	NUMCCLRQU	VERSION
	NUMCYCLE	VERSIONDATE
	NUMEXCRQUAVG	VERSIONNUM
	NUMEXCRQUMAX	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.2.5 CEEACE Block

Refer to the *Application Control Environment (ACE) User's Guide* for more information about this block.

4.3 Input/Output Link Interface Module Block

Related topics

"IOLIM block" on page 269

4.3.1 IOLIM block

Description	Identifies the primary and secondary I/O Link Interface Modules (IOLIM) and associated I/O Link that serve as the communication bridge between the control system and the Process Manager (PM) Input/Output Processors (IOPs).	
Function	Supports C200 Controller Redundant Chassis Pair hardware configurations. Supports both the publish/subscribe and the client/server communication methods to access process data and maintenance information from the IOP devices Publishes parameters describing the status and configuration of the IOPs. Reports run-time diagnostics and statistical information for the IOLIMs. Serves as a faceplate for any parameters whose scope corresponds to that of the entire PM I/O. Secondary waits to take control if the "Primary" fails.	
Inputs	Communications bridge between Integrated Control Processor (ICP)/ControlNet and I/O Link	
Outputs	See above.	
Parameters	BUFXMAX[0..5] BUFXTOTAL[0..5] BUFXUSED[0..5] COMPNVSCMD CPUFREEAVG CPUFREEMIN CURTIME DISPRATEAVG DISPRATEMAX DRIVERNAME ENCMDS IOLIMSTATE MAXIMR MODISREDUN NAME NETWORKTYPE NOTRATEAVG NOTRATEMAX NUMCCLRQU NUMCPMINCON NUMCPMOUTCON NUMIMR	NVSUSED PEERRATEAVG PEERRATEMAX PURGEDBCMD RDNCAPABILITY RDNCHASSISID RDNCMPT RDNDELAYAVG RDNDELAYMAX RDNLOS RDNSYNCSTATE RDNXFERAVG RDNXFERMAX SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL SHUTDOWNCMD SLOTNUMBER STATRESET ULCNBMAC
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.4 Input/Output Link Block

Related topics

"IOLINK block" on page 270
"IOLINK Block (C300 Controller)" on page 272
"IOLINK Block (C300 - 20mS CEE)" on page 275

4.4.1 IOLINK block

Description	Provides interface functionality for associated I/O Link Interface Module (IOLIM) block.
Function	Provides supervisory scanning, diagnostic, and performance throughput information for monitoring I/O Link network status.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ACTPRIM[1..40]	PHYSIOPSTS1
	CFIFORATEA	PHYSMODTYPE1
	CFIFORATEM	PRICHNERRA
	COMMAND	PRICHNERRB
	CRCYCLEOVRN[0..40]	PRICHNSILA
	DBVALID	PRICHNSILB
	DESC	PRICOMMERR
	DUPIOLADDR	PRIIFCARD
	FMWREV[1..8]	PRIIFCHNERRA
	HDWREV[1..8]	PRIIFCHNERRB
	IOLCHNFAILA	PRIIFCHNSILA
	IOLCHNFAILB	PRIIFCHNSILB
	IOLCHNSTSA	PRIIFCOMMERR
	IOLCHNSTSB	PRIIFFILE
	IOLDAUGHSF	PRIIFRCVCHN
	IOLFREE	PRIIFSTS
	IOLMAXERR	PRIRCVCHN
	IOMCMD[1..40]	SCANASSOCDSP
	IOMSTS[1..40]	SCANCTRLLVL
	IOMTYPE[1..40]	SCANGRPDTL
	IOPORCUR[1..40]	SCANPNTDTL
	IOPORPREV[1..40]	SCANRATE [1…40]
	IOSSTALLTIME	SECCHNERRA
	IOSTKNDROP	SECCHNERRB
	LASTIOLCMD	SECCOMMERR
	LINKNUM	SECIFCHNERRA
	LSCYCLEOVRN[0..40]	SECIFCHNERRB
	NAME	SECIFCHNSILA
	NOTACTSUPV	SECIFCHNSILB
	NUMCACHE[1..40]	SECIFCOMMERR
	OVERRUNSCUR	SECIFRCVCHN
	OVERRUNSPREV	SECRCVCHN
	PARTMISMATCH	STATE
	PARTNOTVIS	STATRESET
	PDFIFORATEA	SYNCHSTS
	PDFIFORATEM	TOTCHNERRA
	PERSWAPENB	TOTCHNERRB
	PERSWAPTHRES	TOTCHNSILA
	PHYCHNERRA1	TOTCHNSILB
	PHYCHNERRB1	WITHBIAS[1..40]
	PHYCHNSILA1	WITHBIASENM[1..40]
	PHYCHNSILB1	WRFIFORATEA

	PHYLHFSTA PHYRCVCHN1 PHYSDSA1	WRFIFORATEM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.4.2 IOLINK Block (C300 Controller)

Description	Provides interface functionality for associated I/O Link Interface.
Function	Provides supervisory scanning, diagnostic, and performance throughput information for monitoring I/O Link network status.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ACTPRIM	PHYCHNSILB1
	AUXDESC	PHYCOMMERR1
	AVGWRTIME	PHYRCVCHN1
	BLCKCOMMENT1	PHYSDSA1
	BLCKCOMMENT2	PHYSIOPSTS1
	BLCKCOMMENT3	PHYSMODTYPE1
	BLCKCOMMENT4	PRIBLOCKNAME
	CFIFORATEA	PRICHNERRA
	CFIFORATEM	PRICHNERRB
	COMMAND	PRICHNSILA
	CREATEDBY	PRICHNSILB
	CTRLCONFIRM	PRICOMMERR
	DATECREATED	PRIIFCARD
	DBVALID	PRIIFCHNERRA
	DESC	PRIIFCHNERRB
	DUPIOLADDR	PRIIFCHNSILA
	ESIG	PRIIFCHNSILB
	FMWREV1	PRIIFCOMMERR
	FREESMSOVRRUN	PRIIFFILE
	GROUP	PRIIFRCVCHN
	GROUP.NUMBER	PRIIFSTS
	GROUP.NUMPARAMS	PRIMARYSIG
	GROUP.PARAM	PRIRCVCHN
	GROUP.POSITION	QUALSTATE
	HDWREV1	REASONSET
	HIPRIRATE	SCANAREA
	HIST	SCANASSOCDSP
	HIST.EXTD	SCANCTRLLVL
	HIST.FAST	SCANEUHI
	HIST.GATEPARAM	SCANEULO
	HIST.GATESTRING	SCANGRPDTL
	HIST.GATEVALUE	SCANPNTDTL
	HIST.NUMPARAMS	SCANRATE
	HIST.PARAM	SCRIPTOR
	HIST.STD	SECBLOCKNAME
	IOLCHNFAILA	SECCHNERRA
	IOLCHNFAILB	SECCHNERRB
	IOLCHNHISTA	SECCHNSILA
	IOLCHNHISTB	SECCHNSILB
	IOLCHNSTSA	SECCOMMERR
	IOLCHNSTSB	SECIFCHNERRA
	IOLDAUGHSF	SECIFCHNERRB

	IOLFREE	SECIFCHNSILA
	IOLMAXERR	SECIFCHNSILB
	IOLOVRRUN	SECIFCOMMERR
	IOLSOFTFAIL	SECIFRCVCHN
	IOMCOMMAND	SECONDARYSIG
	IOMSTS	SECRCVCHN
	IOMTYPE	SECSIGSECLVL
	IOPBLOCKNAME	STATE
	IOPORCUR	STATRESET
	IOPORPREV	STTEXT
	IOSSTALLTIME	STTEXTA
	IOSTKNDROP	STTEXTB
	LASTIOLCMD	SUPVFIFORATEA
	LINKNUM	SUPVFIFORATEM
	LOPRIRATE	SWTCHACT
	MAXNUMPRIIOP	SYNCHSTS
	MEDPRIRATE	TOTCHNERRA
	MODIFIEDBY	TOTCHNERRB
	MONREADPARAM	TOTCHNSILA
	MONSTATES	TOTCHNSILB
	NOTACTSUPV	TREEBITMAP
	NUMCACHE	TREND
	NUMPRIIOP	TREND.NUMBER
	NUMSIGS	TREND.NUMPARAMS
	OVERRUNSCUR	TREND.PARAM
	OVERRUNSPREV	TREND.POSITION
	OVRRUNCURHR	VERSION
	OVRRUNPREVHR	VERSIONDATE
	OVRSIGNA	VERSIONNUM
	PARTMISMATCH	WITHBIAS
	PARTNOTVIS	WITHBIASENM
	PDFIFORATEA	WRFIFORATEA
	PDFIFORATEM	WRFIFORATEM
	PERSWAPENB	
	PERSWAPTHRES	
	PHYCHNERRA1	
	PHYCHNERRB1	
	PHYCHNSILA1	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.4.3 IOLINK Block (C300 - 20mS CEE)

Description	Provides interface functionality for associated I/O Link Interface.
Function	Provides supervisory scanning, diagnostic, and performance throughput information for monitoring I/O Link network status. Reports a “Pre-Fetch Overrun” diagnostic alarm if rate of increase of pre-fetch overruns exceeds 1 in 200 seconds. This indicates that the end-to-end response time is larger than expected. Once this alarm is reported, it will RTN after a period of 200 seconds provided no new overrun occurs during this interval.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ACTPRIM	PFOVERRUNSCUR
	AUXDESC	PFOVERRUNSPREV
	AVGWRTIME	PHYCHNSILB1
	BASEPERIOD	PHYCOMMERR1
	BLCKCOMMENT1	PHYRCVCHN1
	BLCKCOMMENT2	PHYSDSA1
	BLCKCOMMENT3	PHYSIOPSTS1
	BLCKCOMMENT4	PHYSMODTYPE1
	CFIFORATEA	PRIBLOCKNAME
	CFIFORATEM	PRICHNERRA
	COMMAND	PRICHNERRB
	CREATEDBY	PRICHNSILA
	CTRLCONFIRM	PRICHNSILB
	DATECREATED	PRICOMMERR
	DBVALID	PRIIFCARD
	DESC	PRIIFCHNERRA
	DUPIOLADDR	PRIIFCHNERRB
	ESIG	PRIIFCHNSILA
	FMWREV1	PRIIFCHNSILB
	FREESMSOVRRUN	PRIIFCOMMERR
	GROUP	PRIIFFILE
	GROUP.NUMBER	PRIIFRCVCHN
	GROUP.NUMPARAMS	PRIIFSTS
	GROUP.PARAM	PRIMARYSIG
	GROUP.POSITION	PRIRCVCHN
	HDWREV1	QUALSTATE
	HIPRIRATE	REASONSET
	HIST	SCANAREA
	HIST.EXTD	SCANASSOCDSP
	HIST.FAST	SCANCTRLLVL
	HIST.GATEPARAM	SCANEUHI
	HIST.GATESTRING	SCANEULO
	HIST.GATEVALUE	SCANGRPDTL
	HIST.NUMPARAMS	SCANPNTDTL
	HIST.PARAM	SCANRATE
	HIST.STD	SCRIPTOR
	IOLCHNFAILA	SECBLOCKNAME
	IOLCHNFAILB	SECCHNERRA
	IOLCHNHISTA	SECCHNERRB
	IOLCHNHISTB	SECCHNSILA
	IOLCHNSTSA	SECCHNSILB
	IOLCHNSTSB	SECCOMMERR

	IOLDAUGHSF IOLFREE IOLINKTYPE IOLMAXERR IOLOVRRUN IOLSOFTFAIL IOMCOMMAND IOMSTS IOMTYPE IOPBLOCKNAME IOPORCUR IOPORPREV IOSSTALLTIME IOSTKNDROP LASTIOLCMD LINKNUM LOPRIRATE MAXNUMPRIIOP MEDPRIRATE MODIFIEDBY MONREADPARAM MONSTATES NOTACTSUPV NUMCACHE NUMPRIIOP NUMSIGS OVERRUNSCUR OVERRUNSPREV OVRRUNCURHR OVRRUNPREVHR OVRSIGNA PARTMISMATCH PARTNOTVIS PDFIFORATEA PDFIFORATEM PERSWAPENB PERSWAPTHRES PHYCHNERRA1 PHYCHNERRB1 PHYCHNSILA1	SECIFCHNERRA SECIFCHNERRB SECIFCHNSILA SECIFCHNSILB SECIFCOMMERR SECIFRCVCHN SECONDARYSIG SECRCVCHN SECSIGSECLVL STATE STATRESET STTEXT STTEXTA STTEXTB SUPVFIFORATEA SUPVFIFORATEM SWTCHACT SYNCHSTS TOTCHNERRA TOTCHNERRB TOTCHNSILA TOTCHNSILB TREEBITMAP TREND TREND.NUMBER TREND.NUMPARAMS TREND.PARAM TREND.POSITION VERSION VERSIONDATE VERSIONNUM WITHBIAS WITHBIASENM WRFIFORATEA WRFIFORATEM
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.5 OLE for Process Control Server Block

Related topics

"OPC block" on page 278

4.5.1 OPC block

Description	Provides the representation of an OPC server to the control system. It does not have an associated Control Execution Environment block or any blocks assigned to it.	
Function	Serves as an independent block to provide a communications path to an OPC server. Does not have a corresponding run-time object in the system and no run-time status is obtained directly from the OPC server.	
Inputs	OPC compatible data exchange	
Outputs	See above.	
Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 DESC EXTREF EXTREFSTRUCT GROUP.NUMPARAMS HIST.NUMPARAMS HOSTIPPRI	HOSTNAMEPRI NAME PROGID[0..40] SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL TREND.NUMPARAMS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.6 Inter Cluster Gateway Block

Related topics

"Inter Cluster Gateway (ICG) Block" on page 279

4.6.1 Inter Cluster Gateway (ICG) Block

Description	Provides a path for the transfer of data between ACE nodes in two separate Experion clusters.
Function	This block acts as the means of communicating between Experion clusters. It makes CDA data from one Experion cluster available to the other Experion cluster. It can act as the client to the corresponding block in the other Experion cluster.
Inputs	Regulatory control data from OPC Gateways or Inter Cluster Gateways residing in ACE nodes in another Experion cluster
Outputs	Set point data pushed to regulatory control points resident in its Experion cluster

Parameters	ALMENBSTATE	GWPID
	BLCKCOMMENT1	GWSTATSRESET
	BLCKCOMMENT2	GWUTCCONVERT
	BLCKCOMMENT3	HIST.NUMPARAMS
	BLCKCOMMENT4	HOSTIPPRI
	CDISPAVGPPS	HOSTNAMEPRI
	CDISPAVGPPSCONN	IMAGEVER
	CDISPAVGSPSCONN	INALM
	CDISPMAXPPS	IPEERCONNERRCODE
	CDISPMAXPPSCONN	IPEERCONNERRINFO
	CDISPMAXSPSCONN	IPEERCONNSTS
	CPEERAVGPPS	IPEERNAME
	CPEERAVGPPSCONN	IPEERPATH
	CPEERAVGSPSCONN	JOURNALONLY
	CPEERMAXPPS	MODIFIEDBY
	CPEERMAXPPSCONN	NAME
	CPEERMAXSPSCONN	NUMACCRQUAVG
	CREATEDBY	NUMACEINCON
	CTRLCONFIRM	NUMACTIVEPOINTS
	DATECREATED	NUMCLIENTCONN
	DESC	NUMCPMINCON
	EEGSTATE	NUMFAILEDGETITEM
	GWCOMMAND	NUMFIMINCON
	GWHCIFLAG	NUMIOLMINCON
	GWHOSTIPPRI	NUMNTFRQUAVG
	GWHOSTNAMEPRI	NUMPARRSPAVG
	GWOPCCONNSTATUS	NUMPARRSPMAX
	GWOPCCONNSTR	NUMPOINTSREADS
	GWOPCDCPS	NUMPOINTSWRITES
	GWOPCGETERR	NUMSCEINCON
	GWOPCGSTATE	NUMSIGS
	GWOPCPARAMCNT	NUMSIOLMINCN
	GWOPCPMONERR	PRIMARYSIG
	GWOPCPMONNAME	RDISPDEGIMRCONN
	GWOPCPMONQUAL	REASONSET
	GWOPCPMONSTR	RPEERNAME
	GWOPCSRVSTATE	SCANCTRLLVL
	GWOPCSTOREERR	SECONDARYSIG
	GWOPCSTOREPS	SECSIGSECLVL
	GWOPCVERSION	TOTALNUMGETITEM
		TOTALNUMPOINTSREADS
		TREND.NUMPARAMS

		VERSIONDATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.7 Redundancy Module Block

Related topics

"RM block" on page 282

4.7.1 RM block

Description	Identifies the Primary and Secondary Redundancy Modules connected by a dedicated redundancy cable in a Redundant Chassis Pair (RCP). It associates the Primary RM with its "partner" Secondary RM block. This block always runs at an execution period of 2 seconds. It is redundancy compliant.
Function	Provides parameters describing the status and configuration of the RM. Handles notification reporting. Serves as a faceplate for any parameters whose scope corresponds to that of the RM as a whole.
Inputs	Integrated Control Protocol (ICP) communications Redundancy communications through the redundancy cable.
Outputs	See above.

Parameters	ALMENBSTATE AREVISION AUTOSYNCCMD AUTOSYNCOPT AUTOSYNCSTAT AVERSION BECMPRICMD BREVISION BVERSION CHANINUSE[0..31] CJDISABLE CLKADJUST CLKTIME CLKZONE CONFIGURED DISPPOS DISQSECCMD DRIVERNAME ENTERSBYCMD ERRFL ERRORCODE ERRORMSGIDX EUDESC GENSTATE GENSTATEA HIALM INTISWCMD IPADDRESS KEYWORD LASTSYNCABRT LASTSYNCARES MAJRECFAULT MAJURECFAULT MINRECFAULT MINURECFAULT MODCOMPATA MODTYPEA	MULREDUNSTAT NETWORKTYPE NUMSLOTS PREVISION PRODCODE PRODTYPE PRODTYPEA PROGCMD PROGCMDRECOG QUALPROGA READINESS READINESSA RECOVMSGIDX REDUNSTATE REDUNSTATEA REFRESHMS SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SECMODNAME SECNAMESTRING SECTMPNAME SERIALNUM SLOTNUMBER STDTIME SWAPCTRLCMD SWAPPOSCMD SWAPSBYCMD SYNCSECCMD ULCNBMAC VENDORID WCTCLKTIME
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.8 Fault Tolerant Ethernet Bridge Module Block

Related topics

"FTEB block" on page 284

4.8.1 FTEB block

Description	Provides representation of the Fault Tolerant Ethernet (FTE) Bridge module to support supervisory level communications over Honeywell's Fault Tolerant Ethernet (FTE) network.
Function	Provides parameters describing the status and configuration of the FTE Bridge module. Serves as a faceplate for any parameters whose scope corresponds to that of the Fault Tolerant Ethernet Bridge module as a whole. Supports non-redundant and redundant C200 Controller and Fieldbus Interface Module chassis configurations. Supports direct communication between Series A chassis I/O and C300 Supports C300 communication with ControlNet devices when mounted in a Series A chassis having a CNI Secondary waits to take control if the "Primary" fails. The device index of the Secondary FTE Bridge equals the address of the Primary FTE Bridge plus one.
Inputs	TCP/IP communications and Integrated Control Protocol (ICP) communications
Outputs	See Above

Parameters	BADIPCSUM	MAXFTENODES
	BADUDPCSUM	MAXNODEID
	BLCKCOMMENT1	MODIFIEDBY
	BLCKCOMMENT2	MODISREDUN
	BLCKCOMMENT3	NAME
	BLCKCOMMENT4	NONFTEMARTADDRCOUNT
	CONNCLOSEFAIL	NONFTEMARTAVGDEPTH
	CONNERR	NONFTEMARTCOLLCOUNT
	CONNOPENFAIL	NONFTEMARTMAXDEPTH
	CONNOPENREJECT	NUMFTENODES
	CONNSENDERR	NUMIOM
	CONSTAT	NUMSIGS
	CPUFREEAVG	ORIAPPCONNID[1..48]
	CPUFREEMIN	ORICONNSTATE[1..48]
	CREATEDBY	ORIPATH[1..48]
	CTRLCONFIRM	ORITCLASS[1..48]
	DATECREATED	ORMUXED[1..48]
	DESC	PRIMARYSIG
	DEVICEIDX	RDNCHASSISID
	FTEBBLOCK	RDNCMPT
	FTEMARTADDRCOUNT	RDNLOS
	FTEMARTAVGDEPTH	RDNSYNCSTATE
	FTEMARTCOLLCOUNT	REASONSET
	FTEMARTMAXDEPTH	RECVCLOSEREQ
	GROUP.NUMPARAMS	RECVCLOSERSP
	HIST.NUMPARAMS	RECVNAKS
	ICMPINDESTUNREACHS	RECVOPENREQ
	ICMPINECHOREPS	RECVOPENRSP
	ICMPINECHOS	SCANASSOCDSP
	ICMPINERRORS	SCANCTRLLVL
	ICMPINMSGS	SCANGRPDTL
	ICMPOUTDESTUNREACHS	SCANPNTDTL
	ICMPOUTECHOREPS	SECNAMESTRG
	ICMPOUTECHOS	SECONDARYSIG
	ICMPOUTERRORS	SECSIGSECLVL
	ICMPOUTMSGS	SENDCLOSEREQ
	ICPSTATRESET	SENDCLOSERESP
	IMAGEVER	SENDOPENREQ
	INTERLANFAILED	SENDOPENRSP
	IOMASAPH	SLOTNUMBER
	IOMGRCLI	STATRESET
	IOMNAME	TCPACTIVEOPENS

	IPADDRESS IPFRAGCREATES IPFRAGFAILS IPFRAGOKS IPINADDRERRORS IPINDELIVERS IPINDISCARDS IPINHDRERRORS IPINRECEIVES IPINUNKNOWNPORTS IPOUTDISCARDS IPOUTNOROUTES IPOUTREQUESTS IPREASSEMFAILS IPREASSEMOKS IPREASSEMREQS IPROUTINGDISCARDS LANAFAILED LANARXRATE LANARXRATEMAX LANATXRATE LANATXRATEMAX LANBFAILED LANBRXRATE LANBRXRATEMAX LANBTXRATE LANBTXRATEMAX	TCPATTEMPTFAILS TCPCONNTABLE TCPCURRESTAB TCPESTABRESETS TCPINERRS TCPINSEGS TCPOUTRESETS TCPOUTSEGS TCPPASSIVEOPENS TCPRETRANSSEGS TGTAPPCONNID[1..24 TGTCONNSTATE[1..24 TGTTCLASS[1..24] TREND.NUMPARAMS UDPINDGRAMS UDPINERRORS UDPLISTENERS UDPNOPORTS UDPOUTGRAMS UNCONNSENDERR VERSIONDATE XOVERFAILED
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9 Input Type I/O Module Blocks

> **Attention**
> Model numbers beginning with the prefix "TK" are for a coated version of the I/O module.

Related topics

"TC-HAI081/TK-HAI081 (8 channel HART - 10V / 4 to 20mA - Analog Input)" on page 287
"TC-IAH061/TK-IAH061 (6 Channel - 10V / 4 to 20mA Isolated - Analog Input)" on page 287
"TC-IAH161/TK-IAH161 (16 Channel - 10V / 4 to 20mA Non-Isolated - Analog Input)" on page 288
"TC-IDA161/TK-IDA161 (16 Channel - 120Vac Non-Isolated - Digital Input)" on page 289
"TC-IDD321/TK-IDD321 (32 Channel - 24Vdc Non-Isolated - Digital Input)" on page 290
"TC-IDJ161/TK-IDJ161 (16 Channel - 24Vdc Isolated - Digital Input)" on page 291
"TC-IDK161/TK-IDK161 (16 Channel - 120Vac Isolated - Digital Input)" on page 292
"TC-IDW161/TK-IDW161 (16 Channel - 220Vac Isolated - Digital Input)" on page 293
"TC-IDX081/TK-IDX081 (8 Channel - 120Vac Diagnostic Input)" on page 294
"TC-IDX161/TK-IDX161 (16 Channel - 24Vdc Diagnostic Input)" on page 295
"TC-IXL061/TK-IXL061 (6 Channel - Thermocouple Input)" on page 296
"TC-IXL062/TK-IXL062 (6 Channel - Thermocouple Input)" on page 297
"TC-IXR061/TK-IXR061 (6 Channel - RTD Input)" on page 298

4.9.1 TC-HAI081/TK-HAI081 (8 channel HART - 10V / 4 to 20mA - Analog Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.	
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	BADCAL[0..numChans-1] CALBIAS[0..numChans-1] COMMFAULT COMMTHRESHOLD DIGFILTER[0..numChans-1] HANDLETIMEOUT HART[0..numChans-1] HIGHENG[0..numChans-1]	HIGHSIGNAL[0..numChans-1] INPUTRANGE[0..numChans-1] IOMLOCATION LOWENG[0..numChans-1] LOWSIGNAL[0..numChans-1]NOTCHFILTER[0..numChans-1] PV[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.2 TC-IAH061/TK-IAH061 (6 Channel - 10V / 4 to 20mA Isolated - Analog Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO BADCAL[0..numChans-1] CALBIAS[0..numChans-1] CATNUMBER CEESTATE CHANTEXT CJDISABLE CJOFFSET CJOFFSET[0..numChans-1] DESC DIGFILTER[0..numChans-1] DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM HIGHENG[0..numChans-1] HIGHSIGNAL[0..numChans-1] IFTRANS INALM INPUTRANGE[0..numChans-1] IOMSLOT IOMTYPE KEYWORD LOWENG[0..numChans-1] LOWSIGNAL[0..numChans-1] MAJORREV	MINORREV NOTCHFILTER[0..numChans-1] NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN OHMOFFSET[0..numChans-1] ORDERINCEE ORDERINCM OVERRANGE[0..numChans-1] PERIOD PHASE PRODTYPE PVRAW[0..numChans-1] RTPPRESENT SAMPLERATE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SENSORTYPE[0..numChans-1] SIPTYPE[0..numChans-1] TEMPMODE ULCNBMAC UNDERRANGE[0..numChans-1] UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.3 TC-IAH161/TK-IAH161 (16 Channel - 10V / 4 to 20mA Non-Isolated - Analog Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV NUMCHANS	NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.4 TC-IDA161/TK-IDA161 (16 Channel - 120Vac Non-Isolated - Digital Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE	MAJORREV
	ASACONNSTS	MINORREV
	ASAERRCODE	NOFIELDPWR[0..numChans-1]
	ASAERRINFO	NUMCHANS
	CATNUMBER	NUMCONN
	CEESTATE	NUMDISCONN
	CHANTEXT	NUMSHUTDOWN
	COS[0..numChans-1]	ORDERINCEE
	DESC	ORDERINCM
	DLCNBSLOT	PERIOD
	ELOF[0..numChans-1]	PHASE
	ESTWEIGHT	PRODTYPE
	EUDESC	PVVAL[0..numChans-1]
	EWIREOFF[0..numChans-1]	SCANASSOCDSP
	EXECSTATE	SCANCTRLLVL
	FILTERHDR[0..numChans-1]	SCANEUHI
	FILTEROFF[0..numChans / 8]	SCANEULO
	FILTERON[0..numChans / 8]	SCANGRPDTL
	HIALM	SCANPNTDTL
	HWFAULT[0..numChans-1]	ULCNBMAC
	INALM	UPDATOPT
	IOMSLOT	VENDOR
	IOMTYPE	WIREOFF[0..numChans-1]
	KEYWORD	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.5 TC-IDD321/TK-IDD321 (32 Channel - 24Vdc Non-Isolated - Digital Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 32 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device. Data is“triggered”, or is current digital (Boolean) value.)
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT COS[0..numChans-1] DESC DLCNBSLOT ELOF[0..numChans-1] ESTWEIGHT EUDESC EWIREOFF[0..numChans-1] EXECSTATE FILTERHDR[0..numChans-1] FILTEROFF[0..numChans / 8] FILTERON[0..numChans / 8] HIALM HWFAULT[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD	MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVVAL[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR WIREOFF[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.6 TC-IDJ161/TK-IDJ161 (16 Channel - 24Vdc Isolated - Digital Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device. Data is“triggered”, or is current digital (Boolean) value.)
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT COS[0..numChans-1] DESC DLCNBSLOT ELOF[0..numChans-1] ESTWEIGHT EUDESC EWIREOFF[0..numChans-1] EXECSTATE FILTERHDR[0..numChans-1] FILTEROFF[0..numChans / 8] FILTERON[0..numChans / 8] HIALM HWFAULT[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD	MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVVAL[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR WIREOFF[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.7 TC-IDK161/TK-IDK161 (16 Channel - 120Vac Isolated - Digital Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT COS[0..numChans-1] DESC DLCNBSLOT ELOF[0..numChans-1] ESTWEIGHT EUDESC EWIREOFF[0..numChans-1] EXECSTATE FILTERHDR[0..numChans-1] FILTEROFF[0..numChans / 8] FILTERON[0..numChans / 8] HIALM HWFAULT[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD	MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVVAL[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR WIREOFF[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.8 TC-IDW161/TK-IDW161 (16 Channel - 220Vac Isolated - Digital Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT COS[0..numChans-1] DESC DLCNBSLOT ELOF[0..numChans-1] ESTWEIGHT EUDESC EWIREOFF[0..numChans-1] EXECSTATE FILTERHDR[0..numChans-1] FILTEROFF[0..numChans / 8] FILTERON[0..numChans / 8] HIALM HWFAULT[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD	MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVVAL[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR WIREOFF[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.9 TC-IDX081/TK-IDX081 (8 Channel - 120Vac Diagnostic Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC and provides selected diagnostic information for associated channels.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Supports the following diagnostics, which are user configurable for each channel: • Open Wire Detection: Senses when current input for a given channel falls below a certain value. When an input uses dry contacts, you must include a bleed resistor in the input. You may not need a bleed resistor for solid state contacts. • Loss of Field Power: Senses when field power of a group of channels is lost. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR;
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.10 TC-IDX161/TK-IDX161 (16 Channel - 24Vdc Diagnostic Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC and provides selected diagnostic information for associated channels.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Supports the following diagnostic, which is user configurable for each channel: • Open Wire Detection: Senses when current input for a given channel falls below a certain value. When an input uses dry contacts, you must include a bleed resistor in the input. You may not need a bleed resistor for solid state contacts. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.11 TC-IXL061/TK-IXL061 (6 Channel - Thermocouple Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.12 TC-IXL062/TK-IXL062 (6 Channel - Thermocouple Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC. This module is a replacement for the previous TC-IXL061 module.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE	NOTCHFILTER
	ASACONNSTS	NUMCHANS
	ASAERRCODE	NUMCONN
	ASAERRINFO	NUMDISCONN
	CALBIAS	NUMSHUTDOWN
	CATNUMBER	ORDERINCEE
	CEESTATE	ORDERINCM
	CHANNUM	PERIOD
	CHANTEXT	PHASE
	CHANTYPE	PRODTYPE
	DESC	PUBRATE
	DIGFILTER	SAMPLERATE
	DLCNBSLOT	SCANASSOCDSP
	ESTWEIGHT	SCANCTRLLVL
	EUDESC	SCANEUHI
	EXECSTATE	SCANEULO
	HIALM	SCANGRPDTL
	INALM	SCANPNTDTL
	IOMSLOT	SENSORTYPE
	IOMTYPE	SIPTYPE
	INPUTRANGE	TEMPMODE
	KEYWORD	ULCNBMAC
	MAJORREV	UPDATOPT
	MINORREV	VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.9.13 TC-IXR061/TK-IXR061 (6 Channel - RTD Input)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10 Output Type I/O Module Blocks

Attention
Model numbers beginning with the prefix“TK” are for a coated version of the I/O module.

Related topics

“TC-HAO081/TK-HAO081 (8 channel HART - 10V / 4 to 20mA - Analog Output)” on page 300
“TC-OAH061/TK-OAH061 (6 Channel - 4 to 20mA - Analog Output)” on page 301
“TC-OAV061/TK-OAV061 (6 Channel - 10V - Analog Output)” on page 301
“TC-OAV081/TK-OAV081 (8 Channel - 10V / 4 to 20mA Non-Isolated - Analog Output)” on page 302
“TC-ODA161/TK-ODA161 (16 Channel - 120/220Vac Non-Isolated - Digital Output)” on page 303
“TC-ODD321/TK-ODD321 (32 Channel - 24Vdc Non-Isolated Digital Output)” on page 304
“TC-ODJ161/TK-ODJ161 (16 Channel - 24Vdc Isolated Digital Output)” on page 305
“TC-ODK161/TK-ODK161 (16 Channel - 120/220Vac Isolated - Digital Output)” on page 306
“TC-ODX081/TK-ODX081 (8 Channel - 120Vac- Diagnostic Output)” on page 307
“TC-ODX161/TK-ODX161 (16 Channel - 24Vdc- Diagnostic Output)” on page 308
“TC-ORC081/TK-ORC081 (8 Channel - 8 n.c., 8 n.o. 5-150Vdc, 10-265Vac Isolated - Relay Output)” on page 309
“TC-ORC161/TK-ORC161 (16 Channel, 5-150Vdc, 10-265Vac Isolated - Contact Output)” on page 310

4.10.1 TC-HAO081/TK-HAO081 (8 channel HART - 10V / 4 to 20mA - Analog Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.	
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC.	
Outputs	Real-time data transmission to physical device.	
Parameters	BADCAL[0..numChans-1] CALBIAS[0..numChans-1] COMMFAULT COMMTHRESHOLD FAULTVALUE[0..numChans-1] HART[0..numChans-1] HIGHENG[0..numChans-1] HIGHSIGNAL[0..numChans-1] IDLEMODE[0..numChans-1] IOMLOCATION	LOWENG[0..numChans-1] LOWSIGNAL[0..numChans-1] OP[0..numChans-1] PV[0..numChans-1] SHEDMODE[0..numChans-1] SIPTYPE[0..numChans-1] SLOT0[0..numChans-1] SLOT1[0..numChans-1] SLOT2[0..numChans-1] SLOT3[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.2 TC-OAH061/TK-OAH061 (6 Channel - 4 to 20mA - Analog Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.	
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC.	
Outputs	Real-time data transmission to physical device.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO BADCAL[0..numChans-1] CALBIAS[0..numChans-1] CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE FAULTVALUE[0..numChans-1] HIALM HIGHENG[0..numChans-1] HIGHSIGNAL[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD LOWENG[0..numChans-1] LOWSIGNAL[0..numChans-1]	MAJORREV MINORREV NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN OPFINAL[0..numChans-1] ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SAMPLERATE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SIPTYPE[0..numChans-1] ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.3 TC-OAV061/TK-OAV061 (6 Channel - 10V - Analog Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 6 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC.	
Outputs	Real-time data transmission to physical device.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO BADCAL[0..numChans-1] CALBIAS[0..numChans-1] CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE FAULTVALUE[0..numChans-1] HIALM HIGHENG[0..numChans-1] HIGHSIGNAL[0..numChans-1] INALM IOMSLOT IOMTYPE KEYWORD LOWENG[0..numChans-1]	LOWSIGNAL[0..numChans-1] MAJORREV MINORREV NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN OPFINAL[0..numChans-1] ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SAMPLERATE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SIPTYPE[0..numChans-1] ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.4 TC-OAV081/TK-OAV081 (8 Channel - 10V / 4 to 20mA Non-Isolated - Analog Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC.
Outputs	Real-time data transmission to physical device.

Parameters	ALMENBSTATE	MAJORREV
	ASACONNSTS	MINORREV
	ASAERRCODE	NUMCHANS
	ASAERRINFO	NUMCONN
	BADCAL[0..numChans-1]	NUMDISCONN
	CALBIAS[0..numChans-1]	NUMSHUTDOWN
	CATNUMBER	OPFINAL[0..numChans-1]
	CEESTATE	ORDERINCEE
	CHANTEXT	ORDERINCM
	DESC	PERIOD
	DLCNBSLOT	PHASE
	ESTWEIGHT	PRODTYPE
	EUDESC	SAMPLERATE
	EXECSTATE	SCANASSOCDSP
	FAULTVALUE[0..numChans-1]	SCANCTRLLVL
	HIALM	SCANEUHI
	HIGHENG[0..numChans-1]	SCANEULO
	HIGHSIGNAL[0..numChans-1]	SCANGRPDTL
	INALM	SCANPNTDTL
	IOMSLOT	SIPTYPE[0..numChans-1]
	IOMTYPE	ULCNBMAC
	KEYWORD	UPDATOPT
	LOWENG[0..numChans-1]	VENDOR
	LOWSIGNAL[0..numChans-1]	WIREOFF[0..7]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.5 TC-ODA161/TK-ODA161 (16 Channel - 120/220Vac Non-Isolated - Digital Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC.
Outputs	Real-time data transmission to physical device.

Parameters	ALMENBSTATE	NOLOAD[0..numChans-1]
	ASACONNSTS	NUMCHANS
	ASAERRCODE	NUMCONN
	ASAERRINFO	NUMDISCONN
	CATNUMBER	NUMSHUTDOWN
	CEESTATE	ORDERINCEE
	CHANTEXT	ORDERINCM
	DESC	PERIOD
	DLCNBSLOT	PHASE
	ENOLOAD[0..numChans-1]	PRODTYPE
	ESTWEIGHT	PVSTS
	ETRANS	PVVAL[0..numChans-1]
	EUDESC	PWMPERIOD[0..numChans-1]
	EVERIFY[0..numChans-1]	SAFESTATE[0..numChans-1]
	EXECSTATE	SCANASSOCDSP
	EZCROSS[0..numChans-1]	SCANCTRLLVL
	FAILSTATE[0..numChans-1]	SCANEUHI
	HIALM	SCANEULO
	INALM	SCANGRPDTL
	IOMSLOT	SCANPNTDTL
	IOMTYPE	SHORT[0..numChans-1]
	KEYWORD	ULCNBMAC
	MAJORREV	UPDATOPT
	MINORREV	VENDOR
	NOFIELDPWR[0..numChans-1]	VERIFYLOST[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.6 TC-ODD321/TK-ODD321 (32 Channel - 24Vdc Non-Isolated Digital Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 32 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC. Data is “triggered”, or is current digital (Boolean) value or pulsed (real) value.
Outputs	Real-time data transmission to physical device.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ENOLOAD[0..numChans-1] ESTWEIGHT ETRANS EUDESC EVERIFY[0..numChans-1] EXECSTATE EZCROSS[0..numChans-1] FAILSTATE[0..numChans-1] HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NOLOAD[0..numChans-1]	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVSTS PVVAL[0..numChans-1] PWMPERIOD[0..numChans-1] SAFESTATE[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SHORT[0..numChans-1] ULCNBMAC UPDATOPT VENDOR VERIFYLOST[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.7 TC-ODJ161/TK-ODJ161 (16 Channel - 24Vdc Isolated Digital Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC. Data is“triggered”, or is current digital (Boolean) value or pulsed (real) value.
Outputs	Real-time data transmission to physical device.

Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ENOLOAD[0..numChans-1] ESTWEIGHT ETRANS EUDESC EVERIFY[0..numChans-1] EXECSTATE EZCROSS[0..numChans-1] FAILSTATE[0..numChans-1] HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV NOFIELDPWR[0..numChans-1] NOLOAD[0..numChans-1]	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE PVSTS PVVAL[0..numChans-1] PWMPERIOD[0..numChans-1] SAFESTATE[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL SHORT[0..numChans-1] ULCNBMAC UPDATOPT VENDOR VERIFYLOST[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.8 TC-ODK161/TK-ODK161 (16 Channel - 120/220Vac Isolated - Digital Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC.
Outputs	Real-time data transmission to physical device.

Parameters	ALMENBSTATE	NOLOAD[0..numChans-1]
	ASACONNSTS	NUMCHANS
	ASAERRCODE	NUMCONN
	ASAERRINFO	NUMDISCONN
	CATNUMBER	NUMSHUTDOWN
	CEESTATE	ORDERINCEE
	CHANTEXT	ORDERINCM
	DESC	PERIOD
	DLCNBSLOT	PHASE
	ENOLOAD[0..numChans-1]	PRODTYPE
	ESTWEIGHT	PVSTS
	ETRANS	PVVAL[0..numChans-1]
	EUDESC	PWMPERIOD[0..numChans-1]
	EVERIFY[0..numChans-1]	SAFESTATE[0..numChans-1]
	EXECSTATE	SCANASSOCDSP
	EZCROSS[0..numChans-1]	SCANCTRLLVL
	FAILSTATE[0..numChans-1]	SCANEUHI
	HIALM	SCANEULO
	INALM	SCANGRPDTL
	IOMSLOT	SCANPNTDTL
	IOMTYPE	SHORT[0..numChans-1]
	KEYWORD	ULCNBMAC
	MAJORREV	UPDATOPT
	MINORREV	VENDOR
	NOFIELDPWR[0..numChans-1]	VERIFYLOST[0..numChans-1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.9 TC-ODX081/TK-ODX081 (8 Channel - 120Vac- Diagnostic Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC and provides selected diagnostic information for associated channels.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Supports the following diagnostics, which are user configurable for each channel with the exception of the Short Circuit Protection/Overload diagnostic which is always enabled: • Short Circuit Protection/Overload: Senses when current draw for a given channel is above the limit and protects the device from damage. • Loss of Field Power: Senses lack of power for a channel, if zero-crossing on the ac-line power is not detected which causes the output state to change. • No Load/Hardware Point Fault: Senses when the output current draw falls below the threshold or a hardware output failure occurs. It only works when the output is in the OFF state. • Output Verification: Verifies if the actual output state matches the commanded output state for field side verification. It only works when the output is in the ON state. • Pulse Test: Periodically checks the output to verify that it still has the ability to change states without causing the load to transition. (This function only operates in systems with software version R120 or greater.) Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC. Data is“triggered”, or is current digital (Boolean) value or pulsed (real) value.	
Outputs	Real-time data transmission to physical device.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.10 TC-ODX161/TK-ODX161 (16 Channel - 24Vdc- Diagnostic Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC and provides selected diagnostic information for associated channels.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Supports the following diagnostics, which are user configurable for each channel with the exception of the Short Circuit Protection/Overload diagnostic, which is always enabled. • Short Circuit Protection/Overload: Senses when current draw for a given channel is above the limit and protects the device from damage. • No Load/Hardware Point Fault: Senses when the output current draw falls below the threshold or a hardware output failure occurs. It only works when the output is in the OFF state. • Output Verification: Verifies if the actual output state matches the commanded output state for field side verification. It only works when the output is in the ON state. • Pulse Test: Periodically checks the output to verify that it still has the ability to change states without causing the load to transition. (This function only operates in systems with software version R120 or greater.) Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 16 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC. Data is“triggered”, or is current digital (Boolean) value or pulsed (real) value.	
Outputs	Real-time data transmission to physical device.	
Parameters	ALMENBSTATE ASACONNSTS ASAERRCODE ASAERRINFO CATNUMBER CEESTATE CHANTEXT DESC DLCNBSLOT ESTWEIGHT EUDESC EXECSTATE HIALM INALM IOMSLOT IOMTYPE KEYWORD MAJORREV MINORREV	NUMCHANS NUMCONN NUMDISCONN NUMSHUTDOWN ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.11 TC-ORC081/TK-ORC081 (8 Channel - 8 n.c., 8 n.o. 5-150Vdc, 10-265Vac Isolated - Relay Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.

Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.	
Inputs	Real-time data transmission from configured IOC.	
Outputs	Real-time data transmission to physical device.	
Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 DESC DLCNBSLOT EXECSTATE FAILSTATE[0..numChans-1] IOMSLOT IOMTYPE IOCONNSTATUS MAJORREV MINORREV	NAME SAFESTATE[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.10.12 TC-ORC161/TK-ORC161 (16 Channel, 5-150Vdc, 10-265Vac Isolated - Contact Output)

Description	Identifies the physical IOM for the CPM to provide links to associated IOC.
Function	Defines type of IOM, number of channels, execution state, and communications path for data. Provides link to IOC through IO manager software resident in the CPM. Executes once every cycle. Includes IOC assignment to one of 8 channels (points), as part of IOM configuration.
Inputs	Real-time data transmission from configured IOC.
Outputs	Real-time data transmission to physical device.

Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 DESC DLCNBSLOT EXECSTATE FAILSTATE[0..numChans-1] IOMSLOT IOMTYPE IOCONNSTATUS MAJORREV MINORREV	NAME SAFESTATE[0..numChans-1] SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC UPDATOPT VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.11 Serial Interface Module (SIM) I/O Module Block

Related topics

"TC-MUX021/TK-MUX021 (Up to 32 Array Channel Function Blocks)" on page 312

4.11.1 TC-MUX021/TK-MUX021 (Up to 32 Array Channel Function Blocks)

Description	Identifies the physical Serial Interface Module (SIM) for the CPM to provide links to associated Array Channel blocks and provides selected diagnostic events for associated channels.	
Function	Provides configuration and communication software to enable devices to communicate via an ASCII serial protocol to perform bi-directional data exchange directly with the Experion PKS Control Processor. Stores are not guaranteed during a failover. That is, the store attempt may occur in the primary, but not reach the IO Module before the failover occurs. The secondary will not attempt to re-send the information.	
Inputs	Real-time data transmission from configured FTAs.	
Outputs	Real-time data transmission to physical device.	
Parameters	ALMENBSTATE ASACONNSTS[0..7] ASAERRCODE[0..7] ASAERRINFO[0..7] CATNUMBER CEESTATE CHANINUSE[0..31] CHANSTS[0..31] DESC DLCNBSLOT ERRCODE[0..31] ERRFL[0..31] ESTWEIGHT EUDESC EXECSTATE FTAAOVRNFL FTAASTS FTABOVRNFL FTABSTS INALM IOMSLOT IOMTYPE	KEYWORD MAJORREV MINORREV NUMCHANS NUMCONN[0..7] NUMDISCONN[0..7] NUMSHUTDOWN[0..7] ORDERINCEE ORDERINCM PERIOD PHASE PRODTYPE SCANASSOCDSP SCANCTRLLVL SCANEUHI SCANEULO SCANGRPDTL SCANPNTDTL ULCNBMAC VENDOR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.12 Process Manager Input/Output (PMIO) Blocks

Related topics

"AICHANNEL (PMIO) block" on page 313
"HAICHANNEL (PMIO) block" on page 314
"AOCHANNEL (PMIO) block" on page 316
"HAOCHANNEL (PMIO) block" on page 317
"DICHANNEL (PMIO) block" on page 319
"DOCHANNEL (PMIO) block" on page 320
"HLAI block" on page 320
"HLAIHART block" on page 322
"LLMUX block" on page 323
"STI_MV block" on page 324
"AO16 block" on page 325
"AO16HART block" on page 327
"DI24V block" on page 328
"DISOE block" on page 329
"DI block" on page 330
"DO32 block" on page 331

4.12.1 AICHANNEL (PMIO) block

Description	The AI channel block represents a single analog input point on one of the following I/O Processors: HLAI, HLAI-100, LLAI, LLMUX, RHMUX, and STI. The analog input point converts an analog signal received from a field sensor to engineering units for use by control function blocks in the Experion PKS system. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection

Parameters	CHANNUM	PVEXEULO
	CJTACT	PVEXHIFL
	COMMAND	PVEXLOFL
	DAMPING	PVRAW
	DECONF	PVRAWHI
	DESC	PVRAWLO
	INPTDIR	PVSTS
	IOP	PVTEMP
	IOPTYPE	SECVAR
	LASTPV	SENSRTYP
	LOCUTOFF	SERIALNO
	LRL	SLWSRCID
	LRV	STATE
	NAME	STI_EU
	PIUOTDCF	STIDBDISCRE
	PNTFORM	STIPVNUMBER
	PTEXECST	STISCRATCHPAD
	PV	STISTATUS
	PVCALC	STISWVER
	PVCHAR	STITAG
	PVCLAMP	TCRNGOPT
	PVEUHI	TF
	PVEULO	URL
	PVEXEUHI	URV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.2 HAICHANNEL (PMIO) block

Description	The HAI channel block represents a single analog input point on the HLAIHART I/O Processor. The analog input point converts an analog signal received from a field sensor to engineering units for use by control function blocks in the Experion PKS system. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection Additionally, the HAI channel supports HART digital data received from HART capable devices. Device Id data is read from the device and cached in the IOP. Dynamic and device variable data and device status is collected from the device for use by the control system.

Parameters	ACCEPTDEV	HCMD16
	ACCEPTRNG	HCMD48BT[1..200]
	CHANNUM	HCMD48NOTIFY[1..200]
	CJTACT	HCMD48STRNGS[1..200]
	COMMAND	HCMDFAIL
	CONTAINEDIN	HCMDRESP
	DEVICE	HCOMERFL
	DECONF	HCOMFAIL
	DESC	HCOMHYS
	HCFGDEV	HCOMSTS
	HCMD00	HCOMTHRS
	HCMD12	HDAY
	HCMD13	HSWREV
	HCMD14	HTAG
	HDESC	HTDEU
	HDEVID	HTDLRL
	HDEVIDFL	HTDMINSPAN
	HDEVIDCD	HTDSN
	HDEVMFG	HTDURL
	HDEVMISM	HUCMDREV
	HDEVMSG	HWRTPRCTCODE
	HDEVREV	HYEAR
	HDEVST	INPTDIR
	HDEVTYP	IOP
	HDVMFGCD	IOPTYPE
	HDVREVCD	LASTPV
	HDVTYPCD	LOCUTOFF
	HDVTYPCDNAME	LRL
	HDYNDESC[1..4]	LRV
	HDYNDVC[1..4]	NAME
	HDYNEU[1..4]	PIUOTDCF
	HDYNNAME[1..4]	PNTFORM
	HDYNVAL[1..4]	PTEXECST
	HENABLE	PV
	HEU	PVCALC
	HFASSYNO	PVCHAR
	HFLAGS	PVCLAMP
	HHWREV	PVEUHI
	HISHART5	PVEULO
	HMONTH	PVEXEUHI
	HNCOMERR	PVEXEULO
	HNMSMINPRE	PVEXHIFL

	HREVMISM	PVEXLOFL
	HPHYSIG	PVRAW
	HPVALMCODE	PVRAWHI
	HPVCHAR	PVRAWLO
	HPVDAMP	PVSTS
	HPVLRV	PVTEMP
	HPVMISM	RESETHCOMERR
	HPVTLDST	SECVAR
	HPVURV	SENSRTYP
	HPVXFRCODE	SLWSRCID
	HSCANCFG	STATE
	HSCANOVR	TCRNGOPT
	HSLOTDSC[1..4]	TF
	HSLOTDVC[1..4]	URL
	HSLOTEU[1..4]	URV
	HSLOTNAME[1..4]	
	HSLOTVAL [1..4]	
	HSMSTRFL	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.3 AOCHANNEL (PMIO) block

Description	The AO channel block converts the output value (OP) to a 4-20 mA output signal for operating final control elements such as valves and actuators in the field. The OP parameter value can be controlled from a Experion PKS regulatory point, the operator, or an SCM. To convert the OP value to a 4-20 mA signal, the AO channel performs: • Direct/Reverse Output Function • Nonlinear Output Characterization The AO channel block can be associated with either an AO8 or AO16 IOP.

Parameters	CHANNUM COMMFAILFL DESC FAILOPT INITREQ INITVAL IOP IOPTYPE NAME OP OPCHAR OPFINAL OPIN0 OPIN1	OPIN2 OPIN3 OPIN4 OPIN5 OPOUT0 OPOUT1 OPOUT2 OPOUT3 OPOUT4 OPOUT5 OPTDIR PNTFORM PTEXECST STDBYMAN
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.4 HAOCHANNEL (PMIO) block

Description	The HAO channel block represents a single analog output point on the AO16HART I/O Processor. The AO channel block converts the output value (OP) to a 4-20 mA output signal for operating final control elements such as valves and actuators in the field. The OP parameter value can be controlled from a Experion PKS regulatory point, the operator, or an SCM. To convert the OP value to a 4-20 mA signal, the AO channel performs: • Direct/Reverse Output Function • Nonlinear Output Characterization Additionally, the HAO channel supports HART digital data received from HART capable devices. Device Id data is read from the device and cached in the IOP. Dynamic and device variable data and device status is collected from the device for use by the control system.

Parameters	ACCEPTDEV	HDAY
	CHANNUM	HDESC
	COMMFAILFL	HDEVID
	CONTAINEDIN	HDEVIDFL
	DESC	HDEVIDCD
	DEVICE	HDEVMFG
	FAILOPT	HDEVMISM
	HCFGDEV	HDEVMSG
	HCMD00	HDEVREV
	HCMD12	HDEVST
	HCMD13	HDEVTYP
	HCMD14	HDVMFGCD
	HCMD16	HDVREVCD
	HCMD48BT[1..200]	HDVTYPCD
	HCMD48NOTIFY[1..200]	HDVTYPCDNAME
	HCMD48STRNGS[1..200]	HDYNDESC[1..4]
	HCMDFAIL	HDYNDVC[1..4]
	HCMDRESP	HDYNEU[1..4]
	HCOMERFL	HDYNNAME[1..4]
	HCOMFAIL	HDYNVAL[1..4]
	HCOMHYS	HENABLE
	HCOMSTS	HEU
	HCOMTHRS	HFASSYNO
	HHWREV	HFLAGS
	HISHART5	ICONSTATE
	HMONTH	INITREQ
	HNCOMERR	INITVAL
	HNMSMINPRE	IOP
	HREVMISM	IOPTYPE
	HPHYSIG	NAME
	HPVALMCODE	OP
	HPVDAMP	OPCHAR
	HPVLRV	OPFINAL
	HPVTLDST	OPIN0
	HPVURV	OPIN1
	HPVXFRCODE	OPIN2
	HSCANCFG	OPIN3
	HSCANOVR	OPIN4
	HSLOTDSC[1..4]	OPIN5
	HSLOTDVC[1..4]	OPOUT0
	HSLOTEU[1..4]	OPOUT1
	HSLOTNAME[1..4]	OPOUT2

	HSLOTVAL [1..4]	OPOUT3
	HSMSTRFL	OPOUT4
	HSWREV	OPOUT5
	HTAG	OPTDIR
	HTDEU	PNTFORM
	HTDLRL	PTEXECST
	HTDMINSPAN	RESETHCOMERR
	HTDSN	
	HTDURL	
	HUCMDREV	
	HWRTPRCTCODE	
	HYEAR	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.5 DICHANNEL (PMIO) block

Description	The DI channel block represents a single discrete input point on a DI, DI24V, or DISOE I/O Processors. A digital input point converts a digital PVRAW signal received from the field to a PV that can be used by other data points in the control strategy. Control strategies can test for a bad Digital Input PV. Parameter BADPVFL is set ON when: • The PV source has been switched to Substituted, and the point is inactive or the module status is Idle. • The PV source is AUTO and the PV is not being updated, because, either the point is inactive, the module is idle, there is a slot soft failure, or the FTA is missing. The digital input point is a single-input point that can be configured as a status input or a latched input, as described in the following sections.	
Parameters	ALMOPT	IOPTYPE
	BADPVFL	NAME
	CHANNUM	OFFNRMFL
	DEBOUNCE	PNTFORM
	DESC	PTEXECST
	DITYPE	PV
	DLYTIME	PVCHGDLY
	EVTOPT	PVNORMAL
	HIGHAL	PVRAW
	INPTDIR	PVSOURCE
	IOP	PVSRCOPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.6 DOCHANNEL (PMIO) block

Description	The DO channel block represents a single discrete input point on a DO32 I/O Processor. The digital output point provides a digital output to the field based on the origin of the input and the configured parameters. The digital output point does not have any modes.	
Parameters	OFFPULSE	
CHANNUM	ONPULSE	
COMMFAILFL	OP	
DESC	OPTDIR	
DOTYPE	PERIOD	
FAILOPT	PNTFORM	
INITREQ	PTEXECST	
INITVAL	SO	
IOP	SOINITVAL	
IOPTYPE	SOREADFAIL	
NAME	STDBYMAN	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.7 HLAI block

Description	The analog input point converts an analog PV signal received from a field sensor to engineering units for use by other data points in the control strategy, as shown in the following figure. To accomplish this function, the analog input point performs the following functions. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection

Parameters	CALIBSTS DBVALID DBVALIDCMD DESC FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB IOMCARDA IOMCARDB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPT	NAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS PTEXECST PV PVSTS REDDATAA REDDATAB RESETERRORSCMD SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL SCANRATE SELECTCABLEACMD SELECTCABLEBCMD SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.8 HLAIHART block

<table>
<tr><td>Description</td><td>High Level Analog Input IOP, HART Capable, 16 Channel.

Each input channel is capable of scanning (100 ms sampling) a standard analog input (0 to 100%) and supporting digital data transfer using HART communications protocol.

The analog input point converts an analog PV signal received from a field sensor to engineering units for use by other data points in the control strategy. To accomplish this function, the HLAIHART point performs the following functions.

• Analog-to Digital Conversion
• PV Characterization
• Range Checking and PV Filtering
• PV Source Selection
• Alarm Detection

Additionally, the IOP can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOP. Dynamic and device variable data and device status is collected from the device for use by the control system. IOP allows for servicing of any pass-through commands issued from host/master devices.</td></tr>
</table>

Parameters	CALIBSTSDBVALID DBVALIDCMD DESC DEVICELOCATION FTACONNA FTACONNB FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB ICONSTATE HCUAVAIL IOMPLREVA IOMPLREVB IOPLOCATION IOMCARDA IOMCARDB IOMDESCA IOMDESCB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA	IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPTNAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS REDDATAA REDDATAB RESETERRORSCMD SCANRATE SELECTCABLEACMD SELECTCABLEBCMD SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.9 LLMUX block

Description	Low Level Multiplexer IOP, generally used for Data Acquisition points.

Parameters	CALIBFTA1STS	NAME
	CALIBFTA2STS	NONREDFTABA
	DBVALID	NOTREDCNFGA
	DBVALIDCMD	NOTSAMEFTAA
	DESC	NUMCHANS
	FREQ6050	PTEXECST
	FTAPRESA	PV
	FTAREVERSEDA	PVSTS
	FWINVALIDA	RESETERRORSCMD
	IOMCARDA	SCANASSOCDSP
	IOMFILEA	SCANCTRLLVL
	IOMFWREVA	SCANGRPDTL
	IOMHWREVA	SCANPNTDTL
	IOMLHFSTA	SCANRATE
	IOMNUM	SELECTCABLEACMD
	IOMSTATE	SELECTCABLEBCMD
	IOMSTSA	TYPEINVALIDA
	IOMTYPE	WARMSTRTA
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.10 STI_MV block

Description	The STIMV IOP supports all the Smart Transmitters listed above and multi-PV Smart Transmitter types such as the following: • SCM3000 Smart Flow Transmitter (Coriolis method) • Drexelbrook SLT Level Transmitter • SMV 3000 Multivariable Pressure Transmitter • SGC 3000 Gas Chromatograph An STIMV IOP allows up to four multi-PV transmitters or a mix of multi-PV and single PV transmitter inputs that total no more than 16. A multi-PV transmitter is configured as if it were in“n” contiguous slots where“n” equals the number of PVs expected. The STITAG parameter value for each contiguous slot must be identical

Parameters	DBVALID	IOREDOPT
	DBVALIDCMD	NAME
	DESC	NONREDFTABA
	FREQ6050	NONREDFTABB
	FTAPRESA	NOTREDCNFGA
	FTAPRESB	NOTREDCNFGB
	FTAREVERSEDA	NOTSAMEFTAA
	FTAREVERSEDB	NOTSAMEFTAB
	FWINVALIDA	NUMCHANS
	FWINVALIDB	PTEXECST
	IOMCARDA	PV
	IOMCARDB	PVSTS
	IOMFILEA	REDDATAA
	IOMFILEB	REDDATAB
	IOMFWREVA	RESETERRORSCMD
	IOMFWREVB	SCANASSOCDSP
	IOMHWREVA	SCANCTRLLVL
	IOMHWREVB	SCANGRPDTL
	IOMLHFSTA	SCANPNTDTL
	IOMLHFSTB	SCANRATE
	IOMNUM	SELECTCABLEACMD
	IOMOPERA	SELECTCABLEBCMD
	IOMOPERB	SWAPPRIMARYCMD
	IOMSTATE	TYPEINVALIDA
	IOMSTSA	TYPEINVALIDB
	IOMSTSB	WARMSTRTA
	IOMTYPE	WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.11 AO16 block

Description	Analog Output IOP, 16 channel.

Parameters	CALIBSTS DBVALID DBVALIDCMD DESC FAILOPT FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB INITVAL IOMCARDA IOMCARDB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPT	NAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS OP OPFINAL PTEXECST REDDATAA REDDATAB RESETERRORSCMD SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL SCANRATE SELECTCABLEACMD SELECTCABLEBCMD STDBYSTSA STDBYSTSB SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.12 AO16HART block

Description	Analog Output HART IOP, 16 channel. Each channel is capable of supplying a standard analog output (4 to 20mA) and supporting digital data transfer using HART communications protocol. To accomplish this function, the AO16HART point performs the following functions. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection Additionally, the IOP can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOP. Dynamic and device variable data and device status is collected from the device for use by the control system. IOP allows for servicing of any pass-through commands issued from host/master devices.

Parameters	CALIBSTS DBVALID DBVALIDCMD DESC DEVICELOCATION EUDESC FAILOPT FTACONNA FTACONNB FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB HAUTODET[1..16] HCUAVAIL ICONSTATE IOMCARDA IOMCARDB IOMDESCA IOMDESCB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM	IOMOPERA IOMOPERB IOMPLREVA IOMPLREVB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPT IOPLOCATION NAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS REDDATAA REDDATAB RESETERRORSCMD SCANRATE SELECTCABLEACMD SELECTCABLEBCMD STDBYSTS SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.13 DI24V block

Description	Digital Input IOP, 24 Vdc.

Parameters	BADPVFL DBVALID DBVALIDCMD DESC FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB IOMCARDA IOMCARDB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPT	NAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS PTEXECST PV PVSTS REDDATAA REDDATAB RESETERRORSCMD SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL SCANRATE SELECTCABLEACMD SELECTCABLEBCMD SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.14 DISOE block

Description	Digital Input Sequence of Events

Parameters	BADPVFL CHNLNAME[1..NUMCHAN] DBVALID DBVALIDCMD DESC FTAPRESA FTAPRESB FTAREVERSEDA FTAREVERSEDB FWINVALIDA FWINVALIDB IOMCARDA IOMCARDB IOMFILEA IOMFILEB IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMSTATE IOMSTSA IOMSTSB IOMTYPE IOREDOPT	NAME NONREDFTABA NONREDFTABB NOTREDCNFGA NOTREDCNFGB NOTSAMEFTAA NOTSAMEFTAB NUMCHANS PTEXECST PV PVSTS REDDATAA REDDATAB RESETERRORSCMD SCANASSOCDSP SCANCTRLLVL SCANGRPDTL SCANPNTDTL SCANRATE SELECTCABLEACMD SELECTCABLEBCMD STMCHLASTOVERRUNTIMEA (Partner A) STMCHLASTOVERRUNTIMEB (Partner B) STMCHMAXOVRRUNTIMEA (Partner A) STMCHMAXOVRRUNTIMEA (Partner B) STMCHOVRRUNSA (Partner A) STMCHOVRRUNSB (Partner B) SWAPPRIMARYCMD TYPEINVALIDA TYPEINVALIDB WARMSTRTA WARMSTRTB
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.12.15 DI block

Description	
Parameters	DBVALID DBVALIDCMD

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.

4.12.16 DO32 block

Description	Digital Output, 32 channel.	
Parameters	DBVALID	NAME
	DBVALIDCMD	NONREDFTABA
	DESC	NONREDFTABB
	FAILOPT	NOTREDCNFGA
	FTAPRESA	NOTREDCNFGB
	FTAPRESB	NOTSAMEFTAA
	FTAREVERSEDA	NOTSAMEFTAB
	FTAREVERSEDB	NUMCHANS
	FWINVALIDA	OP
	FWINVALIDB	PTEXECST
	IOMCARDA	REDDATAA
	IOMCARDB	REDDATAB
	IOMFILEA	RESETERRORSCMD
	IOMFILEB	SCANASSOCDSP
	IOMFWREVA	SCANCTRLLVL
	IOMFWREVB	SCANGRPDTL
	IOMHWREVA	SCANPNTDTL
	IOMHWREVB	SCANRATE
	IOMLHFSTA	SELECTCABLEACMD
	IOMLHFSTB	SELECTCABLEBCMD
	IOMNUM	SO
	IOMOPERA	SOINITVAL
	IOMOPERB	SWAPPRIMARYCMD
	IOMSTATE	TYPEINVALIDA
	IOMSTSA	TYPEINVALIDB
	IOMSTSB	WARMSTRTA
	IOMTYPE	WARMSTRTB
	IOREDOPT	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the PMIO Blocks.	

4.13 Series C Input/Output (I/O) Blocks

Related topics

"AICHANNEL (Series C)" on page 332
"AOCHANNEL Series C" on page 333
"DICHANNEL Series C" on page 335
"DOCHANNEL Series C" on page 335
"AI-HART" on page 336
"AI-LLMUX" on page 337
"AI-LLAI" on page 337
"AO-HART" on page 339
"DI-HV" on page 339
"DI-24" on page 340
"DO-24B" on page 340

4.13.1 AICHANNEL (Series C)

Description	The AI channel block represents a single analog input point on one of the following Series C Processors: "AI-HART" on page 336; "AI-LLMUX" on page 337,"AI-LLAI" on page 337.
Function	The analog input channel converts an analog PV signal received from a field sensor to engineering units for use by other function blocks in the C300, and by the rest of Experion.
Inputs	Floating point value in engineering units.
Outputs	Floating point value in engineering units.
Associated Block	Prior to loading, block must be"associated" with 1 channel of the corresponding "AI-HART" on page 336 or "AI-LLMUX" on page 337 or"AI-LLAI" on page 337 block that interfaces with the physical AI hardware module at execution runtime.

Parameters	ACCEPTDEV ACCEPTRNG ALMENBSTATE ASSOCASSET BADPVFL CHANNUM CJTACT PJCOMMAND CONTAINEDIN DAMPING DECONF DEVICELOCATION DVRNGEXT EURNGEXT HARTVERSION HCFGDEV HCMD00 HCMD12 HCMD13 HCMD14 HCMD16 HCMD48BT[1..200] HCMD48NOTIFY[1..200] HDEVID HDEVMFG HDEVREV HDEVST HDEVSTSTATUS HDEVTYPE HDEVTYPENAME HDYNCC[1..4] HDYNEU[1..4] HDYNST[1..4] HENABLE	HEXTDEVST HLOCKBYPRIMARYMASTER HLOCKPERMANENT HLOCKSTATUS HMAINTREQ HMAXDEVVARS HNCFGCHG HNSMMINPRE HPVCHNFLAGS HPVMISM HSCANCFG HSCANOVR HSLOTCC[1..4] (HART Revision 6.0) HSLOTST [1..4] HSLOTVAL [1..4] HTAG HVARALERT INPTDIR IOP IOPTYPE JOURNALONLY LRL LRV PNTFORM PNTTYPE PTEXECST PV PVCHAR PVRAWHI PVRAWLO PVSTS REDTAG SENSRTYP URL URV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.2 AOCHANNEL Series C

Description	The AO channel block represents a single analog input point on the Series C "AO-HART" on page 339 Processor.

Function	The AO channel block converts the output value (OP) to a 4-20 mA output signal for operating final control elements such as valves and actuators in the field. The OP parameter value can be controlled from a Experion regulatory point, the operator, or an SCM.	
Inputs	OP value from • a single Regulatory Control block • an operator input • a program • an SCM block	
Outputs	Floating point value in engineering units	
Associated Block	Prior to loading, block must be "associated" with 1 channel of the corresponding "AO-HART" on page 339 block that interfaces with the physical AO hardware module at execution runtime.	
Parameters	ACCEPTDEV CHANNUM COMMFAILFL CONTAINEDIN DEVICELOCATION FAULTOPT FAULTVALUE HALARMENABLE HARTVERSION HCFGDEV HCMD00 HCMD12 HCMD13 HCMD14 HCMD16 HCMD48BT[1..200] HCMD48NOTIFY[1..200] HDEVREV HDEVST HDEVST HDEVSTSTATUS HDEVTYPE HDEVTYPENAME HDYNCC[1..4] HDYNEU[1..4] HDYNST[1..4] HENABLE HEXTDEVST	HLOCKBYPRIMARYMASTER HLOCKPERMANENT HLOCKSTATUS HMAINTREQ HMAXDEVVARS HNCFGCHG HNSMMINPRE HPVCHNFLAGS HSCANCFG HSCANOVR HSLOTCC[1..4] (HART Revision 6.0) HSLOTST[1..4] (HART Revision 6.0) HSLOTVAL [1..4] HTAG HVARALERT IOP IOPTYPE INITREQ INITVAL OP OPCHAR PNTFORM PNTTYPE PTEXECST REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.3 DICHANNEL Series C

Description	The DI channel block represents a single discrete input point on a Series C "DI-HV" on page 339, or "DI-24" on page 340 Processor.	
Function	The DI channel block converts a PVRAW signal received from the field to a PV that can be used by other data points in the Experion system.	
Inputs	Digital (PV) signals received from the field.	
Outputs	PV status value that can be used by other data points in system.	
Associated Blocks	Prior to loading, block must be "associated" with 1 channel of corresponding "DI-HV" on page 339 or "DI-24" on page 340 block that interfaces with the physical DI hardware module at execution runtime.	
Parameters	ALMOPT	IOP
	BADPVFL	IOPTYPE
	CHANNUM	PNTFORM
	CONTAINEDIN	PNTTYPE
	DEBOUNCE	PTEXECST
	DEVICELOCATION	PV
	DITYPE	PVSOURCE
	DLYTIME	PVSRCOPT
	EVTOPT	REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.4 DOCHANNEL Series C

Description	The DO channel block represents a single discrete input point on a Series C "DO-24B" on page 340 I/O Processor.
Function	The DO channel block provides a digital output to the field based on the origin of the input and the configures parameters
Inputs	SO, PO, ONPULSE, or OFFPULSE value from • a single Regulatory Control block • an operator input • a program • an SCM block
Outputs	Digital (Boolean) value or pulsed (real) value.
Associated Blocks	Prior to loading, block must be "associated" with 1 channel of corresponding "DO-24B" on page 340 block that interfaces with the physical DO hardware module at execution runtime.

Parameters	CHANNUM COMMFAILFL CONTAINEDIN DEVICELOCATION DOSTYPE DOTYPE FAULTOPT FAULTVALUE	INITREQ IOP IOPTYPE OP PNTFORM PNTTYPE PTEXECST REDTAG SO SOREADFAIL
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.5 AI-HART

Description	High Level Analog Input with HART, 16 Channel. This block is used for the following Series C IOMs: CU-PAIH01 CC-PAIH01 Each input channel is capable of scanning (100 ms sampling) a standard analog input (0 to 100%) and supporting digital data transfer using HART communications protocol. The analog input point converts an analog PV signal received from a field sensor to engineering units for use by other data points in the control strategy. To accomplish this function, the AI-HART point performs the following functions. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection Additionally, the IOM can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOM. Dynamic and device variable data and device status is collected from the device for use by the control system. IOM allows for servicing of any pass-through commands issued from host/master devices.	
Parameters	CALIBALL CALIBSTS CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB	IOMBTREVA IOMBTREVA IOMPLREVA IOMPLREVB REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the*Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.6 AI-LLMUX

Description	Low Level Mux Input, 64 Channel This block is used for the following Series C IOMs: CU-PAIM01 CC-PAIM01	
Parameters	CALIBALL CALIBSTS CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB	HCUAVAIL IOMBTREVA IOMBTREVA PVSTS REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.7 AI-LLAI

Description	Low Level Analog Input, 16 Channel, is designed to operate with the low voltage devices such as Thermocouples and RTDs. This block is used for the Series C IOMs CC-PAIM51.
Function	AI-LLAI channel accepts a new RTD input type CU:50 ohm (CU50rtd) in addition to all the inputs that are supported by the AI-LLMUX. In addition, the operating temperature range of each channel is extended from (0 to +60 deg C) to (-40 to +70 deg C).

Parameters	ACCEPTRNG BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM CONTAINEDIN CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB DESC DEVICELOCATION HCUAVAIL HPVMISM INPTDIR IOMBTREVA IOMBTREVA IOPTYPE LASTPV LOCUTOFF LRV LRL OWDENBL	PV PVAUTO PVAUTOST PVCALC PVCHAR PVCLAMP PVEUHI PVEULO PVEXECST PVEXEUHI PVEXHIFL PVEXEULO PVEXLOFL PVRAW PVRAWHI PVRAWLO PVSRCOPT PVSOURCE PVSTS PVTEMP SENSRTYP TF TCRNGOPT URL URV
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.8 AO-HART

Description	Analog Output with HART IOM, 16 channel. This block is used for the following Series C IOMs: CU-PAOH01 CC-PAOH01 Each channel is capable of supplying a standard analog output (4 to 20mA) and supporting digital data transfer using HART communications protocol. To accomplish this function, the AO-HART point performs the following functions. • Analog-to Digital Conversion • PV Characterization • Range Checking and PV Filtering • PV Source Selection • Alarm Detection The IOM can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOM. Dynamic and device variable data and device status is collected from the device for use by the control system. IOM allows for servicing of any pass-through commands issued from host/ master devices.	
Parameters	CALIBALL CALIBSTS CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB HAUTODET[1..16]	HCUAVAIL IOMBTREVA IOMBTREVA IOMPLREVA IOMPLREVB REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.9 DI-HV

Description	High Voltage Digital Input (IOM supports both 120 and 240 volts AC), 32 Channel This block is used for the following Series C IOMs: • CU-PDIH01 • CC-PDIH01	
Parameters	BADPVFL CPUFREEAVGA CPUFREEAVGB CPUFREEMINA	CPUFREEMINB IOMBTREVA IOMBTREVA REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.10 DI-24

Description	Low Voltage Digital Input (24 volts DC); 32 Channels This block is used for the following Series C IOMs: • CU-PDIL01 • CC-PDIL01	
Parameters	BADPVFL CPUFREEAVGA CPUFREEAVGB CPUFREEMINA	CPUFREEMINB IOMBTREVA IOMBTREVA REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.13.11 DO-24B

Description	Bussed Low Voltage Digital Output (24 volts DC); 32 Channels This block is used for the following Series C IOMs: CU-PDOB01 CC-PDOB01	
Parameters	CPUFREEAVGA CPUFREEAVGB CPUFREEMINA CPUFREEMINB IOMBTREVA IOMBTREVA REDTAG	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Series C I/O User's Guide* for more information on the Series C IO Blocks.	

4.14 Series C IEC 61850 Interface Module (850M) blocks

Related topics

"IEC61850M/IEC61850MSEC block" on page 341
"IEC61850LINK block" on page 341
"IED" on page 341
"Logical device block" on page 341
"Logical node block" on page 341

4.14.1 IEC61850M/IEC61850MSEC block

Refer to the *Series C IEC 61850 Interface Module User's Guide* for information of this block.

4.14.2 IEC61850LINK block

Refer to the *Series C IEC 61850 Interface Module User's Guide* for information of this block.

4.14.3 IED

Refer to the *Series C IEC 61850 Interface Module User's Guide* for information of this block.

4.14.4 Logical device block

Refer to the *Series C IEC 61850 Interface Module User's Guide* for information of this block.

4.14.5 Logical node block

Refer to the *Series C IEC 61850 Interface Module User's Guide* for information of this block.

4.15 Speed Protection Module (SPM)

Related topics

"Speed Protection Module (SPM) Block" on page 342
"SP_AI" on page 342
"SP_AO" on page 343
"SP_DI" on page 344
"SP_DO" on page 345
"SP_SPDVOTE" on page 346
"SP_SPEED" on page 348

4.15.1 Speed Protection Module (SPM) Block

Refer to the following documents for Turbine Control-related documentation for this block.

- *Turbine Control User's Guide*
- *Honeywell Turbine Control Solution Parameter Reference*

4.15.2 SP_AI

Description	The AI channel block represents a single analog input point on the Speed Protection (SP) Module.
Function	The analog input channel converts an analog PV signal received from a field sensor to engineering units for use by other function blocks in the C300 - 20msCEE Controller, and by the rest of Experion PKS. To accomplish this function, the AI channel performs the following operation on the analog PV signal. • Analog-to digital conversion • PV characterization • Range Checking and PV filtering • PV source selection
Inputs	Floating point value in engineering units.
Outputs	Floating point value in engineering units.
Associated Block	Prior to loading, block must be "associated" with 1 channel of the corresponding SP module that interfaces with the physical SP hardware module at execution runtime.

Parameters	BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM CONTAINEDIN DESC DEVICELOCATION INPTDIR IOP IOPTYPE LASTPV LOCUTOFF NAME OWDENBL PNTTYPE PTEXECST PV	PVAUTO PVAUTOSTS PVCALC PVCHAR PVCLAMP PVEUHI PVEULO PVEXHIFL PVEXLOFL PVEXEUHI PVEXEULO PVRAW PVSOURCE PVSRCOPT PVSTS SENSRTYP TF
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.15.3 SP_AO

Description	The AO channel block represents a single analog output point on the Speed Protection (SP) Module.
Function	The AO channel block converts the output value (OP) to a 4-20 mA output signal for operating control elements such as valves and actuators in the field. To convert the OP value to a 4-20 mA signal, the AO channel performs the following functions. • Direct/Reverse Output Function • Linear or non-linear Output Characterization In addition, the SP_ AO channel supports input connections from SP_SPEED, SP_AI, and SP_SPDVOTE channels within the same SPM IOM. This can be configured by connecting the OP parameter to one of the following parameters. • VOTPVx of voting logic channel, where x can be 1 or 2 • PV of any SP_SPEED channel • PV of any SP_AI channel The default values for MODE and MODEATTR parameters of SP_AO channel are CAS and PROGRAM, respectively.
Inputs	OP value from the • a single Regulatory Control block • an operator input • SP_AI channel • SP_SPDVOTE channel • SP_SPEED channel

Outputs	Floating point value in engineering units.	
Associated Block	Prior to loading, block must be “associated” with 1 channel of the corresponding SP module that interfaces with the physical SP hardware module at execution runtime.	
Parameters	BLCKCOMMENT1	NMODATTR
	BLCKCOMMENT2	NMODE
	BLCKCOMMENT3	OP
	BLCKCOMMENT4	OPCHAR
	CHANNUM	OPFINAL
	COMMFAILFL	OPIN0
	CONTAINEDIN	OPIN1
	DESC	OPIN2
	DEVICELOCATION	OPIN3
	FAULTOPT	OPIN4
	FAULTVALUE	OPIN5
	INITREQ	OPOUT0
	INITREQLATCH	OPOUT1
	INITVAL	OPOUT2
	IOP	OPOUT3
	IOPTYPE	OPOUT4
	MODE	OPOUT5
	MODEATTR	OPTDIR
	MODEPERM	PNTTYPE
	NAME	PTEXECST
		REDTAG
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.15.4 SP_DI

Description	The DI channel block represents a single digital input point on the Speed Protection (SP) Module.
Function	This digital input channel converts a digital PVRAW signal received from the field to a PV that can be used by other data points in the Experion PKS.
Inputs	Digital (PV) signals received from the field.
Outputs	PV value that can be used by other data points in system.
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding SP module that interfaces with the physical SP hardware module at execution runtime.

Parameters	BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM CONTAINEDIN DESC DEVICELOCATION DITYPE NAME INPTDIR	IOP IOPTYPE OWDENBL PNTTYPE PTEXECST PV PV.FLWRST PVAUTO PVRAW PVSOURCE PVSRCOPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.15.5 SP_DO

Description	The SP_DO channel block represents a single discrete output point on the Speed Protection (SP) Module.
Function	The SP_DO channel provides a digital output to the field based on the origin of the input and the configured parameters. This channel supports upto 8 interlock inputs that can be used to trip the output. The interlock input source can be one of following flags. • PV and PV.FLWRST of any DI Channel block of the same SP IOM only. • XXXX. FL, where XXXX can be any one of VOTPVxHIALM, VOTROCxPOSHIALM, VOTPVxHHALM or VOTROCxPOSHHALM where x=1,2 of the SP_SPDVOTE channel of the same SPM IOM. • XXXX. FLWRST, where XXXX can be any one of VOTPVxHHALM or VOTROCxPOSHHALM where x=1,2 of the SP_SPDVOTE channel of the same SP IOM. • YYYY. FL, where YYYY can be any one of PVHIALM, ROCPOSHIALM, PVHHALM or ROCPOSHHALM of the SP_SPEED channel of the same SP IOM. • YYYY. FLWRST, where YYYY can be any one of PVHHALM or ROCPOSHHALM of the SP_SPEED channel of the same SP IOM.
Inputs	SO, or ONPULSE value from • a single Device Control block • an operator input • a program
Outputs	Digital (Boolean) value or pulsed (real) value.
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding SP module that interfaces with the physical SP hardware module at execution runtime.

Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM COMMFAILFL CONTAINEDIN DESC DEVICELOCATION DOTYPE FAULTOPT FALUTVALUE I1 I1INPTDIR I1STS I2 I2INPTDIR I2STS I3 I3INPTDIR I3STS I4 I4INPTDIR I4STS I5 I5INPTDIR I5STS I6 I6INPTDIR I6STS	I7 I7INPTDIR I7STS I8 I8INPTDIR I8STS INLCKINPTSRC [1..8] INITREQ INITREQLATCH IOP IOPTYPE LASTTRIPREASON LASTTRIPTIME MODE MODEATTR MODEPERM NMODATTR NMODE ONPULSE OP OPINITVAL OPTDIR PERIOD PNTTYPE PTEXECST READY REDTAG SO SOINITVAL SOREADFAIL STDBYMAN
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.15.6 SP_SPDVOTE

Description	The SP_SPDVOTE channel accepts four inputs from the Speed Channel.
Function	The SP_SPDVOTE channel computes the Voted PVs and Voted ROCs, and supports alarm flags for over speed and over acceleration. The Voted PV and Voted ROC can be connected as an input to C300 function blocks or to the OP parameter of local SP_AO channel. Alarm flags are available as output pins for use as interlock input parameters of the SP_DO channel.
Inputs	Accepts four inputs (PVs) from the SP_SPEED channels of the parent IOM.

Outputs	Voted PV and Voted ROC values.	
Associated Block	Prior to loading, block must be“associated” with 1 channel of the corresponding SP module that interfaces with the physical SP hardware module at execution runtime.	
Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM CONTAINEDIN DESC DEVICELOCATION GRP1IGNORD GRP2IGNORD GRP1IGNORDFL GRP2IGNORDFL GRP1NMIN GRP2NMIN GRP1VOTCHENB [1..4] GRP2VOTCHENB [1..4] NAME IOP IOPTYPE LASTTRIPREASON LASTTRIPTIME LASTVOTPV1 LASTVOTPV2 LASTVOTROC1 LASTVOTROC2 MEDOPT1 MEDOPT2 PNTTYPE PTEXECST PV1 PV2 PV3 PV4 VOTALG1 VOTALG2	VOTPV1 VOTPV2 VOTPV1HHALM.FL VOTPV2HHALM.FL VOTPV1HHALM.FLWRST VOTPV2HHALM.FLWRST VOTPV1HHALM.TP VOTPV2HHALM.TP VOTPV1HIALM.FL VOTPV2HIALM.FL VOTPV1HIALM.TP VOTPV2HIALM.TP VOTPV1OVRTSTENB VOTPV2OVRTSTENB VOTPV1STS VOTPV2STS VOTPVMAX VOTPVONTRIP VOTROC1 VOTROC2 VOTROC1POSHHALM.FL VOTROC2POSHHALM.FL VOTROC1POSHHALM.FLWRST VOTROC2POSHHALM.FLWRST VOTROC1POSHHALM.TP VOTROC2POSHHALM.TP VOTROC1POSHIALM.FL VOTROC2POSHIALM.FL VOTROC1POSHIALM.TP VOTROC2POSHIALM.TP VOTROCPOSHHONTRIP VOTROC1STS VOTROC2STS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.15.7 SP_SPEED

Description	The SP_SPEED channel accepts a pulse input on the Speed Protection (SP) Module.
Function	The SP_SPEED channel converts pulse signals received from a field sensor to a PV value in RPM. The PV value is used by the SPM and other function blocks in C300 - 20msCEE Controller. Additionally, the channel also computes ROC of speed. In addition, with R410, SP_SPEED channel supports flow measurement. To accomplish this, you need to configure the MEASUREMENTTYPE parameter as “Flow_Measurement.” The SP_SPEED channel measures the flow in Engineering Unit (EU). SP_Speed channel performs the following functions: • PV computation and Diagnostics • PV source selection When the SP_SPEED channel is configured for Flow_Measurement, the following options are not supported in the SP_SPEED channel. • Connection with SP_AO channel • Connection with voting logic channel (SP_SPDVOTE) • GEARRATIO parameter • NMSPEED parameter • Rate Of Change (ROC) parameters – LASTROCPOSHHDRNMSPD – ROCPOSHHALM.FL – ROCPOSHHALM.FLWRST – ROCPOSHHALM.TP – ROCPOSHIALM.FL – ROCPOSHIALM.TP – ROCPV
Inputs	Floating point value in engineering units.
Outputs	Floating point value in engineering units.
Associated Block	Prior to loading, block must be “associated” with 1 channel of the corresponding SP module that interfaces with the physical SP hardware module at execution runtime.

Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 BADPVFL CHANNUM CONTAINEDIN DEFMPULFL DESC DEVICELOCATION EDGEDETECT GEARRATIO IOP IOPTYPE LASTPV LASTROCPOSHHDRNMSPD KFACTOR ORDEROFKFACTOR MEASUREMENTTYPE MISGPULFL NAME NMSPEED	PNTTYPE PV PVAUTO PVAUTOSTS PVCALC PTEXECST PVHHALM.FL PVHHALM.FLWRST PVHHALM.TP PVHIALM.FL PVHIALM.TP PVRAW PVSOURCE PVSRCOPT PVSTS ROCPOSHHALM.FL ROCPOSHHALM.FLWRST ROCPOSHHALM.TP ROCPOSHIALM.FL ROCPOSHIALM.TP ROCPV SENSRTYP TOOTHCNT ZEROSPDFL
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.16 Servo Valve Positioner (SVP) Module

Related topics

"Servo Positioner Valve Module (SVPM) Block" on page 350
"SVP_AI" on page 350
"SVP_AO" on page 351
"SVP_DI" on page 352
"SVP_REGCTL" on page 353

4.16.1 Servo Positioner Valve Module (SVPM) Block

Refer to the following documents for Turbine Control-related documentation for this block.

- *Turbine Control User's Guide*
- *Honeywell Turbine Control Solution Parameter Reference*

4.16.2 SVP_AI

Description	The SVP_AI accepts a single analog input or LVDT/RVDT/ Resolver inputs on the Servo Valve Positioner (SVP) Module.
Function	The analog input channel converts an analog PV signal received from a field sensor or LVDT/RVDT to engineering units for use by other function blocks in the C300 - 20msCEE Controller, and by the rest of Experion PKS. With R410, SVP_AI channel supports angular measurement using the Resolver. To accomplish this, you need to configure the SENSRTYP parameter as "Resolver." To accomplish this function, the AI channel performs the following operation on the analog PV signal. • Analog-to-digital conversion • PV characterization • Range Checking and PV filtering • PV source selection When the SENSRTYP parameter is configured as "Resolver," the SVP_AI channel does not support the following parameters. • LVDTCOREFALLOUT • OWDENBL • TF • LOCUTOFF • INPTDIR • PVCHAR • PVCLAMP • PVEUHI • PVEULO • PVEXHIFL • PVEXLOFL • PVEXEUHI • PVEXEULO
Inputs	Floating point values in engineering units.

Outputs	Floating point values in engineering units.	
Associated Block	Prior to loading, block must be “associated” with 1 channel of the corresponding SVP module that interfaces with the physical SVP hardware module at execution runtime.	
Parameters	ACTUALANGLE ANGLEOFFSET APPLYOFFSET BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CALIBVAL CHANNUM CONTAINEDIN DESC DEVICELOCATION EXCITNAMPFL EXCITNFBAFL EXCITNFBBFL EXCITNFREQDRIFTFL EXCITNVLTG FBINPUTSFL INPTDIR IOP IOPTYPE LASTPV LOCUTOFF LVDTCOREFALLOUTFL NAME	OWDENBL PNTTYPE PTEXECST PV PVAUTO PVAUTOSTS PVCALC PVCHAR PVCLAMP PVEUHI PVEULO PVEXHIFL PVEXLOFL PVEXEUHI PVEXEULO PVRAW PVSOURCE PVSRCOPT PVSTS SENSRTYP TF UNSTABLEINPUTFL VALVECALIBSTS VALVECALIBENB VDTMODE XMTRWIRESLCT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.16.3 SVP_AO

Description	Two AO channels are present in the Servo Valve Positioner (SVP) Module. The AO channel supports unipolar and bipolar current output in addition to standard 4-20 mA analog output.
Function	The AO channel converts the output value (OP) to an output signal for operating final control elements such as valves and actuators in the field. To convert the OP value to a configured signal value, the AO channel performs: • Direct/Reverse Output Function • Linear or nonlinear Output Characterization
Inputs	Accepts values from AUXILIARY function blocks executing in C300 - 20msCEE Controller or OP value from a local SVP_REGCTL block.

Outputs	Output signals are connected to servo valves or external servo valve positioner.	
Associated Block	Prior to loading, block must be “associated” with 1 channel of the corresponding SVP module that interfaces with the physical SVP hardware module at execution runtime.	
Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNUM COMMFAILFL CONTAINEDIN DESC DEVICELOCATION DITHERAMPL DITHERFREQ FAULTOPT FAULTVALUE INITREQ INITREQLATCH INITVAL INTRLOCKFAILOPT1 INTRLOCKFAILOPT2 IOP IOPTYPE MODE MODEATTR MODEPERM NAME NMODE NMODEATTR OP OPACTION	OPBIASCUR OPCHAR OPFINAL OPTDIR OPHICURRENT OPIN0 OPIN1 OPIN2 OPIN3 OPIN4 OPIN5 OPLOCURRENT OPOUT0 OPOUT1 OPOUT2 OPOUT3 OPOUT4 OPOUT5 OPTYPE PNTTYPE PRCSAFEOP1 PRCSAFEOP2 PRCSINTRLOCK1 PRCSINTRLOCK2 PTEXECST REDTAG STROKENB SECDIAGCURR
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.16.4 SVP_DI

Description	The DI channel block represents a single discrete input point on the Servo Valve Positioner (SVP) Module.
Function	The DI channel block converts a PVRAW signal received from the field to a PV that can be used by other data points in the Experion system.
Inputs	Digital (PV) signals received from the field.
Outputs	PV status value that can be used by other data points in system.
Associated Block	Prior to loading, block must be“associated” with 1 channel of the corresponding SVP module that interfaces with the physical SVP hardware module at execution runtime.

Parameters	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 BADPVFL CHANNUM CONTAINEDIN DESC DEVICELOCATION DITYPE INPTDIR IOP	IOPTYPE OWDENBL PNTTYPE PTEXECST PV PVAUTO PVRAW PVSOURCE PVSRCOPT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.16.5 SVP_REGCTL

Description	The SVP_REGCTL channel operates as Proportional Integral Derivative (PID) controller and supports the ideal form of calculating the PID terms.
Function	Accepts 2 analog inputs by default - process variable (PV1) and set point (SP); produces output calculated to reduce the difference between PV1 and SP. Provides anti-windup protection, and control initialization. Note: SVP_REGCTL supports an optional analog input PV2, which can be used as an alternate to PV1 based on the PV1STS. The SVP_Regctl block functionality is similar to C300 PID with reduced features. SVP_Regctl block performs the following functions. 1. Input Processing **PV Processing:** PV processing fetches the input value, status and range from the configured AI channels and updates appropriate PV parameters. **SP Processing:** SP processing is performed to execute SP limit checking. 2. Mode Processing Mode processing identifies the source of stores which may be accepted by SVP_REGCTL on SP and OP parameters. 3. Initial Control Processing This function verifies if a SVP_REGCTL cascade strategy has been broken, when the SVP_REGCTL blocks are in a cascade strategy. If the cascade strategy is broken, this function initializes the blocks, and builds an initialization request for its primary. 4. Algorithm Calculation SVP_REGCTL blocks only support Equation A (incremental algorithm) and E (full value algorithm). 5. Output Processing This function derives the control output (OP) from the algorithm's calculated variable (CV). 6. Feedback Propagation This function drives the ARWSTS parameter to provide the information to the upstream block in the cascade mode to stop the integral calculation.

Inputs	Accepts PV as input from any of the local SVP_AI channels.
Outputs	OP value can be connected only to the SVP_AO channels, with OPACTION configured as incremental. Note: Use of control equation A or E is independent of OPACTION parameter of SVP_AO. Block has following initializable outputs: • OP = Calculated output in percent. • OPEU = Calculated output in engineering units that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/ OPEUX) connection when required.
Associated Block	Prior to loading, block must be“associated” with 1 channel of the corresponding SVP module that interfaces with the physical SVP hardware module at execution runtime.

Parameters	ARWNET	OPEXLOLM
	ARWOP	OPEXHIFL
	BLCKCOMMENT1	OPEXLOFL
	BLCKCOMMENT2	OPHIFL
	BLCKCOMMENT3	OPHILM
	BLCKCOMMENT4	OPLOFL
	BADCTLFL	OPLOLM
	BADCTLOPT	OPMINCHG
	BADPVFL	OPTOL
	CHANNUM	OUTIND
	CONTAINDEIN	PNTTYPE
	CTLACTN	PREFPVSRC
	CTLEQN	PRIMDATA.ARWSTS
	CTLSTATE	PRIMDATA.INITSTS
	CV	PRIMDATA.INITVAL
	CVEUHI	PTEXECST
	CVEULO	PV
	DESC	PV1
	DEVICELOCATION	PV1STS
	EQNEUNITSOPT	PV2
	EUDESC	PV2STS
	GAINHILM	PVEUHI
	GAINLOLM	PVEULO
	GAINOPT	PVSTS
	INITMAN	PVSTSFL.BAD
	INITREQ	PVSTSFL.NORM
	INITVAL	PVTRAKOPT
	IOP	PVTRAKOPTAI
	IOPTYPE	REDTAG
	K	SECDATA
	LASTGOODPV	SECDATA.ARWSTS
	MODE	SECDATA.INITSTS
	MODEATTR	SECDATA.INITVAL
	MODEATTRFL.NORM	SP
	MODEATTRFL.OPER	SPHIFL
	MODEATTRFL.PROG	SPHILM
	MODEFL.CAS	SPLOFL
	MODEFL.MAN	SPLOLM
	MODEFL.NORM	SPTOL
	MODEPERM	T1
	NMODE	T1HILM
	NMODATTR	T1LOLM

	OP	T2
	OPBIAS	T2HILM
	OPBIAS.FIX	T2LOLM
	OPBIAS.RATE	TMOUTFL
	OPEU	TMOUTMODE
	OPEXHILM	TMOUTTIME
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.17 Universal Input/Output (UIO) Module

Related topics

"UIO module" on page 357

4.17.1 UIO module

<table>
<tr><td>Description</td><td>UIO module supports 32 input/output channels. These channel types can be configured as one of the following:
• analog input channels
• analog output channels
• digital input channels
• digital output channels</td></tr>
<tr><td>Function</td><td>The UIO channel supports multiple functionalities. It can function as an analog input/output channel and/or a digital input/output channel at the same time. The function of the UIO channel is identical to the existing analog input/output channels and/or the digital input/output channels depending on the configured channel type. In addition, it monitors the temperature of the module.
With R430, UIO module supports the following functions.
• DO channel ganging
• Pulse counting using DI channel</td></tr>
<tr><td>Inputs</td><td>The input to the UIO channel varies depending on the channel configuration. The following table defines the inputs for each channel type.
<table>
<tr><th>Channel Type</th><th>Input</th></tr>
<tr><td>Analog input channel</td><td>Floating point value in engineering units</td></tr>
<tr><td>Analog output channel</td><td>OP value from
• a single Regulatory Control block
• an operator input
• a program
• an SCM block</td></tr>
<tr><td>Digital input channel</td><td>Digital (PV) signals received from the field</td></tr>
<tr><td>Digital output channel</td><td>SO, PO, ONPULSE, or OFFPULSE value from
• a single Regulatory Control block
• an operator input
• a program
• an SCM block</td></tr>
</table></td></tr>
<tr><td>Outputs</td><td>The output of the UIO channel varies depending on the channel configuration. The following table defines the outputs for each channel type.
<table>
<tr><th>Channel Type</th><th>Output</th></tr>
<tr><td>Analog input channel</td><td>Floating point value in engineering units.</td></tr>
<tr><td>Analog output channel</td><td>Floating point value in engineering units.</td></tr>
<tr><td>Digital input channel</td><td>PV status value that can be used by other data points in system.</td></tr>
</table></td></tr>
</table>

	Digital output channel	Digital (Boolean) value or pulsed (real) value.
Associated Block	Prior to loading UIO block, the block must be “associated” with one channel of corresponding UIO block that interfaces with the physical UIO hardware module at execution runtime.	

Parameters	AICHNLNAME	IOMSTATE
	AIPTEXECST	IOMSTSA
	AOCHNLNAME	IOMSTSB
	AOPTEXECST	IOMTYPE
	ASSOCASSET	IOPDESCA
	BADPVFL	IOPDESCB
	BLCKCOMMENT1	IOPLOCATION
	BLCKCOMMENT2	IOREDOPT
	BLCKCOMMENT3	LRL
	BLCKCOMMENT4	LRV
	CHNLNAME	NAME
	CPUFREEAVGA	MODIFIEDBY
	CPUFREEAVGB	NUMCHANS
	CPUFREEEMINA	NUMSIGS
	CPUFREEMINB	OP
	CREATEDBY	OPFINAL
	CTRLCONFIRM	OPTINITVAL
	DATECREATED	PARTNERINCOMPATIBLEA
	DBVALID	PARTNERINCOMPATIBLEB
	DESC	PNTTYPE
	DICHNLNAME	PRIMARYSIG
	DIPTEXECST	PV
	DOCHNLNAME	PVFL
	DOOP	PVSTS
	DOPTEXECST	RDNAUTOSYNC
	FWINVALIDA	REASONSET
	FWINVALIDB	REDDATAA
	GROUP.NUMPARAMS	REDDATAB
	HCUAVAIL	SCANCTRLLVL
	HDEVID	SCANRATE
	HDEVMFG	SCANASSOCDSP
	HDEVREV	SCANGRPDTL
	HDEVTYPE7	SCANPNTDTL
	HDEVTYPENAME	SECONDARYSIG
	HENABLE	SECSIGSECLVL
	HIST.NUMPARAMS	SERIALNUMA
	HSCANCFG	SERIALNUMB
	HSCANOVR	SO
	HTAG	SOINITVAL
	INITVAL	SOREADFAIL
	IOLINK	STATRESETA
	IOLINKCOLOR	TEMPCURA

	IOMBTREVA IOMBTREVB IOMCOMMAND IOMFWREVA IOMFWREVB IOMHWREVA IOMHWREVB IOMLHFSTA IOMLHFSTB IOMNUM IOMOPERA IOMOPERB IOMPLD1REVA IOMPLD1REVB IOMPLD2REVA IOMPLD2REVB	TEMPCURB TEMPHILM TEMPLOLM TEMPMAXA TEMPMAXB TEMPMINA TEMPMINB TREND.NUMPARAMS TYPEINVALIDA TYPEINVALIDB URL URV VERSIONDATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Control Builder Components Theory* for more information on the UIO Blocks.	

4.18 Peer Control Data Interface (PCDI) Blocks

Related topics

"PCDI_MASTER (Peer Control Data Interface Master device) Block" on page 361
"PCDIFLAGARRCH (Peer Control Data Interface Flag Array Channel) Block" on page 364
"PCDINUMARRCH (Peer Control Data Interface Numeric Array Channel) Block" on page 365
"PCDITEXTARRCH (Peer Control Data Interface Text Array Channel) Block" on page 366

4.18.1 PCDI_MASTER (Peer Control Data Interface Master device) Block

Description	Identifies the physical Safety Manager or Modbus TCP end device or gateway for the C300 to provide links to associated Array Request Channel blocks and provides selected diagnostic events for associated channels.
Function	Provides configuration and communication software to enable devices to communicate over Honeywell's Fault Tolerant Ethernet media to perform bi-directional data exchange directly with the *Experion C300 Controller*. The request for the data in Honeywell's Safety Manager or Modbus end device, whether it is a native Modbus TCP device or a Modbus RTU device connected to the serial bus of a gateway, is generated by a PCDI Array Request Channel function block. These blocks will be instantiated in a Control Module (CM).
Inputs	Real-time data transmission from configured peer device
Outputs	Real-time data transmission to physical device.

Parameters		
	ALMENBSTATE AVGRCVBYTESPERSEC AVGRCVMSGPERSEC AVGSMITBYTESPERSEC AVGSMITBYTESPERSEC AVGXMITMSGPERSEC BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANFBNAME[0..63] CHANLASTMBERR[0..63] CHANLASTMBERRTIME[0..63] CHANMBERRCNT[0..63] CHANMBERRFL CHANREQRSPRCVD[0..63] CHANREQSENT[0..63] CHANRSTSTATS CHANSTS[0..63] CONINUSE CONINUSEMON CONNSTS[0..1] CONTOUSE CREATEDBY CTRLCONFIRM CYCLETIME[0..63] DATECREATED DEFTIMOUT DESC DEVSTS DEVSUPCMDS[0..15][0..39] DEVTYPE DIAGREQRSPRCVD[0-15] DIAGREQSENT[0..15]	NUMDISCONN[0..1] NUMSIGS NUMUIDS ORPHANRESPCNT PREFERREDCONN PRIMARY PRIMARYSIG PRIMCONNSTS PRIMERRCNT PRIMERRCODE PRIMERRFL PRIMERRINFO PRIMIP PRIMIPMON PRIMLASTCONNTIME PRIMLASTDISCONNTIME PRIMLASTERRTIME PRIMLOOPDATA PRIMNUMCONN PRIMNUMCONNATMPT PRIMNUMDISCONN PRIMNUMDISCONNTIME PRIMTCP PRIMTCPMON QUEUEDREQCNT[0-15] REASONSET REDSWITCHPERIOD REQRSPRCVD REQRTRY REQRTRYCNT[0-15] RSTERRCNT RSTSTATS SCANASSOCDSP SCANCTRLLVL SCANPNTDTL SECCONNSTS SECERRCNT SECERRCODE SECERRFL SECERRINFO SECIP SECIPMON

	ENABLEDEVICE[0..15]	SECLASTCONNTIME
	ENITITYNAME	SECLASTDISCCONNTIME
	ERRCNT[0..1]	SECLASTERRTIME
	ERRCODE[0..1]	SECLOOPDATA
	ERRFL[0..1]	SECNUMCONN
	ERRINFO[0..1]	SECNUMCONNATMPT
	GROUP.NUMPARAMS	SECNUMDISCONN
	HIST.NUMPARAMS	SECONDARYSIG
	INALM	SECSIGSECLVL
	IOMSTATE	SECTCP
	LASTCONNTIME[0..1]	SECTCPMON
	LASTDISCONNTIME[0..1]	STATUS
	LASTERRTIME[0..1]	STUBIOM
	LASTMBERR[0..15]	TIMOUT[0..15]
	LASTMBERRTIME[0.. 15]	TIMOUTCNT[0-15]
	LOOPADDR	TOTALREQSENT
	LOOPDATA	TREND.NUMPARAMS
	LOOPDATACHG	UIDDEVTYPE[0..15]
	LOOPMODE	UIDDEVTYPEMON[0..15]
	LOOPRATE	UIDLOOPADDR[0..15]
	LOOPTYPE	UIDLOOPDATA[0..15]
	MASTERID	UIDLOOPDATACHG
	MASTERSTATE	UIDLOOPMODE[0..15]
	MAXPENDREQ	UIDLOOPRATE[0..15]
	MAXRCVBYTESPERS EC	UIDLOOPTYPE[0..15]
	MAXRCVMSGPERSE C	UIDORPHANRESPCNT[0-15]
	MAXUIDREQ	UIDREQRSPRCVD[0-15]
	MAXXMITBYTESPER SEC	UIDREQSENT[0..15]
	MAXXMITMSGPERS EC	UNITID[0..15]
	MBERRCNT [0..15]	UNITIDMON[0..15]
	MBERRFL[0..15]	USEKEEPALIVE
	MODIFIEDBY	VENDOR[0..15]
	MSGDELAY	VERSIONDATE
	NAME	
	NUMCHANS	
	NUMCONN[0..1]	
	NUMCONNATMPT[0.. 1]	

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Peer Control Data Interface Reference* for information about configuring and using the PCDI_MASTER block.

4.18.2 PCDIFLAGARRCH (Peer Control Data Interface Flag Array Channel) Block

Description	Provides a read/write interface to a Boolean array of data from Honeywell's Safety Manager, Modbus TCP native device or a serial device through a Modbus TCP Gateway/Bridge.	
Function	• Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block. • Supports up to 2000 Boolean values (PVFL[1..2000] from the device. • The start index along with the read or writes defines the function code for the request. The bit types represent read and write forms depending on the address range as follows: Provides access to the array of data by other blocks - one element at a time or whole array access. – 000001-065535: read or write from 1 to 2000 bit flags. – 100001-165535: read only from 1 to 2000 bit flags. • Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRCODE). • Provides bad PV flag (BADPVFL).	
Inputs	Boolean value from device or another block	
Outputs	Boolean value	
Parameters	ACCLOCK AUTOTRIGGER BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNAME CHANSTS DEVADDR DONEFL ERRCODE ERRFL IOCNUMBER IOCSTATE LASTERRTIME MASTERID	NAME NFLAG ORDERINCM PVFL PVSTS RDYFL REQSTATE SENDFL SIMMODE SIMVALUE[1..1968] STARTINDEX SUBVAL SUBVALTYPE USERSYMNAME WRITEOPT
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding PCDI_MASTER block that interfaces with physical peer device at execution runtime. For optimum performance, assign channels to PCDI_MASTER block for given device contiguously. For example, if you have four PCDIFLAGARRCH blocks to use with the device, assign them to PCDI_MASTER block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.	

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Peer Control Data Interface Reference* for information about configuring and using the PCDI_MASTER block.

4.18.3 PCDINUMARRCH (Peer Control Data Interface Numeric Array Channel) Block

Description	Provides a read/write interface to a Numeric array of data from Modbus TCP native device or a serial device through a Modbus TCP Gateway /Bridge.
Function	• Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block. • The start index along with the read or writes defines the function code for the request. The floating point data types represent all word and byte swapped forms depending on the address range as follows: – 200001-265535: read or write from 1 to 60 contiguous Word Swapped IEEE floating point numbers. – 300001-365535: read from 1 to 120contiguous Signed Integer numbers. This is read only addresses. Read/Write Option setting is ignored. – 400001-465535: read from 1 to 120 or write a single Signed Integer numbers. – 700001-765535: read or write from 1 to 60 contiguous IEEE floating point numbers. – 800002-865535: read or write from 1 to 60 contiguous IEEE floating point numbers Addr-1. – 900001-965535: read from 1 to 120 or write a single Unsigned Integer numbers. • Provides access to the array of data by other blocks - one element at a time or whole array access. • Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRCODE). • Provides bad PV flag (BADPVFL).
Inputs	Up to 248 bytes of Real, Integer, or Byte type data from the device.
Outputs	See above.

Parameters	ACCLOCK AUTOTRIGGER BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNAME CHANSTS CONVTOLONG DEVADDR DONEFL ERRCODE ERRFL EUDESC IOCNUMBER IOCSTATE LASTERRTIME MASTERID NAME	NNUMERIC ORDERINCM PV PVEUHI PVEULO PVRAW PVRAWHI PVRAWLO PVSTS RDYFL REQSTATE SENDFL SIMMODE SIMVALUE[1..120] STARTINDEX SUBVAL SUBVALTYPE USERSYMNAME WRITEOPT
Associated Block	Prior to loading, block must be “associated” with 1 channel of corresponding PCDI_MASTER block that interfaces with physical peer device at execution runtime. For optimum performance, assign channels to PCDI_MASTER block for given device contiguously. For example, if you have four PCDINUMARRCH blocks to use with the device, assign them to PCDI_MASTER block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Peer Control Data Interface Reference* for information about configuring and using the PCDINUMARRCH block.	

4.18.4 PCDITEXTARRCH (Peer Control Data Interface Text Array Channel) Block

Description	Provides a read/write interface to a Text (or String) array of data from Modbus TCP native device or a serial device through a Modbus TCP Gateway /Bridge.

Function	• Reads data from the connected block and writes data to the associated field device. Or, reads data from the associated field device and makes it available to the connected block. • Supports up to 16 Text values (STR[1..16]) from the device. Since the maximum size of the interface to the device is 128 bytes, the valid range of values depends on the combination of number of string values (NSTRING) and length of string values (STRLEN) as follows. – If NSTRING is 1 and STRLEN is 128, valid STR[1..16] range is 1. – If NSTRING is 2 and STRLEN is 64, valid STR[1..16] range is 1 to 2. – If NSTRING is 4 and STRLEN is 32, valid STR[1..16] range is 1 to 4. – If NSTRING is 8 and STRLEN is 16, valid STR[1..16] range is 1 to 8. – If NSTRING is 16 and STRLEN is 8, valid STR[1..16] range is 1 to 16. • The start index along with the read or writes defines the function code for the request. The ASCII encoded bytes represent read and write forms depending on the address range as follows: – 500001-565535: read or write from 16 of 8, 8 of 16, 4 of 32, 2 of 164, or 1 of 128 ASCII encoded bytes. – 600000: Loopback test – 600001-665535: write vendor information. – 699999: read vendor information • Provides access to the array of data by other blocks - one element at a time or whole array access. • Sets an overall error flag (ERRFL) ON when the array data is invalid and generates a detailed error code (ERRCODE). • Provides bad PV flag (BADPVFL).	
Inputs	Up to 8 string values depending on whether the length of the string is 8, 16, 32, 64, or 128 characters.	
Outputs	See above.	
Parameters	ACCLOCK AUTOTRIGGER BADPVFL BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4 CHANNAME CHANSTS CONVTOASCII DEVADDR DONEFL ERRCODE ERRFL IOCNUMBER IOCSTATE LASTERRTIME	MASTERID NSTRING ORDERINCM RDYFL REQSTATE SENDFL SIMMODE SIMVALUE[1..16] STARTINDEX STR STRLEN SUBVAL SUBVALTYPE USERSYMNAME WRITEOPT
Associated Block	Prior to loading, block must be “associated"” with 1 channel of corresponding PCDI_MASTER block that interfaces with physical peer device at execution runtime. For optimum performance, assign channels to PCDI_MASTER block for given device contiguously. For example, if you have four PCDITEXTARRCH blocks to use with the device, assign them to PCDI_MASTER block channels 0, 1, 2, and 3 rather than 0, 2, 4, and 6.	

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter. Refer to the *Peer Control Data Interface Reference* for information about configuring and using the. PCDITEXTARRCH block.

4.19 PROFIBUS Gateway Module (PGM) Blocks

Related topics

"PROFIBUS Module Gateway (PGM) Block" on page 369
"PROFIBUS Gateway Module (PGM) Secondary Block" on page 369
"Protocol Block (PBLINK)" on page 369
"Device Support Block (DSB)" on page 369
"PROFIBUS HART Input/Output Module (PBHIOM) Function Block" on page 369
"PROFIBUS Input/Output Module (PIOM) Block" on page 370
"PROFIBUS Interface (PBUSIF) Library" on page 31
"PBAICHANNEL" on page 370
"PBAOCHANNEL" on page 370
"PBDICHANNEL" on page 370
"PBDOCHANNEL" on page 370

4.19.1 PROFIBUS Module Gateway (PGM) Block

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.2 PROFIBUS Gateway Module (PGM) Secondary Block

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.3 Protocol Block (PBLINK)

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.4 Device Support Block (DSB)

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.5 PROFIBUS HART Input/Output Module (PBHIOM) Function Block

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.5.1 PROFIBUS HART I/O channel

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.6 PROFIBUS Input/Output Module (PIOM) Block

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.7 PROFIBUS Interface (PBUSIF) Library

The PROFIBUS Interface Library includes the interface blocks associated with linking PROFIBUS devices with the *Experion* system through the interface modules. Refer to the *PROFIBUS Interface Implementation Guide* for complete details about the PROFIBUS Interface components.

4.19.8 PBAICHANNEL

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.9 PBAOCHANNEL

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.10 PBDICHANNEL

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.19.11 PBDOCHANNEL

Refer to the following documents for PGM-related documentation for this block.

- *PROFIBUS Gateway Module (PGM) User's Guide*
- *PROFIBUS Gateway Module (PGM) Parameter Reference*

4.20 Foundation FieldBus Interface Module (FIM) Blocks

Related topics

"FIM Block" on page 371
"FIMS Block" on page 371
"FIM4 Block" on page 371
"FIM4SEC Block" on page 371
"FIM8 Block" on page 371
"FIM Block" on page 371
"FFLINK" on page 371

4.20.1 FIM Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.2 FIMS Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.3 FIM4 Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.4 FIM4SEC Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.5 FIM8 Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.6 FIM Block

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.20.7 FFLINK

For more information about FIM and device-related operations, refer to the following documents:

- *Series C FIM User's Guide*

4.21 Enhanced High-Performance Process Manager (EHPM) Block

Related topics

"EHPM block" on page 372

4.21.1 EHPM block

Description	The EHPM block represents the EHPM hardware and is used for defining the EHPM Controller in Control Builder. EUCN points and strategies are executed in the EHPM Controller environment. The EHPM block has standard tabs like Main tab, Statistics tab, peer-to-peer communication and so on. These tabs represent the parameters which are mostly for viewing the current status of EHPM and statistics values. A new tab has been introduced, the EUCN tab, which shows the EHPM and ENIM specific information and their relationship. Parameters and information in this tab can be modified or added as per the requirements. The EHPM block is designed to be stand-alone and it belongs to the independent block category.

Function	• The EHPM block defines the EHPM controller in the Control Builder and represents this controller in the Experion server.
	• In the Control Builder Project tree view, identification data and certain configuration parameters can be configured. The EHPM state and statistical parameters are view or read only.
	• On the Main tab, the FTE Device Index shows the unique identification of the module. The value of redundant module parameter indicates whether the EHPM is redundant or non-redundant and how it should be loaded.
	• Once the EHPM block is loaded to the Monitoring view, the EHPM state reflects the state of the EHPM Controller.
	• The EHPM block functionality is similar to that of a C300 controller block. However, a CEE is not associated with the EHPM block and hence the EHPM block can perform only basic execution of its parameters. Also, the EHPM controller state cannot be changed from the Control Builder; it can be changed only from TPN.
	• The EHPM block can directly process parameter read and write requests initiated by CDA peer devices (C300 and ACE), only after it is loaded to the Monitoring view.
	• The EHPM Data Access parameter allows you to optionally download the EHPM points to the Experion server so that standard history can be collected through CDA. All the EHPM points under an EHPM Controller are configured as per the value of EHPM Data Access parameter (**Peer to Peer Only** or **Peer to Peer and ExpServer**) defined on the EHPM block in Control Builder.
	• EHPM points and control functions are configured using TPN configuration tools. The EHPM block must be configured in Control Builder and loaded to the monitoring side to enable:
	  – CDA peer access between the ACE/C300 Controller and the EHPM Controller
	  – CDA data access to the EHPM points from the Experion server
	• Beginning with Experion R431, you can enable automatic import of EHPM points from ENIM into ERDB using the auto-import service. This is accomplished by configuring the owning ENIM which contains the EHPM points and then selecting the **Enable automatic point import** checkbox on the EUCN tab.
	
	Note
	Loading of the EHPM block to the Monitoring view is limited by the **EHPM Connections** license. For more information on EHPM license, contact your Honeywell representative.

Inputs	CDA communication
Outputs	Refer to the description.

Parameters	AUTHSERVERIPADDR AUTHSTATE AUTOMATICPOINTIMPORT AVGRDRRESPONSETIMEUS CWRITEAVGPPS CWRITEMAXPPS CHIGHREADAVGPPS CHIGHREADMAXPPS CLOWREADAVGPPS CLOWREADMAXPPS CDISPAVGCHPPS CDISPMAXCHPPS DATEPTIMPORT # DISPSUBCACHEFRFCTR DISPLAYSUBSCRMAXPPS DISPOVRLOAD EAGN EHPMDATAACCESS EHPMSTATE EHPM_ERRCODE EHPM_SELFTEST ENIMIPADDRESS HPM_CNTRLPERSREV HPM_CNTRLPERSVER HPM_COMMPERSREV HPM_COMMPERSVER	HPM_FWREV HPM_HWREV HPM_IOL_HWREV HPMNODENO IMPORTSTATUS # IMRAVGRESPTIMEP2PREAD IMRAVGRESPTIMEP2PWRITE IMRAVGRESPTIMEDISPREAD IMRAVGSTDDISPREAD IMRMAXRESPTIMEP2PREAD IMRMAXRESPTIMEP2PWRITE IMRMAXRESPTIMEDISPREAD IMRMAXSTDDISPREAD NOPERDB # OWNINGENIM RCVNONRDRFRAMECOUNT RCVRDRREQUESTCOUNT RCVRDRRESPONSECOUNT RDRRETRYCOUNT SIMULATEDIO TOTALREQRESPCOUNT UCNADDRESS UCNNW UNEXPECTEDRDRRESPONSECOUNT
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22 EtherNet/IP blocks

This topic provides reference information about EtherNet/IP I/O devices, PowerFlex drives, E3, and E3 plus relays.

Related topics

"Input and output type EtherNet/IP I/O module blocks" on page 374
"EtherNet/IP drive and relay module blocks" on page 383

4.22.1 Input and output type EtherNet/IP I/O module blocks

Related topics

"1738–AENT (ArmorPoint adapter)" on page 374
"1738-IB4DM12 (ArmorPoint 24V DC 4-channel Digital Input Module with Diagnostics)" on page 375
"1738-IB8M12 (ArmorPoint 24V DC 8-channel Digital Input Module)" on page 375
"1738-IE2CM12 (Armor Point 2-Channel 24V DC Analog Input Module)" on page 376
"1738-IE4CM12 (ArmorPoint 4-Channel 24V DC Analog Input Module)" on page 376
"1738-IR2M12 (Armor Point, 2-Channel RTD Analog Input)" on page 377
"1738-IT2IM12 (Armor Point, 2-Channel Thermocouple Analog Input)" on page 377
"1738-OA2M12AC3 (Armor Point - 2 channel - 120/230V ac Output Module w/ 2 AC 3 pin)" on page 378
"1738-OB2EPM12 (Armor Point - 2 channel - 24V dc Digital Output Module)" on page 379
"1738-OB8EM12 (Armor Point - 8 channel, 24V dc Digital Output Module)" on page 379
"1738-OE2CM12 (Armor Point 2-Channel 24V dc Analog Output Module)" on page 380
"1738-OE4CM12 (Armor Point 4-Channel 24V dc Analog Output Module)" on page 380
"1732E-IB16M12DR (ArmorBlock 24V DC 16-channel Digital Input Dual-Port EtherNet Module with Diagnostics)" on page 381
"1732E-IF4M12R (ArmorBlock 4-Channel 24V DC Analog Input Dual-Port EtherNet/IP Module)" on page 381
"1732E-IR4IM12R (Armor Block, 4 - Channel RTD Analog Input)" on page 382
"1732E-IT4IM12R (Armor Block, 4 – Channel Thermocouple Analog Input)" on page 382
"1732E-OF4M12R (Armor Block- 4 channel – 24V DC Analog Output Module)" on page 383

4.22.1.1 1738–AENT (ArmorPoint adapter)

Description	Identifies the physical adapter that Armor Point I/O modules communicate through.
Function	Defines the number of IOMs in this chassis behind the adapter, communication path for all IOMs attached to this adapter, and execution state of the adapter. Provides link to IOC through IO manager software resident in the CPM/C300. Executes at the rate of CEE Base Execution Period.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • EIOMTYPE • EXECSTATE • FILTEROFF • FILTERON	• IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • OPENWIRE • OTRPI • OWDENBL • PVFL • SHORTCIRCUIT • SLOT • TORPI
Associate Blocks	1738-IB4DM12, 1738-IB8M12, 1738-IE2CM12, 1738-IE4CM12, 1738-IR2M12, 1738-IT2IM12, 1738-OA2M12AC3, 1738-OB2EPM12, 1738-OB8EM12, 1738-OE2CM12, 1738-OE4CM12	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.2 1738-IB4DM12 (ArmorPoint 24V DC 4-channel Digital Input Module with Diagnostics)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module hardware is read and displayed.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • AUTOBAUDDISABLE • BADPVFL • DESC • DRIVERNAME • EIOMTYPE • EXECSTATE	• FILTEROFF • FILTERON • IOCONNSTATUS • IPADDRESS • OPENWIRE • OTRPI • OWDENBL • PVFL • SHORTCIRCUIT • SLOT • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.3 1738-IB8M12 (ArmorPoint 24V DC 8-channel Digital Input Module)

Description	Represents the physical IOM for C300.
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	• ALMENBSTATE • DESC • DRIVERNAME • EIOMTYPE • EXECSTATE • FILTEROFF • FILTERON	• IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • OTRPI • PVFL • SLOT • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.4 1738-IE2CM12 (Armor Point 2-Channel 24V DC Analog Input Module)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module hardware is read and displayed. The Input raw value obtained from IO module hardware is scaled between the PVRAW ranges and sent to the IOC.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • DIGFILTER • DRIVERNAME • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS	• NOTCHFILTER • OTRPI • OVERRANGE • PV • PVRAW • PVRAWHI • PVRAWLO • PVSTS • RANGE • SLOT • TORPI • UNDERRANGE • UPDATERATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.5 1738-IE4CM12 (ArmorPoint 4-Channel 24V DC Analog Input Module)

Description	Represents the physical IOM for C300.
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO hardware module is read and displayed. The Input raw value obtained from IO module hardware is scaled between the PVRAW ranges and sent to the IOC.
Inputs	Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC.

Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • DIGFILTER • DRIVERNAME • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS	• NOTCHFILTER • OTRPI • OVERRANGE • PV • PVRAW • PVRAWHI • PVRAWLO • PVSTS • RANGE • SLOT • TORPI • UNDERRANGE • UPDATERATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.6 1738-IR2M12 (Armor Point, 2-Channel RTD Analog Input)

Description	Identifies the physical IOM for the CPM/C300 to provide links to the associated IOC.	
Function	Defines the type of IOM, number of channels, execution state and communications path for data. Provides link to IOC through IO manager software resident in the CPM/C300. Executes at the rate of CEE Base Execution Period.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMENBSTATE • DESC • DIGFILTER[0 - 1] • DRIVERNAME • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • NOTCHFILTER	• OTRPI • OVERRANGE[0 - 1] • PVRAW[0 - 1] • PV[0 - 1] • PVSTS[0 - 1] • PVLOSIGNAL[0 - 1] • PVHISIGNAL[0 - 1] • SENSORTYPE[0 - 1] • SLOT • TEMPUNITS[0 - 1] • TORPI • UNDERRANGE[0 - 1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.7 1738-IT2IM12 (Armor Point, 2-Channel Thermocouple Analog Input)

Description	Identifies the physical IOM for the CPM/C300 to provide links to the associated IOC.

Function	Defines the type of IOM, number of channels, execution state and communications path for data. Provides link to IOC through IO manager software resident in the CPM/C300. Executes at the rate of CEE Base Execution Period. The channel status information sent back by the IO module hardware is read and displayed.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMENBSTATE • DESC • CJNOTCHFLT • CJMODE • CJENABLE[0 - 1] • CJOFFSET[0 - 1] • CJTEMP • CJUNDERRANGE • CJOVERRANGE • DIGFILTER[0 - 1] • DRIVERNAME • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • NOTCHFILTER[0 - 1]	• OTRPI • OVERRANGE[0 - 1] • PVRAW[0 - 1] • PV[0 - 1] • PVSTS[0 - 1] • PVLOSIGNAL[0 - 1] • PVHISIGNAL[0 - 1] • SLOT • SENSORTYPE[0 - 1] • TEMPUNITS[0 - 1] • TORPI • UNDERRANGE[0 - 1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.8 1738-OA2M12AC3 (Armor Point - 2 channel - 120/230V ac Output Module w/ 2 AC 3 pin)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The output obtained from the channel is sent to the IOM. Performs fault handling based on the configuration made.	
Inputs	Real-time data transmission from the associated channel.	
Outputs	Real-time data transmission to the physical device.	
Parameters	DESC EIOMTYPE IPADDRESS SLOT TORPI OTRPI EXECSTATE ALMENBSTATE IOCONNSTATUS IOCONNEXTSTS	DRIVERNAME FAULTVALUE FAULTMODE FAULTSTATE OPFL OPINITVAL

Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.

4.22.1.9 1738-OB2EPM12 (Armor Point - 2 channel - 24V dc Digital Output Module)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module is read and displayed. The final output obtained from the channel is sent to the IOM. Performs fault handling based on the configuration made.	
Inputs	Real-time data transmission from the associated channel.	
Outputs	Real-time data transmission to the physical device.	
Parameters	DESC EIOMTYPE IPADDRESS SLOT TORPI OTRPI EXECSTATE ALMENBSTATE IOCONNSTATUS IOCONNEXTSTS	DRIVERNAME FAULTVALUE FAULTMODE FAULTSTATE ENBNOLOAD RESETMODE ENBLATALM OPFL OPINITVAL CHNSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.10 1738-OB8EM12 (Armor Point - 8 channel, 24V dc Digital Output Module)

Description	Represents the physical IOM for C300.
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module is read and displayed. The final output obtained from the channel is sent to the IOM. Performs fault handling based on the configuration made.
Inputs	Real-time data transmission from the associated channel.
Outputs	Real-time data transmission to the physical device.

Parameters	DESC EIOMTYPE IPADDRESS SLOT TORPI OTRPI EXECSTATE ALMENBSTATE IOCONNSTATUS IOCONNEXTSTS	DRIVERNAME FAULTVALUE FAULTMODE FAULTSTATE ENBNOLOAD RESETMODE ENBLATALM OPFL OPINITVAL CHNSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.11 1738-OE2CM12 (Armor Point 2-Channel 24V dc Analog Output Module)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module is read and displayed. The output obtained from channel is scaled between the OPFINAL ranges and sent to the IOM. Performs Fault handling, based on the configuration made.	
Inputs	Real-time data transmission from the associated channel.	
Outputs	Real-time data transmission to the physical device.	
Parameters	DESC EIOMTYPE IPADDRESS SLOT TORPI OTRPI EXECSTATE ALMENBSTATE IOCONNSTATUS IOCONNEXTSTS	DRIVERNAME RANGE OPFINALLORANGE OPFINALHIRANGE FAULTVALUE FAULTACT OP INITVAL OPFINAL CHNFLTSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.12 1738-OE4CM12 (Armor Point 4-Channel 24V dc Analog Output Module)

Description	Represents the physical IOM for C300.
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module is read and displayed. The output obtained from channel is scaled between the OPFINAL ranges and sent to the IOM. Performs Fault handling, based on the configuration made.

Inputs	Real-time data transmission from the associated channel.	
Outputs	Real-time data transmission to the physical device.	
Parameters	DESC EIOMTYPE IPADDRESS SLOT TORPI OTRPI EXECSTATE ALMENBSTATE IOCONNSTATUS IOCONNEXTSTS	DRIVERNAME RANGE OPFINALLORANGE OPFINALHIRANGE FAULTVALUE FAULTACT OP INITVAL OPFINAL CHNFLTSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.13 1732E-IB16M12DR (ArmorBlock 24V DC 16-channel Digital Input Dual-Port EtherNet Module with Diagnostics)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module hardware is read and displayed.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • EIOMTYPE • EXECSTATE • FILTEROFF • FILTERON	• IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • OPENWIRE • OTRPI • OWDENBL • PVFL • SHORTCIRCUIT • SLOT • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.14 1732E-IF4M12R (ArmorBlock 4-Channel 24V DC Analog Input Dual-Port EtherNet/IP Module)

Description	Represents the physical IOM for C300.
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module hardware is read and displayed. The Input raw value obtained from IO module hardware is scaled between the PVRAW ranges and sent to the IOC.
Inputs	Real-time data transmission from physical device.

Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • DIGFILTER • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS	• IPADDRESS • OTRPI • OVERRANGE • PV • PVRAW • PVRAWHI • PVRAWLO • PVSTS • RANGE • SLOT • TORPI • UNDERRANGE • UPDATERATE
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.15 1732E-IR4IM12R (Armor Block, 4 - Channel RTD Analog Input)

Description	Identifies the physical IOM for the CPM/C300 to provide links to the associated IOC.	
Function	Defines the type of IOM, number of channels, execution state and communications path for data. Provides link to IOC through IO manager software resident in the CPM/C300. Executes at the rate of CEE Base Execution Period.	
Inputs	Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC.	
Parameters	• ALMENBSTATE • DESC • DIGFILTER[0 - 3] • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • NOTCHFILTER	• OTRPI • OVERRANGE[0 - 3] • PVRAW[0 - 3] • PV[0 - 3] • PVSTS[0 - 3] • PVLOSIGNAL[0 - 3] • PVHISIGNAL[0 - 3] • SENSORTYPE[0 - 3] • TEMPUNITS[0 - 3] • TORPI • UNDERRANGE[0 - 3]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.16 1732E-IT4IM12R (Armor Block, 4 – Channel Thermocouple Analog Input)

Description	Identifies the physical IOM for the CPM/C300 to provide links to the associated IOC.
Function	Defines the type of IOM, number of channels, execution state and communications path for data. Provides link to IOC through IO manager software resident in the CPM//C300. Executes at the rate of CEE Base Execution Period.
Inputs	Real-time data transmission from physical device.

Outputs	Real-time data transmission to configured IOC.	
Parameters	OTRPI OVERRANGE[0 - 3] PVSTS[0 - 3] SENSORTYPE[0 - 3] TEMPUNITS[0 - 3] TORPI • ALMENBSTATE • CJENABLE[0 - 3] • CJOFFSET[0 - 3] • CJMODE • CJSTATUS[0 – 3] • CJTEMP[0 - 3] • DESC • DIGFILTER[0 - 3] • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • NOTCHFILTER[0 - 3]	• OTRPI • OVERRANGE[0 - 3] • PVLOSIGNAL[0 - 3] • PVHISIGNAL[0 - 3] • PVSCALEFACTOR[0 - 3] • PVRAW[0 - 3] • PV[0 - 3] • SENSORTYPE[0 - 3] • TEMPUNITS[0 - 3] • TORPI • UNDERRANGE[0 - 1]
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.1.17 1732E-OF4M12R (Armor Block- 4 channel – 24V DC Analog Output Module)

Description	Represents the physical IOM for C300.	
Function	Defines the type of IOM, number of channels, and execution state. Executes at the rate of Base Execution Period. The channel status information sent back by the IO module is read and displayed. The final output obtained from the channel is sent to the IOM. Performs fault handling	
Inputs	Real-time data transmission from the associated channel.	
Outputs	Real-time data transmission to the physical device.	
Parameters	• DESC • EIOMTYPE • IPADDRESS • SLOT • TORPI • OTRPI • EXECSTATE • ALMENBSTATE • IOCONNSTATUS •	• IOCONNEXTSTS • RANGE • OPFINALLORANGE • FAULTVALUE • FAULTACT • OP • INITVAL • OPFINAL • CHNFLTSTS
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.2 EtherNet/IP drive and relay module blocks

Related topics

“PF755 (PowerFlex 755 AC Drive)” on page 384

"193DNEN_E3 (E3 Overload Relay)" on page 384
"193DNEN_E3P (E3 Plus Overload Relay)" on page 385

4.22.2.1 PF755 (PowerFlex 755 AC Drive)

Description	Represents the physical drive for C300.	
Function	Defines the type of drive module and the execution state Executes at the rate of the base execution period. The Input status information and reference sent back by the module hardware is read and displayed. The output obtained from the output channel is scaled between the EULO/EUHI ranges and sent to the drive. Additionally, commands obtained from the output channel or the drive module is sent to the drive.	
Inputs	Real-time data transmission from the associated channel. Real-time data transmission from physical device.	
Outputs	Real-time data transmission to the configured IOC. Real-time data transmission to the physical device.	
Parameters	• ALMACTIVE • ALMDESC • ALMENBSTATE • ALMPRI • ALMSEV • ALMSTR • DESC • EIOMTYPE • EXECSTATE • FILTEROFF • FILTERON	• IOCCONNEXTSTS • IOCONNSTATUS • IPADDRESS • OPENWIRE • OTRPI • OWDENBL • PVFL • SHORTCIRCUIT • SLOT • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.2.2 193DNEN_E3 (E3 Overload Relay)

Description	Represents the physical E3 Relay connected by using the 193-DNENCATR adaptor for C300.
Function	Defines the type of E3 Relay module and execution state. Executes at the rate of Base Execution Period. The Input status information sent back by the E3 Relay hardware through the 193-DNENCATR adaptor is read and displayed. The output / commands obtained from the Output channel or the 193DNEN_E3 Module is sent to E3 Relay Hardware through the 193-DNENCATR adaptor. Supports 2 digital input as status and 1 digital output as command.
Inputs	Real-time data transmission from the associated Channel. Real-time data transmission from physical device.
Outputs	Real-time data transmission to configured IOC. Real-time data transmission to the physical device.

Parameters	• ALMENBSTATE • DESC • DL_RPI • EUDESC • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS	• IOCONNSTATUS • IPADDRESS • OTRPI • OWDENBL • OPENWIRE • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.22.2.3 193DNEN_E3P (E3 Plus Overload Relay)

Description	Represents the physical E3 Relay connected by using the 193-DNENCATR adaptor for C300.	
Function	Defines the type of E3 Plus Relay module and execution state. Executes at the rate of Base Execution Period. The Input status information sent back by the E3 Plus Relay hardware through the 193-DNENCATR adaptor is read and displayed. The output / commands obtained from the Output channel or the 193DNEN_E3 plus module is sent to E3 Plus Relay Hardware through the 193-DNENCATR adaptor. Supports 4 digital input as status and 2 digital output as command.	
Inputs	Real-time data transmission from the associated Channel. Real-time data transmission from physical device.	
Outputs	Real-time data transmission to configured IOC. Real-time data transmission to the physical device.	
Parameters	• ALMENBSTATE • DESC • DL_RPI • EUDESC • EIOMTYPE • EXECSTATE • IOCCONNEXTSTS	• IOCONNSTATUS • IPADDRESS • OTRPI • OWDENBL • OPENWIRE • TORPI
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.23 ControlLogix and UDT blocks

Related topics

"User Defined Tag block" on page 386
"ControlLogix gateway block" on page 387

4.23.1 User Defined Tag block

Description	Provides a data interface to read from/write to a UDT defined in the Control Logix PLC. This block will be loaded and run in a control module and will have the ability to read from or write to a UDT defined in the Control Logix PLC.	
Function	• Reads data from the UDT defined in the PLC that this block is mapped to and makes it available to the connected block or reads data from the connected block (or a user entered value) and writes data to the mapped UDT in the PLC. • Supports the following data types: – INT32 – FLOAT32 – INT16 – INT8 – BOOL – STRING (Only Read, writes of string are not supported.) • Supports any combination (arrayed or non-arrayed) of the above mentioned data types, provided they are mapped exactly to an existing UDT in the PLC. • Provides information on status of the block execution through the TAGEXECSTATUS parameter. • This is not an alarming block and thus will not generate alarms by itself.	
Inputs	This Block needs to be configured with the information of the gateway block that it will be associated with, through which it will obtain the information of the address of the PLC to establish communication with. Information about the User defined tag in the PLC that this block represents should also be provided. Communication will be based on the **Tag Name**. Therefore, it is required to configure the Tag name to match the tag name provided in the PLC. The Configuration information cannot be modified at run-time. If changes must be made to the configuration of a UDT Block, these changes must be made in the Project Database and the block must be reloaded to the controller for the changes to take effect.	
Outputs	Read/write of data	
Parameters	• BLOCKTYPEID • CLX_GATEWAY • CLX_TAG • DESC • EUDESC • READSTATUS	• TORPI • INBADOPT • TAGEXECSTATUS • TAGRDGENSTATUS • TAGRDEXTSTATUS •
Associated Blocks	CONTROLLOGIXMODULE block	
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.	

4.23.2 ControlLogix gateway block

Description	The Gateway block is a representation of the Control Logix PLC, with which communication from C300 must be established.
Function	This block does not get loaded to the controller. However the address information is passed on to the associated UDT blocks during load.
Inputs	This Block needs to be configured with the address information of the PLC, the IPADDRESS of the ENET module and Slot number of the PLC in the chassis, which would form a unique path to the PLC that is intended to be communicated with.
Outputs	—
Parameters	• IPADDRESS • SLOT • MATRIKON_OPC_HOST • MATRIKON_OPC_PROGID • MATRIKON_OPC_ABPLUGIN • MATRIKON_OPC_PLC
Associated Blocks	“User Defined Tag block” on page 386
Reference	Refer to the *Control Builder Parameter Reference* for definitions of each parameter.

5 Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

5.1 Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the "Support and other contacts" section of this document.

5.2 How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

5.3 Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

5.4 Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.