Honeywell

Experion PKS

SIM-FFD User Guide

EPDOC-X134-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-X134-en-431A	431	0	February 2015

Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Contents

1 About This Document

This document provides information about the simulation environment for Series C FIM Foundation Fieldbus Devices (FFD) and modules. The simulation environment is referred to as SIM-FFD. The document covers the Configuration and Operations performed using the SIM-FFD environment.

Revision	Date	Description
A	February 2015	Initial release of the document.

Related topics

"Prerequisite skills" on page 8
"References" on page 9

1.1 Prerequisite skills

Knowledge of the following:

- Experion PKS
- Experion Control Builder
- Series C Fieldbus Interface Module
- C300 controller

1.2 References

The following list identifies all documents that may be sources of reference for material discussed in this publication.

Document Title
Series C Interface Module User's Guide
Control Building User's Guide
Series C Fieldbus Interface Module User's Guide
Experion specifications document

2 SIMFFD Purpose

The SIM-FFD is a simulation environment for Foundation Fieldbus Devices. It can simulate the following:

- Interface Modules like Series C FIM (FIM)
- LINKS of the FIM
- FF Devices connected to Interface Modules
- Standard Foundation Fieldbus Function Blocks (FB) representation and execution
- Interchange of parameter data between various FBs as specified by the Control
- Module strategy
- AI, PID, AO, Resource, and Transducer blocks
- Peer-to-peer communication with SIM-C200E and SIM-C300

Related topics

"Benefits of SIM-FFD" on page 12
"System topology" on page 13
"Features supported by SIM-FFD" on page 14

2.1 Benefits of SIM-FFD

You can use SIM-FFD to perform the following activities:

- Verify the control strategies for Foundation Fieldbus Devices without requiring FIM hardware and Foundation Fieldbus Devices.
- Configure Foundation Fieldbus Devices offline.
- Move the Foundation Fieldbus Device configuration effortlessly between the onprocess systems and the Simulation environment.
- Train new and experienced operators through a completely simulated system and process environment.

2.2 System topology

The following figure depicts how SIM-FFD fits into the Experion architecture. The SIM-FFD installed on a Windows Server or Windows operating system, simulates the functions of Foundation Fieldbus interface module and the Devices. In a simulation environment, the devices need not be physically connected to the network.

Attention

Refer to the latest Experion specifications document for the operating system specifications on which the SIM-FFD is installed.

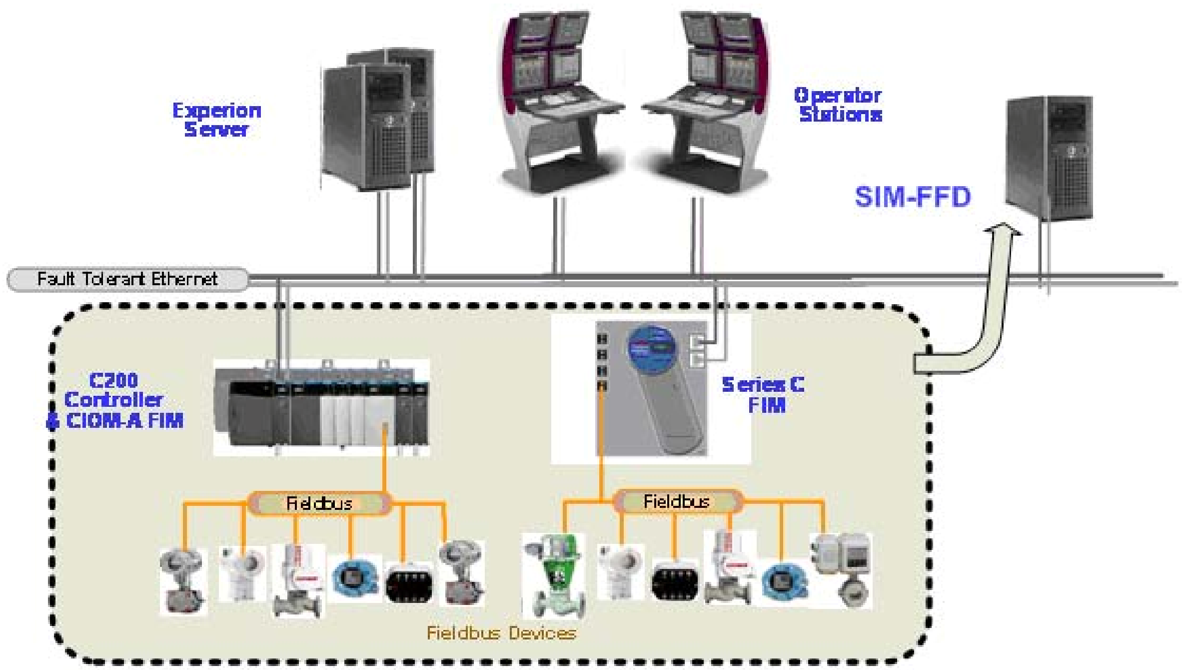


Figure 1: SIM-FFD System topology

2.3 Features supported by SIM-FFD

SIM-FFD supports the following features of Foundation Fieldbus and their integration with Experion PKS:

- Configuring, loading, and monitoring FIM blocks, LINKS, Devices, and Control Strategies
- Foundation Fieldbus function blocks like AI, AO, and PID. Parameter reads and writes are supported for all Foundation Fieldbus FBs
- Standard function block parameters and algorithm
- Honeywell-specific parameters for read/write operation
- Block instantiation
- Execution order and macrocycle periodicity
- Peer-to-peer communication

The following table lists the functionalities supported by SIM-FFD:

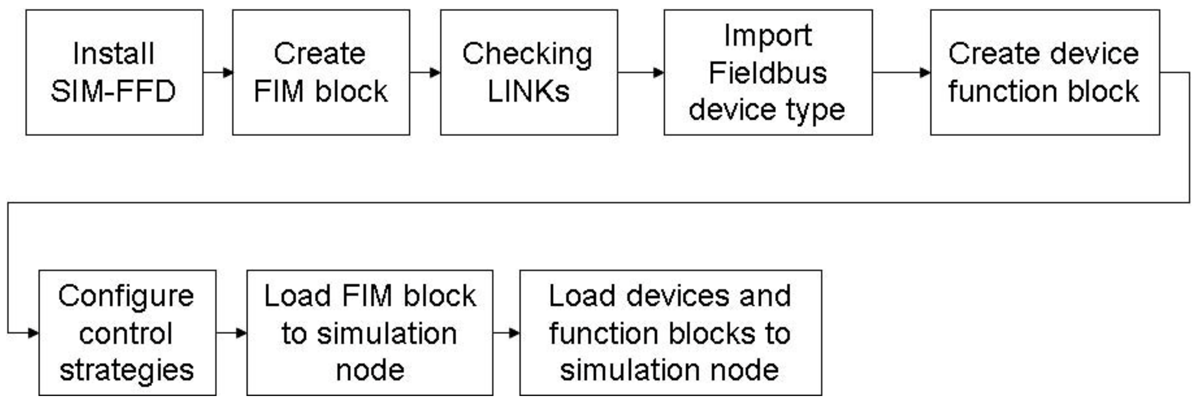
Feature	Supported by SIM-FFD
Link Behavior	Supports only Parameter substitution, and Status handling
FIM	Supports only Parameter substitution, Status handling, and Active/Inactive
FB Instantiation	Supports as standard FF function blocks
SMIB (System management)	Supports only System Management forms, and parameter read/write
NMIB (Network management)	Supports only Network Management forms, and parameter read/write
Resource block	Supports only Parameter read/write, OOS functionality, and propagating the OOS logic to Device FBs
Transducer block	Supports only Parameter read/write and OOS functionality
Function blocks, Input Class	Supports Parameter read/write, algorithm, status handling, mode handling, data handling, Simulate enable/disable, and Active/Inactive
Function blocks, Output Class	Supports Parameter read/write, algorithm, status handling, mode handling, data handling, Simulate enable/disable, and Active/Inactive
Function blocks, Control and Computational Class	Supports features like Parameter read/write, algorithm, status handling, mode handling, data handling, Simulate enable/disable, and Active/Inactive
Function block execution timing	Supports Macro cycle execution, Execution order, and Peer-to-peer communication
Alarms and Events	Supports only Raising only the standard FF specific alarms, execution handling, and Status and mode handling
Communication Failure	You can perform this by changing the Resource block mode to OOS

The SIM-FFD does not support the following functionalities:

- Communication Stack behavior
- Device recognition
- Device matching
- Device commissioning

3 Planning

The following diagram shows the tasks you have to perform to get started with using SIM-FFD:



Related topics

"Configuring FIM4 blocks for simulation" on page 20

"Checking the FFLINK configuration for simulation" on page 22

"Making Fieldbus Device Type from vendor DD files" on page 24

"Creating devices" on page 26

"Configuring and assigning control strategies to LINKS" on page 28

"Loading the FIM4 platform block" on page 30

"Loading of devices and function blocks" on page 30

Related topics

"Scenarios for using a simulation environment" on page 16
"Peer-to-peer communication" on page 17
"Alarms and Events handling" on page 18

3.1 Scenarios for using a simulation environment

You can use simulation for familiarization and training on FIM modules and Foundation Fieldbus devices. The simulation environment provides the same features for validation, configuration, and feedback, as in an on-process Experion PKS System.

3.1.1 Offline device configuration

You can use the simulation environment as a test bed for offline device configuration and testing. You can then download the configuration to an on-process system where the devices are available and connected.

Related Topics

"Making Fieldbus Device Type from vendor DD files" on page 24

"Modifying imported device information" on page 25

3.1.2 Switch control strategies from an on-process system to simulation environment

You can use the simulation environment to test changes in configuration and control strategies before loading them to an on-process system.

Related topics

"Configuring FIM4 blocks for simulation" on page 20

"Checking the FFLINK configuration for simulation" on page 22

3.1.3 SIM-FFD integration with SIM-C300

SIM-FFD cannot directly interact with a C300 controller, as there is restricted peer-to-peer access to the parameter data. However, this limitation is overcome by SIM-FFD interacting with SIM-C300 using peer-to-peer communication without any restriction. You can use heterogeneous CMs for integrating SIM-FFD with SIM-C300.

Heterogeneous CMs contain both Foundation Fieldbus specific function blocks and Experion function blocks.

Related topics

"Loading heterogeneous Control Modules for simulation" on page 31

3.2 Peer-to-peer communication

The simulation environment supports peer-to-peer communication with other simulation environments and restricted peer-to-peer communication with on-process environments.

- A simulation environment can read from the on-process system, but cannot write to the on-process system.
- A simulation environment can read and write to any other simulation environment.
- An on-process system can write to the simulation environment but cannot read value from a simulation environment.
- An on-process system can read and write to any other on-process system.

The following table provides the details of the peer-to-peer communication support:

<table>
<tr><th colspan="2"></th><th colspan="14">Responder</th></tr>
<tr><th colspan="2">CEE based platforms peer to peer relations</th><th>C200</th><th>ACE</th><th>C300</th><th>FIM</th><th>FIM4</th><th>IOLIM
(chassis)</th><th>EHG</th><th>OPC</th><th>SIM-C200E</th><th>SIM-ACE</th><th>SIM-C300</th><th>SIM-IOLIM</th><th>SIM-EHG</th><th>SIM-FFD
(FIM)</th></tr>
<tr><td>Initiator</td><td>SIM-FFD
(FIM)</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>NC</td><td>NC</td><td>NC</td><td>F</td><td>F</td><td>F</td><td>NC</td><td>NC</td><td>F</td></tr>
<tr><td colspan="8">F = Supports full peer-to-peer communication.</td><td colspan="8">R = Supports read in a peer-to-peer communication but cannot write.</td></tr>
<tr><td colspan="8">N = Does not support any peer-to-peer communication because the node cannot initiate.</td><td colspan="8">W = Supports write in a peer-to-peer communication but cannot read.</td></tr>
<tr><td colspan="8">NC = Not configurable.</td><td colspan="8"></td></tr>
</table>

3.3 Alarms and Events handling

The SIM-FFD environment can simulate the alarms and events of an Experion PKS system. It supports clearing events and regeneration of alarm functionality. However, simulation alarms are not reported to the Station.

For more details about alarms, see Function Block and Parameter Reference.

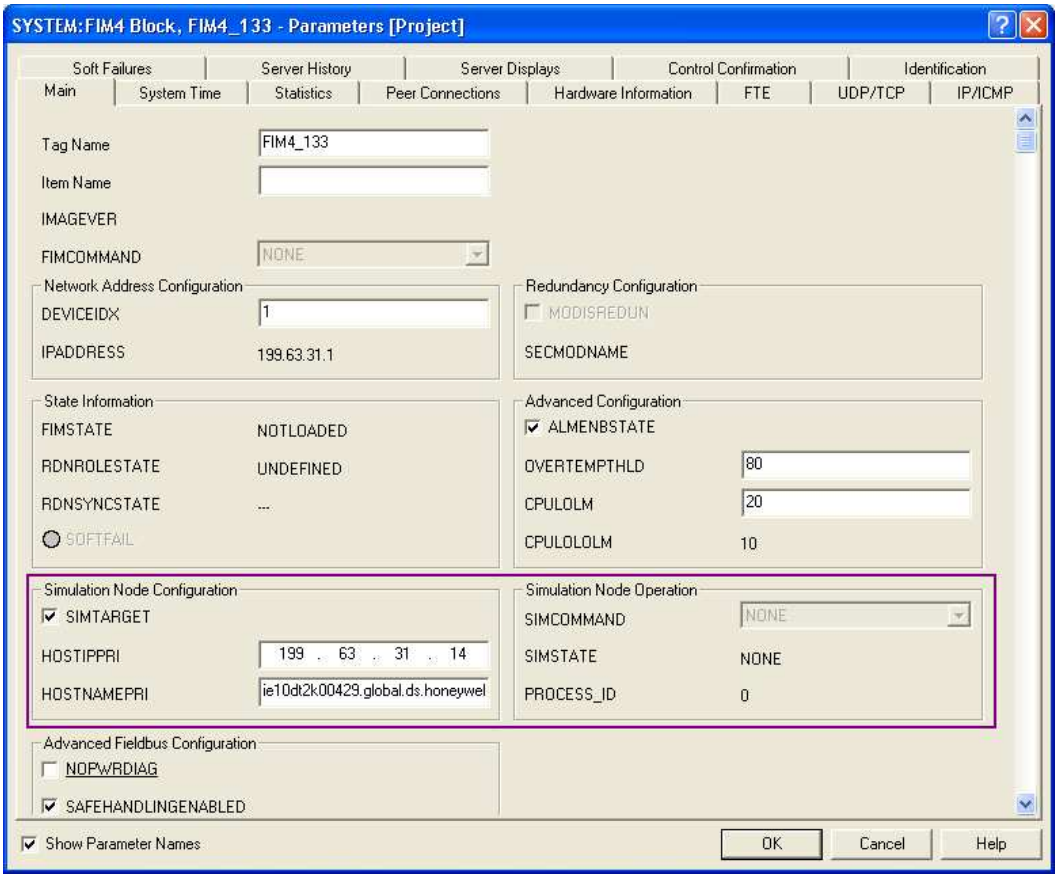
4 Configuring SIM-FFD

Related topics

"Configuring FIM4 blocks for simulation" on page 20
"Checking the FFLINK configuration for simulation" on page 22
"Making Fieldbus Device Type from vendor DD files" on page 24
"Modifying imported device information" on page 25
"Creating devices" on page 26
"Assigning a Device to FFLINK in a Project" on page 27
"Configuring and assigning control strategies to LINKS" on page 28
"Viewing and optimizing link schedule configuration" on page 29
"Loading the SIM-FFD configuration to the simulation node" on page 30
"Loading heterogeneous Control Modules for simulation" on page 31

4.1 Configuring FIM4 blocks for simulation

To configure a simulation node, you need to configure some parameters in the FIM4 block. The following figure highlights the simulation-specific parameters:



For details about creating FIM4 blocks, see 'Adding FIM4 Block to Project' in the 'Series C FIM Configuration' section of the Series C Fieldbus Interface Module User's Guide.

4.1.1 FIM4 block configuration parameters reference

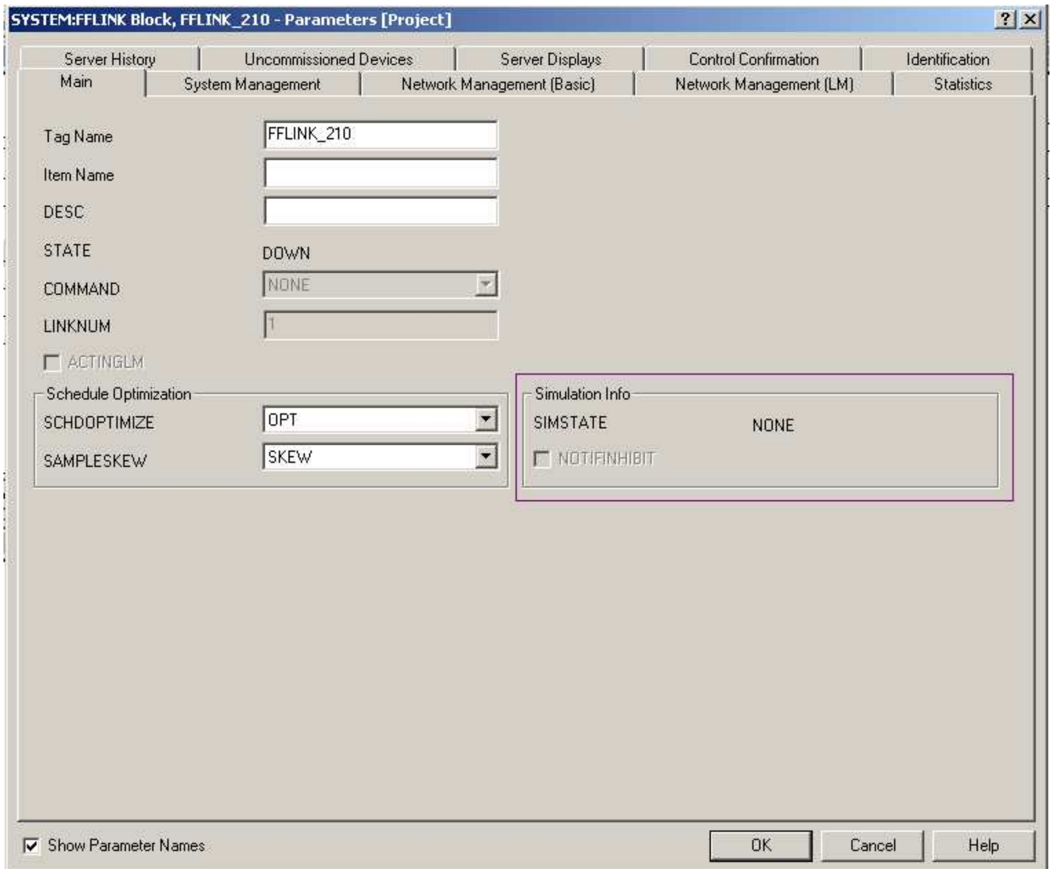
The following table describes the simulation parameters:

Parameter	Description
SIMTARGET	Select this check box to switch from the on-process system to the simulation environment. Clear this check box to switch from the simulation environment to the on-process system.

Parameter	Description
SIMCOMMAND	Select any one of the following options from the drop-down list to initiate the Simulation Command from the Shadow Plant: • **SIMFREEZE**: Freezes operation and stops simulation. • **SIMRUN**: Command to start running simulation or stepping through different execution cycles. • **SIMDISABLE**: Disables simulation because Shadow Plant has exited or the SIMDISABLE command has been issued. This changes SIMSTATE to SIMRUN. • **NONE**: No command has been issued or the issued command has been processed.
SIMSTATE	Indicates the current state of the simulation.
HOSTIPRI	Enter the Internet Protocol (IP) address for the node hosting the program. When **SIMTARGET** is selected, • You can enter the IP address (**HOSTIPPRI**) and the windows machine (**HOSTNAMEPRI**). • **Device index** and the **IPAddress** issued by BOOTP server are unavailable. When **SIMTARGET** is not selected, the IP address and the windows machine name are unavailable.
HOSTNAMEPRI	Enter the network name for the node hosting the program. **Note**: Enter either the IP address or the host name of the machine where the SIM-FFD is installed. The other is automatically traced and retrieved.
PROCESS_ID	This parameter indicates the process id of windows executable SIMFFDFIM4.exe. This is a read-only parameter that you can use to attach the debugger when multiple SIMFFDFIM4 controllers are running on the same windows node.

4.2 Checking the FFLINK configuration for simulation

To configure a simulation node, few parameters in the FFLINK configuration form must be checked. The following figure highlights the simulation-specific parameters:



For details about configuring the LINK, see 'Checking FFLink configuration in the Series C FIM Configuration' section of the *Series C Fieldbus Interface Module User's Guide*.

4.2.1 LINK configuration parameters reference

Parameter	Description
NOTIFINHIBIT	• Select to stop reporting the alarms generated in the simulation environment to the Station. • Clear to report the alarms generated in the simulation environment to the Station.

Parameter	Description
COMMAND	Select any one of the following options from the drop-down list to initiate the Simulation Command from the Shadow Plant: SIMFREEZE: Freezes operation and stops simulation. SIMRUN: Command to start running simulation or stepping through different execution cycles. SIMDISABLE: Disables simulation because Shadow Plant has exited or the SIMDISABLE command has been issued. This changes SIMSTATE to SIMRUN. NONE: No command has been issued or the issued command has been processed.
SIMSTATE	Indicates the current state of the simulation.

4.3 Making Fieldbus Device Type from vendor DD files

You can import Fieldbus Device templates and related function blocks from a vendor's DD file. Using the function blocks, you can create control strategies for Foundation Fieldbus Devices.

For details about importing Fieldbus Device templates, see 'Making Fieldbus Device Type from vendor DD files' in the 'Series C FIM configuration' section of the *Series C Fieldbus Interface Module User's Guide*.

4.4 Modifying imported device information

You can modify imported DD files to make changes to parameters, if necessary.

For details about modifying the imported device information, see 'Editing Device Block Parameters' in the 'Series C FIM configuration' section of the *Series C Fieldbus Interface Module User's Guide*.

4.5 Creating devices

You can create devices after creating the FIM block, LINK and the device type.

4.5.1 Adding a Fieldbus device to Project

You can add Fieldbus devices to the Project view by dragging and dropping the device function blocks from the library.

For details, see 'Adding a Fieldbus Device to Project' in the 'Series C FIM Configuration' section of the *Series C Interface Module User's Guide*.

4.5.2 Adding Instantiable blocks

You can also add devices with instantiable function blocks. The DD file for instantiable function blocks limits the number of instances of the function block that can be used in a device.

For details about adding instantiable blocks, see 'Block Instantiation Support' section in the *Series C Fieldbus Interface Module User's Guide*.

4.6 Assigning a Device to FFLINK in a Project

You can assign a device to the FFLINK in a project by dragging and dropping a device from the library.

For details about adding instantiable blocks, see 'Block Instantiation Support' section in the *Series C Fieldbus Interface Module User's Guide*.

4.7 Configuring and assigning control strategies to LINKS

You can configure and assign a control strategy to the LINKS by defining the associated function blocks in a Control Module (CM).

For details about Configuring control strategies, see 'Making fieldbus block template and assigning function block to device' in the 'Series C FIM Configuration' section of the *Series C Interface Module User's Guide*.

4.8 Viewing and optimizing link schedule configuration

You can automatically optimize the link schedule configuration to determine the execution order of the function blocks in the Control Module.

For details, see 'Viewing and Optimizing Link schedule configuration' in the 'Series C FIM Operation' section of the *Series C Interface Module User's Guide*.

4.9 Loading the SIM-FFD configuration to the simulation node

Loading SIM-FFD configuration to the simulation node is identical to loading the configuration to a Series C FIM module. It involves loading the following in sequence:

- FIM4 platform block
- Devices and function blocks

4.9.1 Loading the FIM4 platform block

For details about loading the configuration, see 'Loading FIM4 Components Online' in the 'Series C FIM Configuration' section of the *Series C Interface Module User's Guide*.

4.9.2 Loading of devices and function blocks

To load the devices and the function blocks in a simulation environment, only the following information is required:

- Unique node address of the simulation node

You must create devices and function blocks before loading the configuration.

For details about loading of devices and function blocks, see 'Loading FFLink Contents or Fieldbus Device' in the 'Series C FIM Configuration' section of the *Series C Interface Module User's Guide*.

4.10 Loading heterogeneous Control Modules for simulation

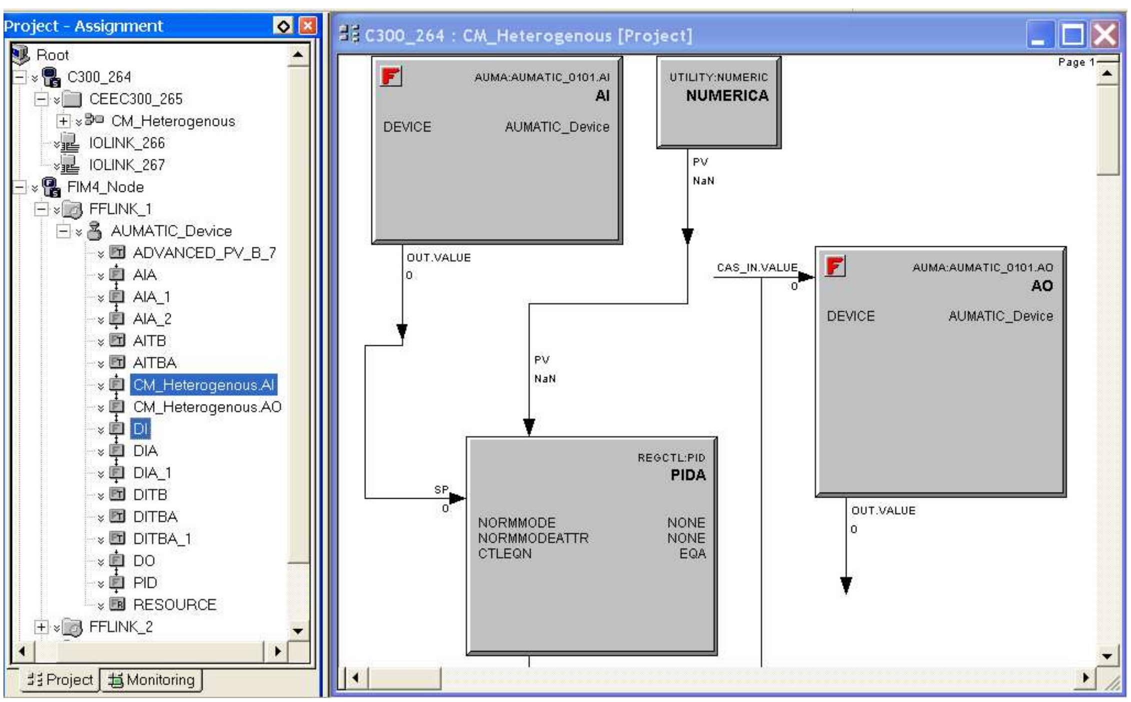
The heterogeneous CMs reside in the controller environment.

Use the following checklist for loading heterogeneous control module:

Task	Done?
Configure SIM-FFD using the simulation parameters of the FIM4 configuration form.	
Create and configure the SIM-C300 controller.	
Create and assign a Control Module to the SIM-C300 controller.	
Create and assign new Foundation Fieldbus devices to the FFLINK.	
You can also use existing devices in the library.	
In the SIM-C300 Control Module, create heterogeneous control strategies using Foundation Fieldbus function blocks and Experion function blocks.	
Load SIM-C300.	
Load SIM-FFD.	
Load the devices for which you created and assigned control strategies.	
Load the Heterogeneous Control Module.	

4.10.1 Example control strategy using a heterogeneous CM

The following figure shows an example control strategy created using a heterogeneous CM:



5 Operations

Related topics

“Switching from a simulated environment to an onprocess system” on page 34
“Switching from an on-process system to simulated environment” on page 35
“Exporting and importing project configuration” on page 36
“Monitoring the status of the simulation environment from Control Builder” on page 37
“Monitoring the status of the simulation environment from Station” on page 38
“Activating or Inactivating Fieldbus blocks or links” on page 40

5.1 Switching from a simulated environment to an onprocess system

Use this checklist to perform the following tasks in Control Builder:

Task	Done?
Inactivate SIM-C300 the simulated controller and delete the heterogeneous control modules.	
Delete the loaded SIM-FFD from the Monitoring view.	
Disable simulation configuration parameters from the FIM4 and FFLINK configuration forms.	
Use the FIM4 and FFLINK configuration forms to configure Series C FIM.	
Make Fieldbus Device Type from vendor DD files.	
Create Fieldbus devices.	
Assign devices to the FFLINK.	
Configure and assign control strategies.	
Load the FIM4 and FFLINK blocks.	
Connect the Fieldbus devices physically. • The newly connected un-commissioned devices appear in the Monitoring view.	
Match the uncommissioned devices to the devices configured in the Project view.	
Load FIM4 block to activate the on-process system.	
Load the heterogeneous control modules to the real controller environment.	

5.2 Switching from an on-process system to simulated environment

Use this checklist to perform the following tasks in Control Builder:

Task	Done?
Inactivate the C300 controller and delete the heterogeneous control modules.	
Delete the loaded FIM4 configuration from the **Monitoring** view.	
Configure simulation parameters of the FIM4 configuration form. Ensure that the SIM-FFD execution environment and the necessary Experion PKS environment are available in the target simulation node.	
Make Fieldbus Device Type from vendor DD files.	
Create Fieldbus devices.	
Assign devices to the FFLINK.	
Configure and assign control strategies.	
Load SIM-FFD to activate the simulation environment.	
Load the heterogeneous CMs to the simulation environment.	

5.3 Exporting and importing project configuration

You can import/export project configuration between the simulation environment and the on-process system.

5.3.1 Exporting function block configuration

For details, see 'Exporting function block configurations' in the 'Control Builder Operations' section of the *Control Building User's Guide*.

5.3.2 Importing function block configuration

For details, see 'Importing function block configurations' in the 'Control Builder Operations' section of the *Control Building User's Guide*.

5.4 Monitoring the status of the simulation environment from Control Builder

You can view the status of simulated FIM, its Links, and the assigned fieldbus devices from the Monitoring view of the Control Builder. In a simulation environment, the Monitoring view does not display the uncommissioned devices on the H1 segment.

For details, see 'On-line Monitoring using Control Builder' in the 'Control Builder Operations' section of the *Control Building User's Guide*.

5.5 Monitoring the status of the simulation environment from Station

You can view the status of the simulated FIM, its Links, and the assigned fieldbus devices using the Station displays. In a simulation environment, the text **Simulation** appears in the faceplate of all the relevant displays.

The following figure shows the status of a simulated FIM:

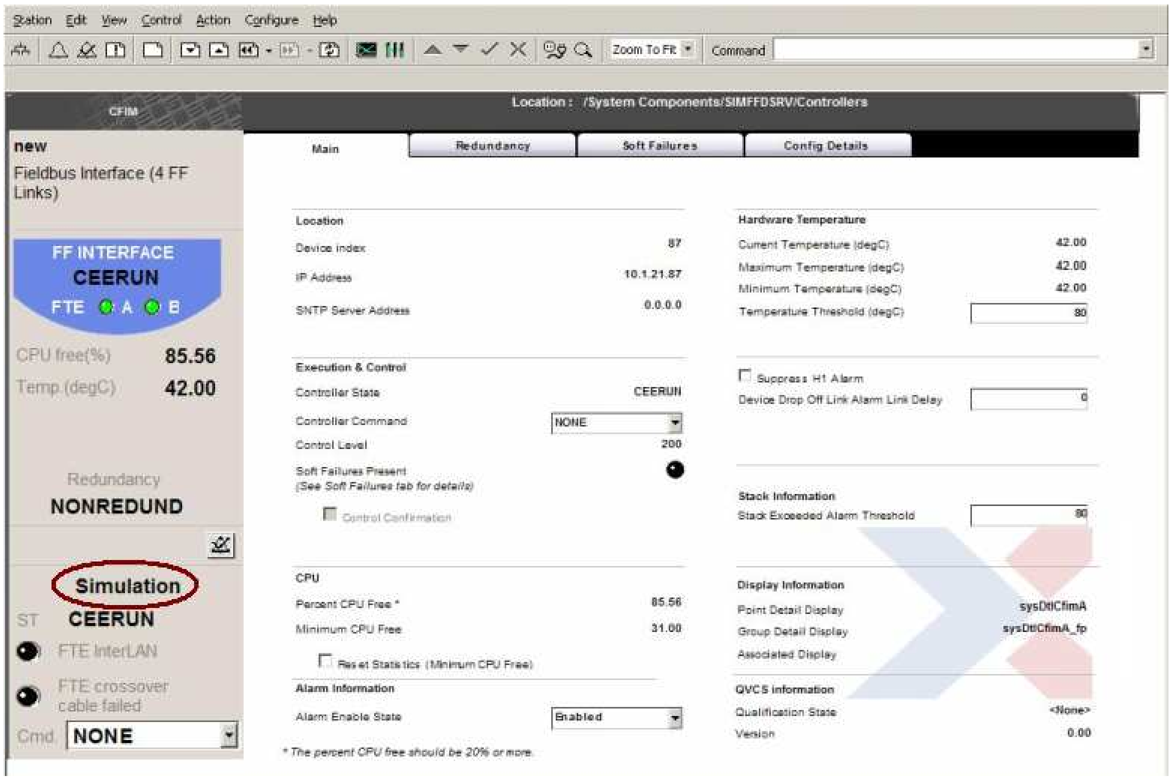


Figure 2: FIM Detail Display

The following figure shows the status of a simulated FFLINK:

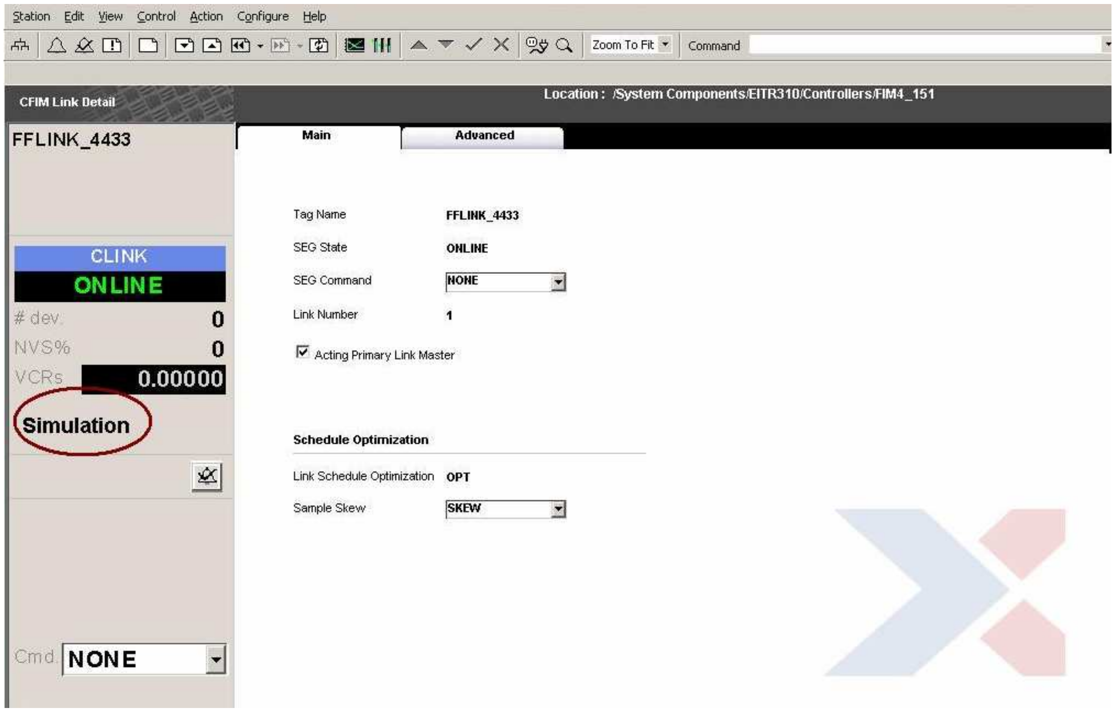


Figure 3: FFLINK Detail Display

The following figure shows the status of a device in a simulation environment:

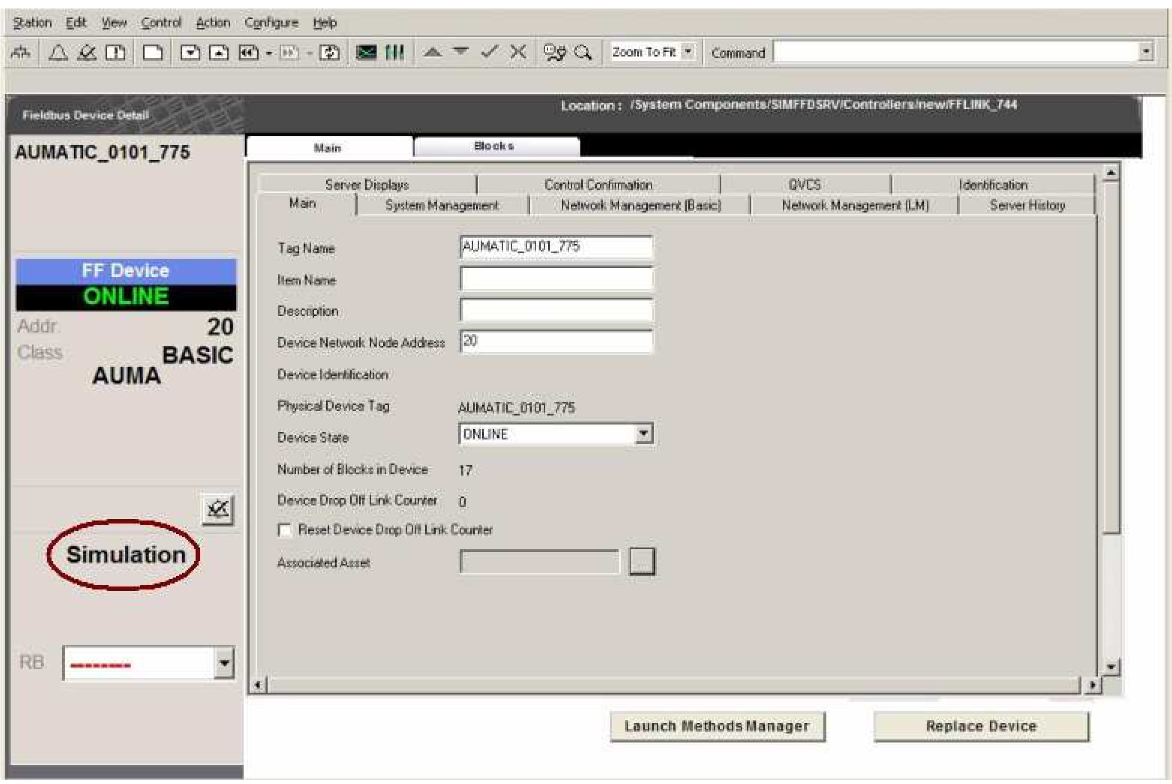


Figure 4: Device Detail Display

5.6 Activating or Inactivating Fieldbus blocks or links

You can activate/inactivate the FIM link and the associated Fieldbus devices from the Monitoring view. You can also perform this from the corresponding Detail Display in Station.

For details, see 'Activating or Inactivating Fieldbus blocks' in the 'Series C FIM Operation' section of the *Series C Interface Module User's Guide*.

6 Function Block and Parameter Reference

This section describes the function blocks supported by the simulation environment. SIM-FFD supports the following standard function blocks and their algorithms for creating control strategies:

- AI - Analog Input function block
- PID - Proportional/Integral/Derivative function block
- AO - Analog Output block

SIM-FFD supports read from and/or write to the parameters of the following function blocks. Their algorithms are not supported.

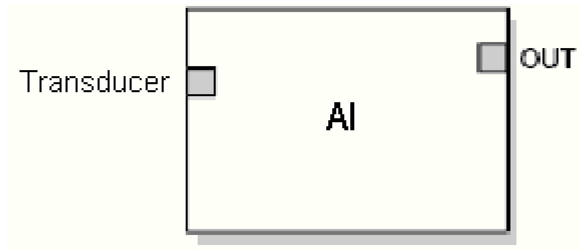
- Transducer function block
- Resource function block (SIM-FFD supports only the mode functionality of this block.)

Related topics

"Analog Input function block" on page 42
"PID function block" on page 50
"Analog Output function block" on page 60
"Discrete Input function block" on page 66
"Discrete Output function block" on page 70

6.1 Analog Input function block

The Analog Input (AI) function block processes field device measurements. This output serves as input for other function blocks. The output value from the AI block is in engineering units and contains a status indicating the quality of the measurement. The channel number is used for selecting the measurement value.



The AI block supports alarming, signal scaling, signal filtering, signal status calculation, mode control, and simulation. In the Automatic mode, the block's output parameter (OUT) indicates the process variable (PV) value and the status. In the Manual mode, you can manually set OUT. The output status indicates the Manual mode.

The **FIELD_VAL.STATUS** of the AI block shows the status values as Good, Non- Specified, and Unlimited. The **PV.STATUS** holds the same value **as FIELD_VAL.STATUS**. In addition, **OUT.STATUS** can show Bad, Out of Service statuses.

6.1.1 Input - Analog Input function block

The Analog Input block receives an input from the Transducer block.

6.1.2 Output - Analog Input function block

The calculated value from the Analog Input block serves as input to other function blocks.

6.1.3 Simulation - Analog Input function block

To work with the AI block in a simulation environment, you can perform any one of the following:

- change the mode of the block to manual and adjust the output value, or
- enable the simulation (set simulate.enable_disable= simulation active) on the configuration form and manually enter a value for the measurement value and its status.

By default, simulate.enable_disable is set to simulation active in simulation environment.

6.1.4 Parameters - Analog Input function block

The following table lists the AI block parameters and their use in simulation:

Table 1: Definitions of Analog Input function block system parameters

Parameter	Units	Description	Support and modifications in simulation
ACK_OPTION	None	Selects whether alarms associated with the block are automatically acknowledged.	Supported

Parameter	Units	Description	Support and modifications in simulation
ALARM_HYS	%	Defines the amount of change a PV value must attain within the alarm limits before the alarm condition clears.	Supported
ALARM_SUM	None	Detects the current alert status, unacknowledged states, and disable states of the alarms associated with the block	Supported
ALERT_KEY	None	The identification number of the plant unit. This data may be used in the host for sorting alarms.	Supported **Note**: The parameter value must be greater than 1 for reporting the alarms.
BLOCK_ALM	None	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alarm is entered in the sub-code field. The first alarm to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alarm-reporting task, another block alarm may be reported without clearing the Active status, if the sub-code has changed.	Supported **Note**: The Hardware information is not applicable in the simulation environment.
BLOCK_ERR	None	Reflects the error status associated with the hardware or software components associated with a block. It is a bit string that can show multiple errors.	Supported. Some block errors are not supported in simulation. See the " AI BLOCK_ERR conditions" table.

Parameter	Units	Description	Support and modifications in simulation
CHANNEL	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Simulation algorithm does not use this parameter. However, you must configure a valid value for this parameter.
FIELD_VAL	%	Represents the raw value from the field device in percent of transducer span, with a status reflecting the transducer condition, before signal characterization (L_TYPE) or filtering (PV_FTIME).	As the block is in simulation, by default it contains the simulated input value and the status. If the block is not in simulation field_val.value is set to default value 0 and field_val.status is set to default value bad.
GRANT_DENY	None	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm parameters.	Supported
HI_ALM	None	Identifies the status and time stamp associated with the high alarm.	Supported
HI_HI_ALM	None	Identifies the status and time stamp associated with the high high alarm.	Supported
HI_HI_LIM	EU of PV_S CALE	Defines the high high alarm limit setting in engineering units.	Supported
HI_HI_PRI	None	Defines priority of the high high alarm.	Supported
HI_LIM	EU of PV_S CALE	Defines the high alarm limit setting in engineering units.	Supported
O_OPTS	None	Identifies user-selectable options for altering the Input and Output block processing.	Supported Low cutoff is the only I/O option supported by AI block. You can set the I/O option in **Manual** or **Out Of Service** mode only. Other options can be selected but has no impact in the simulation.

Parameter	Units	Description	Support and modifications in simulation
L_TYPE	None	Determines whether the values passed by the Transducer block to the Analog Input block may be used directly (DIRECT) or, if the value is in different units, must be converted linearly (indirectly); or with square root (Ind Sqr Root), using the input range defined by the transducer and associated output range.	L_TYPE must be set to Direct, Indirect, or Indirect Square Root and cannot be left at initial value of 0.
LO_ALM	None	Represents the status of the low alarm and its associated time stamp.	Supported
LO_LIM	EU of PV_S CALE	Defines the setting for the low alarm in engineering units.	Supported
LO_LO_ALM	None	Represents the status of the low low alarm and its associated time stamp.	Supported
LO_LO_LIM	EU of PV_S CALE	Defines the setting for the low low alarm in engineering units.	Supported
LO_LO_PRI	None	Represents the priority of the low low alarm.	Supported
LO_PRI	None	Represents the priority of the low alarm.	Supported
LOW_CUT	%	Represents the limit used for the flow sensor input processing by the Analog Input block, if the Low Cutoff selection is chosen in IO_OPTS. If the calculated PV falls below this limit, the PV value is set to zero (0).	Supported
MODE_BLK	None	Represents the mode record of the block. Contains the Actual, Target, Permitted, and Normal modes.	Supported

Parameter	Units	Description	Support and modifications in simulation
OUT	EU of OUT_ SCALE	Represents the primary analog value calculated as a result of executing the function.	Supported
OUT_SCALE	None	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters that have the same scaling as OUT.	Supported
PV	EU of XD_S CALE	Represents either the primary analog value for use in executing the function, or a process value associated with it. It may also be calculated from the READBACK value of an Analog Output block.	Supported
PV_FTIME	Seconds	Defines the time constant of a single exponential filter for the Process Variable in seconds.	Supported
SIMULATE	None	Allows the transducer analog input or output to the block to be annually supplied, when SIMULATE is enabled. When SIMULATE is disabled, the simulate value and status track the actual value and status.	Enable/disable is selected by default in simulation. The simulation active alarm is not reported.
STRATEGY	None	Assists in grouping blocks. This data is not checked or processed by the block.	Supported

Parameter	Units	Description	Support and modifications in simulation
ST_REV	None	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.	Supported
TAG_DESC	None	Serves as user defined description of the block.	Supported
UPDATE_EVT	None	Represents an alert generated by any change to the static data.	Supported

6.1.5 Filtering - Analog Input function block

The filtering feature changes the response time of the output. You can adjust the Filter time constant (in seconds) using the **PV_FTIME** parameter. To disable the filter feature, **set PV_FTIME**=0.

6.1.6 Signal conversion - Analog Input function block

The signal conversion type can be set with the Linearization Type (**L_TYPE**) parameter. You can view the converted signal (in percent of **XD_SCALE**) using the **FIELD_VAL** parameter.

$$\text{Field\_val} = \frac{100 * (\text{Channel Value} - \text{EU□@0\%})}{\text{EU□@100\%} - \text{EU@0\%}} \text{ *XD\_SCALE values}$$

In a simulation environment, **FIELD_VAL** holds the simulated value and the status. You can choose from Direct, Indirect, or Indirect square root signal conversion with the **L_TYPE** parameter.

Direct

Direct signal conversion allows the simulated value to pass through without any conversion.

Indirect

Indirect signal conversion converts the simulated value linearly from its specified range (**XD_SCALE**) to the range and units of the **PV** and **OUT** parameters (**OUT_SCALE**).

$$\text{PV values} = \frac{\text{FIELD\_VAL}}{100} \text{ * EU**@100\% – EU**@0\%)+ EU**@0\% **OUT\_SCALE}$$

Indirect square root

Indirect square root signal conversion calculates the square root of the value computed with the indirect signal conversion and scales it to the range and units of the PV and OUT parameters.

$$PV = \text{square root}\left\{\frac{FIELD\_VAL}{100}\right\} * EU^{**}@100\% - EU^{**}@0\%) + EU^{**}@0\%$$

**OUT_SCALE values

When the converted input value < the limit specified by the LOW_CUT parameter, and the low cutoff I/O option (IO_OPTS) is selected (True), a value of zero is used for the converted value (PV).

6.1.7 Analog Input function block errors

The following table lists the conditions reported in the BLOCK_ERR parameter. The last column describes the errors supported in simulation.

Table 2: AI BLOCK_ERR conditions

Condition Number	Condition Name and Description	Supported in simulation
0	Other	No
1	Block Configuration Error: The selected channel carries a measurement that is incompatible with the engineering units selected in XD_SCALE, the L_TYPE parameter is not configured, or CHANNEL = zero.	Yes
2	Link Configuration Error	No. Algorithm must never see a link configuration error.
3	Simulate Active: Simulation is enabled and the block is using a simulated value in its execution.	No
4	Local Override	No
5	Device Fault State Set	No
6	Device Needs Maintenance Soon	No
7	Input Failure/Process Variable has Bad Status: The hardware is bad, or a bad status is being simulated.	Yes
8	Output Failure: The output is bad based on a bad input.	Yes
9	Memory Failure	No
10	Lost Static Data	No
11	Lost NV Data	No
12	Readback Check Failed	No
13	Device Needs Maintenance Now	No
14	Power Up	No
15	Out of Service: The actual mode is out of service.	Yes

6.1.8 Analog Input function block modes

The Analog Input function Block supports three modes of operation as defined by the MODE_BLK parameter:

- **Manual (Man)**— You can manually set the block output (OUT).
- **Automatic (Auto)**— OUT indicates the analog input measurement or the simulated value in the simulation (simulate.enable_disable= simulation active) environment. You can set a good status for simulate.simulate_status and a valid value in simulate.simulate_value.

If simulation is not selected (simulate.enable_disable= simulation disabled), OUT indicates a bad status.

- **Out of Service (O/S)**— The block is not processed. FIELD_VAL and PV are not updated and the OUT status is set to Bad: Out of Service. The BLOCK_ERR parameter indicates Out of Service. In this mode, you can change all configured parameters. The target mode of a block may be restricted to one or more of the supported modes.

6.1.9 Alarm detection - Analog Input function block

A block alarm is generated when the BLOCK_ERR has an error bit set. The types of block error for the AI block are defined in the "AI BLOCK_ERR conditions" table.

The Process alarm detection is based on the OUT value. You can configure the alarm limits of the following standard alarms:

- High (HI_LIM)
- High high (HI_HI_LIM)
- Low (LO_LIM)
- Low low (LO_LO_LIM)

To avoid alarm chattering when the variable is oscillating around the alarm limit, an alarm hysteresis in percent of the PV span can be set using the ALARM_HYS parameter. The priority of each alarm is set in the following parameters:

- HI_PRI
- HI_HI_PRI
- LO_PRI
- LO_LO_PRI

Table 3: Shows the five alarm priority levels.

Priority Number	Priority Description
0	The priority of an alarm condition changes to 0 after the condition that caused the alarm is corrected.
1	An alarm condition with a priority of 1 is recognized by the system, but is not reported to the operator.
2	An alarm condition with a priority of 2 is reported to the operator, but does not require operator attention. Examples include diagnostics and system alerts.
3-7	Alarm conditions of priority 3 to 7 are advisory alarms of increasing priority.
8-15	Alarm conditions of priority 8 to 15 are critical alarms of increasing priority.

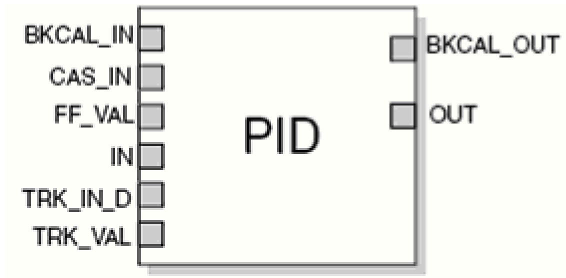
6.1.10 Status handling - Analog Input function block

The mode of the block reverts to **Manual**, if the input status on the AI block is **Bad**. In addition, you can select the **Target to Manually if Bad IN** status option to direct the target mode to revert to the manual mode. You can set the status option in **Manual or Out of Service** mode only.

Attention
The Target to Manual if Bad IN is the only status option supported by the AI function block.

6.2 PID function block

The PID function block combines all the logic to perform the proportional/integral/derivative (PID) control. The block supports mode control, signal scaling and limiting, feedforward control, override tracking, alarm limit detection, and signal status propagation. The following figure depicts the primary inputs and outputs of the PID function block:



The PID block for a Honeywell device supports two forms of the PID equation: Ideal and Robust. You can choose the appropriate equation using the PID_FORM parameter. The Ideal PID equation is the default selection.

The following is the definition of the algorithm including the form and the equation type.

$$\frac{OUT_{(s)}}{e_{(s)}} = K * (1 + \frac{1}{T_i * s} + T_d * s) * \frac{1}{T_f * s + 1};$$

$$K = K_{ln} * (1 + K_{nl} * \left|SP_{(n)} - PV_{(n)}\right|)$$

Figure 5: PID Laplace Transform

Where: Kln - linear gain

Knl - nonlinear gain

Ti - integral action time constant (RATE parameter) in seconds

S - laplace operator

Td- derivative action time constant (RATE parameter)

6.2.1 Input - PID function block

PV - Process Value

SP - Set Point

6.2.2 Output - PID function block

OP - Control Output

6.2.3 Parameters - PID function block

The following table lists the PID block parameters and their use in simulation:

Table 4: Definitions of PID function block system parameters

Parameter	Units	Description	Support in simulation
ACK_OPTION	None	Selects whether alarms associated with the block will be automatically acknowledged.	Supported
ALARM_HYS	%	Defines the amount of change a PV value must attain within the alarm limits before the alarm condition clears.	Supported
ALARM_SUM	None	Detects the current alert status, unacknowledged states, and disable states of the alarms associated with the block	Supported
ALERT_KEY	None	The identification number of the plant unit. This data may be used in the host for sorting alarms.	Supported. The value must be greater than 1 for reporting the alarms.
BAL_TIME	Seconds	Specifies the time in seconds for the internal working value of Bias or Ratio to return to the operator set value. In PID block, specifies the time constant to be used to move the integral term to obtain balance, when the output is limited and the mode is Auto, Cas, or RCas.	Supported
BIAS	EU of OUT_SCALE	Specifies the Bias value in engineering units to be used in computing the function block output.	Supported
BKCAL_HYS	%	Defines the amount of change an output value must attain from the limit before the limit status is turned OFF.	Supported
BKCAL_IN	EU of OUT_SCALE	The value and status from a lower block's BKCAL_OUT that is used to prevent reset windup and to initialize the control loop.	Supported
BKCAL_OUT	EU of PV_SCALE	The value and status required by an upper block's BKCAL_IN so the upper block may prevent reset windup and provide bumpless transfer to closed loop control.	Supported
BLOCK_ERR	None	Reflects the error status associated with the hardware or software components associated with a block. It is a bit string that can show multiple errors.	Supported The block errors for the PID block are Block configuration error, Input Failure/IN with Bad Status, local override, device fault state and Out of Service.
BYPASS	None	Provides the means to bypass the normal control algorithm. When BYPASS is On, the set point value is directly transferred to the output. To prevent a bump upon BYPASS switching, the set point automatically initializes to the output value or process variable and sets the path broken flag for one execution.	Supported. If this value is not configured then block configuration error occurs.
CAS_IN	EU of PV_SCALE	Represents the remote set point value that must come from another fieldbus block or a distributed control system (DCS) block through a defined link.	Supported

Parameter	Units	Description	Support in simulation
CONTROL_O PTS	None	Represents bit string for control options to alter the calculations done in an applicable function block.	Supported
DV_HI_ALM	None	Identifies the status and time stamp associated with the high deviation alarm.	Supported
DV_HI_LIM	EU of PV_SCALE	Defines the high deviation alarm limit setting in engineering units.	
DV_HI_PRI	None	Defines priority of the high deviation alarm.	Supported
DV_LO_ALM	None	Identifies the status and time stamp associated with the low deviation alarm.	Supported
DV_LO_LIM	EU of PV_SCALE	Defines the low deviation alarm limit setting in engineering units.	Supported
DV_LO_PRI	None	Defines priority of the low deviation alarm.	Supported
ERROR	EU of PV_SCALE	Indicates the error (SP-PV) value that identifies the control action.	Supported
FF_ENABLE	None	Enables the use of feedforward calculations.	Supported
FF_GAIN	None	Defines the gain value used to multiply the feed-forward signal before it is added to the calculated control output.	Supported
FF_SCALE	None	Defines the feed-forward input high and low scale values, engineering units code, and number of digits to the right of the decimal.	Supported
FF_VAL	EU of FF_SCALE	Represents the feed-forward value.	Supported
GAIN	None	Represents dimensionless gain used by several different algorithms.	Supported
GRANT_DENY	None	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm parameters.	Supported
HI_ALM	None	Identifies the status and time stamp associated with the high alarm.	Supported
HI_HI_ALM	None	Identifies the status and time stamp associated with the high high alarm.	Supported
HI_HI_LIM	EU of PV_SCALE	Defines the high high alarm limit setting in engineering units.	Supported
HI_HI_PRI	None	Defines priority of the high high alarm.	Supported
HI_LIM	EU of PV_SCALE	Defines the high alarm limit setting in engineering units.	Supported
HI_PRI	None	Defines priority of the high alarm.	Supported
IN	EU of PV_SCALE	Represents the primary input value of the block. Blocks that filter the input to get the PV require this parameter.	Supported
LO_ALM	None	Represents the status of the low alarm and its associated time stamp.	Supported
LO_LO_ALM	None	Represents the status of the low low alarm and its associated time stamp.	Supported
LO_LO_LIM	EU of PV_SCALE	Defines the setting for the low low alarm in engineering units.	Supported

Parameter	Units	Description	Support in simulation
LO_LO_PRI	None	Represents the priority of the low low alarm.	Supported
LO_LIM	EU of PV_SCALE	Defines the setting for the low alarm in engineering units.	Supported
LO_PRI	None	Represents the priority of the low alarm.	Supported
MODE_BLK	None	Represents the mode record of the block. Contains the Actual, Target, Permitted, and Normal modes.	Supported
OUT	EU of OUT_SCALE	Represents the primary analog value calculated as a result of executing the function.	Supported
OUT_HI_LIM	EU of OUT_SCALE	Defines the maximum output value limit in all modes, unless the CONTROL_OPTS selection No Out limits in Manual is chosen.	Supported
OUT_LO_LIM	EU of OUT_SCALE	Defines the minimum output value limit in all modes, unless the CONTROL_OPTS selection No Out limits in Manual is chosen.	Supported
OUT_SCALE	None	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters that have the same scaling as OUT.	Supported
PV	EU of PV_SCALE	Represents either the primary analog value for use in executing the function, or a process value associated with it. It may also be calculated from the READBACK value of an Analog Output block.	Supported
PV_FTIME	Seconds	Defines the time constant of a single exponential filter for the Process Variable in seconds.	Supported
PV_SCALE	None	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters that have the same scaling as PV.	Supported
RATE	Seconds	Defines the derivative time constant in seconds.	Supported
RCAS_IN	EU of PV_SCALE	Represents target set point and status provided by a supervisory host to the analog control or output block.	Supported
RCAS_OUT	EU of PV_SCALE	Represents block set point and status after ramping. It serves as input to a supervisory host for back calculation that allows action to be taken under limiting conditions or mode change.	Supported
RESET	Seconds Per repeat	Represents the Integral time constant in seconds. It is the inverse of repeats per minute.	Supported

Parameter	Units	Description	Support in simulation
ROUT_IN	EU of OUT_SCALE	Represents target output and status provided by a host to a control block for use as the block's output in ROUT mode.	Supported
ROUT_OUT	EU of OUT_SCALE	Represents block output and status. It serves as input to a host for back calculation in ROut mode that allows action to be taken under limited conditions or mode change.	Supported
SHED_OPT	None	Defines action to be taken on remote control device timeout.	Supported
SP	EU of PV_SCALE	Defines the set point of any analog block.	Supported
SP_HI_LIM	EU of PV_SCALE	Defines the high limit for set point entry.	Supported
SP_LO_LIM	EU of PV_SCALE	Defines the low limit for set point entry.	Supported
SP_RATE_DN	EU of PV_SCALE per second	Defines the downward ramp rate in PV units per second for set point changes to invoke action in the Auto mode.	Supported
SP_RATE_UP	EU of PV_SCA LE per second	Defines the upward ramp rate in PV units per second for set point changes to invoke action in the Auto mode.	Supported
STATUS_OPTS	None	Defines user-selectable options for the block processing of status.	Supported
STRATEGY	None	Assists in grouping blocks. This data is not checked or processed by the block.	Supported However, this data is not checked or processed by the block.
ST_REV	None	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.	Supported
TAG_DESC	None	Serves as user defined description of the block.	Supported
TRK_IN_D	None	Represents the discrete input for initiation of the external tracking function.	Supported
TRK_SCALE	None	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with TRK_VAL.	Supported
TRK_VAL	EU of TRK_SCALE	Represents the input value for external tracking.	Supported
UPDATE_EVT	None	Represents an alert generated by any change to the static data.	Supported
WSP.VALUE	EU of PV_SCALE	Indicates the working setpoint of the block. It is the result of setpoint rate-of-change limiting. The value is converted to percent to obtain the block's OUT value.	Supported

6.2.4 Setpoint selection and limiting- PID function block

The mode determines the setpoint of the PID block. You can configure setpoint limits using **SP_HI_LIM and SP_LO_LIM** parameters.

In **Cascade** or **RemoteCascade** mode, the setpoint is adjusted by another function block or by a host computer, and the output is computed based on the setpoint.

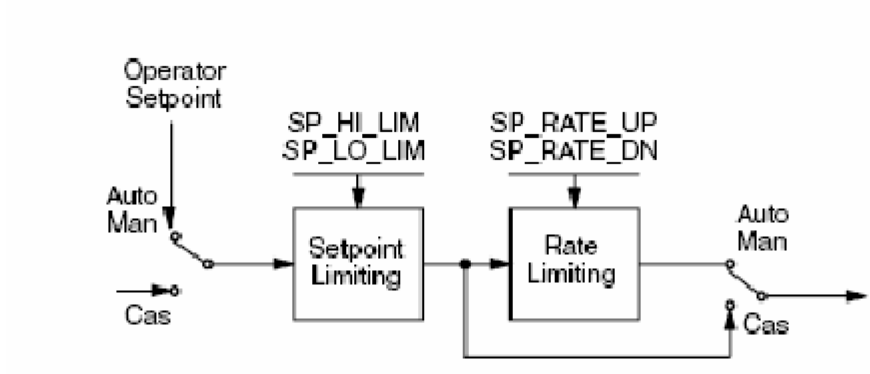
In **Automatic** mode, the operator can manually configure the setpoint, and the output is computed based on the setpoint. In Auto mode, you can adjust the setpoint limit and the setpoint rate of change using the **SP_RATE_UP** and **SP_RATE_DN** parameters.

In **Manual** mode, the operator can manually configure the output, and is independent of the setpoint.

In RemoteOutput mode, the output is entered by a host computer, and is independent of the setpoint.

The following figure illustrates the method for selecting setpoint:

PID function block setpoint



6.2.5 Filtering - PID function block

The filtering feature changes the response time of the device to smooth variations in output reading caused by rapid changes in input. You can configure the filtering feature with the **FILTER_TYPE** parameter, and you can adjust the filter time constant (in seconds) using the **PV_FTIME** or **SP_FTIME** parameters. To disable the filter feature, set the filter time constant to zero.

6.2.6 Feedforward calculation

The feedforward value (FF_VAL) is scaled (FF_SCALE) to a common range for compatibility with the output scale (OUT_SCALE). A gain value (FF_GAIN) is applied to achieve the total feedforward contribution.

6.2.7 Tracking - PID function block

You can enable the output tracking using the control options. You can set the control options in Manual or Out of Service mode only.

To operate the track function, select the **Track Enable** control option.

- When the **Track in Manual** control option is selected and the block is in Manual mode, you can activate and maintain the tracking.
- When the **Track in Manual** control option is not selected and the block is in Manual mode, you can override the tracking function.

On activating the track function, the block's actual mode changes to Local Override.

The **TRK_VAL** parameter indicates the value that must be converted and tracked into the output when the track function is operating. The **TRK_SCALE** parameter indicates the range of TRK_VAL.

When the **TRK_IN_D** parameter and the **Track Enable** control option is selected, the **TRK_VAL** input is converted to the appropriate value and the output is in the units of **OUT_SCALE**.

6.2.8 Output selection and limiting - PID function block

The block's mode and the setpoint determines the Output selection.

In **Automatic**, **Cascade**, or **Remote Cascade** mode, the PID control equation computes output.

In **Manual** and **RemoteOutput** mode, you can manually configure the output. You can limit the output by configuring the **OUT_HI_LIM** and **OUT_LO_LIM** parameters.

6.2.9 Bumpless transfer and setpoint tracking

You can configure the setpoint tracking method by configuring the following control options (CONTROL_OPTS):

SP-PV Track in Man- This control option allows the SP to track the PV when the target mode of the block is **Man**.

SP-PV Track in Local Override (LO) or IMan- This control option allows the SP to track the PV when the actual mode of the block is **LO** or **IMan**.

When one of these options is selected, the SP value is set to the PV value while in the specified mode.

You can select the value that a master controller uses for tracking by configuring the **Use PV for BKCAL_OUT** control option. The **BKCAL_OUT** value tracks the **PV** value. **BKCAL_IN** on a master controller connected to **BKCAL_OUT** on the PID block in an open cascade strategy forces its **OUT** to match **BKCAL_IN**. Therefore, you can track the PV from the slave PID block into its cascade input connection (CAS_IN). If the Use **PV** for **BKCAL_OUT** option is not selected, the working setpoint (SP_WRK) is used for **BKCAL_OUT**.

You can set the control options in **Manual** or **O/S** mode only. In Auto mode, the SP remains at the last value (it does not follow the PV).

6.2.10 PID equation type

You can select one of the following choices:

Equation Type A: Error applies to P, I, and D.

Equation Type B: Error applied to P and I, PV applied to D.

Equation Type C: Error applied only to I, PV applied to P and D.

To configure the PID block to perform the integral only control (regardless of the equation type) set RESET=0. When RESET=0, the equation reduces to an integrator equation with a gain value applied to the error.

The PID block for a non-Honeywell device can support forms and equation type in different parameters.

To customize the block and use in application, you can configure filtering, feedforward inputs, tracking inputs, setpoint and output limiting, PID equation structures, and block output action.

6.2.11 PID_FORM option

The first order filter in the above transfer function:

$$\frac{1}{T_f*s+1}$$

is the output lag filter. The PID_FORM defines how this filter is configured.

1. Regular Ideal PID:

Tf is not adjustable and is set to:

$$T_f = \frac{T_i}{RA}$$

where RA is the rate amplitude set at 24

1. Robust PID

Tf is adjustable by the operator.

6.2.12 Reverse and direct action

To configure the block output action, select the Direct Acting control option. This option defines the relationship between a change in PV and the corresponding change in output. If Direct Acting is selected, an increase in PV increases the output. You can set the control options in Manual or O/S mode only.

Attention

The following are the only control options supported by the PID function block:

- Track Enable
- Track in Manual
- SP-PV Track in Man
- SP-PV Track in LO or IMan
- Use PV for BKCA_OUT
- Direct Acting

6.2.13 Reset limiting

The PID function block provides the following:

- feedback reset limiting that prevents windup when output or input limits are encountered
- proper behavior in selector applications

6.2.14 PID function block errors

The following table lists the conditions reported in the BLOCK_ERR parameter. The last column describes the errors supported in simulation.

Table 5: PID BLOCK_ERR conditions

Condition Number	Condition Name and Description	Supported in simulation
0	Other	No
1	Block Configuration Error: The BY_PASS parameter is not configured and is set to 0, the SP_HI_LIM is less than the SP_LO_LIM, or the OUT_HI_LIM is less than the OUT_LO_LIM or Remote shed option is not set.	Yes
2	Link Configuration Error	No. Algorithm must never see a link configuration error.
3	Simulate Active	No
4	Local Override	Yes
5	Device Fault State Set	Yes. Set based on FAULT_STATE.
6	Device Needs Maintenance Soon	No

Condition Number	Condition Name and Description	Supported in simulation
7	Input Failure/Process Variable has Bad Status: The parameter linked to IN is indicating a Bad status.	Yes
8	Output Failure	No
9	Memory Failure	No
10	Lost Static Data	No
11	Lost NV Data	No
12	Readback Check Failed	No
13	Device Needs Maintenance Now	No
14	Power Up	No
15	Out of Service: The actual mode is out of service.	Yes

6.2.15 PID block modes

The PID function block supports the following modes:

- **Man**\ You can manually set the block output (**OUT**).
- **Auto**\ You can manually set the SP and the block algorithm calculates the **OUT**.
- **Cas** \ The SP is calculated in another block and provided to the PID block through the **CAS_IN** connection.
- **RCas**\ The SP is provided by a host computer that writes to the **RCAS_IN** parameter.
- **Rout**\ The **OUT** is provided by a host computer that writes to the **ROUT_IN** parameter.
- **Local Override** (LO)\ The track function is active. TRK_VAL sets OUT. The BLOCK_ERR parameter shows Local override.
- **IMan**\ The output path is not complete (for example, the cascade-to-slave path might not be open). In **IMan** mode, **OUT** tracks **BKCAL_IN**.
- **O/S**\ The block is not processed. The Out status is set to Bad: Out of Service. The BLOCK_ERR parameter shows Out of Service.

You can configure the Man, Auto, Cas and O/S modes as permitted modes for operator entry.

6.2.16 Alarm detection - PID function block

A block alarm is generated whenever the BLOCK_ERR has an error bit set. The types of block error for the PID block are defined in the "PID BLOCK_ERR conditions" table.

The Process alarm is generated based on the PV value. You can configure the alarm limits of the following standard alarms:

- High (HI_LIM)
- High high (HI_HI_LIM)
- Low (LO_LIM)
- Low low (LO_LO_LIM)

To avoid alarm chattering when the variable is oscillating around the alarm limit, an alarm hysteresis in percent of the PV span can be set using the ALARM_HYS parameter.

The additional process alarm is generated based on the difference between the SP and PV values. You can configure these alarms using the following parameters:

- Deviation High (DEV_HI_LIM)
- Deviation low (DEV_LO_LIM)

You can configure the priority of each alarm using the following parameters:

- HI_PRI
- HI_HI_PRI
- LO_PRI
- LO_LO_PRI
- DEV_HI_PRI
- DEV_LO_PRI

The following table describes the five alarm priority levels:

Table 6: Shows the five alarm priority levels.

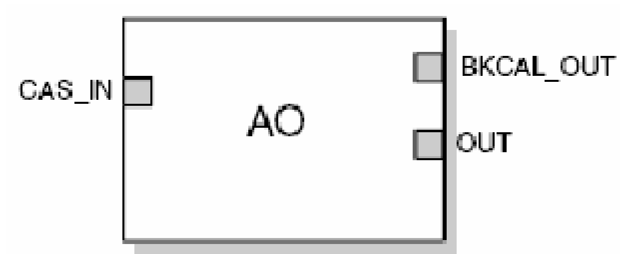
Priority Number	Priority Description
0	The priority of an alarm condition changes to 0 after the condition that caused the alarm is corrected.
1	The system recognizes an alarm condition with a priority of 1, but is not reported to the operator.
2	An alarm condition with a priority of 2 is reported to the operator, but does not require operator attention. Examples include diagnostics and system alerts.
3-7	Alarm conditions of priority 3 to 7 are advisory alarms of increasing priority.
8-15	Alarm conditions of priority 8 to 15 are critical alarms of increasing priority.

6.2.17 Status handling - PID function block

The mode of the block changes to Manual, if the input status on the PID block is Bad. In addition, you can select the Target to Manually if Bad IN status option to direct the target mode to revert to the manual mode. You can set the status option in Manual or Out of Service mode only.

6.3 Analog Output function block

The Analog Output (AO) function block assigns an output value to a field device through a specified I/O channel. The block supports the mode control, signal status calculation, and the simulation. The following figure depicts the primary inputs and outputs of the Analog Output function block:



6.3.1 Input - Analog Output function block

CAS_IN: Indicates the remote setpoint value from another function block.

6.3.2 Output - Analog Output function block

BKCAL_OUT: Indicates the value and the status required by the BKCAL_IN input of another block to prevent reset windup and to provide bumpless transfer to closed loop control.

OUT: Indicates the block output value and the status.

6.3.3 Simulation - Analog Output function block

In a simulation environment, the last value of OUT is maintained and reflected in the field value of the SIMULATE parameter. In this case, the PV and READBACK values and statuses are based on the SIMULATE value and the status that you enter.

6.3.4 Parameters - Analog Output function block

The following table lists the AO block parameters and their use in simulation:

Table 7: Definitions of Analog Output function block system parameters

Parameter	Units	Description	Support in simulation
ACK_OPTION	None	Selects whether alarms associated with the block will be automatically acknowledged.	Supported
ALARM_SUM	None	Detects the current alert status, unacknowledged states, and disable states of the alarms associated with the block	Supported
ALERT_KEY	None	The identification number of the plant unit. This data may be used in the host for sorting alarms.	Supported. The value must be greater than 1 for reporting the alarms.

Parameter	Units	Description	Support in simulation
BLOCK_ALM	None	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alarm is entered in the sub-code field. The first alarm to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alarm-reporting task, another block alarm may be reported without clearing the Active status, if the sub- code has changed.	Supported. The Hardware information is not applicable in the simulation environment.
BKCAL_OUT	EU of PV_SCALE	The value and status required by an upper block's BKCAL_IN so the upper block may prevent reset windup and provide bumpless transfer to closed loop control.	Supported
BLOCK_ERR	None	Reflects the error status associated with the hardware or software components associated with a block. It is a bit string that can show multiple errors.	Supported The block errors for the AO block are block configuration error, Input Failure/IN with Bad Status, local override, device fault state and Out of Service.
CAS_IN	EU of PV_SCALE	Represents the remote set point value that must come from another fieldbus block or a distributed control system (DCS) block through a defined link.	Supported
CHANNEL	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Simulation algorithm does not use this parameter. However, you must configure a valid value for this parameter.
FSTATE_TIME	seconds	Represents the reaction time in seconds from the detection of a failure at the output block remote set point to the output block action, if the condition still exists.	Supported
FSTATE_VAL	None	Defines the preset analog set point to use when a failure occurs. Value is ignored, if the IO_OPTS Fault state to value option is false.	Supported
GRANT_DENY	None	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm parameters.	Supported
IO_OPTS	None	Identifies user-selectable options for altering the Input and Output block processing.	Supported The supported I/O options for the AO function block are SP_PV Track in Man, Increase to Close, and Use PV for BKCAL_OUT.
MODE	None	Indicates the enumerated attribute used for requesting and showing the source of the setpoint and/or output used by the block.	Supported
OUT	EU of XD_SCALE	Represents the primary analog value calculated as a result of executing the function.	Supported

Parameter	Units	Description	Support in simulation
PV	EU of PV_SCALE	Represents either the primary analog value for use in executing the function, or a process value associated with it. It may also be calculated from the READBACK value of an Analog Output block.	Supported
PV_SCALE	None	Defines the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters that have the same scaling as PV.	Supported
READBACK	EU of XD_SCALE	Represents the 'readback' of the actual continuous valve or other actuator position in transducer units.	Supported
RCAS_IN	EU of PV_SCALE	Represents target set point and status provided by a supervisory host to the analog control or output block.	Supported
RCAS_OUT	EU of XD_SCALE	Represents block set point and status after ramping. It serves as input to a supervisory host for back calculation that allows action to be taken under limiting conditions or mode change.	Supported
SHED_OPT	None	Defines action to be taken on remote control device timeout.	Supported
			Supported
SIMULATE	EU of XD_SCALE	Allows the transducer analog input or output to the block to be manually supplied, when SIMULATE is enabled. When SIMULATE is disabled, the simulate value and status track the actual value and status.	Supported
SP	EU of PV_SCALE	Defines the set point of any analog block.	Supported
STATUS_OPTS	None	Defines user-selectable options for the block processing of status.	Supported
STRATEGY	None	Assists in grouping blocks. This data is not checked or processed by the block.	Supported However, this data is not checked or processed by the block.
ST_REV	None	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.	Supported
SP_HI_LIM	EU of PV_SCALE	Defines the high limit for set point entry.	Supported
SP_LO_LIM	EU of PV_SCALE	Defines the low limit for set point entry.	Supported
SP_RATE_DN	EU of PV_SCALE per second	Defines the downward ramp rate in PV units per second for set point changes to invoke action in the Auto mode.	Supported
SP_RATE_UP	EU of PV_SCALE per second	Defines the upward ramp rate in PV units per second for set point changes to invoke action in the Auto mode.	Supported

Parameter	Units	Description	Support in simulation
TAG_DESC	None	Serves as user defined description of the block.	Supported
UPDATE_EVT	None	Represents an alert generated by any change to the static data.	Supported
WSP.VALUE	EU of PV_SCALE	Indicates the working setpoint of the block. It is the result of setpoint rate-of-change limiting. The value is converted to percent to obtain the block's OUT value.	Supported

6.3.5 Setting the output - Analog Output function block

To set the output for the AO block, you must set the mode to define the manner in which the block determines its setpoint.

In **Manual** mode, the user can manually set the value of the output attribute (OUT), and is independent of the setpoint.

In **Automatic** mode, OUT is set automatically based on the value specified by the setpoint (SP) in engineering units and the I/O options attribute (IO_OPTS). In addition, you can limit the SP value and the rate at which a change in the SP is passed to OUT.

In **Cascade** mode, the cascade input connection (CAS_IN) is used for updating the SP. The back calculation output (BKCAL_OUT) is wired to the back calculation input (BKCAL_IN) of the upstream block that provides CAS_IN. This provides bumpless transfer on mode changes and windup protection in the upstream block. The OUT attribute or an analog readback value, such as valve position, is shown by the process value (PV) attribute in engineering units.

To select the manner of processing the SP and the channel output value, configure the setpoint limiting options, the tracking options, and the conversion and the status calculations.

6.3.6 Setpoint selection and limiting - Analog Output function block

To select the source for of the SP value, you can use the **MODE** parameter. In Auto mode, you can use the manually-entered SP. In **Cascade** (Cas) mode, you can derive the SP from another block through the **CAS_IN** input connector. In **Remote Cascade** (RCas) mode, you can derive the SP from a host computer that writes to **RCAS_IN**. The **PV_SCALE** parameter defines the range and the units for the SP.

In **Man** mode, the SP automatically tracks the PV value when you select the **SP_PV Track in Man I/O** option. The SP value is set equal to the PV value when the block is in manual mode, and the option is selected. You can disable this option in **Man** or **O/S** mode only.

The SP value is limited to the range defined by the setpoint high limit parameter **(SP_HI_LIM)** and the setpoint low limit parameter **(SP_LO_LIM)**.

In **Auto** mode, the rate at which a change in the SP is passed to OUT is limited by the values of the setpoint upward rate limit parameter **(SP_RATE_UP)** and the setpoint downward rate limit parameter **(SP_RATE_DN)**. A limit of zero prevents rate limiting, even in Auto mode.

6.3.7 Conversion and status calculation - Analog Output function block

The working setpoint **(WSP.VALUE)** is the setpoint value after limiting. You can reverse the conversion range, which reverses the **PV_SCALE** range to calculate the **OUT** parameter. You can perform this by selecting the **Increase to Close I/O** option. This inverts the **OUT** value with respect to the setpoint based on the **PV_SCALE** and **XD_SCALE**.

In **Auto** mode, the converted SP value is stored in the **OUT** parameter.

In **Man** mode, you can manually set **OUT**, and is used for setting the analog output defined by the **CHANNEL** parameter.

You can access the actuator position associated with the output channel through the **READBACK** parameter (in **OUT** units) and in the PV parameter (in engineering units). If the actuator does not support position feedback, the PV and **READBACK** values are based on the **OUT** parameter.

The working setpoint (WSP.VALUE) is the value normally used for the **BKCAL_OUT** parameter.

However, for those cases where the **READBACK** signal directly (linearly) indicates the **OUT** channel, you can choose to allow the PV to be used for **BKCAL_OUT** by selecting the Use PV for **BKCAL_OUT I/O** option.

Note: SP_PV Track in Man, Increase to Close, and Use PV for BKCAL_OUT are the only I/O options that the AO block supports. You can set I/O options in Manual or Out of service mode only.

6.3.8 Action on fault detection - Analog Output function block

You can configure the following parameters, to define the valve state when the **CAS_IN** input detects a bad status and the block is in **CAS** mode:

FSTATE_TIME: Indicates the AO block wait time to position the **OUT** value to the **FSTATE_VAL** value on detecting of a fault condition. When the block has a target mode of **CAS**, a fault condition is detected:

- if the **CAS_IN** has a **BAD** status

  or

- an **Initiate Fault State** substatus is received from the upstream block

FSTATE_VAL: Indicates **OUT** value after **FSTATE_TIME** elapses and the fault condition has not cleared.

6.3.9 Analog Output function block errors

The following table lists the conditions reported in the BLOCK_ERR parameter. The last column describes the errors supported in simulation.

Table 8: AO BLOCK_ERR conditions

Condition Number	Condition Name and Description	Supported in simulation
0	Other	No
1	Block Configuration Error: Remote shed option is not configured, or CHANNEL = zero.	Yes
2	Link Configuration Error	No. Algorithm must never see a link configuration error.
3	Simulate Active: Simulation is selected and the block is using a simulated value in its execution.	No
4	Local Override	No
5	Device Fault State Set	Yes. Set based on FAULT_STATE.
6	Device Needs Maintenance Soon	No
7	Input Failure/Process Variable has Bad Status: The hardware is bad, or a bad status is being simulated.	Yes
8	Output Failure: The output is bad based on a bad input.	Yes
9	Memory Failure	No
10	Lost Static Data	No
11	Lost NV Data	No

Condition Number	Condition Name and Description	Supported in simulation
12	Readback Check Failed	Yes. based on SIMULATE.status = Bad/ Device_Failure
13	Device Needs Maintenance Now	No
14	Power Up	No
15	Out of Service: The actual mode is out of service.	Yes

6.3.10 Analog Output function block modes

The Analog Output function block supports the following modes:

- **Man**- You can manually set the output to the I/O channel through the **OUT** parameter. You can use this mode for maintenance and troubleshooting.
- **Auto**- The block output (**OUT**) indicates the target operating point as specified by the setpoint (SP) parameter.
- **Cas**- The SP parameter is set by another function block through a connection to **CAS_IN**. You can use the SP value to set the **OUT** parameter.
- **RCas**- The SP is set by a host computer by writing to the **RCAS_IN** parameter. You can use the SP value to set the **OUT** parameter.
- **O/S**- The block is not processed. The output channel is maintained at the last value and the status of OUT is set to Bad: Out of Service. The **BLOCK_ERR** attribute shows Out of Service.
- **Initialization Manual (IMAN)**- The path to the output hardware is broken and the output remains at the last position.
- **Local Override (LO)** - The output of the block is not responding to **OUT** because the resource block is in LO mode or fault state action is active.

The target mode of the block may be restricted to one or more of the following modes: **Man**, **Auto**, **Cas**, **RCas**, or **O/S**.

6.3.11 Status Handling - Analog Output function block

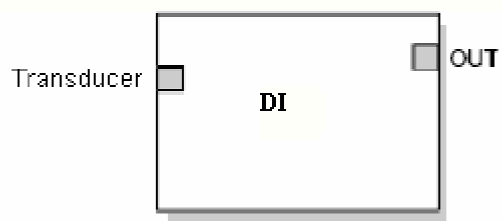
The Output or readback fault detection are indicated in the status of **PV**, **OUT**, and **BKCAL_OUT**. A limited SP condition is indicated in the **BKCAL_OUT** status. In a simulation environment, you can set the value and status for the PV and the READBACK parameter.

When the block is in **Cas** mode and the **CAS_IN** input is Bad, the block switches its mode to the next permitted mode.

6.4 Discrete Input function block

The DI block takes the manufacturer's discrete input data, selected by channel number, and makes it available to other function blocks at its output.

The DI block supports alarming, signal status calculation, mode control, and simulation. In the Automatic mode, the block's output parameter (OUT) indicates the process variable (PV) value and the status. In the Manual mode, you can manually set OUT. The output status indicates the Manual mode.



The FIELD_VAL_D shows the true on/off state of the hardware, using XD_STATE. The Invert I/O option can be used to do a Boolean NOT function between the field value and the output. A discrete value of zero (0) will be considered to be a logical zero (0) and an non-zero discrete value will be considered to be a logical (1) e.g. if invert is selected, the logical NOT of a non-zero field value would result in a zero (0) discrete output, the logical NOT of a zero field value would result in a discrete output value of one (1). PV_FTIME may be used to set the time that the hardware must be in one state before it gets passed to the PV_D. The PV_D is always the value that the block will place in OUT_D if the mode is Auto. If Man is allowed, someone may write a value to OUT_D. The PV_D and the OUT_D always have identical scaling. OUT_STATE provides scaling or PV_D.

6.4.1 Input - Discrete Input function block

The Discrete Input block receives an input from the Transducer block.

6.4.2 Output - Discrete Input function block

The calculated value from the Discrete Input block serves as input to other function blocks.

6.4.3 Simulation - Discrete Input function block

To work with the DI block in a simulation environment, you can perform any one of the following:

- change the mode of the block to manual and adjust the output value, or
- enable the simulation (set simulate.enable_disable= simulation active) on the configuration form and manually enter a value for the measurement value and its status.

By default, simulate.enable_disable is set to simulation active in simulation environment.

6.4.4 Parameters - Discrete Input function block

The following table lists the DI block parameters and their use in simulation:

Table 9: Definitions of Discrete Input function block system parameters

Parameter	Units	Description	Support in simulation
ACK_OPTION	None	Selects whether alarms associated with the block will be automatically acknowledged.	Supported
ALARM_SUM	None	Detects the current alert status, unacknowledged states, and disable states of the alarms associated with the block	Supported
ALERT_KEY	None	The identification number of the plant unit. This data may be used in the host for sorting alarms.	Supported. The value must be greater than 1 for reporting the alarms.
BLOCK_ALM	None	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alarm is entered in the sub-code field. The first alarm to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alarm-reporting task, another block alarm may be reported without clearing the Active status, if the sub- code has changed.	Supported. The Hardware information is not applicable in the simulation environment.
BLOCK_ERR	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Supported. Some block errors are not supported in simulation. See Table 2 .
CHANNEL	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Simulation algorithm does not use this parameter. However, you must configure a valid value for this parameter.
DISC_ALM	None	The status and time stamp associated with the discrete alarm.	Supported
DISC_LIM	None	State of discrete input which will generate an alarm.	Supported
DISC_PRI	None	Priority of the discrete alarm.	Supported
FIELD_VAL_D	None	Raw value of the field device discrete input, with a status reflecting the Transducer condition.	As the block is in simulation, by default it contains the simulated input value and the status. If the block is not in simulation field_val.value is set to default value 0 and field_val.status is set to default value bad.
GRANT_DENY	None	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm Parameters	Supported
IO_OPTS	None	Identifies user selectable options for altering the Input and Output block processing.	Supported
MODE_BLK	None	Represents the mode record of the block. Contains the Actual, Target, Permitted, and Normal modes.	Supported
OUT_D	None	The primary discrete value calculated as a result of executing the function.	Supported

Parameter	Units	Description	Support in simulation
OUT_STATE	None	Index to the text describing the states of a discrete output.	Supported
PV_D	None	Either the primary discrete value for use in executing the function, or a process value associated with it. May also be calculated from the READBACK_D value of a DO block.	Supported
PV_FTIME	Seconds	Defines the time constant of a Single exponential filter for the Process Variable in seconds.	Supported
SIMULATE_D		Allows the transducer discrete input or output to the block to be manually supplied when simulate is enabled. When simulation is disabled, the simulate value and status track the actual value and status.	Supported
STATUS_OPTS		Options which the user may select in the block processing of status.	Supported
STRATEGY	None	Assists in grouping blocks. This data is not checked or processed by the block.	Supported
ST_REV	None	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.	Supported
TAG_DESC	None	Serves as user defined description of the block.	Supported
UPDATE_EVT	None	Represents an alert generated by any change to the static data.	Supported

6.4.5 Discrete Input function block errors

The following table lists the conditions reported in the BLOCK_ERR parameter. The last column describes the errors supported in simulation.

Table 10: Discrete Input function block_ERR conditions

Condition Number	Condition Name and Description	Supported in simulation
0	Other	No
1	Block Configuration Error: The selected channel carries a measurement that is incompatible with the engineering units selected in XD_SCALE, the L_TYPE parameter is not configured , or channel = zero.	Yes
2	Link Configuration Error	No. Algorithm must never see a link configuration error.
3	Simulate Active: Simulation is selected and the block is using a simulated value in its execution.	No
4	Local Override	No
5	Device Fault State Set	No
6	Device Needs Maintenance Soon	No
7	Input Failure/Process Variable has Bad Status: The hardware is bad, or a bad status is being simulated.	Yes
8	Output Failure: The output is bad based on a bad input.	Yes

Condition Number	Condition Name and Description	Supported in simulation
9	Memory Failure	No
10	Lost Static Data	No
11	Lost NV Data	No
12	Readback Check Failed	No
13	Device Needs Maintenance Now	No
14	Power Up	No
15	Out of Service: The actual mode is out of service.	Yes

6.4.6 Discrete Input function block modes

The Discrete Input function Block supports three modes of operation as defined by the MODE_BLK parameter:

- Manual (Man)—You can manually set the block output (OUT).
- Automatic (Auto)—OUT indicates the discrete input measurement or the simulated value in the simulation (simulate.enable_disable= simulation active) environment. You can set a good status for simulate.simulate_status and a valid value in simulate.simulate_value. If simulation is not selected (simulate.enable_disable= simulation disabled), OUT indicates a bad status.
- Out of Service (O/S)—The block is not processed. FIELD_VAL and PV are not updated and the OUT status is set to Bad: Out of Service. The BLOCK_ERR parameter indicates Out of Service. In this mode, you can change all configured parameters. The target mode of a block may be restricted to one or more of the supported modes.

6.4.7 Alarm detection - Discrete Input function block

Standard block alarms and standard discrete alarms applied on OUT_D are supported by the DI block.

A block alarm is generated when the BLOCK_ERR has an error bit set.

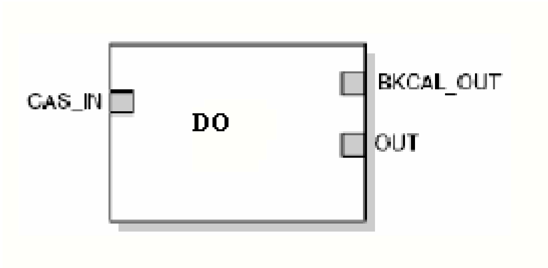
6.4.8 Status Handling - Discrete Input function block

The following STATUS_OPTS is supported by the DI block:

Propagate Fault Forward: If the status from the actuator is Bad, Device failure or Fault State Active or Local Override is active, propagate this as Bad, Device Failure or Good Cascade, Fault State Active or Local Override to BKCAL_OUT respectively without generating an alarm. . The use of these sub-statuses in BKCAL_OUT is determined by this option. Through this option, the user may determine whether alarming (sending of an alert) will be done by the block or propagated upstream for alarming.

6.5 Discrete Output function block

The DO block converts the value in SP_D to something useful for the hardware found at the CHANNEL selection.



The SP_D supports the full cascade sub-function. Cas mode must be used to transfer the output of another block to the SP_D of the DO. There are additional I/O options which will cause the SP_D value to track the PV_D value when the block is in an actual mode of LO or when the target mode for the block is Man. If the hardware supports a readback value, it is used for READBACK_D, which, after accounting for the Invert I/O option, acts as the PV_D for this block. If not supported, READBACK_D is generated from OUT_D. The OUT_D and READBACK_D parameters both use XD_STATE. The PV_D and SP_D use PV_STATE.

6.5.1 Input - Discrete Output function block

SP_D - The discrete setpoint of this block.

CAS_IN_D: Indicates the setpoint value from another function block.

6.5.2 Output - Discrete Output function block

BKCAL_OUT_D: Indicates the value and the status required by the BKCAL_IN input of another block to prevent reset windup and to provide bumpless transfer to closed loop control.

OUT_D: Indicates the block output value and the status.

6.5.3 Simulation - Discrete Output function block

To work with the DI block in a simulation environment, you can perform any one of the following:

- change the mode of the block to manual and adjust the output value, or
- enable the simulation (set simulate.enable_disable= simulation active) on the configuration form and manually enter a value for the measurement value and its status.

By default, simulate.enable_disable is set to simulation active in simulation environment.

6.5.4 Parameters - Discrete Output function block

The following table lists the DO block parameters and their use in simulation:

Table 11: Definitions of Discrete Output function block system parameters

Parameter	Units	Description	Support in simulation
ALERT_KEY	None	The identification number of the plant unit. This data may be used in the host for sorting alarms.	Supported. The value must be greater than 1 for reporting the alarms.

Parameter	Units	Description	Support in simulation
BLOCK_ALM	None	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alarm is entered in the sub-code field. The first alarm to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alarm-reporting task, another block alarm may be reported without clearing the Active status, if the sub- code has changed.	Supported. The Hardware information is not applicable in the simulation environment.
BKCAL_OUT_D	None	The output value and status provided to an upstream discrete block. This information is used to provide bumpless transfer to closed loop control.	Supported
BLOCK_ERR	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Supported Some block errors are not supported in Simulation.
CAS_IN_D	None	This parameter is the remote setpoint value of a discrete block, which must come from another Fieldbus block, or a DCS block through a defined link.	Supported
CHANNEL	None	The number of the logical hardware channel that is connected to this I/O block. It defines the transducer used to connect to the physical world.	Simulation algorithm does not use this parameter. However, you must configure a valid value for this parameter.
FSTATE_TIME	seconds	Represents the reaction time in seconds from the detection of a failure at the output block remote set point to the output block action, if the condition still exists.	Supported
FIELD_VAL_D	None	Raw value of the field device discrete input, with a status reflecting the Transducer condition.	As the block is in simulation, by default it contains the simulated input value and the status. If the block is not in simulation field_val.value is set to default value 0 and field_val.status is set to default value bad.
GRANT_DENY	None	Defines options for controlling access of host computer or local control panels to the block's operating, tuning, and alarm parameters.	Supported
IO_OPTS	None	Identifies user-selectable options for altering the Input and Output block processing.	Supported
MODE_BLK	None	Represents the mode record of the block. Contains the Actual, Target, Permitted, and Normal modes.	Supported
OUT_D	None	The primary discrete value calculated as a result of executing the function.	Supported
PV_D	None	Either the primary discrete value for use in executing the function, or a process value associated with it. May also be calculated from the READBACK_D value of a DO block.	Supported

Parameter	Units	Description	Support in simulation
PV_STATE	None	Index to the text describing the states of a discrete PV.	Supported
RCAS_IN_D	None	Target setpoint and status provided by a supervisory Host to a discrete control or output block.	Supported
RCAS_OUT_D	None	Block setpoint and status after ramping - provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change.	Supported
READBACK_D	None	This indicates the readback of the actual continuous valve or other actuator position, in transducer units.	Supported
SHED_OPT	None	Defines action to be taken on remote control device timeout.	Supported
SIMULATE_D		Allows the transducer analog input or output to the block to be manually supplied, when SIMULATE is enabled. When SIMULATE is disabled, the simulate value and status track the actual value and status.	Enable/disable is selected by default in simulation. The simulation active alarm is not reported.
SP_D	None	The discrete setpoint of this block.	Supported
STATUS_OPTS	None	Options which the user may select in the block processing of status.	Supported
STRATEGY	None	Assists in grouping blocks. This data is not checked or processed by the block.	Supported
ST_REV	None	Defines the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.	Supported
TAG_DESC	None	Serves as user defined description of the block.	Supported
UPDATE_EVT	None	Represents an alert generated by any change to the static data.	Supported

6.5.5 Discrete Output function block errors

The "Definitions of Discrete Input function block system parameters" table lists the conditions reported in the BLOCK_ERR parameter.

6.5.6 Discrete Output function block modes

The DO function block supports the following modes:

- **Man**—You can manually set the block output (**OUT**).
- **Auto**—You can manually set the SP and the block algorithm calculates the **OUT**.
- **Cas** —The SP is calculated in another block and provided to the PID block through the **CAS_IN** connection.
- **RCas**—The SP is provided by a host computer that writes to the **RCAS_IN** parameter.
- **Local Override (LO)**—The track function is active. TRK_VAL sets OUT. The BLOCK_ERR parameter shows Local override.
- **IMan**— The output path is not complete (for example, the cascade-to-slave path might not be open). In **IMan** mode, **OUT** tracks **BKCAL_IN**.

- **O/S**—The block is not processed. The Out status is set to Bad: Out of Service. The BLOCK_ERR parameter shows Out of Service.

You can configure the Man, Auto, Cas and O/S modes as permitted modes for operator entry.

6.5.7 Alarm detection - Discrete Output function block

DO block supports only the Standard block alarms.

6.5.8 Status Handling - Discrete Output function block

The following STATUS_OPTS is supported by the DO block:

Propagate Fault Forward: If the status from the actuator is Bad, Device failure or Fault State Active or Local Override is active, propagate this as Bad, Device Failure or Good Cascade, Fault State Active or Local Override to BKCAL_OUT respectively without generating an alarm. . The use of these sub-statuses in BKCAL_OUT is determined by this option. Through this option, the user may determine whether alarming (sending of an alert) will be done by the block or propagated upstream for alarming.

7 Troubleshooting

For more troubleshooting details, see Series C FIM Troubleshooting section in the *Series C Fieldbus Interface Module User's Guide*.

Related topics

"Understanding SIM-FFD error messages" on page 76

7.1 Understanding SIM-FFD error messages

7.1.1 3364

Problem	Maximum number of device Instances Exceeded@@DEVLoadError
Cause	There are already 16 devices present in the link.
Resolution	Try deleting at least one of the device from the monitoring view and re-load the device.

7.1.2 3365

Problem	No Memory Manager Buffers Available@@NoMemManBuf
Cause	There are no sufficient memory buffers available in the SIM-FFD environment.
Resolution	Try loading the device.

7.1.3 10760

Problem	The device cannot be loaded because the device seeding failed@@CL_FF_DEV_VALD_SEED_FAILED
Cause	You cannot create the device in the simulation environment.
Resolution	Try loading the device.

7.1.4 10825

Problem	Exceeds Instantiable Blocks limit for this Device.@@ERR_CL_FF_DEV_CAPACITY_EXCEEDED
Cause	The number of Instantiable blocks in the Project device reached the maximum limit.
Resolution	Delete some of the Instantiable blocks for the Project device.

7.1.5 10139

Problem	Block Load FAILED@@CL_BLKLOADFAIL
Cause	• Parent block may not be loaded. • There is a block configuration error.
Resolution	• Load the parent block. • Ensure that the block configuration is correct and complete.

7.1.6 10342

Problem	Load of Block Parent has failed, Load Canceled for the block.@@CL_PARNTLDFAIL
Cause	If the parent block fails to load, you cannot load the child block.

Resolution	Ensure that the parent block is loaded.

7.1.7 3191

Problem	Parameter check error@@FMSParamCheck
Cause	The parameter value may be: • Incorrect • invalid data type • exceeded applicable limits
Resolution	Ensure to enter correct value within limits of the specified data type.

7.1.8 3192

Problem	Exceeds parameter limits@@FMSParamLimits
Cause	The parameter value exceeds the applicable limits.
Resolution	Enter correct value within the limits.

7.1.9 3193

Problem	Wrong mode for request@@FMSWrongMode
Cause	Cannot change the parameter value in the particular mode.
Resolution	Set the target mode to OOS and try to change the value.

7.1.10 3194

Problem	Write is prohibited by write lock switch@@FMSWriteLock
Cause	Write lock is set on the Resource block.
Resolution	Set writelock to unlocked and try changing the value.

7.1.11 3195

Problem	Data value is read-only@@FMSReadOnly
Cause	Cannot change the parameter value because it is read-only.
Resolution	None.

7.1.12 3196

Problem	Instantiate not supported or unsuccessful@@FMSNoInstantiate
Cause	Instantiation is not supported for this device.
Resolution	Use a different device if you want to use block instantiation.

7.1.13 10195

Problem	Can not perform the operation while loaded in Monitor. @@ERR_CL_BLOCK_LOADED_IN_MONITOR
Cause	The block is loaded in a Monitoring view.
Resolution	Delete the block from the Monitoring view and change the parameter.

7.1.14 10199

Problem	This operation is not supported on redundant module. @@ERR_CL_OP_NOT_SUPPORTED_IN_RED
Cause	The function block is configured as a redundant block.
Resolution	Ensure that the function block is configured as a non-redundant block. Ensure not to perform the operation on a redundant block.

7.1.15 3171

Problem	FMS Service Illegal Parameter@@FMSServIllegalParam
Cause	The parameter does not support the value.
Resolution	Ensure that a valid value is assigned to the parameter.

7.1.16 2072

Problem	Read Only 'Parameter@@ReadOnlyPar' Indicates that the parameter is read-only and cannot be changed.
Cause	By design, many block parameters, including many internal parameters, have this characteristic.
Resolution	None

7.1.17 3305

Problem	All links must be inactive@@LinkActive
Cause	The links are active.
Resolution	Ensure that all links are inactive before you shutdown.

7.1.18 2007

Problem	Access Level 'Invalid@@InvalidAccess' Indicates a user access level violation.
Cause	Indicates that the user does not have access rights to perform the operation.
Resolution	In most cases, an access lock error indicates that this operation must be performed by a higher authority. Note the following relationship with respect to access level permissions: ENGINEER > SUPERVISOR > OPERATOR.

7.1.19 7019

Problem	Data Value could not be converted into the type specified by the client
Cause	The value assigned to the parameter does not match with the data type of the parameter.
Resolution	Ensure that you enter a correct value that matches with the data type of the parameter.

7.1.20 3303

Problem	Not in Proper State For Transition@@WrongState
Cause	• The transition Link is not in correct state. • The link does not support the operation.
Resolution	Select the correct link state. For example, • If the link state is up, set state=Down • If the link state is down, set state=startup

7.1.21 7022

Problem	Not allowed to write to a SR parameter while monitoring
Cause	The parameter is a SR parameter and it cannot be changed from monitoring.
Resolution	None

7.1.22 3312

Problem	MIB Not Online@@NotOnline
Cause	Link is not active to perform the operation. Therefore, you cannot seed the device.
Resolution	Activate the link.

8 Performance and capacity considerations

The following specification considerations are arrived at by conducting Design of Experiments on a Simulation Node installed on Dell Inc., PowerEdge SC 1430 machine.

Attention

The Specification and Technical information is subject to change without notice and is superseded by information in applicable Experion product Specification and Technical data documents. Hence, for each Experion release, you are recommended to refer the applicable Specification and Technical data documents.

Description	FIM	SIM-FFD [FIM]
Number of H1 Networks per FIM	4	4
Maximum No. of FF Devices per H1 Network	16	16
Maximum Number of FIMs per Controller	15	15
Maximum Number of FIMs per Server [L1 FTE]	125	125
Maximum Number of FIMs per Server [L1 ControlNet]	NA	NA
Maximum Number of Blocks per FF Device	30	30
Maximum Number of Blocks per FIM	1200	* 1188
Agents per Link	100	100
Max. No. of VCR on H1/Link	128	128
Max. No. of Block Types per FIM	100	* 30
Macro cycle periods supported	250, 500ms; 1, 2, 4, 8,16, & 32 sec.	250, 500 ms; 1, 2, 4, 8, 16, & 32 sec.
Max. Peer connections per FIM	5	5
Max number of SIM-FFDs per Simulation Node	NA	11
Average Display PPS observer when 11 SIM-FFD loaded on one Simulation Node	Na	~900

Note:

- The performance and capacity specifications are identical for SIM-FFD and FIM.
- * indicates the supported value.