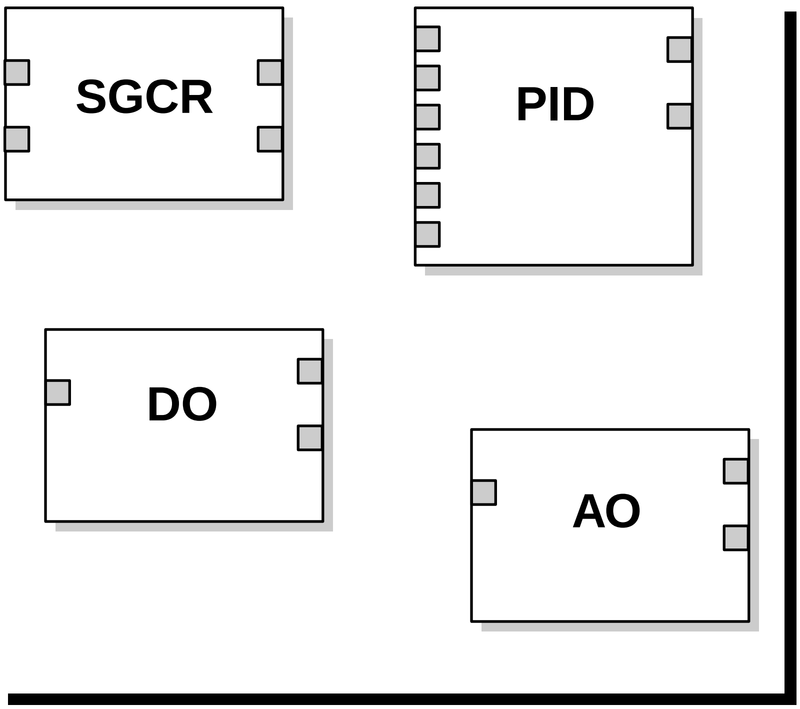
00809-0100-4783
English
Rev. BA

Foundation™ Fieldbus Blocks



ROSEMOUNT®
FISHER-ROSEMOUNT™

FOUNDATION

Product
Manual

Foundation™ Fieldbus Blocks

NOTICE

Read this manual before working with the product. For personal and system safety, and for optimum product performance, make sure you thoroughly understand the contents before installing, using, or maintaining this product.

Within the United States, Rosemount Inc. has two toll-free assistance numbers:

Customer Central
Technical support, quoting, and order-related questions.

1-800-999-9307 (7:00 am to 7:00 pm CST)

North American Response Center
Equipment service needs.

1-800-654-7768 (24 hours—includes Canada)

Outside of the United States, contact your local Rosemount representative.

⚠CAUTION

The products described in this document are NOT designed for nuclear-qualified applications. Using non-nuclear qualified products in applications that require nuclear-qualified hardware or products may cause inaccurate readings.

For information on Rosemount nuclear-qualified products, contact your local Rosemount Sales Representative.

Rosemount Inc.
8200 Market Boulevard
Chanhassen, MN 55317 USA
Tel 1-800-999-9307
Fax (952) 949-7001

Fisher-Rosemount satisfies all obligations coming from legislation to harmonise product requirements in the European Union.

00809-0100-4783 Rev.BA

www.rosemount.com

ROSEMOUNT®
FISHER-ROSEMOUNT™

Table of Contents

SECTION 5
Discrete Input (DI) Function Block

SECTION 6
Discrete Output (DO) Function Block

SECTION 7
Integrator (INT) Function Block

SECTION 8
Input Selector (ISEL) Function Block

SECTION 9
Proportional/Integral/ Derivative (PID) Function Block

Section 1

FOUNDATION™ Fieldbus Technology and Fieldbus Function Blocks

OVERVIEW

This section introduces fieldbus systems that are common to all fieldbus devices.

INTRODUCTION

A fieldbus system is a distributed system composed of field devices and control and monitoring equipment integrated into the physical environment of a plant or factory. Fieldbus devices work together to provide I/O and control for automated processes and operations. The Fieldbus FOUNDATION provides a framework for describing these systems as a collection of physical devices interconnected by a fieldbus network. One of the ways the physical devices are used is to perform their portion of the total system operation by implementing one or more function blocks.

Function Blocks

Function blocks within the fieldbus device perform the various functions required for process control. Because each system is different, the mix and configuration of functions are different. Therefore, the Fieldbus FOUNDATION has designed a range of function blocks, each addressing a different need.

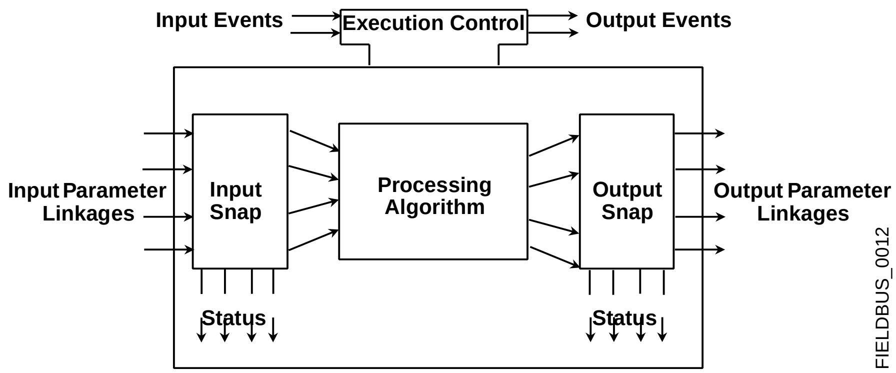
Function blocks perform process control functions, such as analog input (AI) and analog output (AO) functions as well as proportional-integral-derivative (PID) functions. The standard function blocks provide a common structure for defining function block inputs, outputs, control parameters, events, alarms, and modes, and combining them into a process that can be implemented within a single device or over the fieldbus network. This simplifies the identification of characteristics that are common to function blocks.

The Fieldbus FOUNDATION has established the function blocks by defining a small set of parameters used in all function blocks called universal parameters. The FOUNDATION has also defined a standard set of function block classes, such as input, output, control, and calculation blocks. Each of these classes also has a small set of parameters established for it. They have also published definitions for transducer blocks commonly used with standard function blocks. Examples include temperature, pressure, level, and flow transducer blocks.

The FOUNDATION specifications and definitions allow vendors to add their own parameters by importing and subclassify specified classes. This approach permits extending function block definitions as new requirements are discovered and as technology advances.

Figure 1-1 illustrates the internal structure of a function block. When execution begins, input parameter values from other blocks are snapped-in by the block. The input snap process ensures that these values do not change during the block execution. New values received for these parameters do not affect the snapped values and will not be used by the function block during the current execution.

Figure 1-1. Function Block Internal Structure



Once the inputs are snapped, the algorithm operates on them, generating outputs as it progresses. Algorithm executions are controlled through the setting of contained parameters. Contained parameters are internal to function blocks and do not appear as normal input and output parameters. However, they may be accessed and modified remotely, as specified by the function block.

Input events may affect the operation of the algorithm. An execution control function regulates the receipt of input events and the generation of output events during execution of the algorithm. Upon completion of the algorithm, the data internal to the block is saved for use in the next execution, and the output data is snapped, releasing it for use by other function blocks.

A block is a tagged logical processing unit. The tag is the name of the block. System management services locate a block by its tag. Thus the service personnel need only know the tag of the block to access or change the appropriate block parameters.

Function blocks are also capable of performing short-term data collection and storage for reviewing their behavior.

Device Descriptions

Device Descriptions are specified tool definitions that are associated with the function blocks. Device descriptions provide for the definition and description of the function blocks and their parameters.

To promote consistency of definition and understanding, descriptive information, such as data type and length, is maintained in the device description. Device Descriptions are written using an open language called the Device Description Language (DDL). Parameter transfers between function blocks can be easily verified because all parameters are described using the same language. Once written, the device description can be stored on an external medium, such as a CD-ROM or diskette. Users can then read the device description from the external medium. The use of an open language in the device description permits interoperability of function blocks within devices from various vendors. Additionally, human interface devices, such as operator consoles and computers, do not have to be programmed specifically for each type of device on the bus. Instead their displays and interactions with devices are driven from the device descriptions.

Device descriptions may also include a set of processing routines called methods. Methods provide a procedure for accessing and manipulating parameters within a device.

BLOCK OPERATION

In addition to function blocks, fieldbus devices contain two other block types to support the function blocks. These are the resource block and the transducer block. The resource block contains the hardware specific characteristics associated with a device. Transducer blocks couple the function blocks to local input/output functions.

Instrument-Specific Function Blocks

Resource Blocks

Resource blocks contain the hardware specific characteristics associated with a device; they have no input or output parameters. The algorithm within a resource block monitors and controls the general operation of the physical device hardware. The execution of this algorithm is dependent on the characteristics of the physical device, as defined by the manufacturer. As a result of this activity, the algorithm may cause the generation of events. There is only one resource block defined for a device. For example, when the mode of a resource block is “out of service,” it impacts all of the other blocks.

Transducer Blocks

Transducer blocks connect function blocks to local input/output functions. They read sensor hardware and write to effector (actuator) hardware. This permits the transducer block to execute as frequently as necessary to obtain good data from sensors and ensure proper writes to the actuator without burdening the function blocks that use the data. The transducer block also isolates the function block from the vendor specific characteristics of the physical I/O.

Alerts

When an alert occurs, execution control sends an event notification and waits a specified period of time for an acknowledgment to be received. This occurs even if the condition that caused the alert no longer exists. If the acknowledgment is not received within the pre-specified time-out period, the event notification is retransmitted. This assures that alert messages are not lost.

Two types of alerts are defined for the block, events and alarms. Events are used to report a status change when a block leaves a particular state, such as when a parameter crosses a threshold. Alarms not only report a status change when a block leaves a particular state, but also report when it returns back to that state.

NETWORK COMMUNICATION

Figure 1-2 illustrates a simple fieldbus network consisting of a single segment (link).

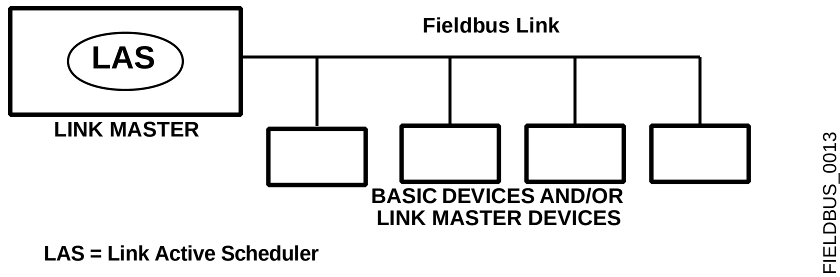


Figure 1-2. Simple, Single-Link Fieldbus Network.

Link Active Scheduler (LAS)

All links have one and only one Link Active Scheduler (LAS). The LAS operates as the bus arbiter for the link. The LAS does the following:

- recognizes and adds new devices to the link.
- removes non-responsive devices from the link.
- distributes Data Link (DL) and Link Scheduling (LS) time on the link. Data Link Time is a network-wide time periodically distributed by the LAS to synchronize all device clocks on the bus. Link Scheduling time is a link-specific time represented as an offset from Data Link Time. It is used to indicate when the LAS on each link begins and repeats its schedule. It is used by system management to synchronize function block execution with the data transfers scheduled by the LAS.
- polls devices for process loop data at scheduled transmission times.
- distributes a priority-driven token to devices between scheduled transmissions.

Any device on the link may become the LAS, as long as it is capable. The devices that are capable of becoming the LAS are called link master devices. All other devices are referred to as basic devices. When a segment first starts up, or upon failure of the existing LAS, the link master devices on the segment bid to become the LAS. The link master that wins the bid begins operating as the LAS immediately upon completion of the bidding process. Link masters that do not become the LAS act as basic devices. However, the link masters can act as LAS backups by monitoring the link for failure of the LAS and then bidding to become the LAS when a LAS failure is detected.

Only one device can communicate at a time. Permission to communicate on the bus is controlled by a centralized token passed between devices by the LAS. Only the device with the token can communicate. The LAS maintains a list of all devices that need access to the bus. This list is called the “Live List.”

Two types of tokens are used by the LAS. A time-critical token, compel data (CD), is sent by the LAS according to a schedule. A non-time critical token, pass token (PT), is sent by the LAS to each device in ascending numerical order according to address.

Device Addressing

Fieldbus uses addresses between 0 and 255. Addresses 0 through 15 are reserved for group addressing and for use by the data link layer. For all Fisher-Rosemount fieldbus devices addresses 20 through 35 are available to the device. If there are two or more devices with the same address, the first device to start will use its programmed address. Each of the other devices will be given one of four temporary addresses between 248 and 251. If a temporary address is not available, the device will be unavailable until a temporary address becomes available.

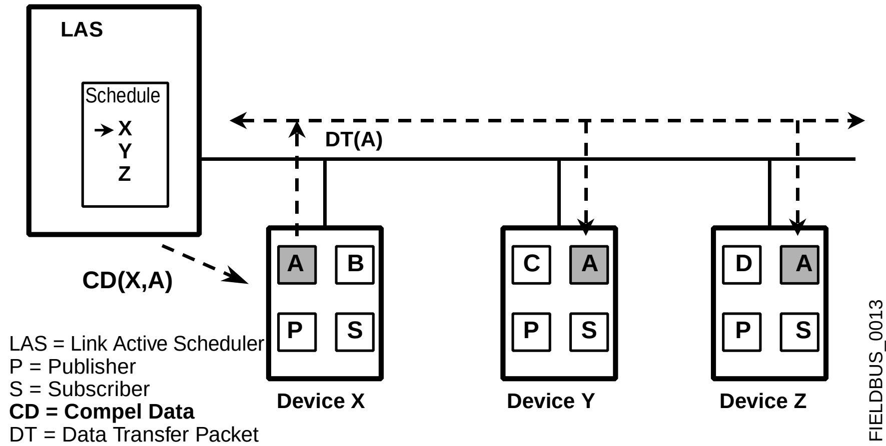
Scheduled Transfers

Information is transferred between devices over the fieldbus using three different types of reporting.

- **Publisher/Subscriber:** This type of reporting is used to transfer critical process loop data, such as the process variable. The data producers (publishers) post the data in a buffer that is transmitted to the subscriber (S), when the publisher receives the Compel data. The buffer contains only one copy of the data. New data completely overwrites previous data. Updates to published data are transferred simultaneously to all subscribers in a single broadcast. Transfers of this type can be scheduled on a precisely periodic basis.
- **Report Distribution:** This type of reporting is used to broadcast and multicast event and trend reports. The destination address may be predefined so that all reports are sent to the same address, or it may be provided separately with each report. Transfers of this type are queued. They are delivered to the receivers in the order transmitted, although there may be gaps due to corrupted transfers. These transfers are unscheduled and occur in between scheduled transfers at a given priority.
- **Client/Server:** This type of reporting is used for request/response exchanges between pairs of devices. Like Report Distribution reporting, the transfers are queued, unscheduled, and prioritized. Queued means the messages are sent and received in the order submitted for transmission, according to their priority, without overwriting previous messages. However, unlike Report Distribution, these transfers are flow controlled and employ a retransmission procedure to recover from corrupted transfers.

Figure 1-3 diagrams the method of scheduled data transfer. Scheduled data transfers are typically used for the regular cyclic transfer of process loop data between devices on the fieldbus. Scheduled transfers use publisher/subscriber type of reporting for data transfer. The Link Active Scheduler maintains a list of transmit times for all publishers in all devices that need to be cyclically transmitted. When it is time for a device to publish data, the LAS issues a Compel Data (CD) message to the device. Upon receipt of the CD, the device broadcasts or “publishes” the data to all devices on the fieldbus. Any device that is configured to receive the data is called a “subscriber.”

Figure 1-3. Scheduled Data Transfer.

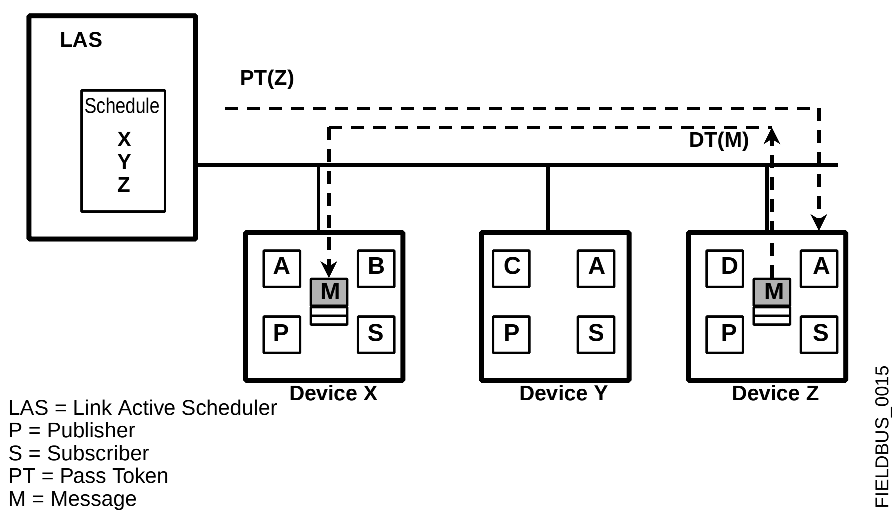


Unscheduled Transfers

Figure 1-4 diagrams an unscheduled transfer. Unscheduled transfers are used for things like user-initiated changes, including set point changes, mode changes, tuning changes, and upload/download. Unscheduled transfers use either report distribution or client/server type of reporting for transferring data.

All of the devices on the fieldbus are given a chance to send unscheduled messages between transmissions of scheduled data. The LAS grants permission to a device to use the fieldbus by issuing a pass token (PT) message to the device. When the device receives the PT, it is allowed to send messages until it has finished or until the "maximum token hold time" has expired, whichever is the shorter time. The message may be sent to a single destination or to multiple destinations.

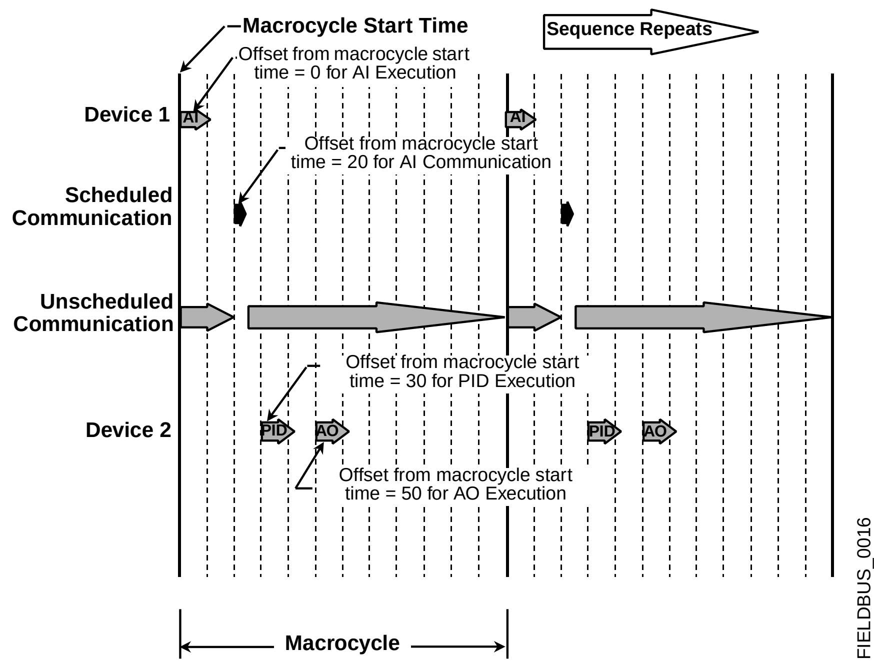
Figure 1-4. Unscheduled Data Transfer.



Function Block Scheduling

Figure 1-5 shows an example of a link schedule. A single iteration of the link-wide schedule is called the macrocycle. When the system is configured and the function blocks are linked, a master link-wide schedule is created for the LAS. Each device maintains its portion of the link-wide schedule, known as the Function Block Schedule. The Function Block Schedule indicates when the function blocks for the device are to be executed. The scheduled execution time for each function block is represented as an offset from the beginning of the macrocycle start time.

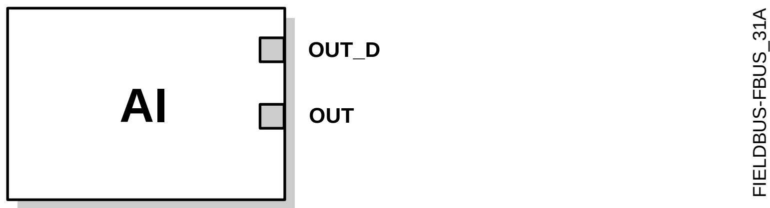
Figure 1-5. Example Link Schedule Showing scheduled and Unscheduled Communication.



To support synchronization of schedules, periodically Link Scheduling (LS) time is distributed. The beginning of the macrocycle represents a common starting time for all Function Block schedules on a link and for the LAS link-wide schedule. This permits function block executions and their corresponding data transfers to be synchronized in time.

Section 2

Analog Input (AI) Function Block



OUT = The block output value and status
OUT_D = Discrete output that signals a selected alarm condition

The Analog Input (AI) function block processes field device measurements and makes them available to 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 measuring device may have several measurements or derived values available in different channels. Use the channel number to define the variable that the AI block processes.

The AI block supports alarming, signal scaling, signal filtering, signal status calculation, mode control, and simulation. In Automatic mode, the block's output parameter (OUT) reflects the process variable (PV) value and status. In Manual mode, OUT may be set manually. The Manual mode is reflected on the output status. A discrete output (OUT_D) is provided to indicate whether a selected alarm condition is active. Alarm detection is based on the OUT value and user specified alarm limits. Figure 2-1 on page 2-5 illustrates the internal components of the AI function block, and Table 2-1 lists the AI block parameters and their units of measure, descriptions, and index numbers.

TABLE 2-1. Definitions of Analog Input Function Block System Parameters

Parameter	Index Number	Units	Description
ACK_OPTION	23	None	Used to set auto acknowledgment of alarms.
ALARM_HYS	24	Percent	The amount the alarm value must return within the alarm limit before the associated active alarm condition clears.
ALARM_SEL	38	None	Used to select the process alarm conditions that will cause the OUT_D parameter to be set.

TABLE 2-1. Definitions of Analog Input Function Block System Parameters

Parameter	Index Number	Units	Description
ALARM_SUM	22	None	The summary alarm is used for all process alarms in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.
ALERT_KEY	04	None	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
BLOCK_ALM	21	None	The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.
BLOCK_ERR	06	None	This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown.
CHANNEL	15	None	The CHANNEL value is used to select the measurement value. Refer to the appropriate device manual for information about the specific channels available in each device. You must configure the CHANNEL parameter before you can configure the XD_SCALE parameter.
FIELD_VAL	19	Percent	The value and status from the transducer block or from the simulated input when simulation is enabled.
GRANT_DENY	12	None	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block. Not used by device.
HI_ALM	34	None	The HI alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.
HI_HI_ALM	33	None	The HI HI alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.

TABLE 2-1. Definitions of Analog Input Function Block System Parameters

Parameter	Index Number	Units	Description
HI_HI_LIM	26	EU of PV_SCALE	The setting for the alarm limit used to detect the HI HI alarm condition.
HI_HI_PRI	25	None	The priority of the HI HI alarm.
HI_LIM	28	EU of PV_SCALE	The setting for the alarm limit used to detect the HI alarm condition.
HI_PRI	27	None	The priority of the HI alarm.
IO_OPTS	13	None	Allows the selection of input/output options used to alter the PV. Low cutoff enabled is the only selectable option.
L_TYPE	16	None	Linearization type. Determines whether the field value is used directly (Direct), is converted linearly (Indirect), or is converted with the square root (Indirect Square Root).
LO_ALM	35	None	The LO alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.
LO_LIM	30	EU of PV_SCALE	The setting for the alarm limit used to detect the LO alarm condition.
LO_LO_ALM	36	None	The LO LO alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.
LO_LO_LIM	32	EU of PV_SCALE	The setting for the alarm limit used to detect the LO LO alarm condition.
LO_LO_PRI	31	None	The priority of the LO LO alarm.
LO_PRI	29	None	The priority of the LO alarm.
LOW_CUT	17	%	If percentage value of transducer input fails below this, PV = 0.
MODE_BLK	05	None	The actual, target, permitted, and normal modes of the block. Target: The mode to “go to” Actual: The mode the “block is currently in” Permitted: Allowed modes that target may take on Normal: Most common mode for target
OUT	08	EU of OUT_SCALE	The block output value and status.
OUT_D	37	None	Discrete output to indicate a selected alarm condition.
OUT_SCALE	11	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT.
PV	07	EU of XD_SCALE	The process variable used in block execution.
PV_FTIME	18	Seconds	The time constant of the first-order PV filter. It is the time required for a 63% change in the IN value.

TABLE 2-1. Definitions of Analog Input Function Block System Parameters

Parameter	Index Number	Units	Description
SIMULATE	09	None	A group of data that contains the current transducer value and status, the simulated transducer value and status, and the enable/disable bit.
STRATEGY	03	None	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ST_REV	01	None	The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in the block is changed.
TAG_DESC	02	None	The user description of the intended application of the block.
UPDATE_EVT	20	None	This alert is generated by any change to the static data.
VAR_INDEX	39	% of OUT Range	The average absolute error between the PV and its previous mean value over that evaluation time defined by VAR_SCAN.
VAR_SCAN	40	Seconds	The time over which the VAR_INDEX is evaluated.
XD_SCALE	10	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with the channel input value. The XD_SCALE units code must match the units code of the measurement channel in the transducer block. If the units do not match, the block will not transition to MAN or AUTO

Simulation

To support testing, you can either change the mode of the block to manual and adjust the output value, or you can enable simulation through the configuration tool and manually enter a value for the measurement value and its status. In both cases, you must first set the ENABLE jumper on the field device.

NOTE
All fieldbus instruments have a simulation jumper. As a safety measure, the jumper has to be reset every time there is a power interruption. This measure is to prevent devices that went through simulation in the staging process from being installed with simulation enabled.

With simulation enabled, the actual measurement value has no impact on the OUT value or the status.

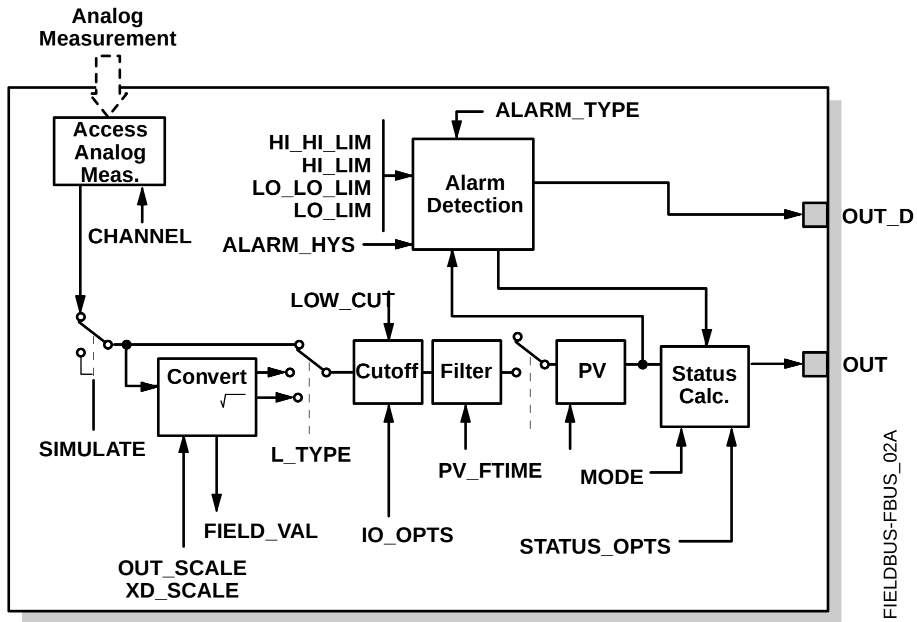


Figure 2-1. Analog Input Function Block Schematic

NOTES:
OUT = block output value and status.
OUT_D = discrete output that signals a selected alarm condition.

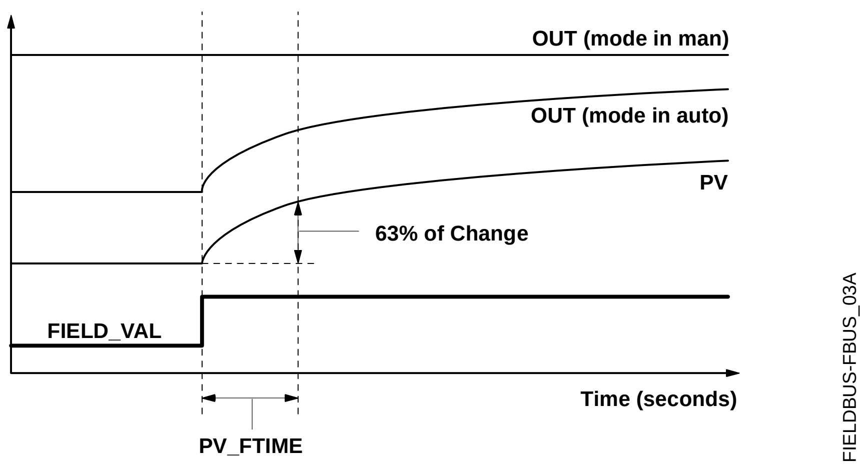


Figure 2-2. Analog Input Function Block Timing Diagram

Filtering

The filtering feature changes the response time of the device to smooth variations in output readings caused by rapid changes in input. You can adjust the filter time constant (in seconds) using the PV_FTIME parameter. Set the filter time constant to zero to disable the filter feature.

Signal Conversion

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

$$FIELD\_VAL = \frac{100 \times (\text{Channel Value} - EU^*@0\%)}{(EU^*@100\% - EU^*@0\%)}$$

*** XD_SCALE values**

You can choose from direct, indirect, or indirect square root signal conversion with the L_TYPE parameter.

Direct

Direct signal conversion allows the signal to pass through the accessed channel input value (or the simulated value when simulation is enabled).

$$PV = \text{Channel Value}$$

Indirect

Indirect signal conversion converts the signal linearly to the accessed channel input value (or the simulated value when simulation is enabled) from its specified range (XD_SCALE) to the range and units of the PV and OUT parameters (OUT_SCALE).

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

**** OUT_SCALE values**

Indirect Square Root

Indirect Square Root signal conversion takes 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.

$$PV = \sqrt{\left(\frac{FIELD\_VAL}{100}\right)} \times (EU^{**}@100\% - EU^{**}@0\%) + EU^{**}@0\%$$

**** OUT_SCALE values**

When the converted input value is below the limit specified by the LOW_CUT parameter, and the Low Cutoff I/O option (IO_OPTS) is enabled (True), a value of zero is used for the converted value (PV). This option is useful to eliminate false readings when the differential pressure measurement is close to zero, and it may also be useful with zero-based measurement devices such as flowmeters.

NOTE
Low Cutoff is the only I/O option supported by the AI block. You can set the I/O option in **Manual** or **Out of Service** mode only.

Block Errors

Table 2-3 lists conditions reported in the BLOCK_ERR parameter. Conditions in *italics* are inactive for the AI block and are given here only for your reference.

TABLE 2-3. BLOCK_ERR Conditions

Condition Number	Condition Name and Description
0	*Other*
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.
2	*Link Configuration Error*
3	**Simulate Active:** Simulation is enabled and the block is using a simulated value in its execution.

TABLE 2-3. BLOCK_ERR Conditions

Condition Number	Condition Name and Description
4	*Local Override*
5	*Device Fault State Set*
6	*Device Needs Maintenance Soon*
7	**Input Failure/Process Variable has Bad Status:** The hardware is bad, or a bad status is being simulated.
8	**Output Failure:** The output is bad based primarily upon a bad input.
9	*Memory Failure*
10	*Lost Static Data*
11	*Lost NV Data*
12	*Readback Check Failed*
13	*Device Needs Maintenance Now*
14	*Power Up*
15	**Out of Service:** The actual mode is out of service.

Modes

The AI Function Block supports three modes of operation as defined by the MODE_BLK parameter:

- **Manual (Man)** The block output (OUT) may be set manually.
- **Automatic (Auto)** OUT reflects the analog input measurement or the simulated value when simulation is enabled.
- **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 shows Out of Service. In this mode, you can make changes to all configured parameters. The target mode of a block may be restricted to one or more of the supported modes.

Alarm Detection

A block alarm will be generated whenever the BLOCK_ERR has an error bit set. The types of block error for the AI block are defined above.

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)

In order 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

Alarms are grouped into five levels of priority:

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 (such as 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.

Status Handling

Normally, the status of the PV reflects the status of the measurement value, the operating condition of the I/O card, and any active alarm condition. In Auto mode, OUT reflects the value and status quality of the PV. In Man mode, the OUT status constant limit is set to indicate that the value is a constant and the OUT status is *Good*.

The **Uncertain** - EU range violation status is always set, and the PV status is set high- or low-limited if the sensor limits for conversion are exceeded.

In the STATUS_OPTS parameter, you can select from the following options to control the status handling:

BAD if Limited – sets the OUT status quality to *Bad* when the value is higher or lower than the sensor limits.

Uncertain if Limited – sets the OUT status quality to *Uncertain* when the value is higher or lower than the sensor limits.

Uncertain if in Manual mode – The status of the Output is set to *Uncertain* when the mode is set to Manual

NOTES
1. The instrument must be in **Manual** or **Out of Service** mode to set the status option.
2. The AI block only supports the **BAD if Limited** option. Unsupported options are not grayed out; they appear on the screen in the same manner as supported options.

Advanced Features

The AI function block provided with Fisher-Rosemount fieldbus devices provides added capability through the addition of the following parameters:

ALARM_TYPE – Allows one or more of the process alarm conditions detected by the AI function block to be used in setting its OUT_D parameter.

OUT_D – Discrete output of the AI function block based on the detection of process alarm condition(s). This parameter may be linked to other function blocks that require a discrete input based on the detected alarm condition.

VAR_SCAN – Time period in seconds over which the variability index (VAR_INDEX) is computed.

VAR_INDEX – Process variability index measured as the integral of average absolute error between PV and its mean value over the previous evaluation period. This index is calculated as a percent of OUT span and is updated at the end of the time period defined by VAR_SCAN.

Troubleshooting

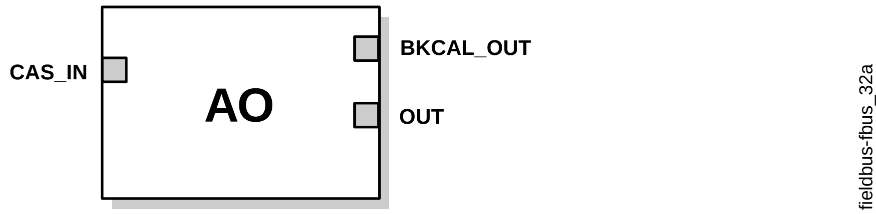
Refer to Table 2-4 to troubleshoot any problems that you encounter.

TABLE 2-4. Troubleshooting

Symptom	Possible Causes	Corrective Action
Mode will not leave OOS	Target mode not set.	Set target mode to something other than OOS.
	Configuration error	BLOCK_ERR will show the configuration error bit set. The following are parameters that must be set before the block is allowed out of OOS: • CHANNEL must be set to a valid value and cannot be left at initial value of 0. • XD_SCALE.UNITS_INDX must match the units in the transducer block channel value. • L_TYPE must be set to Direct, Indirect, or Indirect Square Root and cannot be left at initial value of 0.
	Resource block	The actual mode of the Resource block is OOS. See Resource Block Diagnostics for corrective action.
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
Process and/or block alarms will not work	Features	FEATURES_SEL does not have Alerts enabled. Enable the Alerts bit.
	Notification	LIM_NOTIFY is not high enough. Set equal to MAX_NOTIFY.
	Status Options	STATUS_OPTS has Propagate Fault Forward bit set. This should be cleared to cause an alarm to occur.
Value of output does not make sense	Linearization Type	L_TYPE must be set to Direct, Indirect, or Indirect Square Root and cannot be left at initial value of 0.
	Scaling	Scaling parameters are set incorrectly: • XD_SCALE.EU0 and EU100 should match that of the transducer block channel value. • OUT_SCALE.EU0 and EU100 are not set properly.
Cannot set HI_LIMIT, HI_HI_LIMIT, LO_LIMIT, or LO_LO_LIMIT Values	Scaling	Limit values are outside the OUT_SCALE.EU0 and OUT_SCALE.EU100 values. Change OUT_SCALE or set values within range.

Section **3**

Analog Output (AO) Function Block



CAS_IN = The remote setpoint value from another function block.

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

OUT = The block output and status.

The Analog Output (AO) function block assigns an output value to a field device through a specified I/O channel. The block supports mode control, signal status calculation, and simulation. Figure 3-1 illustrates the internal components of the AO function block, and Table 3-1 lists the definitions of the system parameters.

TABLE 3-1. Analog Output Function Block System Parameters

Parameters	Units	Description
BKCAL_OUT	EU of PV_SCALE	The value and status required by the BKCAL_IN input of another block to prevent reset windup and to provide bumpless transfer to closed loop control.
BLOCK_ERR	None	The summary of active error conditions associated with the block. The block errors for the Analog Output block are **Simulate Active**, **Input Failure/Process Variable has *Bad* Status**, **Output Failure**, **Read back Failed**, and **Out of Service**.
CAS_IN	EU of PV_SCALE	The remote setpoint value from another function block.
IO_OPTS	None	Allows you to select how the I/O signals are processed. 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**.
CHANNEL	None	Defines the output that drives the field device.
MODE	None	Enumerated attribute used to request and show the source of the setpoint and/or output used by the block.
OUT	EU of XD_SCALE	The primary value and status calculated by the block in Auto mode. OUT may be set manually in Man mode.
PV	EU of PV_SCALE	The process variable used in block execution. This value is converted from READBACK to show the actuator position in the same units as the setpoint value.
PV_SCALE	None	The high and low scale values, the engineering units code, and the number of digits to the right of the decimal point associated with the PV.

Parameters	Units	Description
READBACK	EU of XD_SCALE	The measured or implied actuator position associated with the OUT value.
SIMULATE	EU of XD_SCALE	Enables simulation and allows you to enter an input value and status.
SP	EU of PV_SCALE	The target block output value (setpoint).
SP_HI_LIM	EU of PV_SCALE	The highest setpoint value allowed.
SP_LO_LIM	EU of PV_SCALE	The lowest setpoint value allowed.
SP_RATE_DN	EU of PV_SCALE per second	Ramp rate for downward setpoint changes. When the ramp rate is set to zero, the setpoint is used immediately.
SP_RATE_UP	EU of PV_SCALE per second	Ramp rate for upward setpoint changes. When the ramp rate is set to zero, the setpoint is used immediately.
SP_WRK	EU of PV_SCALE	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.

Setting the Output

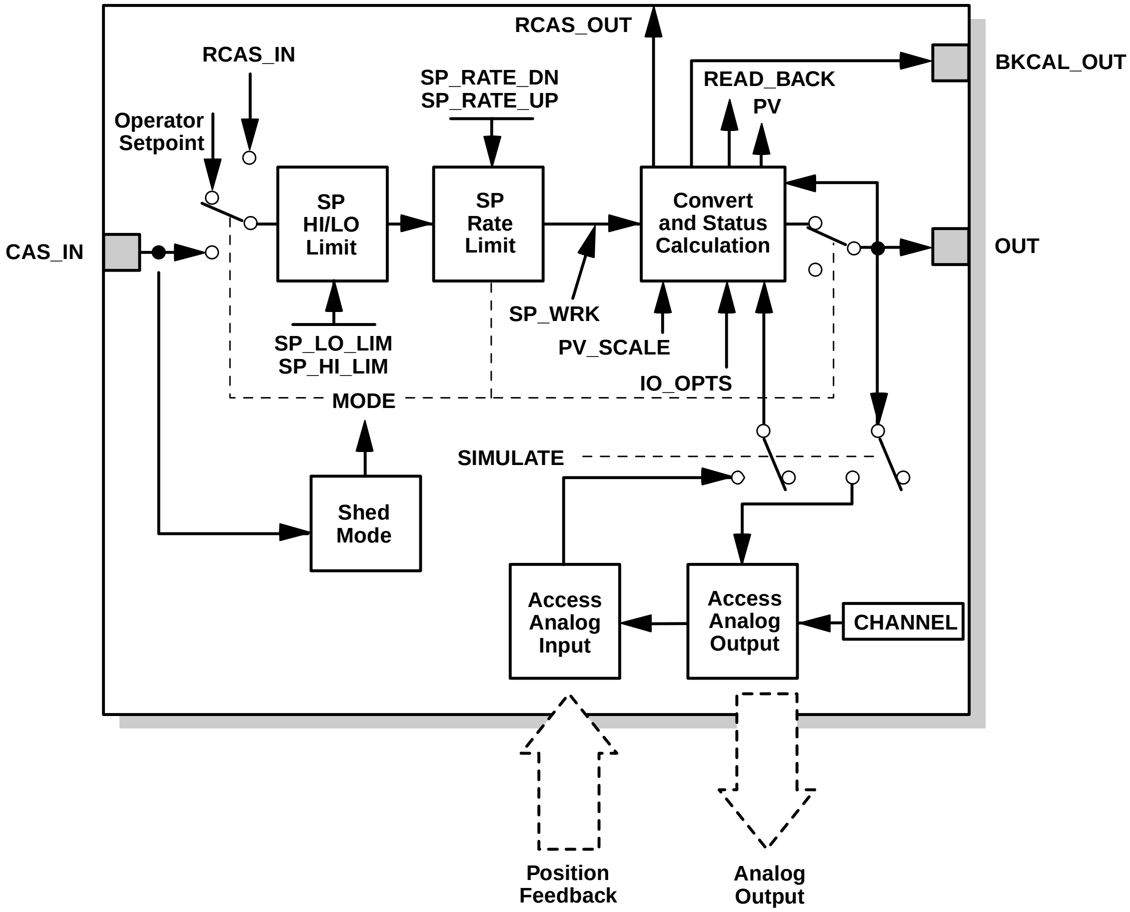
To set the output for the AO block, you must first set the mode to define the manner in which the block determines its setpoint. In Manual mode the value of the output attribute (OUT) must be set manually by the user, and is independent of the setpoint. In Automatic mode, OUT is set automatically based 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 to update 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 support testing, you can enable simulation, which allows you to manually set the channel feedback. There is no alarm detection in the AO function block.

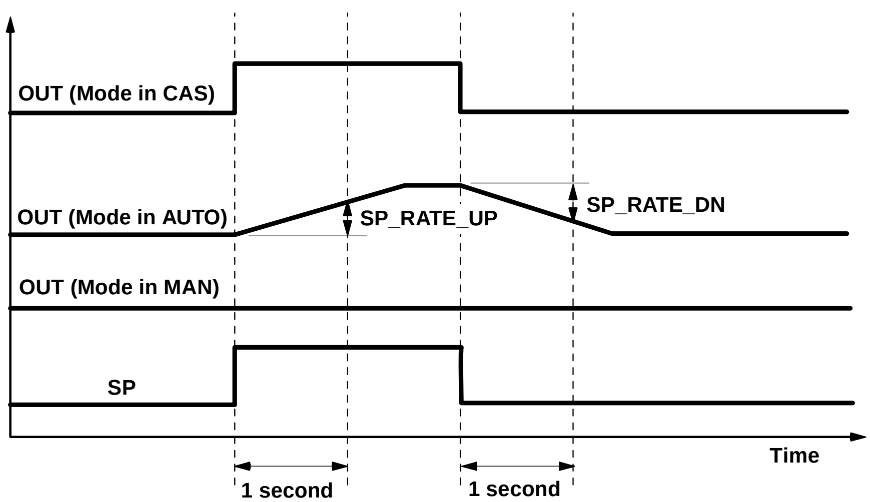
To select the manner of processing the SP and the channel output value configure the setpoint limiting options, the tracking options, and the conversion and status calculations.

Figure 3-1. Analog Output Function Block Schematic



fieldbus-fbus_07a

Figure 3-2. Analog Output Function Block Timing Diagram



fieldbus-fbus_08a

Setpoint Selection and Limiting

To select the source of the SP value use the **MODE** attribute. In Automatic (Auto) mode, the local, manually-entered SP is used. In Cascade (Cas) mode, the SP comes from another block through the CAS_IN input connector. In RemoteCascade (RCas) mode, the SP comes from a host computer that writes to RCAS_IN. The range and units of the SP are defined by the **PV_SCALE** attribute.

In Manual (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 is enabled (True) as a default. 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 attribute (SP_HI_LIM) and the setpoint low limit attribute (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 attribute (SP_RATE_UP) and the setpoint downward rate limit attribute (SP_RATE_DN). A limit of zero prevents rate limiting, even in Auto mode.

Conversion and Status Calculation

The working setpoint (SP_WRK) is the setpoint value after limiting. You can choose to reverse the conversion range, which will reverse the range of **PV_SCALE** to calculate the **OUT** attribute, by selecting the **Increase to Close** I/O option. This will invert 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** attribute. In Man mode, the **OUT** attribute is set manually, and is used to set 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 attribute (in engineering units). If the actuator does not support position feedback, the PV and **READBACK** values are based on the **OUT** attribute.

The working setpoint (SP_WRK) is the value normally used for the **BKCAL_OUT** attribute. However, for those cases where the **READBACK** signal directly (linearly) reflects 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.

Simulation

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

Action On Fault Detection

To define the state to which you wish the valve to enter when the CAS_IN input detects a bad status and the block is in CAS mode, configure the following parameters:

FSTATE_TIME: The length of time that the AO block will wait to position the **OUT** value to the **FSTATE_VAL** value upon the detection of a fault condition. When the block has a target mode of CAS, a fault condition will be detected if the **CAS_IN** has a **BAD** status or an **Initiate Fault State** substatus is received from the upstream block.

FSTATE_VAL: The value to which the OUT value transitions after **FSTATE_TIME** elapses and the fault condition has not cleared. You can configure the channel to hold the value at the start of the failure action condition or to go to the failure action value (FAIL_ACTION_VAL).

Block Errors

The following conditions are reported in the BLOCK_ERR attribute:

Input failure/process variable has *Bad* status – The hardware is bad, the Device Signal Tag (DST) does not exist, or a **BAD** status is being simulated.

Out of service – The block is in Out of Service (O/S) mode.

Output failure – The output hardware is bad.

Readback failed – The readback failed.

Simulate active – Simulation is enabled and the block is using a simulated value in its execution.

Modes

The Analog Output function block supports the following modes:

Manual (Man) – You can manually set the output to the I/O channel through the **OUT** attribute. This mode is used primarily for maintenance and troubleshooting.

Automatic (Auto) – The block output (OUT) reflects the target operating point specified by the setpoint (SP) attribute.

Cascade (Cas) – The SP attribute is set by another function block through a connection to CAS_IN. The SP value is used to set the **OUT** attribute automatically.

RemoteCascade (RCas) – The SP is set by a host computer by writing to the RCAS_IN parameter. The SP value is used to set the **OUT** attribute automatically.

Out of Service (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 will remain at the last position.

Local Override (LO) – The output of the block is not responding to **OUT** because the resource block has been placed into 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.

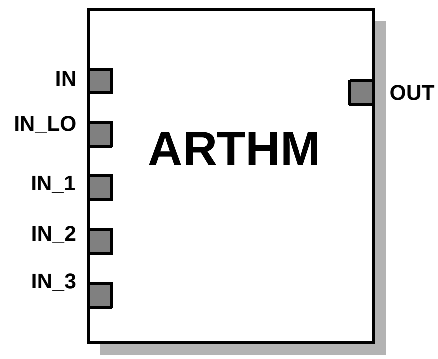
Status Handling

Output or readback fault detection are reflected in the status of PV, OUT, and BKCAL_OUT. A limited SP condition is reflected in the BKCAL_OUT status. When simulation is enabled through the SIMULATE attribute, you can set the value and status for PV and READBACK.

When the block is in Cas mode and the CAS_IN input goes bad, the block sheds mode to the next permitted mode.

Section 4

Arithmetic Function Block



The Arithmetic function block provides the ability to configure a range extension function for a primary input and applies the nine (9) different arithmetic types as compensation to or augmentation of the range extended input. All operations are selected by parameter and input connection.

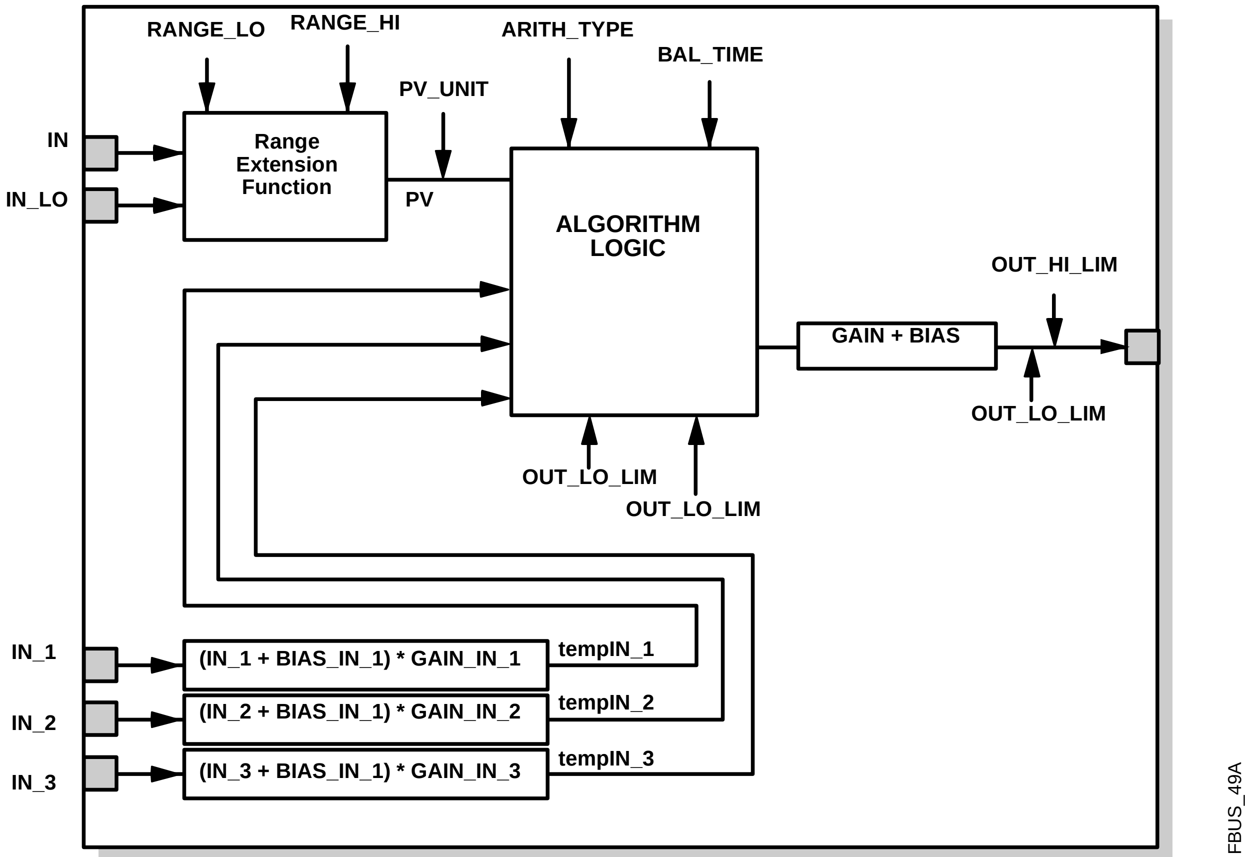
The nine (9) arithmetic functions are Flow Compensation Linear, Flow Compensation Square Root, Flow Compensation Approximate, Btu Flow, Traditional Multiply and Divide, Average, Summer, Fourth Order Polynomial, and Simple HTG Compensate Level.

This Arithmetic function block supports mode control (Auto, Manual, and Out of service). There is no standard alarm detection in this block.

TABLE 4-1. Arithmetic Block Parameters

Index	Parameter	Description
1	ST_REV	The revision level of the static data associated with the function block. The revision value will increment each time a static parameter value in the block is changed.
2	TAG_DESC	The user description of the intended application of the block.
3	STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
4	ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
5	MODE_BLK	The actual, target, permitted, ad normal modes of the block. Target: The mode to “go to”, Actual: The mode the “block is currently in”, Permitted: Allowed modes that target may take, Normal: Most common mode for target.
6	BLOCK_ERR	This parameter reflect the error status associated with the hardware or software componets associated with a block. It is a bit string so that multiple errors may be shown.
7	PV	
8	OUT	The analog output value and status.
9	PRE_OUT	Displays what would be the OUT value if the mode was “Auto” or lower.
10	PV_SCALE	
11	OUT_RANGE	The high and low scale values, engineering units code, and number of digits to the tight of the decimal point associated with OUT.
12	GRANT_DENY	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block. (Not used by the device)
13	INPUT_OPTIONS	Option bit string for handling the status of the auxiliary inputs.
14	IN	The block input value and status.
15	IN_LO	Input of the low range transmitter, in a range extension application.
16	IN	The block input value and status.
17	IN	The block input value and status.
18	IN	The block input value and status.
19	RANGE_HI	Constant value above which the range extension has switch to the high range transmitter.
20	RANGE_LO	Constant value below which the range extension has switch to the high range transmitter.
21	BIAS_IN_1	The bias value for IN_1.
22	GAIN_IN_1	The proportional gain (multiplier) value for IN_1.
23	BIAS_IN_2	The bias value for IN_2.
24	GAIN_IN_2	The proportional gain (multiplier) value for IN_2.
25	BIAS_IN_3	The bias value for IN_3.
26	GAIN_IN_3	The proportional gain (multiplier) value for IN_3.
27	COMP_HI_LIM	Determines the high limit of the compensation input.
28	COMP_LO_LIM	Determines the low limit of the compensation input.
29	ARITH_TYPE	The set of 9 arithmetic functions applied as compensation to or augmentation of the range extended input.
30	BAL_TIME	Specifies the time for a block value to match an input, output, or calculated value or the time for dissipation of the internal balancing bias.
31	BIAS	The bias value is used to calculate the output.
32	GAIN	The gain value is used to calculate the output.
33	OUT_HI_LIM	The maximum output value allowed.
34	OUT_LO_LIM	The minimum output value allowed.
35	UPDATE_EVT	This alert is generated by any changes to the static data.
36	BLOCK_ALM	Used for all configuration, hardware, connection failure, or system problem in the block. The cause of the alert is entered in the subcode field. The first active alarm will set the active status in the status parameter. When the Unreported status is cleared by the alert reporting test other block alert may be reported without clearing the Active status, if the subcode has changed.

Figure 4-2. Signal Characterizer
Function Block Schematic Diagram



Block Execution

The Arithmetic function block provides range extension and compensation through nine (9) arithmetic types.

There are two inputs (IN and IN_LO) used in calculating PV. PV is then combined with up to three inputs (IN_1, IN_2, and IN_3) through the user selected compensation function (ARITH_TYPE) to calculate the value of func. A gain is applied to func and then a bias is added to get the value PRE_OUT. In AUTO, PRE_OUT is used for OUT.

Range Extension and Calculation of PV

When both IN and IN_LO are usable, the following formula is applied to calculate range extension for PV:

$$PV = G \bullet IN + (1 - G) \bullet IN\_LO$$

G has a range from 0 to 1, for IN from RANGE_LO to RANGE_HI.

Compensation Input Calculations

For each of the inputs IN_1, IN_3, IN_4 there is a gain and bias. The compensation terms (t) are calculated as follows:

- When IN_(k) is usable:

$$t(k) = GAIN\_IN(k) \bullet (BIAS\_IN(k) + IN\_(k))$$

- When IN_(k) is not usable, then t(k) gets the value of the last t(k) computed with a usable input.

Modes

The ARTHM block supports the following modes:

Manual (Man)—.

Automatic (Auto)—.

Out of Service (OOS)—.

The target mode of a block bay be restricted to one or more of the supported modes.

Status Handling

IN_x Use Bad

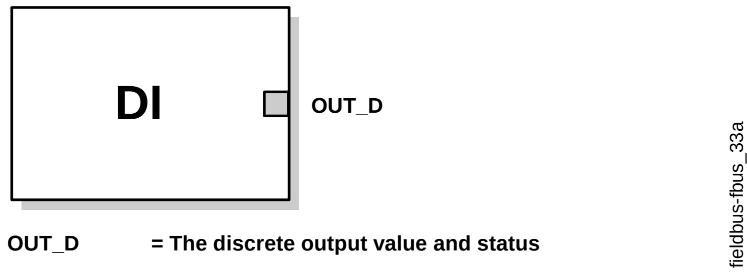
IN_x Use Uncertain

IN_LO Use Uncertain

IN Use Uncertain

For complete descriptions of supported input options, refer to the Option Bitstring Parameters topic.

Discrete Input (DI) Function Block



The Discrete Input (DI) function block processes a single discrete input from a field device and makes it available to other function blocks. You can configure inversion and alarm detection on the input value.

The Discrete Input function block supports mode control, signal status propagation, and simulation.

Normally, the block is used in Automatic (Auto) mode so that the process variable (PV_D) is copied to the output (OUT_D). You can change the mode to Manual (Man) to disconnect the field signal and substitute a manually-entered value for OUT_D. In this case, PV_D continues to show the value that will become OUT_D when the mode is changed to Auto.

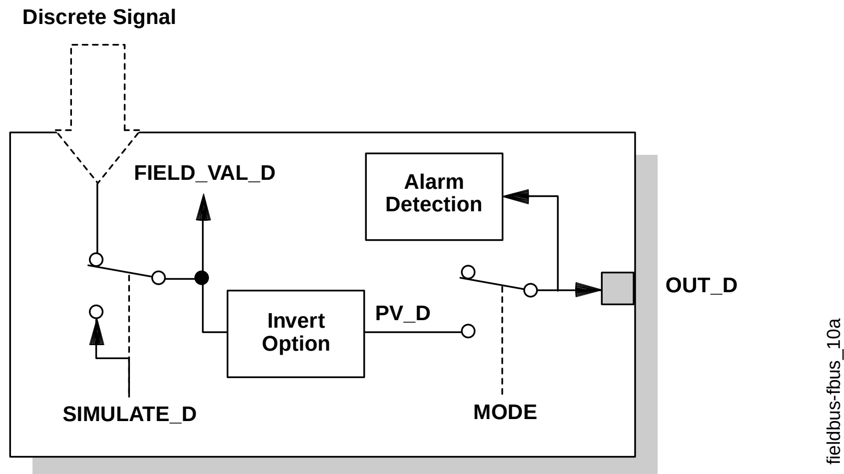
To support testing, you can enable simulation, which allows the measurement value to be supplied manually through the SIMULATE_D parameter. Figure 5-2 illustrates the internal components of the DI function block, and Table 5-1 lists the definitions of the system parameters.

TABLE 5-1. Discrete Input Function Block System Parameters

Parameters	Units	Description
BLOCK_ERR	None	The summary of active error conditions associated with the block. The supported block errors in the Discrete Input function block are Simulate active, Input failure/process variable has *Bad* status, and Out of service. See System Support.
DISC_LIM	None	The state of the discrete input that causes an alarm. Any number from 0 to 255 may be. State 255 specifies that no alarm indication is to be shown.
FIELD_VAL_D	None	The value and status of the discrete input from a field device.
CHANNEL	None	Defines the I/O input used for the field measurement.
IO_OPTS	None	Allows you to select options for I/O value processing. The supported I/O option for the Discrete Input function block is Invert. See System Support.

Parameters	Units	Description
MODE	None	The mode record of the block. Contains the actual, target, permitted, and normal modes. See System Support
OUT_D	None	The discrete output value and status.
PV_D	None	The discrete process variable used in block execution.
SIMULATE_D	None	Enables simulation and allows you to enter an input value and status when SIMULATE_IN_D is not connected.

FIGURE 5-2. Discrete Input Function Block Schematic



I/O Selection

To select the I/O associated with the discrete measurement, configure the value of the **CHANNEL** attribute.

Simulation

To support testing, you can either change the mode of the block to manual and adjust the output value, or you can enable simulation through the configuration tool and manually enter a value for the measurement value and its status. In both cases, you must first set the ENABLE jumper on the field device.

NOTE
All fieldbus instruments have an ENABLE jumper. As a safety measure, the jumper has to be reset every time there is a power interruption. This measure is to prevent devices that went through simulation in the staging process from being installed with simulation enabled.

With simulation enabled, the actual measurement value has no impact on the OUT_D value or the status.

Field Value Processing

You can configure the Invert I/O option (IO_OPTS) to process FIELD_VAL_D. The invert option indicates whether or not the discrete input is logically inverted before it is stored in the process variable (PV_D).

The output of the Invert processor is PV_D. This value goes to the mode switch where it becomes OUT_D when the mode is Auto. OUT_D is also tested for an alarm state. You might choose this option when the field contact is normally closed, so an open contact or a broken wire represents the active state of the condition being sensed.

NOTE
Invert is the only I/O option that the DI block supports. You can set the I/O option in **Manual** or **Out of Service** mode only.

Alarm Detection

To select the state that initiates an input alarm, and to set discrete alarm substatus in the output, configure the **DISC_LIM** attribute. You can enter any value between 0 and 255. A value of 255 disables the alarm.

Block Errors

The following conditions are reported in the BLOCK_ERR attribute:

Simulate active – SIMULATE_D is enabled; OUT_D does not reflect actual process conditions.

Input failure/process variable has ***Bad*** **status** – The hardware is bad, the configured channel is invalid, or a *Bad* status is being simulated.

Out of service – The block is not being processed.

Modes

The Discrete Input function block supports the following modes:

Manual (Man) – The output (OUT_D) is disconnected form the field.

Automatic (Auto) – The block algorithm determines OUT_D.

Out of Service (O/S) – The block is not processed. The output status is set to *Bad: Out of Service*. The BLOCK_ERR attribute shows *Out of Service*.

Status Handling

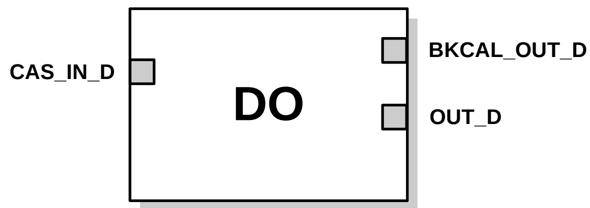
Under normal conditions, a *Good: Non-cascade* status is passed through to OUT_D. The block also support **Status Action on Failure** and **Block Error** indications.

Action on Failure

In case of hardware failure, FIELD_VAL_D, PV_D, and OUT_D change to a *Bad* status and the BLOCK_ERR attribute displays *Bad PV*. When SIMULATE_D is enabled, FIELD_VAL_D, PV_D, and OUT_D change to a simulation status. When the block is set to Man mode, OUT_D is set to *Good: Non-cascade, Constant* status.

Section **6**

Discrete Output (DO) Function Block



fieldbus-fbus_32a

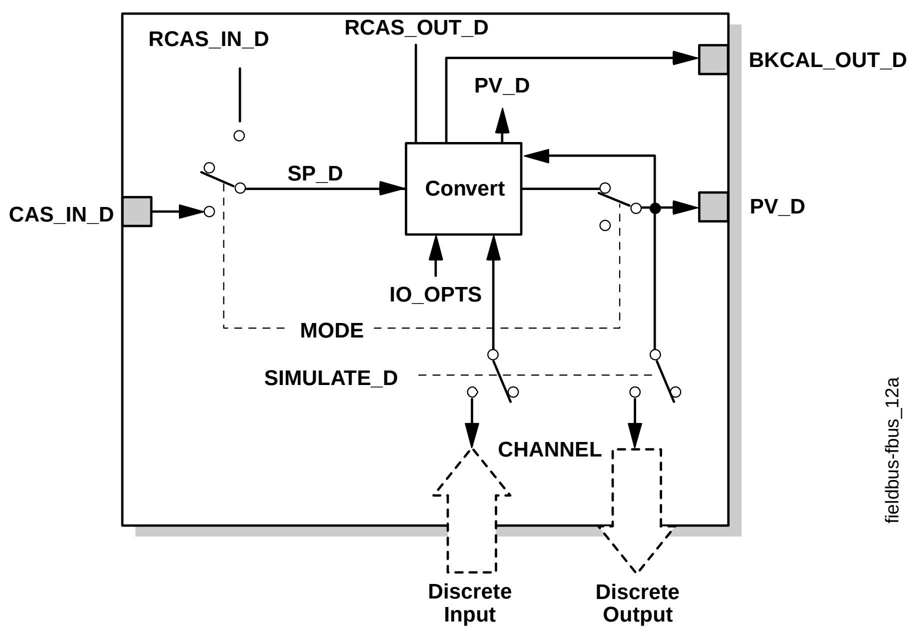
CAS_IN_D	=	**The remote setpoint value from another function block.**
BKCAL_OUT_D	=	**The output value and status required by BKCAL_IN_D input of another block for output tracking.**
OUT_D	=	**The discrete output value and status**

The Discrete Output (DO) function block processes a discrete setpoint and saves it to a specified channel to produce an output signal. The block supports mode control, output tracking, and simulation. There is no process alarm detection in the block. In operation, the DO function block determines its setpoint, sets the output, and, as an option, checks a feedback signal from the field device to confirm the physical output operation. Figure 6-2 illustrates the internal components of the DO function block, and Table 6-1 lists the system parameters.

TABLE 6-1. Discrete Output Function Block System Parameters

Parameters	Units	Description
BKCAL_OUT_D	None	The value and status required by the BKCAL_IN_D input of another block for output tracking.
BLOCK_ERR	None	The summary of active error conditions associated with the block. The supported block errors in the Discrete Output function block are Simulate active, Input failure/process variable has *Bad* status, Output failure, readback failed, and Out of service. See System Support
CAS_IN_D	None	The remote setpoint value from another block.
IO_OPTS	None	Allows you to select how the I/O signals are processed. The supported I/O options for the Discrete Output function block are SP_PV Track in Man, Invert, and Use PV for BKCAL_OUT.
CHANNEL	None	Defines the output that drives the field device.
MODE	None	The mode record of the block. Contains the actual, target, permitted, and normal modes.
OUT_D	None	The discrete output value and status.
PV_D	None	The discrete process variable calculated from READBACK_D.
READBACK_D	None	The discrete feedback from the output.
SIMULATE_D	None	Enables simulation.
SP_D	None	The discrete target block output value (setpoint).

FIGURE 6-2. Discrete Input Function Block Schematic



Setting the Output

To set the output for the DO block, you must first set the mode to define the manner in which the block determines its setpoint. In **Cascade** mode, the setpoint equals the input value (CAS_IN_D). In **Automatic** or **Manual** mode, the setpoint must be entered manually by the user. In **Remote Cascade** mode, the setpoint is determined by a host computer that is writing to the RCAS_IN_D parameter.

To further customize the output, configure the **SP_PV Track in Man**, **Invert**, and **Use PV for BKCAL_OUT** I/O options.

NOTE
SP_PV Track in Man, **Invert**, and **Use PV for BKCAL_OUT** are the only I/O options supported by the DO block. You can configure I/O options in **Manual** or **Out of Service** mode only.

The **SP_PV Track in Man** option permits the setpoint to track the process variable when the block is in **Manual** mode. With this option enabled, the setpoint (SP_D) becomes a copy of the process variable (PV_D), and a manually-entered SP_D value is overwritten on the block's next execution cycle. This option can prevent a state change when transitional from **Manual** to **Automatic** mode. You can disable this option in **Manual** or **Out of Service** mode only.

The **Invert** option inverts the setpoint (SP_D) before it is stored in OUT_D. With this option enabled, OUT_D becomes an inverted copy of SP_D. With this option disabled, OUT_D is a direct copy of SP_D. If discrete output feedback is not supported by the field device, a copy of OUT_D is used in its place (with a delay of one execution time) to become READBACK_D. The readback value is processed through the **Invert** option to become PV_D, which normally matches SP_D in Auto, Cas, or RCas mode.

The **Use PV for BKCAL_OUT** option specifies that BKCAL _OUT equal the value of the process variable (PV_D) instead of the setpoint (SP_D). If you do not enable this option, BKCAL_OUT will equal SP_D.

Simulation

With SIMULATE_D enabled, the specified value and status is reflected in READBACK_D. If SIMULATE_D is not enabled, and the mode is not **Out of Service**, the value of OUT_D is sent to the hardware.

Action on Fault Detection

To determine the state to which the output goes if the block is in CAS mode and the CAS_IN input has a *BAD* status, configure the following parameters:

FSTATE_TIME – The length of time that the AO block delays before setting OUT equal to FSTATE_VAL upon the detection of a fault condition. If the block's target mode is Cascade, a fault condition will be detected if the CAS_IN has a *BAD* status, or an *Initiate Fault State* substatus is received from the upstream block.

FSTATE_VALD – The value to which the OUT_D attribute transitions if the length of time specified in FSTATE_TIME passes and the fault condition has not cleared. You can configure the channel to hold the value at the start of the fault action condition or to go to the Fault Action Value (FAULT_ACTION_VAL).

Block Errors

The following conditions are reported in the BLOCK_ERR attribute:

Simulate active – SIMULATE_D is enabled; therefore, PV_D is not real.

Input failure/process variable has *Bad* status – The readback value is bad.

Output failure – The output hardware or the configured channel is invalid.

Readback failed – The hardware providing readback is bad.

Out of service – The block is not being processed.

Modes

The DO block supports the following modes:

Manual (Man) – The block output (OUT_D) may be entered manually.

Automatic (Auto) – The block algorithm uses the local setpoint value (SP_D) to determine OUT_D.

Cascade (Cas) – The block uses a setpoint supplied by another function block.

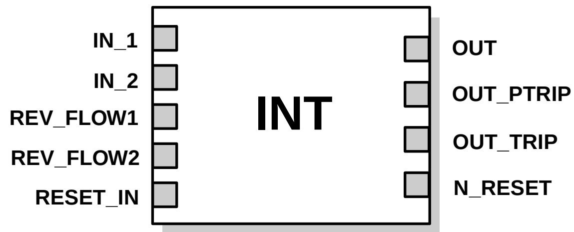
RemoteCascade (RCas) – The block uses a setpoint supplied by a host computer.

Out of Service (O/S) – The block is not processed and the output is not transferred to I/O. The BLOCK_ERR attribute shows **Out of service**.

Status Handling / Action on Failure

Under normal operating conditions, the output statuses (OUT_D and BKCAL_OUT_D) are *Good: Cascade*. If the output hardware fails, the status of BKCAL_OUT_D is set to *Bad: DeviceFail*, and the BLOCK_ERR attribute shows *Output Failure*. If the hardware used for output feedback fails, the status of READBACK_D and PV_D is set to *Bad: DeviceFail*, and the BLOCK_ERR attribute shows *Bad PV* and *Readback Failed*.

Section 7 Integrator (INT) Function Block



IN_1	=	**The first input value and status.**
IN_2	=	**The second input value and status.**
REV_FLOW1	=	**The discrete input that specifies whether IN_1 is positive or negative.**
REV_FLOW2	=	**The discrete input that specifies whether IN_1 is positive or negative.**
RESET_IN	=	**The discrete input that resets the integrator and holds reset until released.**
OUT	=	**The integration output value and status.**
OUT_PTRIP	=	**A discrete value that is set when the pre-trip limit value is reached.**
OUT_TRIP	=	**A discrete value that is set when the trip target value (setpoint) is reached.**
N_RESET	=	**The number of times the integrator function block is initialized or reset.**

FIELDBUS-FBUS_35A

The Integrator (INT) function block integrates one or two variables over time. The block compares the integrated or accumulated value to pre-trip and trip limits and generates discrete output signals when the limits are reached.

The Integrator function block supports mode control, demand reset, a reset counter, and signal status calculation. There is no process alarm detection in the block. Figure 7-1 illustrates the internal components of the INT function block, and Table 7-1 lists the system parameters.

Figure 7-1. Integrator Function Block Schematic

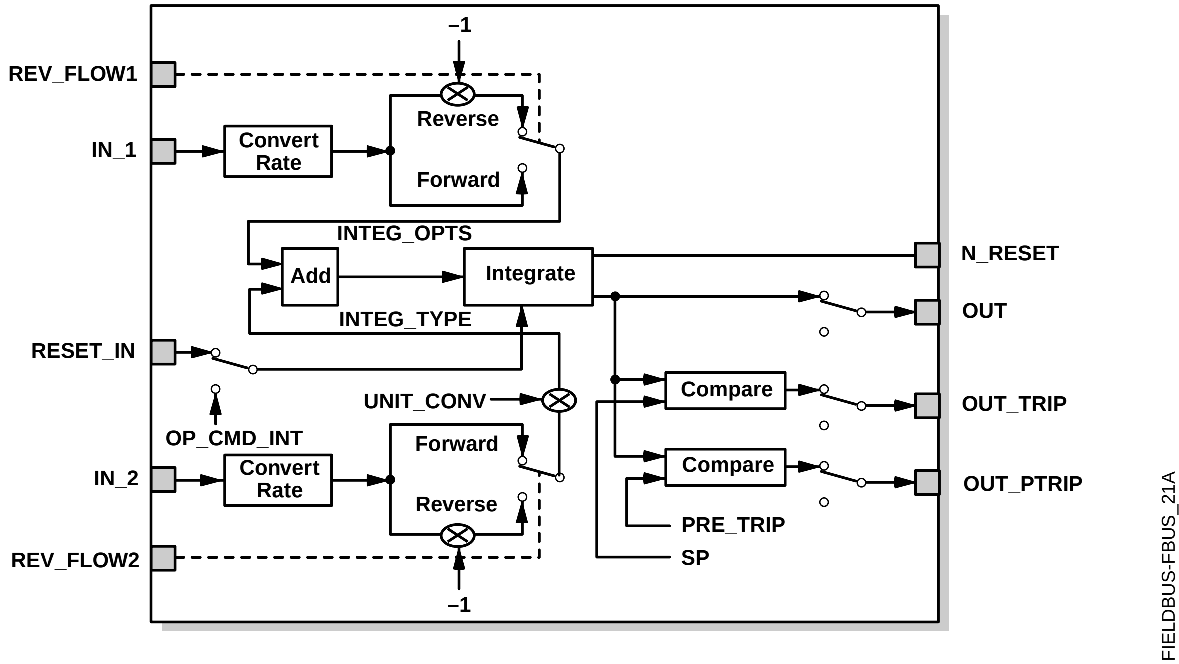


TABLE 7-1. Integrator Function Block System Parameters

Index	Parameters	Definition
1	ST_REV	The revision level of the static data associated with the function block.
2	TAG_DESC	The user description of the intended application of the block.
3	STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked of processed by the block.
4	ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms.
5	MODE_BLK	
6	BLOCK_ERR	The summary of active error conditions associated with the block. The block error for the Integrator function block is Out of service.
7	TOTAL_SP	The set point for a batch totalization.
8	OUT	The block output value and status.
9	OUT_RANGE	The high an low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT.
10	GRAND_DENY	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block (not used by the device).
11	STATUS_OPTS	Allows you to select option for status handling and processing. The supported status option for the Integrator block are: “Uncertain if Manual mode.”
12	IN_1	The block input value and status.
13	IN_2	The block input value and status.
14	OUT_TRIP	The first discrete output.
15	OUT_PTRIP	The second discrete output.
16	TIME_UNIT1	Converts the rate time, units in seconds.
17	TIME_UNIT2	Converts the rate time, units in seconds.
18	UNIT_CONV	Factor to convert the engineering units of IN_2 into the engineering units of IN_1.
19	PULSE_VAL1	Determines the mass, volume or energy per pulse.
20	PULSE_VAL2	Determines the mass, volume or energy per pulse.

TABLE 7-1. Integrator Function Block System Parameters

Index	Parameters	Definition
21	REV_FLOW1	Indicates reverse flow when “true”; 0- Forward, 1- Reverse
22	REV_FLOW2	Indicates reverse flow when “true”; 0- Forward, 1- Reverse
23	RESET_IN	Resets the totalizers
24	STOTAL	Indicates the snapshot of OUT just before a reset
25	RTOTAL	Indicates the totalization of “bad” or “bad” and “uncertain” inputs, according to INTEG_OPTIONS.
26	SRTOTAL	The snapshot of RTOTAL just before a reset
27	SSP	The snapshot of TOTAL_SP
28	INTEG_TYPE	Defines the type of counting (up or down) and the type of resetting (demand or periodic)
29	INTEG_OPTIONS	A bit string to configure the type of input (rate or accumulative) used in each input, the flow direction to be considered in the totalization, the status to be considered in TOTAL and if the totalization residue should be used in the next batch (only when INTEG_TYPE=UP_AUTO or DN_AUTO).
30	CLOCK_PER	Establishes the period for periodic reset, in hours.
31	PRE_TRIP	Adjusts the amount of mass, volume or energy that should set OUT_PTRIP when the integration reaches (TOTAL_SP-PRE_TRIP) when counting up of PRE_TRIP when counting down.
32	N_RESET	Counts the number of resets. It can not be written or reset.
33	PCT_INC	Indicates the percentage of inputs with “good” status compared to the ones with “bad or “uncertain” and “bad” status.
34	GOOD_LIMIT	Sets the limit for PCT_INC. Below this limit OUT receives the status “good”
35	UNCERTAIN_LIMIT	Sets the limit for PCT_INC. Below this limit OUT receives the status “uncertain”
36	OP_CMD_INT	Operator command RESET Resets the totalizer
37	OUTAGE_LIMIT	The maximum tolerated duration for power failure
38	RESET_CONFIRM	Momentary discrete value with can be written by a host to enable further resets, if the option “Confirm reset” in INTEG_OPTIONS is chosen.
39	UPDATE_EVT	This alert is generated by any changes to the static data.
40	BLOCK_ALM	Used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the active status in the status parameter. As soon as the unreported status is cleared by the alert reporting task other block alerts may be reported without clearing the Active status, if the subcode has changed.

Block Execution

The INT function block integrates a variable over time. The integrated or accumulated value (OUT) is compared to pre-trip and trip limits. When the limits are reached, discrete output signals are generated (OUT_PTRIP and OUT_TRIP). You can choose one of six integrator types that determine whether the integrated value increases from zero or decreases from the trip value. The block has two inputs and can integrate positive, negative, or net flow. This capability is useful to calculate volume or mass variation in vessels, or as an optimization tool for flow ratio control.

The transfer equation used in the Integrator function block is:

$$\text{Current Integral} = \left(\frac{\Delta t}{2}\right) \times (x + y + OUT[t-1])$$

Where

Δt: the elapsed time since the previous cycle (in seconds)
x: the converted IN_1 value (based on the options you configure)
y: the converted IN_2 value (based on the options you configure), or zero if you select not to use a second input
OUT[t-1]: the value of OUT from the previous cycle

You can choose integration type options that define the integrate up, integrate down, and reset characteristics of the block. When you select the SP to 0 - auto reset or SP to 0 - demand reset integration type option:

$$\text{Integral} = \text{Integral} + \text{Current Integral}$$

$$OUT = SP - \text{Integral}$$

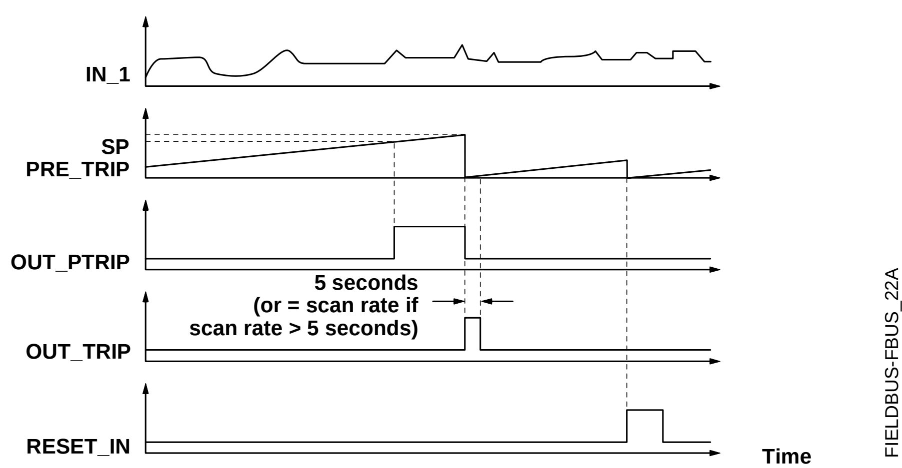
For all other integration types:

$$OUT = \text{Integral}$$

Figure 7-2 illustrates the relationship between the SP, PRE_TRIP, OUT_PTRIP, OUT_TRIP, and RESET_IN parameters in the Integrator function block.

To specify the execution of the INT block, configure input flow and rate time variables, integration type and carryover options, and trip and pre-trip action.

Figure 7-2. Integrator Function Block Timing Diagram



Specifying Rate Time Base

The time unit parameters (TIME_UNIT1 and TIME_UNIT2) specify the rate time base of the inputs (IN_1 and IN_2, respectively). The block uses the following equations to compute the integration increment:

$$x = \frac{IN\_1}{TIME\_UNIT1} \qquad y = \frac{IN\_2}{TIME\_UNIT2}$$

Where

x: the converted IN_1 value (based on the options you configure)
y: the converted IN_2 value (based on the options you configure), or zero if you select not to use a second input
OUT[t-1]: the value of OUT from the previous cycle

The block supports the following options for TIME_UNIT1 and TIME_UNIT2:

- For seconds, TIME_UNIT = 1
- For minutes, TIME_UNIT = 60
- For hours, TIME_UNIT = 3600
- For days, TIME_UNIT = 86400

Setting Reverse Flow at the Inputs

Reverse flow is determined by either the sign of the value at IN_1 or IN_2, or the discrete inputs REV_FLOW1 and REV_FLOW_2. When the REV_FLOW input is True, the block interprets the associated IN value as negative.

Calculating Net Flow

Net flow is calculated by adding the increments calculated for each IN. When ENABLE_IN2 is False, the increment value for IN_2 is considered zero. When ENABLE_IN2 is True, the value of IN_2 is used in the calculation.

To determine the net flow direction that is to be included in the integration, configure the **Flow Forward** and **Flow Reverse** integration options attribute (INTEG_OPTS). When **Flow Forward** is *True*, positive increments are included. When **Flow Reverse** is *True*, negative increments are included. When both **Flow Forward** and **Flow Reverse** are *True*, positive and negative increments are included.

Integration Types

The integration type attribute (INTEG_TYPE) defines the integrate up, integrate down, and reset characteristics of the block. Choose from the following options:

0 to SP - auto reset at SP – Integrates from zero to the setpoint (SP) and automatically resets when the SP is reached.

0 to SP - demand reset – Integrates from zero to the SP and resets when RESET_IN or the operator command to reset the integrator (OP_CMT_INT) transitions to True (1).

SP to 0 - auto reset at SP – Integrates from the SP to zero and automatically resets when zero is reached.

SP to 0 - demand reset – Integrates from the SP to zero and resets when RESET_IN or OP_CMD_INT transitions to True.

0 to ? - periodic reset – Counts upward and resets periodically. The period is set by the CLOCK_PER attribute.

0 to ? - demand reset – Counts upward and is reset when RESET_IN or OP_CMD_INT transitions to True.

0 to ? - periodic & demand reset – Counts upward and is reset periodically or by RESET_IN.

Trip and Pre-trip Action

When the integration value reaches SP – PRE_TRIP (or 0 – PRE_TRIP, depending on the INTEG_TYPE), OUT_PTRIP is set. When the integration value reaches the trip target value (SP or 0), OUT_TRIP is set. OUT_PTRIP remains set until SP or 0 is reached.

Integration Carryover

When the **0 to SP - auto reset at SP** or the **SP to 0 - auto reset at SP** integration type is set, you can enable the **Carry** integration option to carry the excess past the trip point into the next integration cycle as the initial value of the integrator.

Modes

The Integrator function block supports the following modes:

Manual (Man) – The integration calculations are not performed. OUT, OUT_TRIP, and OUT_PTRIP may be set manually.

Automatic (Auto) – The integration algorithm is performed and the result is written to OUT. Reset actions depend on the integration type attribute (INTEG_TYPE) and the inputs.

Out of Service (O/S) – The block does not execute. OUT status is set to *Bad: Out of Service*. The BLOCK_ERR attribute shows **Out of service**.

The integrator initializes with the value in OUT when the mode changes from **Manual** to **Automatic**. The **Manual**, **Automatic**, and **Out of Service** modes may be configured as permitted modes for operator entry.

Status Handling

The output status calculation is based on the accumulation of input statuses. The calculation includes the accumulations for both input channels when IN_2 is enabled.

The input statuses are accumulated in *Good* and *Bad* groups. An input status of *Uncertain* is interpreted as a *Bad* status for the output status calculation. Each time the function block executes, the input status is incremented in the appropriate group. The input status accumulation is reset when the integrator is reset.

The output status is determined with the following logic:

- When less than 25% of the input status accumulation is *Good*, OUT status is set to *Bad*.
- When 25% to less than 50% of the input status accumulation is *Good*, OUT status is set to *Uncertain*.
- When 50% or more of the input status accumulation is *Good*, OUT status is set to *Good*.

Figure 7-3 illustrates output status designations.

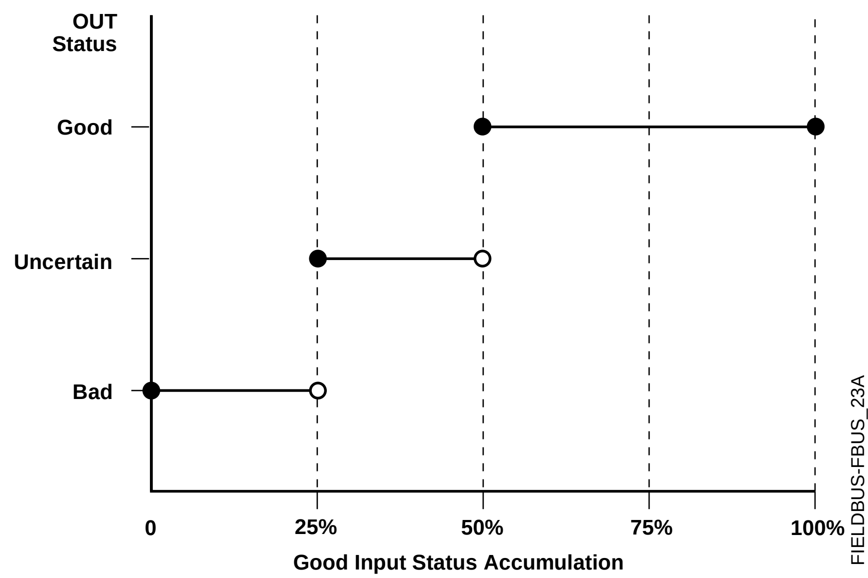


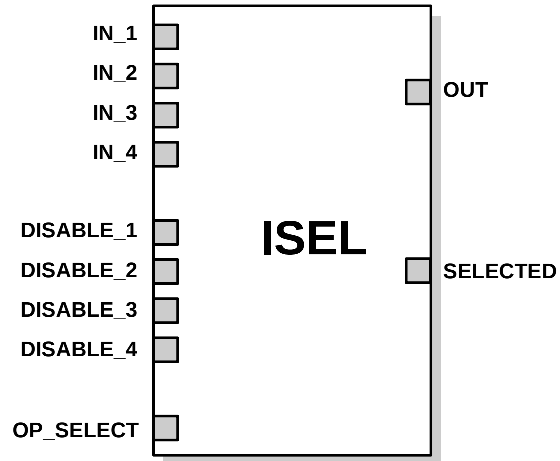
Figure 7-3. Integrator Function Block Output Status Determination

---

NOTE
Default values and data type information for the parameters are available by expanding the Attribute View window.

Section **8**

Input Selector (ISEL) Function Block



IN (1-4) = **Input used in the selection algorithm.**
DISABLE (1-4) = **Discrete input used to enable or disable the associated input channel.**
OP_SELECT = **Input used to override algorithm.**
TRK_VAL = **The value after scaling applied to OUT in Local Override mode.**
SELECTED = **The selected channel number.**
OUT = **The block output and status.**

The Input Selector (ISEL) function block can be used to select the first good, Hot Backup, maximum, minimum, or average of as many as four input values and place it at the output. The block supports signal status propagation. There is no process alarm detection in the Input Selector function block. Figure 8-1 illustrates the internal components of the ISEL function block. Table 8-1 lists the ISEL block parameters and their descriptions, units of measure, and index numbers.

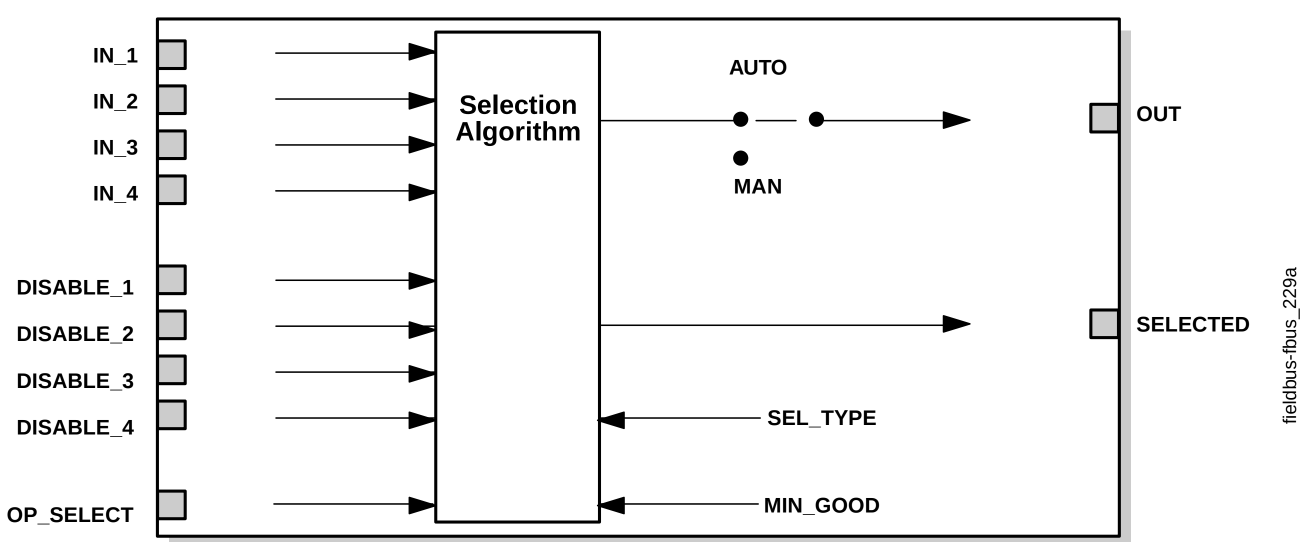
TABLE 8-1. Input Selector Function Block System Parameters

Parameter	Index Number	Description
ST_REV	01	The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in the block is changed.
TAG_DESC	02	The user description of the intended application of the block.
STRATEGY	03	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	04	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
MODE_BLK	05	The actual, target, permitted, and normal modes of the block. Target: The mode to “go to” Actual: The mode the “block is currently in” Permitted: Allowed modes that target may take on Normal: Most common mode for target

TABLE 8-1. Input Selector Function Block System Parameters

Parameter	Index Number	Description
BLOCK_ERR	06	This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown.
OUT	07	The block output value and status.
OUT_RANGE	08	High and low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT
GRANT_DENY	09	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block. Not used by device.
STATUS_OPTIONS	10	Allows you to select options for status handling and processing. The supported status option for the integrator block is: “Use Uncertain as Good”, “Uncertain if Man mode.”
IN_1	11	The block input value and status.
IN_2	12	The block input value and status.
IN_3	13	The block input value and status.
IN_4	14	The block input value and status.
DISABLE_1	15	Parameter to switch off the input from being used 0- Use, 1 - Disable.
DISABLE_2	16	Parameter to switch off the input from being used 0- Use, 1 - Disable.
DISABLE_3	17	Parameter to switch off the input from being used 0- Use, 1 - Disable.
DISABLE_4	18	Parameter to switch off the input from being used 0- Use, 1 - Disable.
SELECT_TYPE	19	Determines the selector action; First good, Minimum, Maximum, Middle, Average.
MIN_GOOD	20	The minimum number of inputs which are “good” is less than the value of MIN_GOOD then set the OUT status to “bad”.
SELECTED	21	The integer indicating the selected input number.
OP_SELECT	22	An operator settable parameter to force a given input to be used.
UPDATE_EVT	23	This alert is generated by any change to the static data.
BLOCK_ALM	24	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.

Figure 8-1. Input Selector Function Block Schematic



Block Errors

Table 8-2 lists conditions reported in the BLOCK_ERR parameter. Conditions in *italics* are inactive for the ISEL block and are given here only for your reference.

TABLE 8-2. BLOCK_ERR Conditions

Condition Number	Condition Name and Description
0	Other: The output has a quality of uncertain.
1	Block Configuration Error
2	Link Configuration Error
3	Simulate Active
4	Local Override: The actual mode is LO.
5	Device Fault State Set
6	Device Needs Maintenance Soon
7	**Input Failure/Process Variable has Bad Status:** One of the inputs is Bad or not connected.
8	Output Failure: The output has a quality of Bad.
9	**Memory Failure:** A memory failure has occurred in FLASH, RAM, or EEROM memory
10	Lost Static Data
11	Lost NV Data
12	Readback Check Failed
13	Device Needs Maintenance Now
14	Power Up: The device was just powered-up.
15	**Out of Service:** The actual mode is out of service.

Modes

The ISEL function block supports three modes of operation as defined by the MODE_BLK parameter:

- **Manual (Man)** The block output (OUT) may be set manually.
- **Automatic (Auto)** OUT reflects the selected value.
- **Out of Service (O/S)** The block is not processed. The BLOCK_ERR parameter shows Out of Service. In this mode, you can make changes to all configurable parameters. The target mode of a block may be restricted to one or more of the supported modes.

Alarm Detection

A block alarm will be generated whenever the BLOCK_ERR has an error bit set. The types of block error for the ISEL block are defined above.

Alarms are grouped into five levels of priority:

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 (such as 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.

Block Execution

The ISEL function block reads the values and statuses of as many as four inputs. To specify which of the six available methods (algorithms) is used to select the output, configure the selector type parameter (SEL_TYPE) as follows:

- **max** selects the maximum value of the inputs.
- **min** selects the minimum value of the inputs.
- **avg** calculates the average value of the inputs.
- **mid** calculates the middle of three inputs or the average of the middle two inputs if four inputs are defined.
- **1st Good** selects the first available good input.
- **Hot Backup** latches on the selected input and continues to use it until it is bad.

If the DISABLE_N is active, the associated input is not used in the selection algorithm.

If the OP_SELECT is set to a value between 1 and 4, the selection type logic is overridden and the output value and status is set to the value and status of the input selected by OP_SELECT.

SELECTED will have the number of the selected input unless the SEL_TYPE is average, in which case it will have the number of inputs used to calculate its value.

Status Handling

In Auto mode, OUT reflects the value and status quality of the selected input. If the number of inputs with Good status is less than MIN_GOOD, the output status will be Bad.

In Man mode, the OUT status high and low limits are set to indicate that the value is a constant and the OUT status is always Good.

In the STATUS_OPTS parameter, you can select from the following options to control the status handling:

- **Use Uncertain as Good**: sets the OUT status quality to Good when the selected input status is Uncertain.
- **Uncertain if in Manual mode**: The status of the Output is set to Uncertain when the mode is set to manual.

NOTE
The instrument must be in Manual or Out of Service mode to set the status option.

Troubleshooting

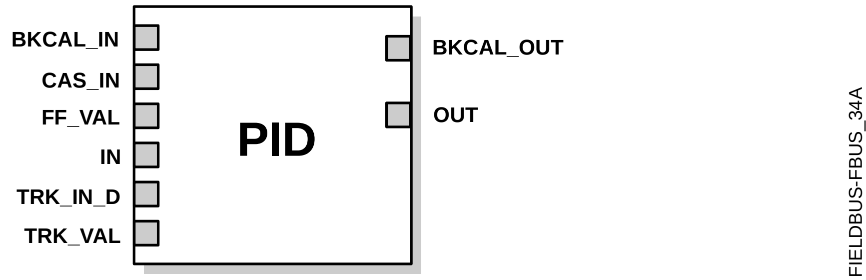
Refer to Table 8-3 to troubleshoot any problems that you encounter.

TABLE 8-3. Troubleshooting

Symptom	Possible Causes	Corrective Action
Mode will not leave OOS	Target mode not set.	Set target mode to something other than OOS.
	Configuration error	BLOCK_ERR will show the configuration error bit set. SELECT_TYPE must be set to a valid value and cannot be left at 0.
	Resource block	The actual mode of the Resource block is OOS. See Resource Block Diagnostics for corrective action.
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
Status of output is BAD	Inputs	All inputs have Bad status
	OP selected	OP_SELECT is not set to 0 (or it is linked to an input that is not 0), and it points to an input that is Bad.
	Min good	The number of Good inputs is less than MIN_GOOD.
Block alarms will not work.	Features	FEATURES_SEL does not have Alerts enabled. Enable the Alerts bit.
	Notification	LIM_NOTIFY is not high enough. Set equal to MAX_NOTIFY.
	Status Options	STATUS_OPTS has Propagate Fault Forward bit set. This should be cleared to cause an alarm to occur.

Section **9**

Proportional/Integral/Derivative (PID) Function Block



BKCAL_IN = **The analog input value and status from another block's BKCAL_OUT output that is used for backward output tracking for bumpless transfer and to pass limit status.**
CAS_IN = **The remote setpoint value from another function block.**
FF_VAL = **The feedforward control input value and status.**
IN = **The connection for the process variable from another function block.**
TRK_IN_D = **Initiates the external tracking function.**
TRK_VAL = **The value after scaling applied to OUT in Local Override mode.**
BKCAL_OUT = **The value and status required by the BKCAL_IN input of another function block to prevent reset windup and to provide bumpless transfer to closed loop control.**
OUT = **The block output and status.**

The PID function block combines all of the necessary logic to perform 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 block supports two forms of the PID equation: Standard and Series. You can choose the appropriate equation using the FORM parameter. The Standard ISA PID equation is the default selection.

$$\text{Standard Out} = \text{GAIN} \times e \times \left(1 + \frac{1}{\tau_r s + 1} + \frac{\tau_d s}{\alpha \times \tau_d s + 1}\right) + F$$

$$\text{Series Out} = \text{GAIN} \times e \times \left[\left(1 + \frac{1}{\tau_r s}\right) + \left(\frac{\tau_d s + 1}{\alpha \times \tau_d s + 1}\right)\right] + F$$

Where

GAIN: proportional gain value
τ_r: integral action time constant (RESET parameter) in seconds
s: laplace operator
τ_d: derivative action time constant (RATE parameter)
α: fixed smoothing factor of 0.1 applied to RATE
F: feedforward control contribution from the feedforward input (FF_VAL parameter)
e: error between setpoint and process variable

To further customize the block for use in your application, you can configure filtering, feedforward inputs, tracking inputs, setpoint and output limiting, PID equation structures, and block output action. Table 9-1 lists the PID block parameters and their descriptions, units of measure, and index numbers, and Figure 9-1 on page 9-5 illustrates the internal components of the PID function block.

TABLE 9-1. PID Function Block System Parameters.

Parameter	Index Number	Units	Description
ACK_OPTION	46	None	Used to set auto acknowledgment of alarms.
ALARM_HYS	47	Percent	The amount the alarm value must return to within the alarm limit before the associated active alarm condition clears.
ALARM_SUM	45	None	The summary alarm is used for all process alarms in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.
ALERT_KEY	04	None	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
ALG_TYPE	74	None	Selects filtering algorithm as Backward or Bilinear.
BAL_TIME	25	Seconds	The specified time for the internal working value of bias to return to the operator set bias. Also used to specify the time constant at which the integral term will move to obtain balance when the output is limited and the mode is AUTO, CAS, or RCAS.
BIAS	66	EU of OUT_SCALE	The bias value used to calculate output for a PD type controller.
BKCAL_HYS	30	Percent	The amount the output value must change away from the its output limit before limit status is turned off.
BKCAL_IN	27	EU of OUT_SCALE	The analog input value and status from another block's BKCAL_OUT output that is used for backward output tracking for bumpless transfer and to pass limit status.
BKCAL_OUT	31	EU of PV_SCALE	The value and status required by the BKCAL_IN input of another block to prevent reset windup and to provide bumpless transfer of closed loop control.
BLOCK_ALM	44	None	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the active status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, and other block alert may be reported without clearing the Active status, if the subcode has changed.
BLOCK_ERR	06	None	This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string so that multiple errors may be shown.
BYPASS	17	None	Used to override the calculation of the block. When enabled, the SP is sent directly to the output.
CAS_IN	18	EU of PV_SCALE	The remote setpoint value from another block.
CONTROL_OPTS	13	None	Allows you to specify control strategy options. The supported control options for the PID block are Track enable, Track in Manual, SP-PV Track in Man, SP-PV Track in LO or IMAN, Use PV for BKCAL_OUT, and Direct Acting
DV_HI_ALM	64	None	The DV HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
DV_HI_LIM	57	EU of PV_SCALE	The setting for the alarm limit used to detect the deviation high alarm condition.
DV_HI_PRI	56	None	The priority of the deviation high alarm.
DV_LO_ALM	65	None	The DV LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
DV_LO_LIM	59	EU of PV_SCALE	The setting for the alarm limit use to detect the deviation low alarm condition.
DV_LO_PRI	58	None	The priority of the deviation low alarm.
ERROR	67	EU of PV_SCALE	The error (SP-PV) used to determine the control action.
FF_ENABLE	70	None	Enables the use of feedforward calculations
FF_GAIN	42	None	The feedforward gain value. FF_VAL is multiplied by FF_GAIN before it is added to the calculated control output.
FF_SCALE	41	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with the feedforward value (FF_VAL).
FF_VAL	40	EU of FF_SCALE	The feedforward control input value and status.
GAIN	23	None	The proportional gain value. This value cannot = 0.
GRANT_DENY	12	None	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block. Not used by the device.

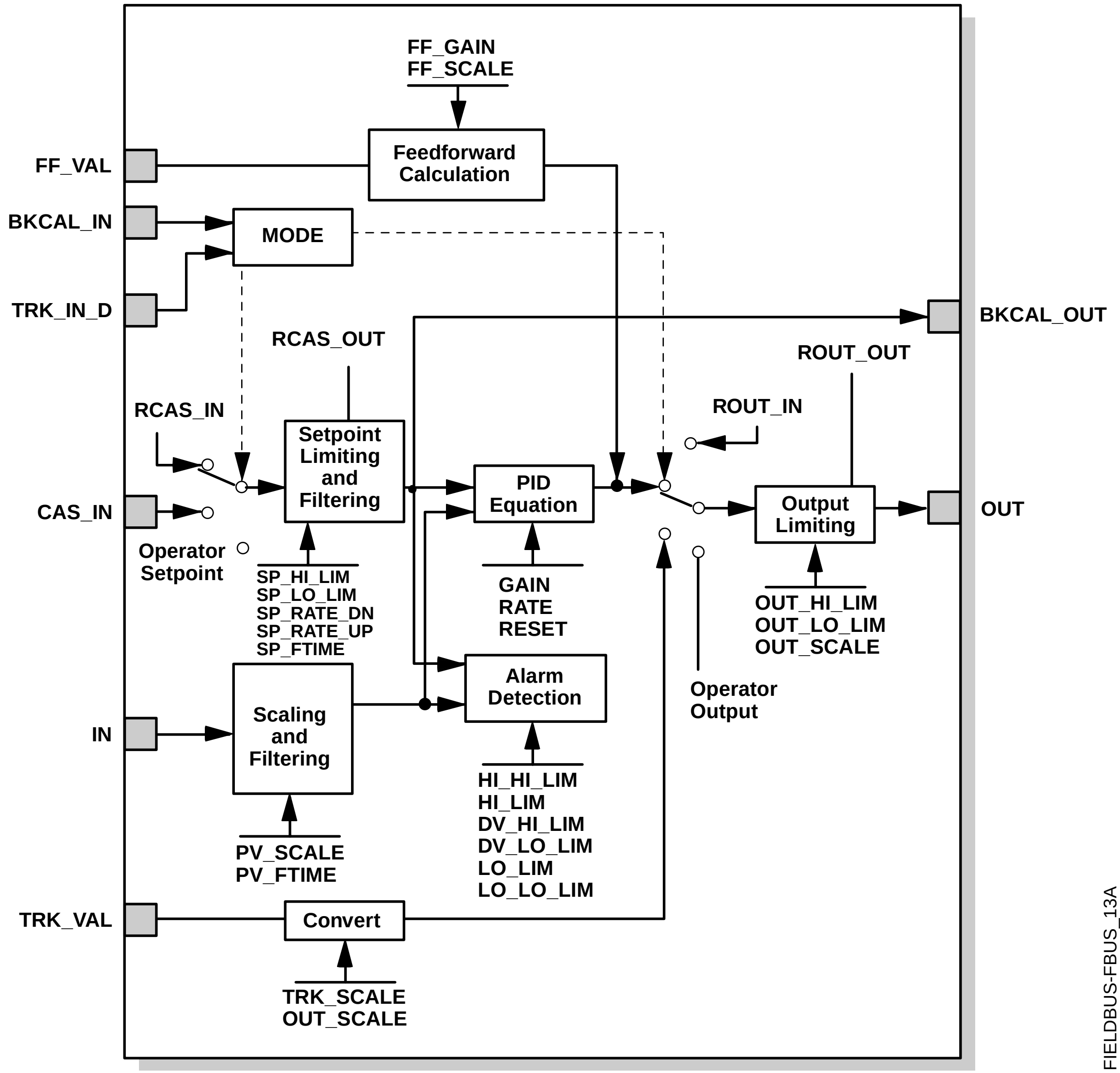
TABLE 9-1. PID Function Block System Parameters.

Parameter	Index Number	Units	Description
HI_ALM	61	None	The HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
HI_HI_ALM	60	None	The HI HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
HI_HI-LIM	49	EU of PV_SCALE	The setting for the alarm limit used to detect the HI HI alarm condition.
HI_HI_PRI	48	None	The priority of the HI HI Alarm.
HI_LIM	51	EU of PV_SCALE	The setting for the alarm limit used to detect the HI alarm condition.
HI_PRI	50	None	The priority of the HI alarm.
IN	15	EU of PV_SCALE	The connection for the PV input from another block.
LO_ALM	62	None	The LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
LO_LIM	53	EU of PV_SCALE	The setting for the alarm limit used to detect the LO alarm condition.
LO_LO_ALM	63	None	The LO LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
LO_LO_LIM	55	EU of PV_SCALE	The setting for the alarm limit used to detect the LO LO alarm condition.
LO_LO_PRI	54	None	The priority of the LO LO alarm.
LO_PRI	52	None	The priority of the LO alarm.
MATH_FORM	73	None	Selects equation form (series or standard).
MODE_BLK	05	None	The actual, target, permitted, and normal modes of the block. Target: The mode to “go to” Actual: The mode the “block is currently in” Permitted: Allowed modes that target may take on Normal: Most common mode for target
OUT	09	EU of OUT_SCALE	The block input value and status.
OUT_HI_LIM	28	EU of OUT_SCALE	The maximum output value allowed.
OUT-LO_LIM	29	EU of OUT_SCALE	The minimum output value allowed
OUT_SCALE	11	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT.
PV	07	EU of PV_SCALE	The process variable used in block execution.
PV_FTIME	16	Seconds	The time constant of the first-order PV filter. It is the time required for a 63 percent change in the IN value.
PV_SCALE	10	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with PV.
RATE	26	Seconds	The derivative action time constant.
RCAS_IN	32	EU of PV_SCALE	Target setpoint and status that is provided by a supervisory host. Used when mode is RCAS.
RCAS_OUT	35	EU of PV_SCALE	Block setpoint and status after ramping, filtering, and limiting that is provided to a supervisory host for back calculation to allow action to be taken under limiting conditions or mode change. Used when mode is RCAS.
RESET	24	Seconds per repeat	The integral action time constant.
ROUT_IN	33	EU of OUT_SCALE	Target output and status that is provided by a supervisory host. Used when mode is ROUT.
ROUT_OUT	36	EU of OUT_SCALE	Block output that is provided to a supervisory host for a back calculation to allow action to be taken under limiting conditions or mode change. Used when mode is RCAS.
SHED_OPT	34	None	Defines action to be taken on remote control device timeout.
SP	08	EU of PV_SCALE	The target block setpoint value. It is the result of setpoint limiting and setpoint rate of change limiting.
SP_FTIME	69	Seconds	The time constant of the first-order SP filter. It is the time required for a 63 percent change in the IN value.
SP_HI_LIM	21	EU of PV_SCALE	The highest SP value allowed.
SP_LO_LIM	22	EU of PV_SCALE	The lowest SP value allowed.
SP_RATE_DN	19	EU of PV_SCALE per second	Ramp rate for downward SP changes. When the ramp rate is set to zero, the SP is used immediately.
SP-RATE_UP	20	EU of PV_SCALE per second	Ramp rate for upward SP changes. When the ramp rate is set to zero, the SP is used immediately.
SP_WORK	68	EU of PV_SCALE	The working setpoint of the block after limiting and filtering is applied.
STATUS_OPTS	14	None	Allows you to select options for status handling and processing. The supported status option for the PID block is Target to Manual if Bad IN.

TABLE 9-1. PID Function Block System Parameters.

Parameter	Index Number	Units	Description
STRATEGY	03	None	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ST_REV	01	None	The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in the block is changed.
STRUCTURE. CONFIG	75	None	Defines PID equation structure to apply controller action.
TAG_DESC	02	None	The user description of the intended application of the block.
TRK_IN_D	38	None	Discrete input that initiates external tracking.
TRK_SCALE	37	None	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with the external tracking value (TRK_VAL).
TRK_VAL	39	EU of TRK_SCALE	The value (after scaling from TRK_SCALE to OUT_SCALE) APPLIED to OUT in LO mode.
UBETA	72	Percent	Used to set disturbance rejection vs. tracking response action for a 2.0 degree of freedom PID.
UGAMMA	71	Percent	Used to set disturbance rejection vs. tracking response action for a 2.0 degree of freedom PID.
UPDATE_EVT	43	None	This alert is generated by any changes to the static data.

Figure 9-1. PID Function Block Schematic



Setpoint Selection and Limiting

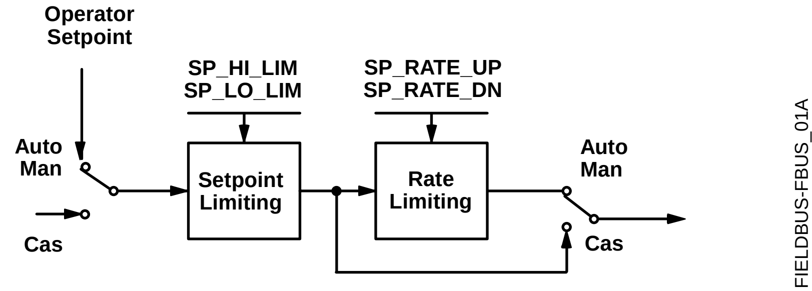
The setpoint of the PID block is determined by the mode. You can configure the SP_HI_LIM and SP_LO_LIM parameters to limit the setpoint. In **Cascade** or **RemoteCascade** mode, the setpoint is adjusted by another function block or by a host computer, and the output is computed based on the setpoint.

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

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

Figure 9-2 illustrates the method for setpoint selection.

Figure 9-2. PID Function Block Setpoint Selection



Filtering

The filtering feature changes the response time of the device to smooth variations in output readings 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. Set the filter time constant to zero to disable the filter feature.

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.

Tracking

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

The **Track Enable** control option must be set to *True* for the track function to operate. When the Track in Manual control option is set to *True*, tracking can be activated and maintained only when the block is in **Manual** mode. When **Track in Manual** is *False*, the operator can override the tracking function when the block is in **Manual** mode. Activating the track function causes the block's actual mode to revert to **Local Override**.

The TRK_VAL parameter specifies the value to be converted and tracked into the output when the track function is operating. The TRK_SCALE parameter specifies the range of TRK_VAL.

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

Output Selection and Limiting

Output selection is determined by the mode and the setpoint. In **Automatic**, **Cascade**, or **RemoteCascade** mode, the output is computed by the PID control equation. In **Manual** and **RemoteOutput** mode, the output may be entered manually. You can limit the output by configuring the OUT_HI_LIM and OUT_LO_LIM parameters.

Bumpless Transfer and Setpoint Tracking

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

SP-PV Track in Man — Permits the SP to track the PV when the target mode of the block is Man.

SP-PV Track in LO or IMan — Permits the SP to track the PV when the actual mode of the block is Local Override (LO) or Initialization Manual (IMan).

When one of these options is set, 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, thus tracking 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 control options in **Manual** or **Out of Service** mode only. When the mode is set to **Auto**, the SP will remain at the last value (it will no longer follow the PV.

PID Equation Structures

Configure the STRUCTURE parameter to select the PID equation structure. You can select one of the following choices:

- PI Action on Error, D Action on PV
- PID Action on Error
- I Action on Error, PD Action on PV

Set RESET to zero to configure the PID block to perform integral only control regardless of the STRUCTURE parameter selection. When RESET equals zero, the equation reduces to an integrator equation with a gain value applied to the error:

$$\frac{GAIN \times e(s)}{s}$$

Where

GAIN: proportional gain value
e: error
s: laplace operator

Reverse and Direct Action

To configure the block output action, enable the **Direct Acting** control option. This option defines the relationship between a change in PV and the corresponding change in output. With **Direct Acting** enabled (True), an increase in PV results in an increase in the output.

You can set control options in **Manual** or **Out of Service** mode only.

NOTE
Track Enable, **Track in Manual**, **SP-PV Track in Man**, **SP-PV Track in LO or IMan**, **Use PV for BKCAL_OUT**, and **Direct Acting** are the only control options supported by the PID function block. Unsupported options are not grayed out; they appear on the screen in the same manner as supported options.

Reset Limiting

The PID function block provides a modified version of feedback reset limiting that prevents windup when output or input limits are encountered, and provides the proper behavior in selector applications.

Block Errors

Table 9-3 lists conditions reported in the BLOCK_ERR parameter. Conditions in *italics* are inactive for the PID block and are given here only for your reference.

TABLE 9-3. BLOCK_ERR Conditions

Condition Number	Condition Name and Description
0	*Other*
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.
2	*Link Configuration Error*
3	*Simulate Active*
4	**Local Override**: The actual mode is LO.
5	*Device Fault State Set*
6	*Device Needs Maintenance Soon*
7	**Input Failure/Process Variable has Bad Status:** The parameter linked to IN is indicating a Bad status.
8	*Output Failure*
9	*Memory Failure*
10	*Lost Static Data*
11	*Lost NV Data*
12	*Readback Check Failed*
13	*Device Needs Maintenance Now*
14	*Power Up*
15	**Out of Service:** The actual mode is out of service.

Modes

The PID function block supports the following modes:

Manual (Man)—The block output (OUT) may be set manually.

Automatic (Auto)—The SP may be set manually and the block algorithm calculates OUT.

Cascade (Cas)—The SP is calculated in another block and is provided to the PID block through the CAS_IN connection.

RemoteCascade (RCas)—The SP is provided by a host computer that writes to the RCAS_IN parameter.

RemoteOutput (Rout)—The OUT IS provided by a host computer that writes to the ROUT_IN parameter

Local Override (LO)—The track function is active. OUT is set by TRK_VAL. The BLOCK_ERR parameter shows Local override.

Initialization Manual (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.

Out of Service (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.

Alarm Detection

A block alarm will be generated whenever the BLOCK_ERR has an error bit set. The types of block error for the AI block are defined above.

Process alarm detection is 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)

Additional process alarm detection is based on the difference between SP and PV values and can be configured via the following parameters:

- Deviation high (DV_HI_LIM)
- Deviation low (DV_LO_LIM)

In order 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
- DV_HI_PRI
- DV_LO_PRI

Alarms are grouped into five levels of priority:

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 (such as 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.

Status Handling

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

NOTE
Target to Manual if Bad IN is the only status option supported by the PID function block. Unsupported options are not grayed out; they appear on the screen in the same manner as supported options.

Troubleshooting

Refer to Table 9-4 to troubleshoot any problems that you encounter.

TABLE 9-4. Troubleshooting

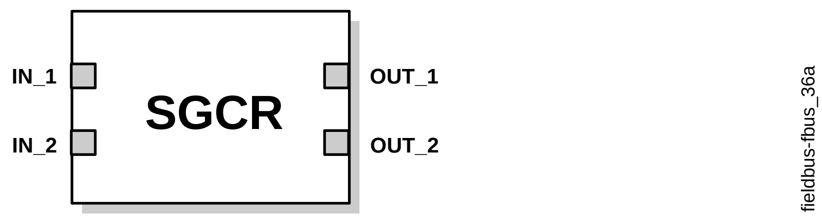
Symptom	Possible Causes	Corrective Action
Mode will not leave OOS	Target mode not set.	Set target mode to something other than OOS.
	Configuration error	BLOCK_ERR will show the configuration error bit set. The following are parameters that must be set before the block is allowed out of OOS: • BYPASS must be off or on and cannot be left at initial value of 0. • OUT_HI_LIM must be less than or equal to OUT_LO_LIM. • SP_HI_LIM must be less than or equal to SP_LO_LIM.
	Resource block	The actual mode of the Resource block is OOS. See Resource Block Diagnostics for corrective action.
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
Mode will not leave IMAN	Back Calculation	BKCAL_IN • The link is not configured (the status would show “Not Connected”). Configure the BKCAL_IN link to the downstream block. • The downstream block is sending back a Quality of “Bad” or a Status of “Not Invited”. See the appropriate downstream block diagnostics for corrective action.
Mode will not change to AUTO	Target mode not set.	Set target mode to something other than OOS.
	Input	IN • The link is not configured (the status would show “Not Connected”). Configure the IN link to the block. • The upstream block is sending back a Quality of “Bad” or a Status of “Not Invited”. See the appropriate upstream block diagnostics for corrective action.
Mode will not change to CAS	Target mode not set.	Set target mode to something other than OOS.
	Cascade input	1. CAS_IN • The link is not configured (the status would show “Not Connected”). Configure the CAS_IN link to the block. • The upstream block is sending back a Quality of “Bad” or a Status of “Not Invited”. See the appropriate up stream block diagnostics for corrective action.
Mode sheds from RCAS to AUTO	Remote Cascade Value	Host system is not writing RCAS_IN with a quality and status of “good cascade” within shed time (see 2 below).
	Shed Timer	The mode shed timer, SHED_RCAS in the resource block is set too low. Increase the value.

TABLE 9-4. Troubleshooting

Symptom	Possible Causes	Corrective Action
Mode sheds from ROUT to MAN	Remote output value	Host system is not writing ROUT_IN with a quality and status of “good cascade” within shed time (see 2 below).
	Shed timer	The mode shed timer, SHED_RCAS, in the resource block is set too low. Increase the value.
Process and/or block alarms will not work.	Features	FEATURES_SEL does not have Alerts enabled. Enable the Alerts bit.
	Notification	LIM_NOTIFY is not high enough. Set equal to MAX_NOTIFY.
	Status Options	STATUS_OPTS has Propagate Fault Forward bit set. This should be cleared to cause an alarm to occur.

Section 10

Signal Characterizer (SGCR) Function Block



IN_1 and IN_2 = The input values to the block.
OUT_1 = The output associated with IN_1.
OUT_2 = The output associated with IN_2.

The Signal Characterizer (SGCR) function block characterizes or approximates any function that defines an input/output relationship. The function is defined by configuring as many as twenty X,Y coordinates. The block interpolates an output value for a given input value using the curve defined by the configured coordinates. Two separate analog input signals can be processed simultaneously to give two corresponding separate output values using the same defined curve.

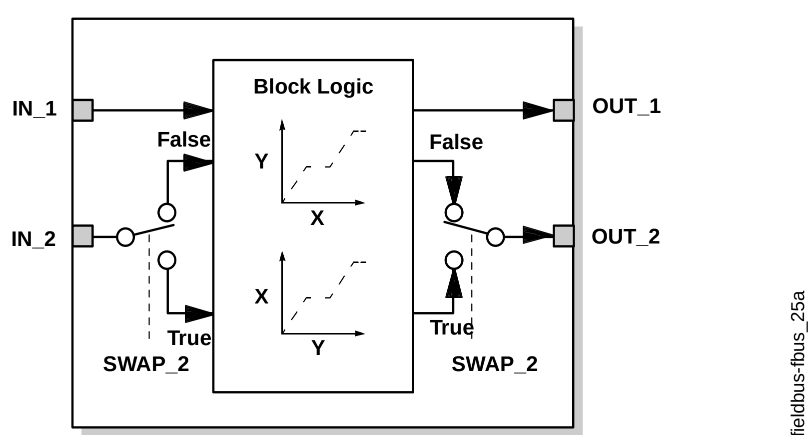
The SGCR block supports signal status propagation. There are no standard alarms in this function block. Custom alarms are supported.

Figure 10-1 illustrates the internal components of the SGCR function block, and Table 10-1 lists the system parameters. The block correlates the input IN_1 to the output OUT_1 and the input IN_2 to the output OUT_2 according to the configured curve. You can configure the curve by defining as many as twenty pairs of X,Y values in the CURVE_X and CURVE_Y parameters. The CURVE_X array defines the input values (X1 to X20) and the CURVE_Y array defines the output values (Y1 to Y20).

TABLE 10-1. Signal Characterizer Function Block System Parameters

Index Number	Parameter	Description
1	ST_REV	The revision level of the static data associated with the function block. The revision value will be incremented in each time a static parameter value in the block is changed.
2	TAF_DESC	The use description of the intended application of the block.
3	STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
4	ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
5	MODE_BLK	The actual, target, permitted, ad normal modes of the block. Target: The mode to “go to” Actual: The mode the “block is currently in” Permitted: Allowed modes that target may take on Normal: Most common mode for target
6	BLOCK_ER R	This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string so that multiple errors may be shown.
7	OUT1	The block output value and status.
8	OUT2	The block output value and status.
9	X_SCALE	The display scaling of the variable corresponding to the x-axis for display. It has no effect on the block.
10	Y_SCALE	The display scaling of the variable corresponding to the y-axis for display. It has no effect on the block.
11	GRANT_DE NY	Options for controlling access of host computers and local control panels to operating, tuning, and alarm parameters of the block. (Not used by the device)
12	IN1	The block input value and status.
13	IN2	The block input value and status.
14	SWAP_2	Changes the algorithm in such a way that IN_2 corresponds to “y” and OUT _2 to “x”.
15	CURVE_X	Curve input points. The “x” points of the curve are defined by an array of 21 points.
16	CURVE_Y	Curve input points. The “y” points of the curve are defined by an array of 21 points.
17	UPDATE_EV T	This alert is generated by any changes to the static data.
18	BLOCK _ALM	The block alarm is used for all configuration, hardware, connection failure, or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the active status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task other block alerts may be reported without clearing the active status, if the subcode has changed.

Figure 10-1. Signal Characterizer Function Block Schematic Diagram



Block Execution

For any given input value, the SGCR block determines where the input lies in CURVE_X and calculates the slope of that segment using the point-slope method:

$$y = mx + b$$

Where:

m: slope of the line
b: y-intercept of the line

Using this formula, the block derives an output value that corresponds to the input. When the input lies beyond the range configured in CURVE_X, the output is clamped to the corresponding limit in the CURVE_Y array.

CURVE_X Values

The CURVE_X values must be defined in ascending order. The X1 element must be the smallest value and each following X value must be larger than the previous X value (see Figure 10-2). If the X values are not configured in ascending order, a block alarm is set and the last X value that is larger than the previous one is used as the curve endpoint.

The curve in Figure 10-3 has an invalid definition because X3 is equal to X4. Between these points, the Y value is undefined because it can be any value from Y3 to Y4. In this configuration, the X3,Y3 pair becomes the endpoint for the curve definition. To use the X4,Y4 pair, you must designate X4 to be greater than X3; it cannot be configured as X3 = X4.

Figure 10-2. Example of Valid Signal Characterizer Function Block CURVE_X Values

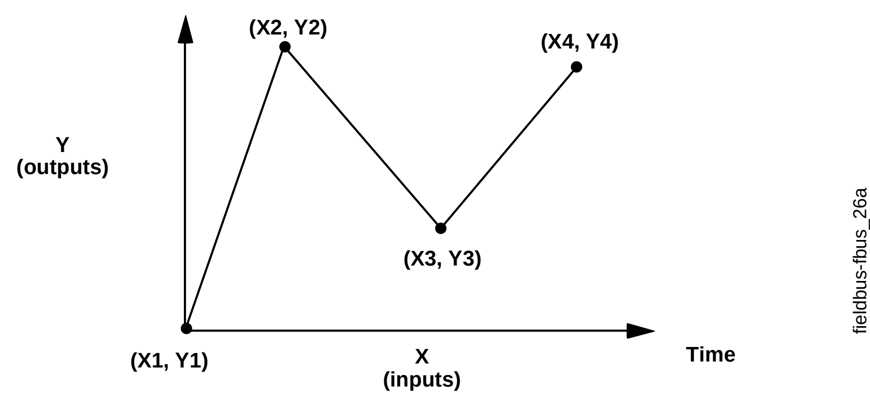
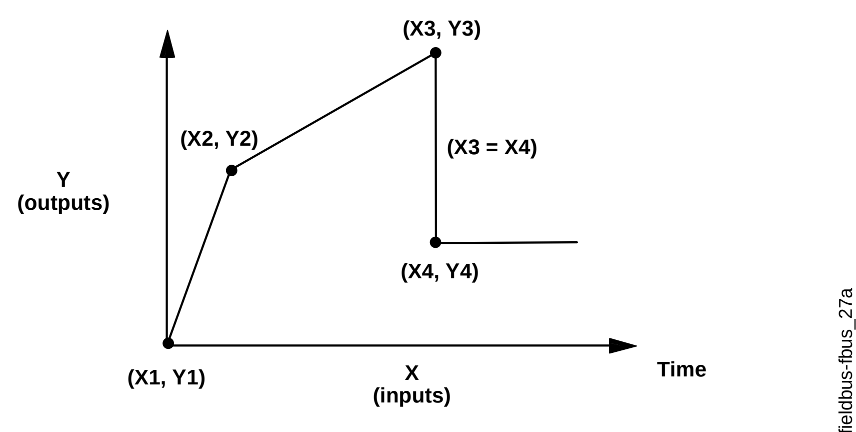


Figure 10-3. Example of Invalid Signal Characterizer Function Block CURVE_X Values



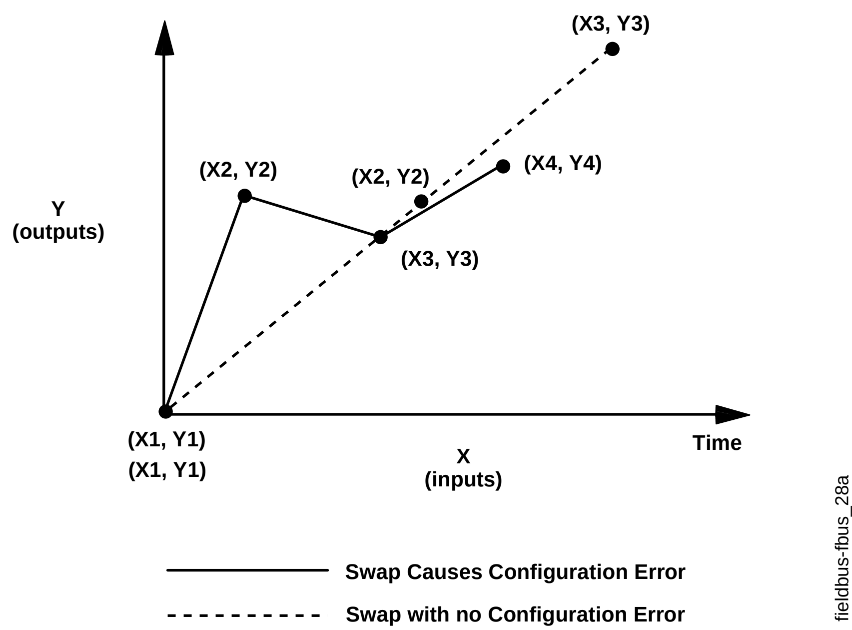
SWAP_2 Attribute

The SWAP_2 attribute swaps the X and Y axes used for OUT_2. When the SWAP_2 attribute is *True*, IN_2 references the CURVE_Y values and OUT_2 references the CURVE_X values. In addition, the IN_2 units change to Y_UNITS and the OUT_2 units change to X_UNITS.

The block sets a configuration error when SWAP_2 is *True* and the CURVE_Y elements are not configured in an increasing manner. Figure 10-4 illustrates how the block configuration error (BLOCK_ERR) is set during a SWAP_2 action.

When swap is in effect, the first curve has an invalid definition because Y3 is less than Y2. In this configuration, the X2,Y2 pair becomes the endpoint for the swapped curve definition when processing IN_2. Note that the X4,Y4 pair is the valid endpoint when processing IN_1.

Figure 10-4. Example of a Signal Characterizer Function Block SWAP_2 Configuration Error



Block Errors

The following conditions are reported in the BLOCK_ERR attribute:

Block configuration error – The curve definition is truncated (fewer than 20 points are defined) and the X value of the X,Y pairs beyond the valid definition are nonzero. The block still performs the signal characterizer function.

Out of service – The block is in Out of Service (O/S) mode.

Modes

The Signal Characterizer function block supports the following modes:

Automatic (Auto) – The outputs are controlled by the block.

Out of Service (O/S) – The block is not processed. The block is placed in O/S mode when both IN_1 and IN_2 inputs have *Bad* statuses. The BLOCK_ERR attribute shows Out of service.

Status Handling

The OUT_1 status is set to the IN_1 status and the OUT_2 status is set to the IN_2 status. When one of the curve limits is reached, the appropriate limit is set in the substatus.

Section

11 Diagnostics (ADB) Transducer Block

The diagnostics block (ADB) is a transducer block that contains two different algorithms, Plugged Impulse Line Detection and Statistical Process Monitoring. Learning, detection and configuration are the three different phases to the Plugged Impulse Line and Statistical Process Monitoring.

TABLE 11-1. Diagnostics Block (ADB) Parameters

Parameter	Index	Description
BLOCK	800	Transducer block characteristics
ST_REV	801	Static revision
TAG_DESC	802	Tag descriptor
STRATEGY	803	Strategy
ALERT_KEY	804	Alert key
MODE_BLK	805	Mode
BLOCK_ERR	806	Block error
UPDATE_EVT	807	Update event
BLOCK_ALM	808	Block alarm
TRANSDUCER_DIRECTORY	809	Transducer directory
TRANSDUCER_TYPE	810	Transducer type
XD_ERROR	811	Transducer error
COLLECTION_DIRECTORY	812	Data collection directory
ADB_STATUS	813	ADB status
DIAG_EVT	814	Diagnostics event

STATISTICAL PROCESS MONITORING

Statistical Process Monitoring algorithm is intended to provide basic information regarding the behavior of process measurement, PID control block and actual valve position. This algorithm can perform higher levels of diagnostics by distribution of computational power to field devices. The two statistical parameters monitored by the Statistical Process Monitoring is mean and standard deviation. By using the mean and standard deviation, the process or control levels and dynamics can be monitored for change over time. The algorithm also provides:

- Configure limits/alarms for changes in mean and standard deviation with respect to the learned levels
- High variation, low dynamics, and mean changes
- Necessary statistical information for Regulatory Control Loop Diagnostics, Root Cause Diagnostics, and Operations Diagnostics

Learning Phase

In the learning phase of Statistical Process Monitoring, the algorithm establishes a baseline of the mean and dynamics of a Statistical Process Monitoring variable. The baseline data is compared to current data for calculating any changes in mean or dynamics of the Statistical Process Monitoring variables.

Monitoring Phase

The monitoring phase starts after a "Training Complete" message is posted for the variable. The algorithm compares the current values to the baseline values of the mean and standard deviation. During this phase the algorithm computes the percent change in mean and standard deviation to determine if the defined limits are violated.

Configuration Phase

The configuration phase is an inactive state or when the ADB mode is OOS. In this phase, the block tags, limits for high variation, low dynamics, and mean change detection can be set by the user.

TABLE 11-2. Statistical Process Monitoring Parameters

Parameter	Index	Units	Description
SPM_ACTIVE	815	NA	Enables/disables the statistical process monitoring alogrithm
SPM_MONITORING_CYCLE	816	min	Baseline length and frequency of mean and Stdev updates
SPM_BYPASS_VERIFICATION	817	NA	Enables/Disables Bypas of process stability checks during learning
SPM1_BLOCK_TYPE	818	NA	The function block type for the SPM variable
SPM1_BLOCK_TAG	819	NA	Block tag for the process variable
SPM1_PARAM_INDEX	820	NA	OD parameter index for the block tag that is entered.
SPM1_STATUS	821	NA	Status of the SPM1 Statistical Process Monitoring
SPM1_TIMESTAMP	822	time	Timestamp of last SPM1 Statistical process Monitoring status change
SPM1_USER_COMMAND	823	NA	User control for the Statistical Process Monitoring session.
SPM1_MEAN	824	NA	Last mean of SPM1
SPM1_MEAN_CHANGE	825	%	Percent change in SPM1 mean with respect to baseline mean
SPM1_STDEV	826	NA	Last Stdev of SPM1
SPM1_STDEV_CHANGE	827	%	Change in SPM1 Stdev with respect to baseline Stdev
SPM1_BASELINE_MEAN	828	NA	Baseline Mean for SPM1
SPM1_BASELINE_STDEV	829	NA	Baseline Stdev for SPM1
SPM1_MEAN_LIM	830	%	Percent change in mean for SPM1 allowed by user
SPM1_HIGH_VARIATION_LIM	831	%	Percent increase in dynamics for SPM1 allowed by user
SPM1_LOW_DYNAMICS_LIM	832	%	Percent decrease in dynamics for SPM1 allowed by user
SPM2_BLOCK_TYPE	833	NA	The function block type for the SPM variable
SPM2_BLOCK_TAG	834	NA	Block tag for the process variable

Parameter	Index	Units	Description
SPM2_PARAM_INDEX	835	NA	OD Parameter index for the Block tag that is entered
SPM2_STATUS	836	NA	Status of the SPM2 Statistical Process Monitoring
SPM2_TIMESTAMP	837	time	Timestamp of last SPM2 Statistical Process Monitoring status change
SPM2_USER_COMMAND	838	NA	User control for the Statistical Process Monitoring session
SPM2_MEAN	839	NA	Last mean of SPM2
SPM2_MEAN_CHANGE	840	%	Percent change in SPM2 mean with respect to baseline mean
SPM2_STDEV	841	NA	Last Stdev of SPM2
SPM2_STDEV_CHANGE	842	%	Change in SPM2 Stdev with respect to baseline Stdev
SPM2_BASELINE_MEAN	843	NA	Baseline mean for SPM2
SPM2_BASELINE_STDEV	844	NA	Baseline Stdev for SPM2
SPM2_MEAN_LM	845	%	Percent change in mean for SPM2 allowed by user
SPM2_HIGH_VARIATION_LIM	846	%	Percent increase in dynamics for SPM2 allowed by user
SPM2_LOW_DYNAMICS_LIM	847	%	Percent decrease in dynamics for SPM2 allowed by user
SPM3_BLOCK_TYPE	848	NA	The function block type for the SPM variable
SPM3_BLOCK_TAG	849	NA	Block tag for the process variable
SPM3_PARAM_INDEX	850	NA	OD Parameter index for the Block tag that is entered
SPM3_STATUS	851	NA	Status of the SPM3 Statistical Process Monitoring
SPM3_TIMESTAMP	852	time	Timestamp of last SPM3 Statistical Process Monitoring status change
SPM3_USER_COMMAND	853	NA	User control for the Statistical Process Monitoring session
SPM3_MEAN	854	NA	Last mean of SPM3
SPM3_MEAN_CHANGE	855	%	Percent change in SPM3 mean with respect to baseline mean
SPM3_STDEV	856	NA	Last Stdev of SPM3
SPM3_STDEV_CHANGE	857	%	Change in SPM3 Stdev with respect to baseline Stdev
SPM3_BASELINE_MEAN	858	NA	Baseline mean for SPM3
SPM3_BASELINE_STDEV	859	NA	Baseline Stdev for SPM3
SPM3_MEAN_LM	860	%	Percent change in mean for SPM3 allowed by user
SPM3_HIGH_VARIATION_LIM	861	%	Percent increase in dynamics for SPM3 allowed by user
SPM3_LOW_DYNAMICS_LIM	862	%	Percent decrease in dynamics for SPM3 allowed by user
SPM4_BLOCK_TYPE	863	NA	The function block type for the SPM variable
SPM4_BLOCK_TAG	864	NA	Block tag for the process variable
SPM4_PARAM_INDEX	865	NA	OD Parameter index for the Block tag that is entered
SPM4_STATUS	866	NA	Status of the SPM4 Statistical Process Monitoring
SPM4_TIMESTAMP	867	time	Timestamp of last SPM4 Statistical Process Monitoring status change

Parameter	Index	Units	Description
SPM4_USER_COMMAND	868	NA	User control for the Statistical Process Monitoring session
SPM4_MEAN	869	NA	Last mean of SPM4
SPM4_MEAN_CHANGE	870	%	Percent change in SPM4 mean with respect to baseline mean
SPM4_STDEV	871	NA	Last Stdev of SPM4
SPM4_STDEV_CHANGE	872	%	Change in SPM4 Stdev with respect to baseline Stdev
SPM4_BASELINE_MEAN	873	NA	Baseline mean for SPM4
SPM4_BASELINE_STDEV	874	NA	Baseline Stdev for SPM4
SPM4_MEAN_LM	875	%	Percent change in mean for SPM4 allowed by user
SPM4_HIGH_VARIATION_LIM	876	%	Percent increase in dynamics for SPM4 allowed by user
SPM4_LOW_DYNAMICS_LIM	877	%	Percent decrease in dynamics for SPM4 allowed by user

PLUGGED IMPULSE LINE DETECTION

Plugged Impulse Line Detection is designed to be used in a continuous liquid flow application that is controlled by a set point.

Learning Phase

In the learning phase, the algorithm establishes a baseline of the process mean and process dynamics. The base line data is used to compare current process data when determining a plugged impulse line.

Detection Phase

The detection phase begins after a “Training Complete” message is posed by the learning phase. To determine if the impulse lines are plugged., the algorithm compares the baseline process variable mean to the standard deviation.

Configuration Phase

The configuration phase is an inactive state and the user can determine the affect on the PV status. Once the configuration is valid, the message “Valid Configuration is posted and the configuration phase stops. The configuration phase is also used to set up the parameters controlling the sensitivity, thresholds, length of leading and detecting phases.

TABLE 11-3. Plugged Impulse Line Parameters

Parameter	Index	Units	Description
PLINE_STATUS	878	NA	Last Impulse line status
PLINE_TIMESTAMP	879		Last Timestamp for PLINE_STATUS
PLINE_ON	880	NA	Turns algorithm on/off
PLINE_RELEARN	881	NA	Resets the algorithm and reinitiates learning
PLINE_SENSITIVITY	882	NA	Detection sensitivity
PLINE_AFFECT_PV_STATUS	883	NA	Determines whether the quality of the pressure measurement will be affected or unaffected
PLINE_HISTORY_STATUS	884	NA	Last plugged line determination status
PLINE_HISTORY_TIMESTAMP	885		Last plugged line determination timestamp
PLINE_LEARN_LENGTH	886	Min	Length of learning and verification cycles in minutes
PLINE_DETECT_LENGTH	887	Min	Length of detection cycle status update in minutes
PLINE_AUTO_RELEARN	888		Enables/disables auto relearn on process mean changes
PLINE_RELEARN_THRESHOLD	889	%URL	Threshold for relearning in%URL of sensor (excludes Range 1 and 2 DP... Those thresholds are fixed at 2 and 5 inH20)
PLINE_LEARN_SENSITIVITY	890		Learning sensitivity check options. Only one of bits 2,3, and 4 is allowed and only one of bits 5 and 6 is allowed.
PLINE_DETECT_SENSITIVITY	891	%	Overrides IL Plugging sensitivity if a non-zero value is entered. Value corresponds to a percentage decrease in standard deviation.

Appendix A Applications

ANALOG INPUT (AI) FUNCTION BLOCK

The configuration of the AI function block and its associated output channels depends on the specific application. A typical configuration for the AI block involves the following parameters:

CHANNEL If the device supports more than one measurement, verify that the selected channel contains the appropriate measurement or derived value.

L_TYPE Select **Direct** when the measurement is already in the engineering units that you want for the block output.

Select **Indirect** when you want to convert the measured variable into another, for example, pressure into level or flow into energy.

Select **Indirect Square Root** when the block I/O parameter value represents a flow measurement made using differential pressure, and when square root extraction is not performed by the transducer.

SCALING **XD_SCALE** provides the range and units of the measurement and **OUT_SCALE** provides the range and engineering units of the output.

Application Example: Temperature Transmitter

Situation

A temperature transmitter with a range of –200 to 450 °C.

Solution

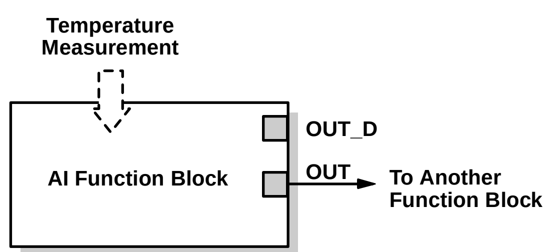
Table A-1 lists the appropriate configuration settings, and Figure A-1 illustrates the correct function block configuration.

.

TABLE A-1. Analog Input Function Block Configuration for a Typical Temperature Transmitter

Parameter	Configured Values
L_TYPE	Direct
XD_SCALE	Not Used
OUT_SCALE	Not Used

Figure A-1. Analog Input Function Block Diagram for a Typical Temperature Transmitter.



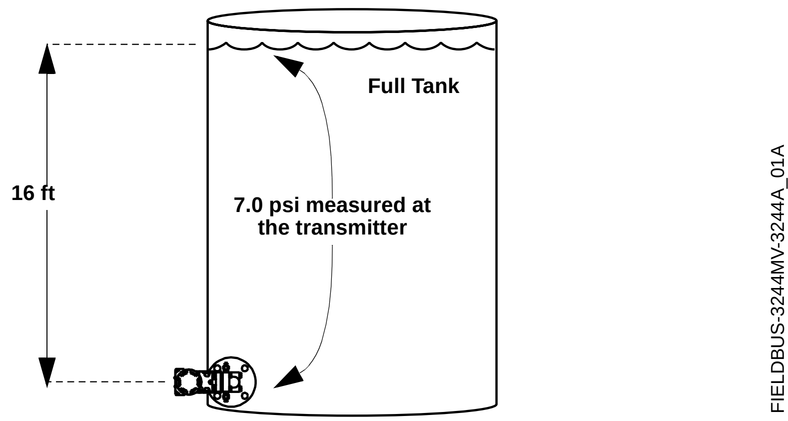
FIELDBUS-FBUS_04A

Application Example: Pressure Transmitter used to Measure Level in an Open Tank

Situation #1

The level of an open tank is to be measured using a pressure tap at the bottom of the tank. The level measurement will be used to control the level of liquid in the tank. The maximum level at the tank is 16 ft. The liquid in the tank has a density that makes the level correspond to a pressure of 7.0 psi at the pressure tap (see Figure A-2).

Figure A-2. Situation #1 Diagram



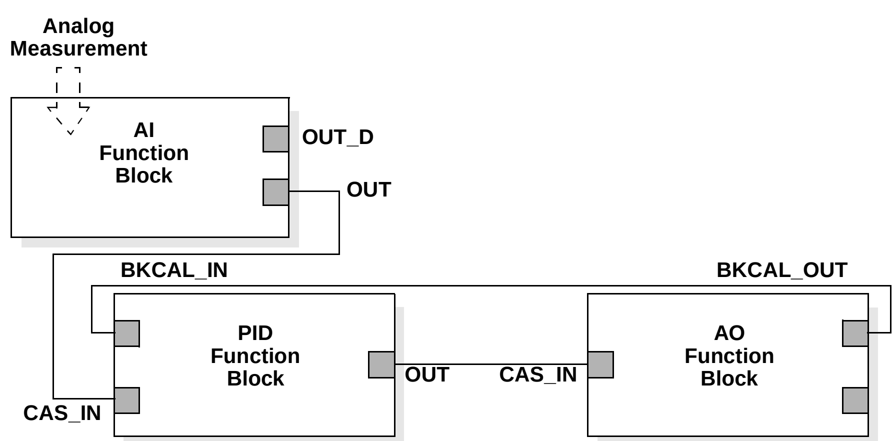
Solution to Situation #1

Table A-2 lists the appropriate configuration settings, and Figure A-3 illustrates the correct function block configuration.

TABLE A-2. Analog Input Function Block Configuration for a Pressure Transmitter used in Level Measurement (situation #1).

Parameter	Configured Values
L_TYPE	Indirect
XD_SCALE	0 to 7 psi
OUT_SCALE	0 to 16 ft

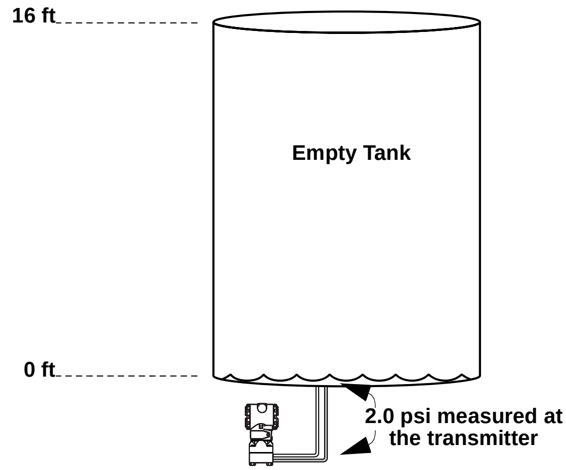
Figure A-3. Function Block Diagram for a Pressure Transmitter used in Level Measurement.



Situation #2

The transmitter in situation #1 is installed below the tank in a position where the liquid column in the impulse line, when the tank is empty, is equivalent to 2.0 psi (see Figure A-4).

Figure A-4. Situation #2 Diagram



Solution

Section : Analog Input Function Block Configuration for a Pressure Transmitter used in Level Measurement (Situation #2) lists the appropriate configuration settings.

TABLE A-3. Analog Input Function Block Configuration for a Pressure Transmitter used in Level Measurement (Situation #2)

Parameter	Configured Values
L_TYPE	Indirect
XD_SCALE	2 to 9 psi
OUT_SCALE	0 to 16 ft

Application Example: Differential Pressure Transmitter to Measure Flow

Situation

The liquid flow in a line is to be measured using the differential pressure across an orifice plate in the line, and the flow measurement will be used in a flow control loop. Based on the orifice specification sheet, the differential pressure transmitter was calibrated for 0 to 20 in$H_2$0 for a flow of 0 to 800 gal/min, and the transducer was not configured to take the square root of the differential pressure.

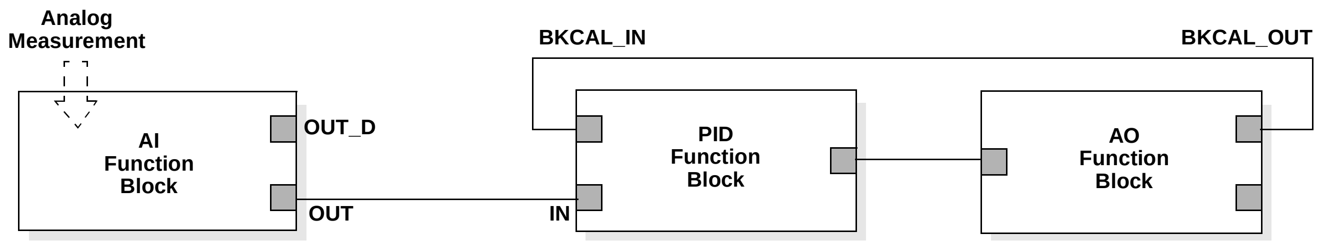
Solution

Table A-4 lists the appropriate configuration settings, and Figure A-5 illustrates the correct function block configuration.

TABLE A-4. Analog Input Function Block Configuration for a Differential Pressure Transmitter

Parameter	Configured Values
L_TYPE	Indirect Square Root
XD_SCALE	0 to 20 in.
OUT_SCALE	0 to 800 gal/min.

Figure A-5. Function Block Diagram for a Differential Pressure Transmitter Used in a Flow Measurement



ANALOG OUTPUT (AO) FUNCTION BLOCK

The configuration of an AO function block and its associated output channels depends on the specific application. A typical configuration for the Analog Output involves the following parameters:

PV_SCALE Set the range and engineering units to values that correspond to the operation range. In many cases, PV_SCALE is set to 0 – 100%.

IO_OPTS Select **Increase to Close** when the actuator is designed to fail open.

BKCAL_OUT If you are using the CAS_IN connector wired from another block, wire the BKCAL_OUT attribute to the other block's BKCAL_IN attribute.

Application Example: Using an AO Block and a Valve to Control Flow in a Pipe

Situation

A regulating valve equipped with an air-operated actuator is connected to the analog output channel to control flow in a pipe.

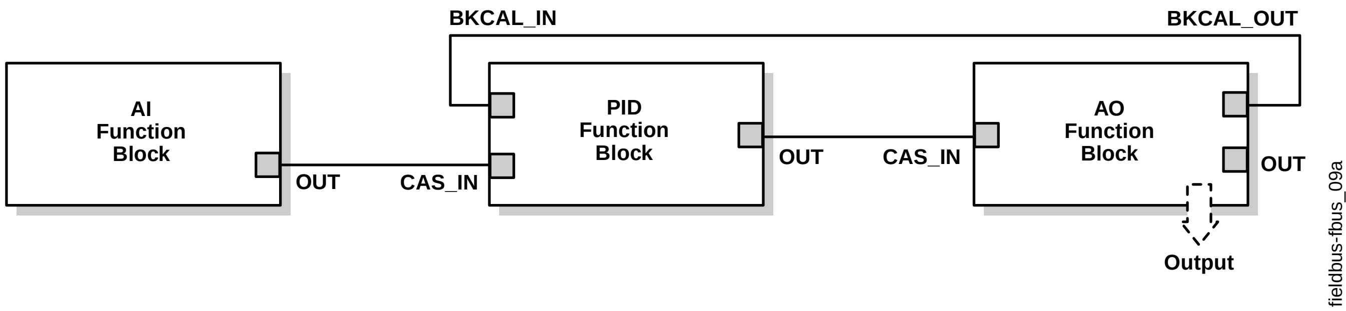
Solution

The Analog Output (AO) function block is used with an AI and a PID function block. The configuration differs depending on whether the valve actuator is designed to allow the valve to fail closed or to fail open upon the loss of power. Table A-5 lists the appropriate settings for each attribute, and Figure A-6 illustrates the correct function block configuration.

TABLE A-5. Analog Output Function Block Configuration Example

Attribute	Configured Values	
	Fail Close	Fail Open
PV_SCALE	0 to 100%	0 to 100%
XD_SCALE	0 to 100%	0 to 100%
IO_OPTS Increase to Close	Not selected	Selected

Figure A-6. Analog Output Function Block Diagram Example



ARITHMETIC (ARTH) FUNCTION BLOCK

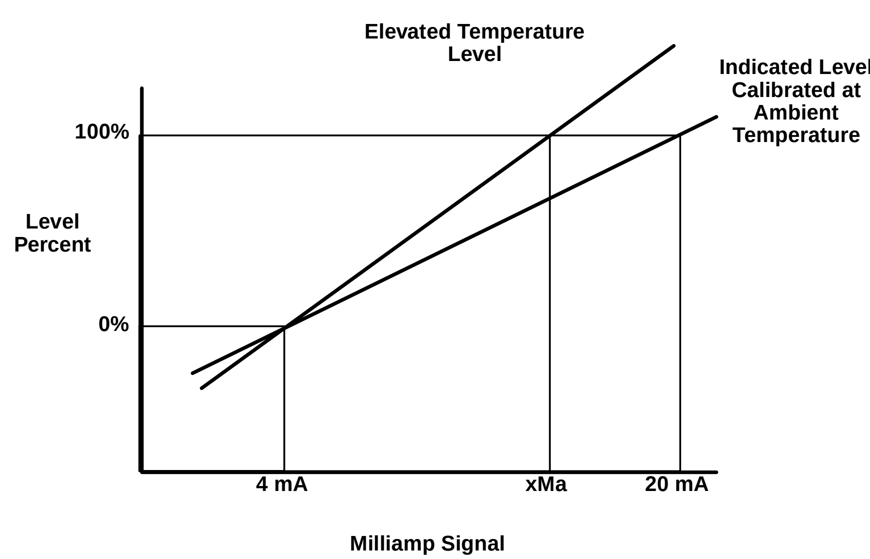
The Arithmetic function block can be used to calculate tank level changes based on greatly changing temperature conditions in devices that depend on the physical properties of the fluid.

For example, a differential pressure cell's analog input can be scaled initially to provide a 4-20 mA signal for 0-100% of level indication. As the temperature of the system rises, the density of the fluid decreases. For a system that requires accurate level indication at widely ranging temperature, changing density proves inconvenient.

The Arithmetic function block allows for the automatic compensation of this change by incorporating gain and bias adjustments to the temperature signal. It then applies both the compensated temperature signal and the level signal to a characteristic system equation. The result is a level that is a true indication of fluid in the vessel.

Different fluids over the same temperature range have different effects on level due to their thermal expansion coefficients. Vessel geometry also plays a major role. As the height of the vessel increases, the effect of thermal expansion becomes more apparent. The following figure shows the relative temperature effects on a level signal.

Figure A-6. Relative Temperature Effects on Level



The calculation is done by applying the level signal to the IN connector, the liquid temperature to the IN_1 connector, and the ambient air temperature to the IN_2 connector. Select the Arithmetic type (ARITH_TYPE) of Flow Compensation - Linear. This allows a ratio to be set up that increases the level indication at block output for an increase in the tank temperature relative to ambient temperature.

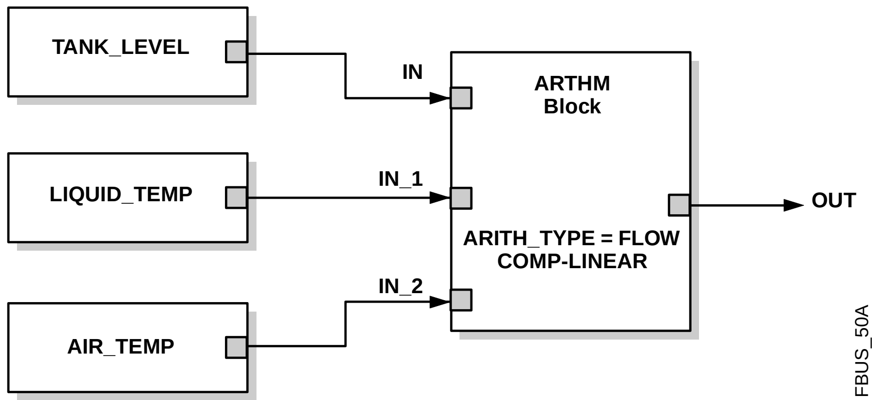


Figure A-7. Arithmetic Function Block Diagram Example

This application can be applied to very large storage tanks whose contents are subject to thermal expansion and contraction during seasonal changes in temperature.

Advanced Topics

Arithmetic Types

The parameter, ARITH_TYPE, determines how PV and the compensation terms (t) are combined. User may select from nine (9) commonly used math functions, depicted below. COMP_HI and COMP_LO are compensation limits.

- Flow Compensation Linear:

$$\text{func} = PV \bullet f$$

$$\text{COMP\_HI}$$
$$f = \frac{t(1)}{t(2)}$$
$$\text{COMP\_LO}$$

- Flow Compensation Square Root:

$$\text{func} = PV \bullet f$$

$$\text{COMP\_HI}$$
$$f = \sqrt{\frac{t(1) \bullet t(3)}{t(2)}}$$
$$\text{COMP\_LO}$$

If there is a divide by zero and the numerator is positive, f is set to COMP_HI; if the numerator is negative, then f is set to COMP_LO. The square root of a negative value will equal the negative of the square root of the absolute value. Imaginary roots are not supported.

- Flow Compensation Approximate:

$$func = PV \bullet f$$

$$f = \left. t(1) \bullet t(2) \bullet t(3)^2 \right|_{COMP\_LO}^{COMP\_HI}$$

- BTU Flow:

$$func = PV \bullet f$$

$$f = \left. t(1) - t(2) \right|_{COMP\_LO}^{COMP\_HI}$$

- Traditional Multiply and Divide:

$$func = PV \bullet f$$

$$f = \left. \frac{t(1)}{t(2)} + t(3) \right|_{COMP\_LO}^{COMP\_HI}$$

If there is a divide by zero and numerator is positive, f will be limited to COMP_HI; if the numerator is negative, f will be limited to COMP_LO.

- Average

$$Sum = PV.Val;\ n = 1$$

$$For\ k = 1, 3 \{ sum = sum + t(k);\ n = n + 1 \} EndFor$$

$$func = \frac{sum}{n}$$

Compensation inputs which are not usable are not included in the calculation. PV is always included.

- Summer

$$Sum = PV$$

$$For\ k = 1, 3 \{ sum = sum + t(k);\ n = n + 1 \} EndFor$$

$$func = sum$$

Compensation inputs which are not configured are not used in the calculation. PV is always used.

- Fourth-Order Polynomial

$$func = PV + t(1)^2 + t(2)^3 + t(3)^4$$

- Simple HTG Compensate Level

$$func = \frac{PV - t(1)}{PV - t(2)}$$

If there is a divide by zero and the numerator is positive, function will be limited to COMP_HI; if the numerator is negative, function will be limited to COMP_LO.

DISCRETE INPUT (DI) FUNCTION BLOCK

The configuration of the DI function block and its associated output channels depends on the specific application. A typical configuration for the Discrete Input involves the following attribute:

IO_OPTS Select Invert to reverse the value supplied for a two-state input.

Application Example: Discrete Input Function Block

Situation

A DI block is used with a DO block to drive a solenoid valve under the active condition.

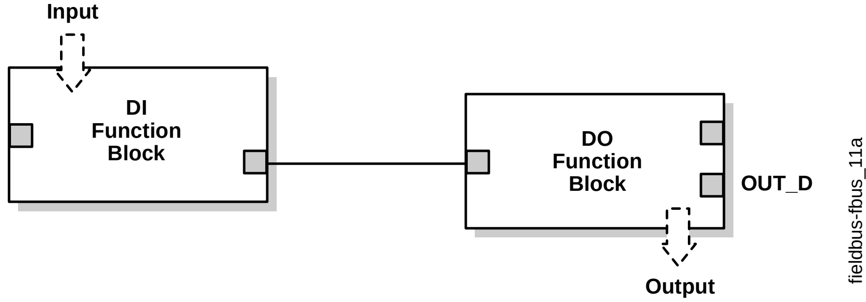
Solution

The configuration differs depending on whether the valve actuator is designed to open or close under the active condition, or to fail open on the loss of power. Table A-8 lists the appropriate settings for each attribute, and Figure A-7 illustrates the correct function block configuration.

TABLE A-8. Analog Output Function Block Configuration Example

Configuration Setting	Open if Active	Close if Active
IO_OPTS Invert	Not selected	Selected

Figure A-7. Discrete Input Function Block Diagram Example



DISCRETE OUTPUT (DO) FUNCTION BLOCK

More information to come.

INPUT SELECTOR (ISEL) FUNCTION BLOCK

You can use the ISEL function block to select the maximum temperature input from four inputs and send it to a PID function block to control a process water chiller (see Figure A-8) or you can use the block to calculate the average temperature of the four inputs (see Figure A-9).

Figure A-8. Input Selector Function Block Application Example (SEL_TYPE = max)

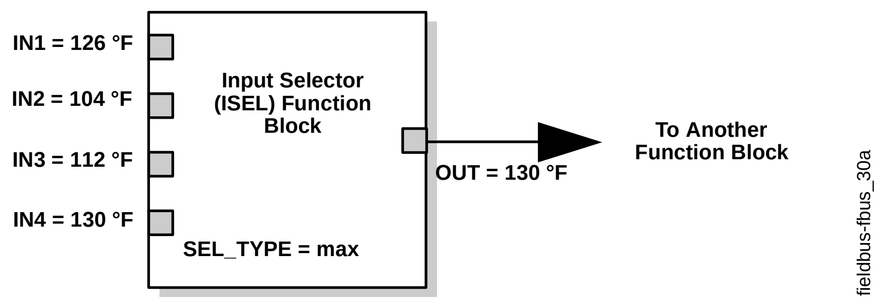


Figure A-9. Input Selector Function Block Application Example (SEL_TYPE = avg)

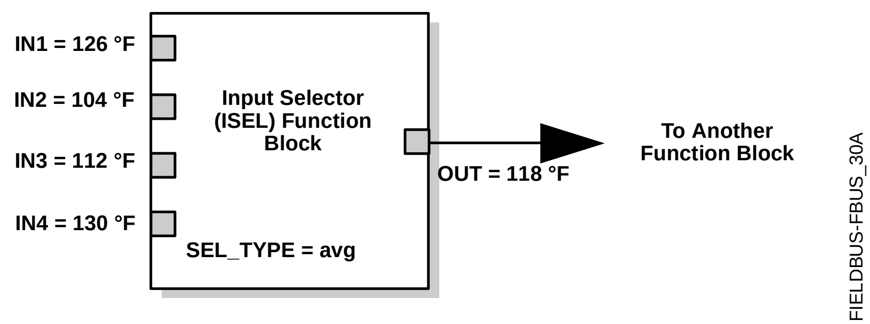
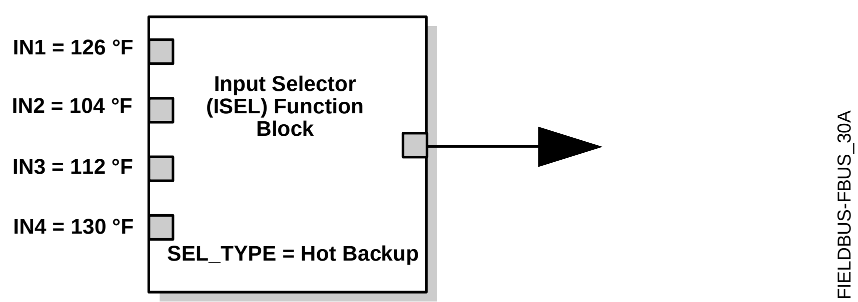


Figure A-10. Input Selector Function Block Application Example (SEL_TYPE = Hot Backup)



	IN1		IN2		Out		Selected	
Time	Value	Status	Value	Status	Value	Status	Value	Status
T_0	Good	20	Good	21	Good	20	Good	1
T_1	Bad	20	Good	21	Good	21	Good	2
T_2	Good	20	Good	21	Good	21	Good	2

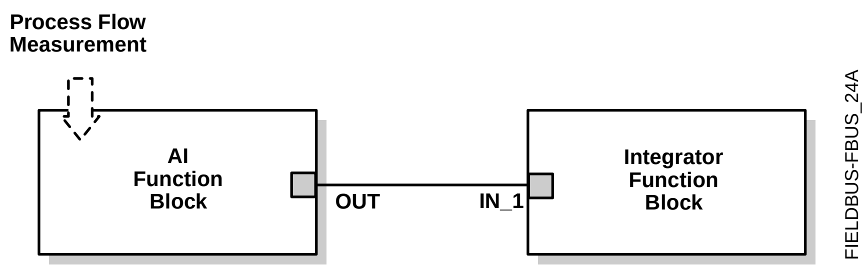
INTEGRATOR (INT) FUNCTION BLOCK

The Integrator function block is useful for calculating total flow, total mass, or volume over time. You can also use it to calculate total power, given the total energy.

Application Example: Flow Integration

To totalize flow over a one-hour period, configure INTEG_TYPE = 0 to **? - periodic reset** and configure CLOCK_PER to **3600**. Figure A-11 illustrates the correct function block configuration.

Figure A-11. Function Block Diagram for Row Integration



PROPORTIONAL/ INTEGRAL/DERIVATIVE (PID) FUNCTION BLOCK

The PID function block is a powerful, flexible control algorithm that is designed to work in a variety of control strategies. The PID block is configured differently for different applications. The following examples describe the use of the PID block for closed-loop control (basic PID loop), feedforward control, cascade control with master and slave, and complex cascade control with override.

Closed Loop Control

To implement basic closed loop control, compute the error difference between the process variable (PV) and setpoint (SP) values and calculate a control output signal using a PID (Proportional Integral Derivative) function block.

The proportional control function responds immediately and directly to a change in the PV or SP. The proportional term **GAIN** applies a change in the loop output based on the current magnitude of the error multiplied by a gain value.

The integral control function reduces the process error by moving the output in the appropriate direction. The integral term **RESET** applies a correction based on the magnitude and duration of the error. Set the RESET parameter to zero for integral-only control. To reduce reset action, configure the RESET parameter to be a large value.

The derivative term **RATE** applies a correction based on the anticipated change in error. Derivative control is typically used in temperature control where large measurement lags exist.

The MODE parameter is a switch that indicates the target and actual mode of operation. Mode selection has a large impact on the operation of the PID block:

- **Manual** mode allows the operator to set the value of the loop output signal directly.
- **Automatic** mode allows the operator to select a setpoint for automatic correction of error using the **GAIN**, **RESET**, and **RATE** tuning values.
- **Cascade** and **Remote Cascade** modes use a setpoint from another block in a cascaded configuration.
- **Remote Out** mode is similar to **Manual** mode except that the block output is supplied by an external program rather than by the operator.
- **Initialization Manual** is a non-target mode used with cascade configurations while transitioning from manual operation to automatic operation.
- **Local Override** is a non-target mode that instructs the block to revert to Local Override when the tracking or fail-safe control options are activated.
- **Out of Service** mode disables the block for maintenance.

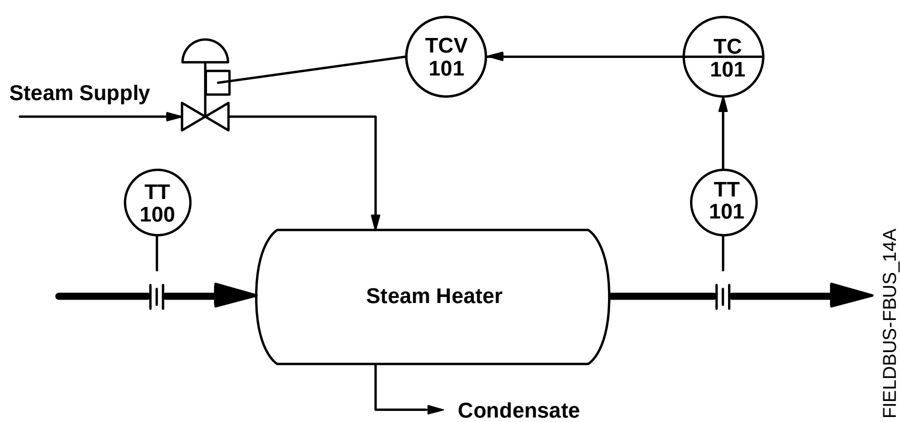
Abrupt changes in the quality of the input signal can result in unexpected loop behavior. To prevent the output from changing abruptly and upsetting the process, select the **SP-PV Track in Man** I/O option. This option automatically sets the loop to **Manual** if a *Bad* input status is detected. While in manual mode, the operator can manage control manually until a *Good* input status is reestablished.

Application Example: Basic PID Block for Steam Heater Control

Situation

A PID block is used with an AI block and an AO block to control the flow steam used to heat a process fluid in a heat exchanger. Figure A-12 illustrates the process instrumentation diagram.

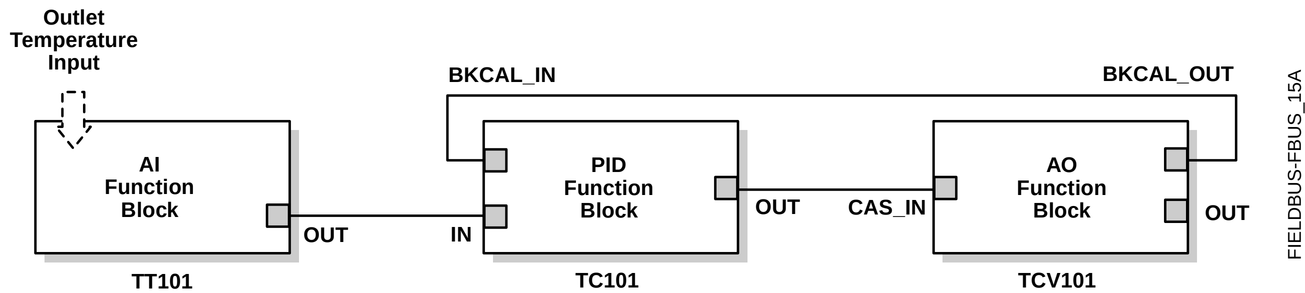
Figure A-12. PID Function Block Steam Heater Control Example



Solution

The PID loop uses TT101 as an input and provides a signal to the analog output TCV101. The BKCAL_OUT of the AO block and the BKCAL_IN of the PID block communicate the status and quality of information being passed between the blocks. The status indication shows that communications is functioning and the I/O is working properly. Figure A-13 illustrates the correct function block configuration.

Figure A-13. PID Function Block Diagram for Steam Heater Control Example



Application Example: Feedforward Control

Situation

In the previous example, control problems can arise because of a time delay caused by thermal inertia between the two flow streams (TT100 and TT101). Variations in the inlet temperature (TT100) take an excessive amount of time to be sensed in the outlet (TT101). This delay causes the product to be out of the desired temperature range.

Solution

Feedforward control is added to improve the response time of the basic PID control. The temperature of the inlet process fluid (TT100) is input to an AI function block and is connected to the FF_VAL connector on the PID block. Feedforward control is then enabled (FF_ENABLE), the feedforward value is scaled (FF_SCALE), and a gain (FF_GAIN) is determined. Figure A-14 illustrates the process instrumentation diagram, and Figure A-15 illustrates the correct function block configuration.

Figure A-14. PID Function Block Feedforward Control Example

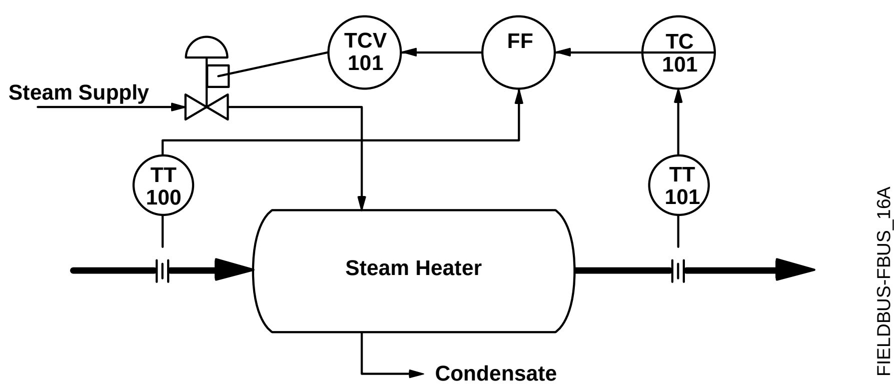
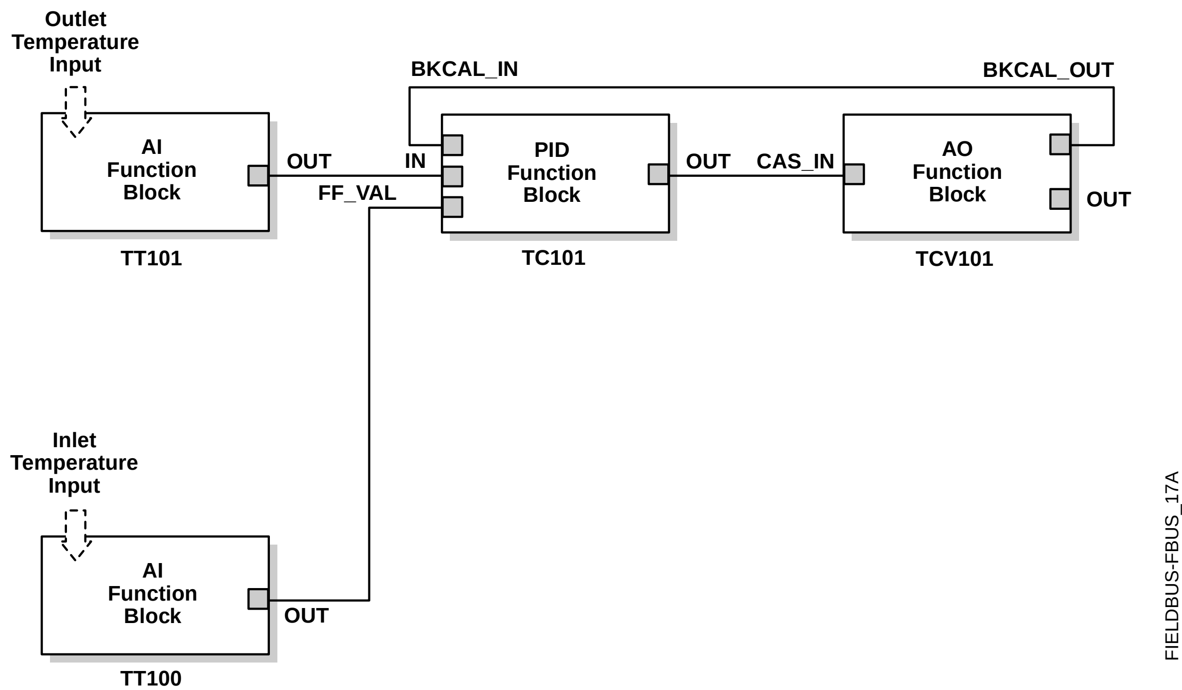


Figure A-15. Function Block Diagram for Feedforward Control

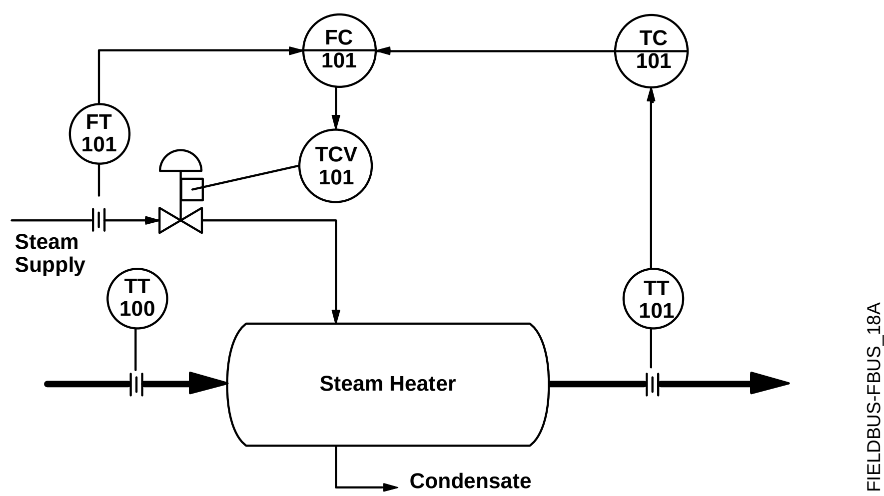


Application Example: Cascade Control with Master and Slave Loops

Situation

A slave loop is added to a basic PID control configuration to measure and control steam flow to the steam heater. Variations in the steam pressure cause the temperature in the heat exchanger to change. The temperature variation will later be sensed by TT101. The temperature controller will modify the valve position to compensate for the steam pressure change. The process is slow and causes variations in the product temperature. Figure A-16 illustrates the process instrumentation diagram.

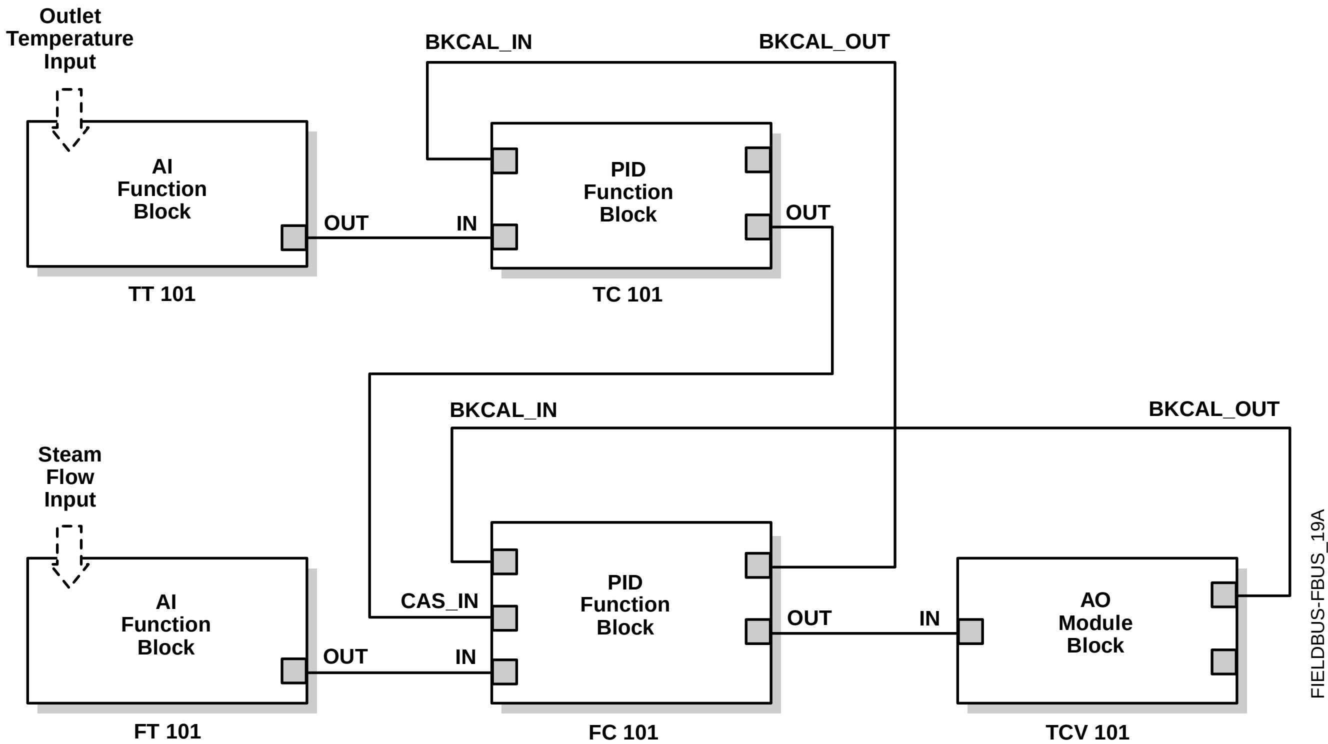
Figure A-16. PID Function Block Cascade Control Example



Solution

If the flow is controlled, steam pressure variations will be compensated before they significantly affect the heat exchanger temperature. The output from the master temperature loop is used as the setpoint for the slave steam flow loop. The BKCAL_IN and BKCAL_OUT connections on the PID blocks are used to prevent controller windup on the master loop when the slave loop is in Manual or Automatic mode, or it has reached an output constraint. Figure A-17 illustrates the correct function block configuration.

Figure A-17. PID Function Block Diagram for Cascade Control Example



Application Example: Cascade Control with Override

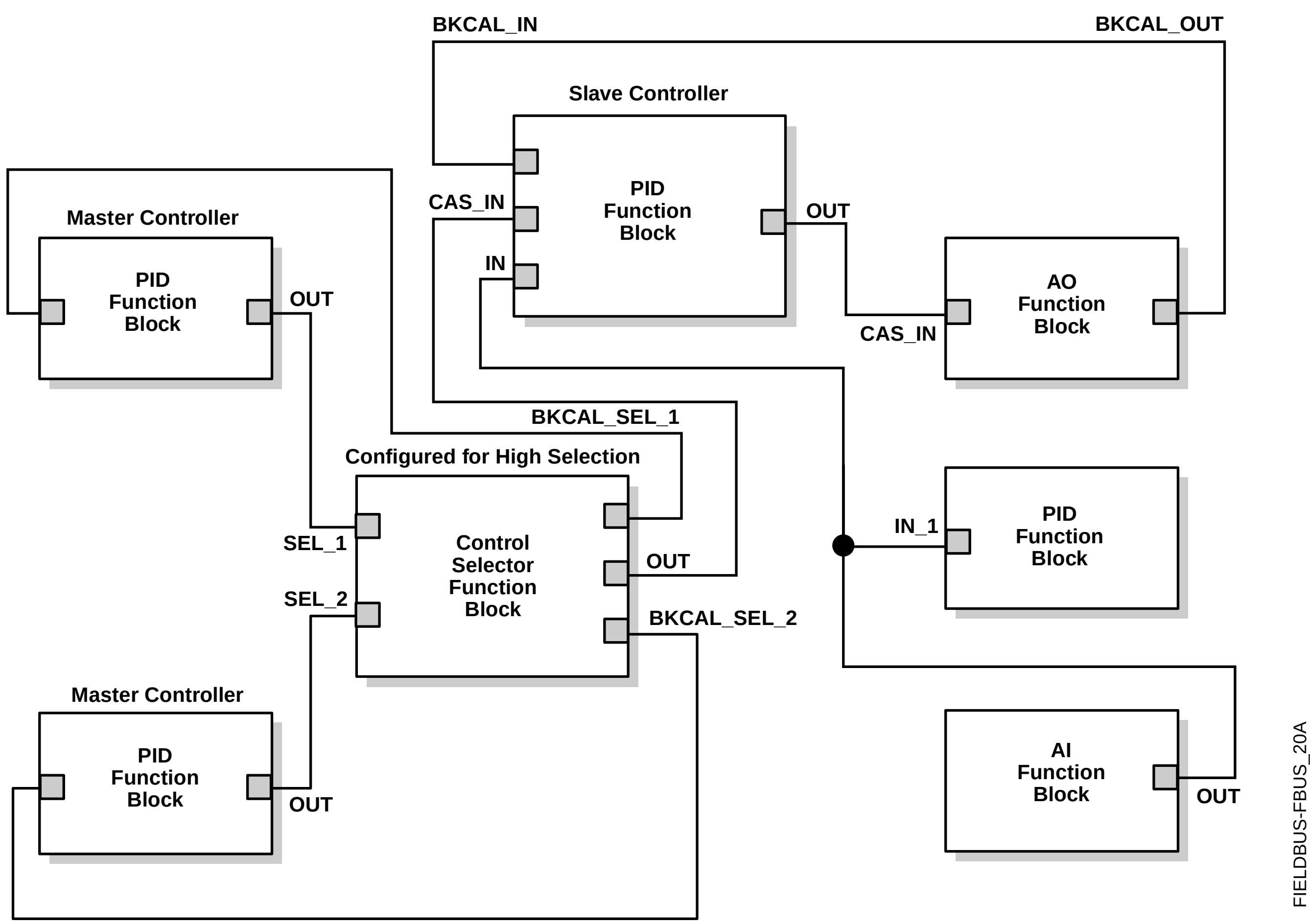
You can use the PID function block with other function blocks for complex control strategies. Figure A-18 illustrates the function block diagram for cascade control with override.

When configured for cascade control with override, if one of the PID function blocks connected to the selector inputs is deselected, that PID block filters the integral value to the selected value (the value at its BKCAL_IN). The selected PID block behaves normally and the deselected controller never winds up. At steady state, the deselected PID block offsets its OUT value from the selected value by the proportional term. When the selected block becomes output-limited, it prevents the integral term from winding further into the limited region.

When the cascade between the slave PID block and the Control Selector block is open, the open cascade status is passed to the Control Selector block and through to the PID blocks supplying input to it. The Control Selector block and the upstream (master) PID blocks have an actual mode of **IMan**.

If the instrument connected to the AI block fails, you can place the AI block in **Manual** mode and set the output to some nominal value for use in the Integrator function block. In this case, IN at the slave PID block is constant and prevents the integral term from increasing or decreasing.

Figure A-18. Function Block Diagram for Cascade Control with Override



SIGNAL CHARACTERIZER (SGCR) FUNCTION BLOCK

You can use the SGCR function block as a curve fitting function. For example, you can scale a 4 to 20 mA input signal to a 0 to 100% output value using the block. You can also use the block to convert measurements from a split-range or other nonlinear device or from a dual-temperature control device used for both heating and cooling.

You can also use the SGCR function block to alter the relationship of a PID function block output to valve position, and thereby gain more linear control over a critical region.

Index

Rosemount Inc.
8200 Market Boulevard
Chanhassen, MN 55317 USA
Tel 1-800-999-9307
Fax (952) 949-7001

http://www.rosemount.com

00809-0100-4783, Rev. BA 5/00

ROSEMOUNT®

FISHER-ROSEMOUNT™