

**Fieldbus Overview**



# Contents

Introduction .....	1
Device Description Files .....	1
Device Identification Assignment .....	1
Fieldbus Components .....	1
Links .....	1
Devices .....	2
Blocks and Parameters .....	3
Linkages .....	3
Loops .....	4
Schedules .....	5
Physical Layer Standards .....	6
Communication Stack .....	6
Data Link Layer (DLL) .....	6
Fieldbus Access Sublayer (FAS) .....	7
Fieldbus Message Specification (FMS) .....	7
User Application Blocks .....	7
Resource Block .....	8
Explanation of Resource Block Parameters .....	9
Transducer Block .....	12
Analog Input (AI) Function Block .....	12
Description .....	12
Explanation of AI Block Parameters .....	13
PID Block .....	17
Description .....	18
Explanation of PID Block Parameters .....	18



# Introduction

This document contains general information about the FOUNDATION fieldbus concept, structure, and operation, which is applicable to all types of fieldbus transmitters and devices.

The FOUNDATION fieldbus is an all digital, serial, two-way communication system that interconnects field devices, such as transmitters, actuators, and controllers and operates at 31.25 kbits/second. It is a Local Area Network (LAN) with built-in capability to distribute control application across the network. This communication technology consists of the Physical Layer, the Communication Stack, and the User Application Blocks.

FOUNDATION fieldbus, created by the Fieldbus Foundation, is designed for reliable, distributed control in industrial process environments. Devices connected to a FOUNDATION fieldbus control a process by exchanging data.

## Device Description Files

Device Description (DD) technology and function block parameters are used to achieve interoperability of fieldbus devices. The DD provides the information that the host requires to interpret the data in the device, including calibration and diagnostics. Each device manufacturer provides Device Description (DD) files for the device that describe the function and transducer blocks contained in a device. A DD file includes both symbolic and functional information that can be used to configure a device.

## Device Identification Assignment

Each device must have a Device ID which comprises at least a tag and address. The manufacturer automatically assigns each device a device tag and address number. The tag and address can be reconfigured but each tag or address must be unique on a wiring segment. Using the physical device tag, a special “set address” is sent to the device, thus forcing the device to move to a network address. A “Find Tag Service” initiates a tag search for the convenience of host systems and maintenance devices.

## Fieldbus Components

A fieldbus network includes six type of components: **links**, **devices**, **blocks** and **parameters**, **linkages**, **loops**, and **schedules**.

### Links

A fieldbus network consists of many devices connected by a serial bus, called **links**.

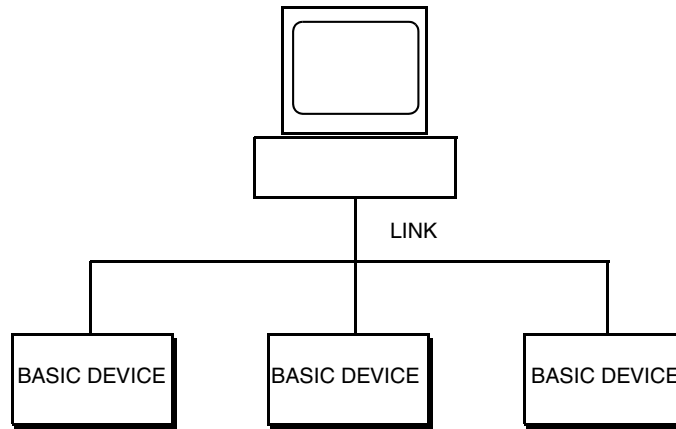


Figure 1. Fieldbus Link

## Devices

A fieldbus network can contain three types of devices: **link master**, **basic devices**, and **bridges**.

- ◆ **Link Master** — A link master controls communication traffic on a **link**. It prevents communication collisions from occurring when multiple devices try to communicate at the same time. A link master can be a system or any device, such as a transmitter, or valve. You can have more than one link master per link, but only one **link master** can be assigned as **Link Active Scheduler (LAS)** at any time.
- ◆ **Basic Device** — A basic device is any fieldbus-compatible device. A **basic device** cannot become the **LAS**.
- ◆ **Bridge** — A bridge is a device that connects two or more **links** or bus segments.

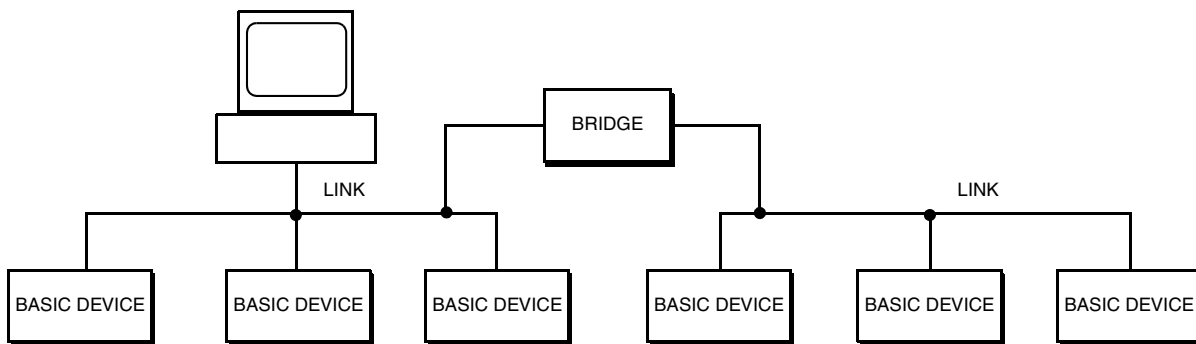


Figure 2. Fieldbus Bridge

Each device on a network is given a unique character string name, or **tag**. This is a user-configured attribute of the device that typically describes the application. The **device ID**, which includes a unique serial number for the device, is another identifier of the device. The **device ID** is assigned by the manufacturer and cannot be user-configured.

## Blocks and Parameters

A **device** normally has a set of associated functions it can perform, which are represented in the device as **function blocks**. Function blocks contain algorithms and parameters. The fieldbus specification defines ten different types of function block:

- ◆ Analog Input (AI)
- ◆ Analog Output (AO)
- ◆ Discrete Input (DI)
- ◆ Discrete Output (DO)
- ◆ Proportional Integral Derivative (PID)
- ◆ Proportional Derivative (PD)
- ◆ Bias Gain (BG)
- ◆ Ratio Control (RC)
- ◆ Control Selector (CS)
- ◆ Manual Loader (ML)

You can change the actions of a block by editing the parameters. Three types of function block parameters are used:

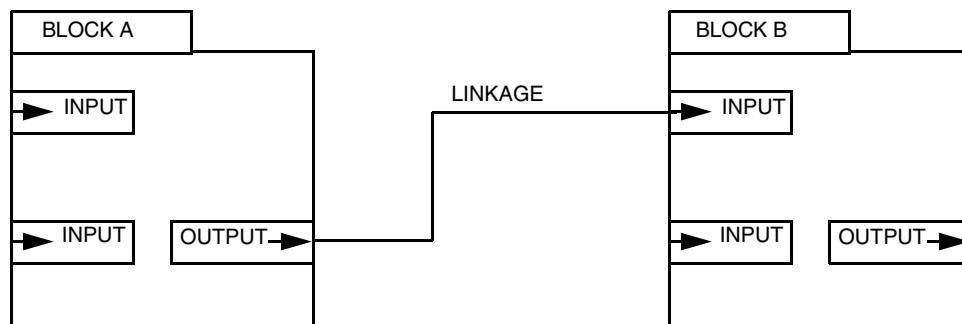
- ◆ **Input** parameters receive data from another block.
- ◆ **Output** parameters send data to another block.
- ◆ **Contained** parameters do not send or receive data.

Input, output, and contained parameters can also be independently classified as **alarm**, **trend**, or **tuning** parameters.

## Linkages

The function blocks that control a process are linked, or connected, by configuration objects inside the devices. The linkages allow you to send data from one block to another. A **linkage** differs from a **link** in that a **linkage** is a software connection between two blocks and a **link** is a physical wire connection between two or more devices.

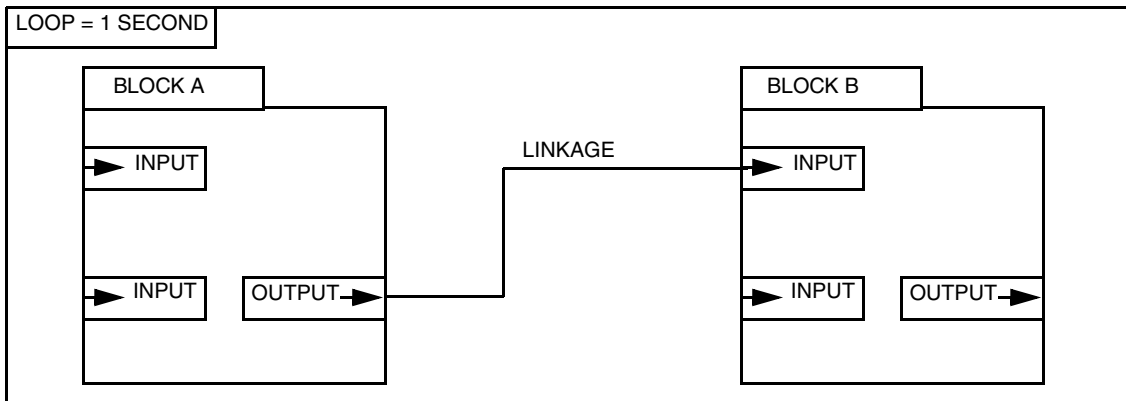
A block parameter written to the bus is called a **published parameter**; one that receives data is called a **subscribed parameter**. An example of a linkage between blocks is shown in Figure 3.



*Figure 3. Linkage between Function Blocks*

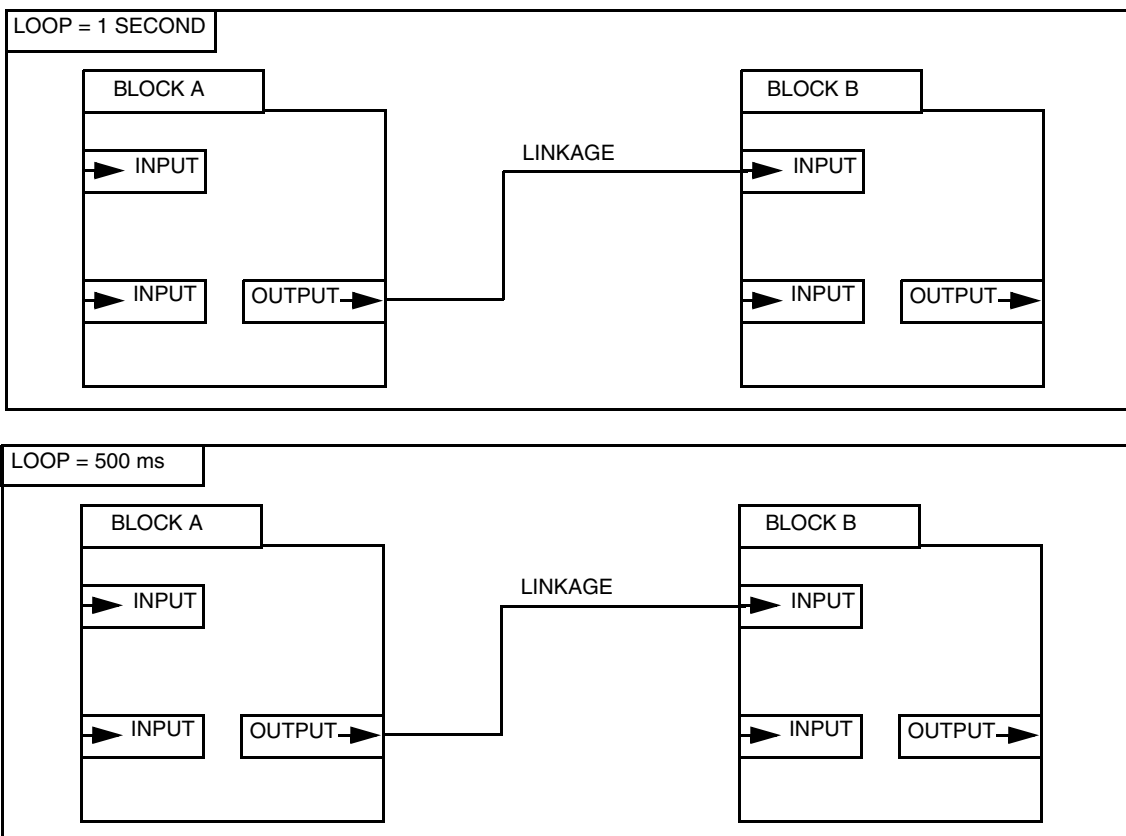
# Loops

A **loop** is a group of function blocks connected by linkages that execute at a fixed, configurable rate. Each block and each linkage execute at the configured rate, as shown in Figure 4.



*Figure 4. Control Loop*

You can have many **loops** on a **link** that execute at different rates, as illustrated in Figure 5.



*Figure 5. Multiple Loops Running at Different Rates*



Even if the **loops** run at different rates, you can send data between them through one or more **linkages**, as shown in Figure 6. All **loops** on a **link** run within one macrocycle, which is the least common multiple of all **loop** times on a specified **link**. In the example shown, the macrocycle time is one second.

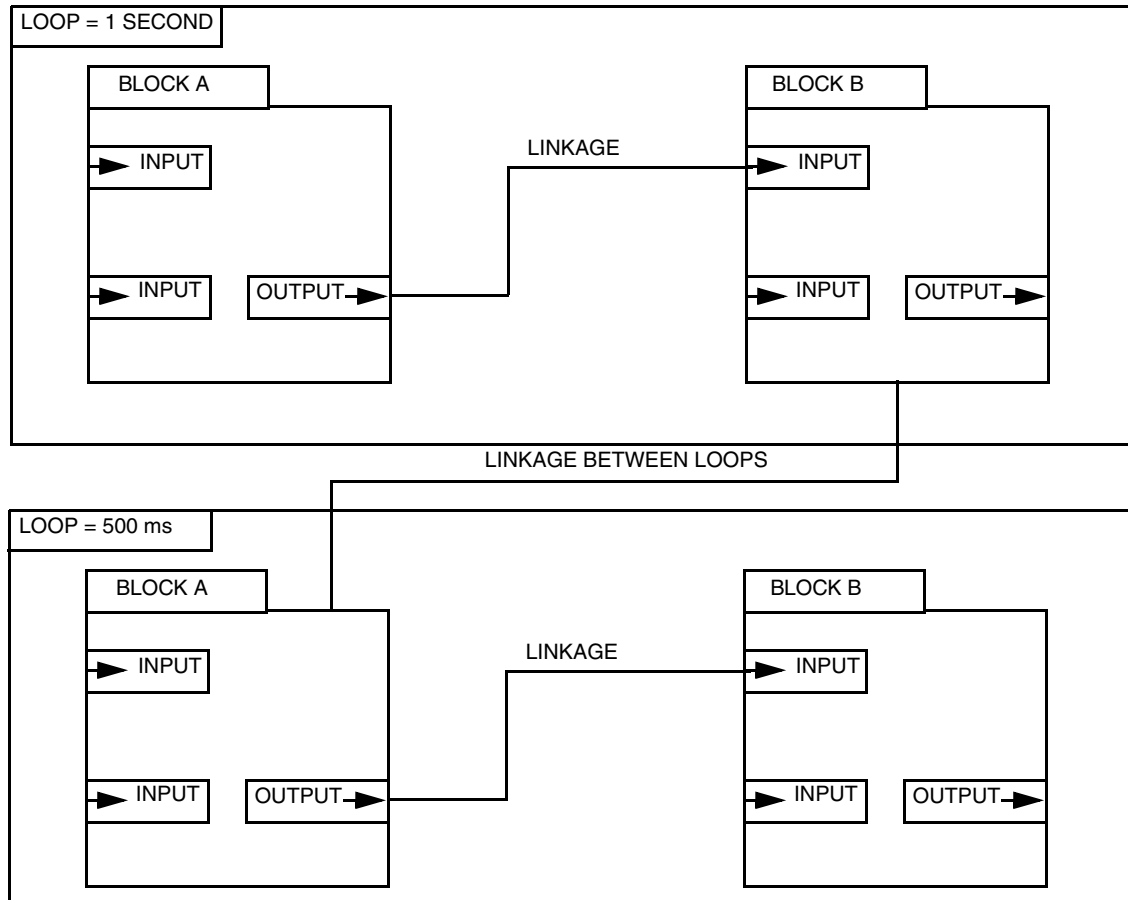


Figure 6. Linkage between Two Loops Running at Different Rates

## Schedules

Data is transferred between a set of devices on a link at the same time during each loop macrocycle. The **schedule** is divided into two parts: a **function block schedule** that determines when a block executes, and a **publishing schedule** that determines when data parameters are published over the fieldbus.

The **function block** is downloaded to the device that contains the function blocks; the **publishing schedule** is downloaded to devices that have **link master capability**. The link master that is assigned the task of running the publishing schedule (and the process) becomes the **Link Active Scheduler (LAS)**.

The reason you download the schedule to many devices with link master capability is to provide backup scheduling control if and when the main LAS fails. If the main LAS fails, another device with link master capability automatically assumes control over all devices on the bus. A typical relationship is illustrated in Figure 7.

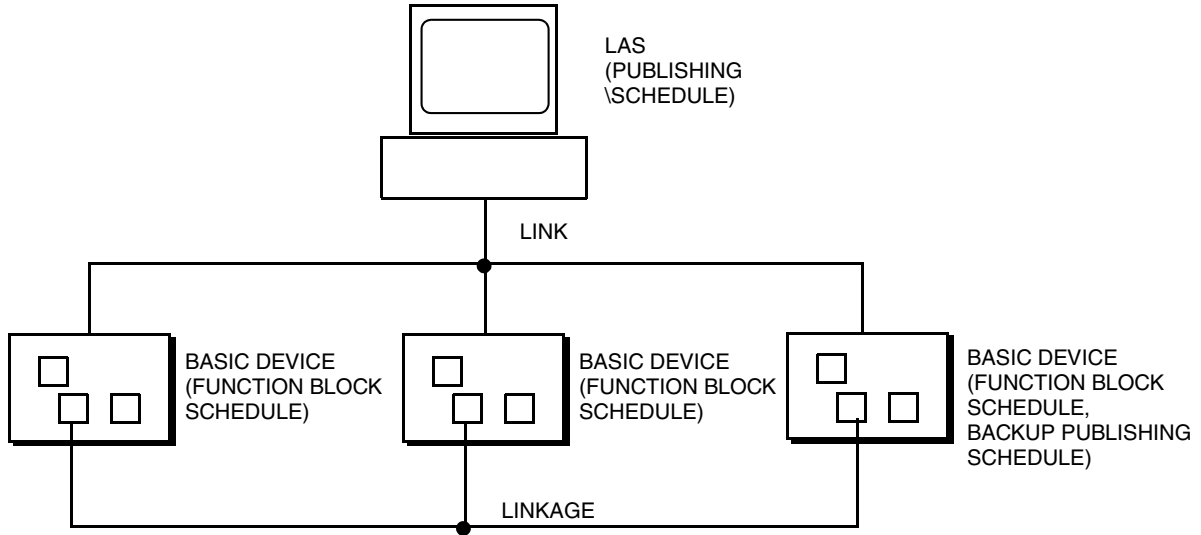


Figure 7. Relationships between Devices with Publishing and Function Block Schedules

## Physical Layer Standards

The Fieldbus Foundation uses IEC 1158-2 and ISA S50.02 Physical Layer Standards as references to define, as a minimum, the signaling and electrical properties of a fieldbus device’s Physical Layer interface. The properties of these transmitters are:

- ◆ Polarity Independent
- ◆ Open Circuit Level of 35 V dc maximum
- ◆ Intrinsically Safe Parameters per table below.

Parameter	Recommended Value
Approval Voltage	24 V dc
Approval Current	250 mA
Input Power	1.2 W
Residual Capacitance	< 5 nF
Residual Inductance	< 20 μH

## Communication Stack

The Communication Stack comprises a Data Link Layer (DLL), a Fieldbus Access Sublayer (FAS), and a Fieldbus Message Specification (FMS).

### Data Link Layer (DLL)

The Data Link Layer (DLL) provides:

- ◆ Scheduled Communication – As authorized by the Link Active Scheduler (LAS)
- ◆ Unscheduled Communication – This allows all devices on the fieldbus to send “unscheduled” messages between the transmission of scheduled messages.

## Fieldbus Access Sublayer (FAS)

The Fieldbus Access Sublayer (FAS) provides:

- ◆ Client/Server VCR (Virtual Communication Relationship) – This is used for queued, unscheduled, user-initiated, one-to-one communication between the devices on the fieldbus; typically user-initiated requests, such as setpoint changes, alarm acknowledgments, and device uploads/downloads.
- ◆ Report Distribution VCR – Typically allows fieldbus devices to send alarm notifications to the operator console.
- ◆ Publisher/Subscriber VCR – Allows fieldbus device to publish data, and send function block I/O data, such as transmitter process variable (PV) and primary output data to the operator console.

## Fieldbus Message Specification (FMS)

FMS services allow user applications to send messages to each other across the fieldbus. The FMS describes communication services, message formats, and protocol required for the user application.

## User Application Blocks

These are software blocks that represent different types of user application functions. The types of blocks used with Invensys Foxboro devices are **Resource Blocks**, **Transducer Blocks**, **Function Blocks**, and sometimes **PID Blocks**. These blocks reside in the transmitter database.

The **mode** parameter in each of these blocks is used to control major states of the block. **OOS mode** (out of service) stops all block execution. When in **OOS mode**, the **actual mode** of the function blocks is changed to OOS but the **target mode** is not changed. The display **IMan** shows that the resource is initialized or receiving a software download.

Certain parameters are unique to a particular block while others are common to all of these blocks. Those common are described immediately below.

### **ACK\_OPTION**

This read/write parameter enables the user to select which alarms associated with the block are automatically acknowledged.

### **ALARM\_SUM**

The **current**, **unacknowledged**, and **unreported** read only subparameters summarize the status of up to 16 alarms. You can use the disabled read/write subparameter to enable or disable each alarm.

### **ALERT\_KEY**

This read/write parameter is an optional user assigned identification number of the plant unit used in the host for sorting alarms and so forth.

### **BLOCK\_ALM**

This **read only parameter** 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 sets the Active status in the Status attribute. As soon as the Unreported status is

cleared by the alert reporting task, another block alert can be reported without clearing the Active status, if the subcode has changed.

#### **BLOCK\_ERR**

This read only parameter reflects the error status associated with the hardware or software components associated with the block. It is a bit string, so that multiple errors can be shown.

#### **BLOCK\_TAG**

This read/write parameter uniquely identifies a block. You can edit block tags, but the tags must be unique to each block on the fieldbus network.

#### **MODE\_BLK**

This read/write parameter establishes the target, actual, permitted, and normal modes for the block.

- ◆ Target: The mode requested by the operator.
- ◆ Actual: The current mode of the block.
- ◆ Permitted: The modes allowed for the block.
- ◆ Normal: The mode of the block during normal operation.

#### **ST\_REV**

This read only parameter shows the revision level of the static data associated with the function block. The revision value is incremented each time a static parameter value in the block is changed.

#### **STRATEGY**

This read/write parameter is optionally used to identify grouping of blocks. This data is not checked or processed by the block.

#### **TAG\_DESC**

This read/write parameter is an optional user description of the block application.

#### **TARGET\_ERROR**

This read only parameter shows a description of the error and the number of the affected parameter.

#### **UPDATE\_EVENT**

This read only parameter is generated by any change to the static data of the block.

## Resource Block

The Resource Block identifies the device. Thus, it provides manufacturer ID, device type, DD resource, device revision level, and DD revision level. This is required to identify and locate the DD so that Device Description Hosting Services can select the correct DD for use with the resource. It is also used to define hardware specific characteristics of the Function Block applications. It also indicates how much resource (memory) is available and controls the overall device. All data is modelled within a controlled space so that no outside inputs into this block are required.

## Explanation of Resource Block Parameters

### CLR\_FSTATE

See **FAULT\_STATE**, **SET\_FSTATE**, and **CLR\_FSTATE**


### CONFIRM\_TIME

This read/write parameter is the time for the resource to wait for confirmation of receipt of a report before trying again. If **CONFIRM\_TIME** = 0, the device does not retry.

### CYCLE\_TYPE and CYCLE\_SEL

The **CYCLE\_TYPE** read only parameter is a bit string that defines the types of cycles that this resource can do. The read/write parameter **CYCLE\_SEL** allows you to choose one or more of them. If a bit in **CYCLE\_SEL** is selected that is not in **CYCLE\_TYPE**, the result is a block alarm for a configuration error.

---

—  **CAUTION** —  
 Changing this parameter may be fatal to communication.

---

### DD\_RESOURCE

This read only parameter shows a string identifying the tag of the resource that contains the Device Description.

### DD\_REV

This read only parameter shows the revision number associated with the resource that contains the Device Description.

### DEV\_REV

This read only parameter shows the revision number associated with the resource.

### DEV\_TYPE

This read only parameter shows the model number associated with the resource.

### FAULT\_STATE, SET\_FSTATE, and CLR\_FSTATE

The **FAULT\_STATE** read only parameter shows the fault state as **active** or **clear**. The fault state can be manually initiated by configuring the read/write **SET\_FSTATE** parameter to **set**. This causes the **FAULT\_STATE** parameter to indicate **active** and causes all output function blocks in the resource to go immediately to the condition chosen by the fault state Type I/O option. It can be cleared by setting the read/write **CLR\_FSTATE** parameter to **clear** if the condition has cleared.

### FEATURES and FEATURE\_SEL

These parameters determine optional behavior of the resource. The **FEATURES** read only parameter defines the available features. The **FEATURE\_SEL** read/write parameter is used to turn on an available feature by configuration. If a bit is set in **FEATURE\_SEL** that is not in **FEATURES**, the result is a block alarm for a configuration error. The available features in **FEATURES** for Foxboro products may include:

- ◆ **Reports:** set to support alert reports. If not set, the master must poll for alerts
- ◆ **Unicode:** allows use of Unicode octet strings
- ◆ **Soft W Lock:** enables write protection via a software selection
- ◆ **Hard W Lock:** enables write protection via a hardware jumper

### **FREE\_SPACE**

The **FREE\_SPACE** read only parameter shows the percentage of configuration memory that is still available.

### **FREE\_TIME**

The **FREE\_TIME** read only parameter shows the approximate percentage of time that the resource has left for processing new function blocks, should they be configured.

### **GRANT\_DENY**

This read/write parameter is used to grant or deny access of host computers and local control panels to operating, tuning, and alarm parameters of the block.

### **HARD\_TYPES**

This read only parameter is a bit string that indicates the types of hardware that is available to this resource. If an I/O block is configured that requires a type of hardware that is not available, the result is a block alarm for a configuration error.

### **ITK\_VER**

This read only parameter shows the major revision number of the interoperability test case used in certifying the device as interoperable. the format and range of the version number is defined and controlled by the Fieldbus Foundation. The value of this parameter is zero if the device has not been registered as interoperable by the Fieldbus Foundation.

### **LIM\_NOTIFY**

See **MAX\_NOTIFY** and **LIM\_NOTIFY**.

### **MANUFAC\_ID**

This read only parameter is used by an interface device to locate the DD file for the resource.

### **MAX\_NOTIFY** and **LIM\_NOTIFY**

The **MAX\_NOTIFY** read only parameter value is the maximum number of alert reports that this resource can have sent without getting a confirmation, corresponding to the amount of buffer space available for alert messages. You can control alert flooding, by adjusting the **LIM\_NOTIFY** read/write parameter value to a number lower than the maximum. If **LIM\_NOTIFY** is set to zero, no alerts are reported.

### **MEMORY\_SIZE**

This read only parameter shows the available configuration memory in the empty resource in kilobytes. It should be checked before attempting a download.

### **MIN\_CYCLE\_T**

This read only parameter shows the manufacturer specified minimum time to execute a cycle. It puts a lower limit on the scheduling of the resource.

### **NV\_CYCLE\_T**

This read only parameter shows the interval of writing copies of nonvolatile parameters to nonvolatile memory.

## RESTART

This read/write parameter allows several degrees of manual restart to be initiated. They are:

- ◆ **Run:** Passive state of the parameter
- ◆ **Restart Resource:** Intended to clear up problems (for example, the memory management resource)
- ◆ **Restart with Defaults:** Intended to wipe configuration memory. It works like a factory initialization.
- ◆ **Restart Processor:** Provides a way to hit the reset button on the processor associated with the resource. This parameter does not appear in a view because it returns to 1 shortly after being written.

## RS\_STATE

This read only parameter contains the operational state of the Function Block application for the resource containing this resource block.

## SET\_FSTATE

See **FAULT\_STATE**, **SET\_FSTATE**, and **CLR\_FSTATE**

## SHED\_RCAS and SHED\_ROUT

These read/write parameters set the time limit for write attempts to RCas / ROut locations. Shedding from RCAS / ROUT does not happen when **SHED\_RCAS** / **SHED\_ROUT** is set to zero.

## WRITE\_LOCK

This read/write parameter can be set to **Locked** or **Not Locked**. Setting this parameter to **Locked**, prevents any external change to the static or nonvolatile data base in the Function Block application of the resource. Block connections and calculation results proceed normally, but the configuration is locked.

To use a hard write lock (on devices that support hard write lock), put the write-lock jumper in the correct position and select the **Hard W Lock** option in **FEATURE\_SEL**. With a hard write lock, the **WRITE\_LOCK** parameter is only an indicator of the write locking. Soft write lock has no impact on the device when hard write lock is enabled in **FEATURE\_SEL**. To deactivate the write lock, the jumper position must be changed.

When hard write lock is not supported or this feature has not been enabled in **FEATURE\_SEL**, then a soft write lock may be set and cleared by writing to the **WRITE\_LOCK** parameter if the **Soft W Lock** option has been enabled in **FEATURE\_SEL**.

Clearing **WRITE\_LOCK** generates the discrete alert **WRITE\_ALM** at the **WRITE\_PRI** priority. Setting **WRITE\_LOCK** clears this alert, if it exists.

## WRITE\_PRI

This read/write parameter enables you to set the priority of the alarm generated by clearing the write lock.

## Transducer Block

The **Transducer Block** decouples the **AI Function Block** from local I/O functions required to read sensors, and to command output devices. It contains all of the manufacturer-specific parameters that define how the device functions. While this block contains many parameters that are specific to the device, it always includes the primary range values and engineering units (EGU), and the sensor range limits.

## Analog Input (AI) Function Block

**Analog Input (AI)** function blocks contain all configurable parameters needed to define the input data for use with other function blocks. It takes the manufacturer input data, selected by channel number, and makes it available to other function blocks at its output.

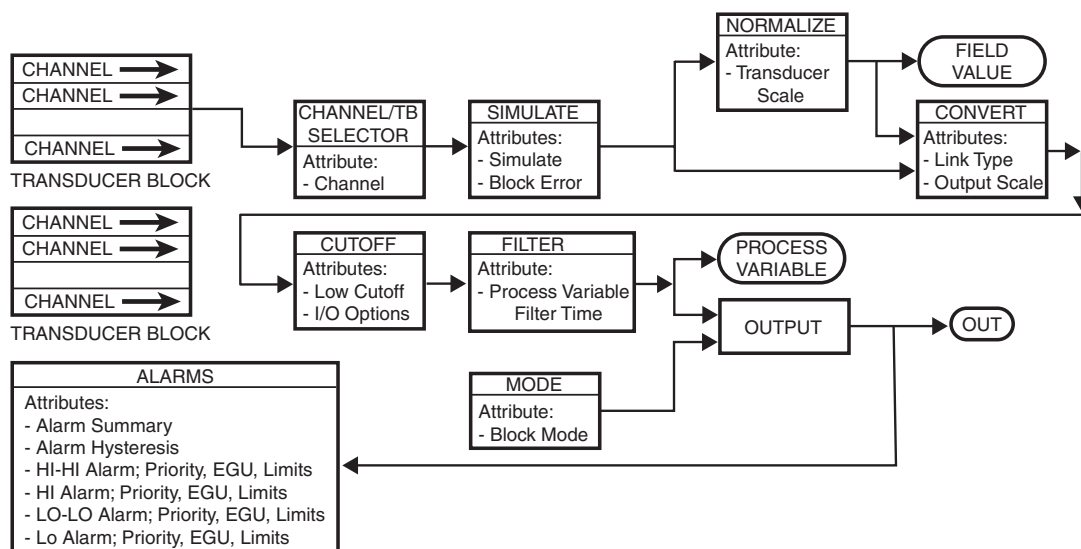


Figure 8. Analog Input (AI) Functional Block Diagram

### Description

Transducer scaling (**XD\_SCALE**) is applied to the value from the channel to produce the **FIELD\_VAL** in percent. The **XD\_SCALE** units code must match the channel units code or the an error message is generated. The **OUT\_SCALE** is normally the same as the transducer, but if **L\_TYPE** is set to **Indirect** or **Ind Sqr Root**, **OUT\_SCALE** determines the conversion from **FIELD\_VAL** to the output. **PV** and **OUT** always have identical scaling.

**OUT\_SCALE** provides scaling for **PV**. The **PV** is always the value that the block places in **OUT** if the mode is **Auto**. If **Man** is allowed, you can write a value to the output. The Status prevents any attempt at closed loop control using the **Man** value, by setting the Limit value to Constant.

The **LOW\_CUT** parameter has a corresponding “Low cutoff” option in the **IO\_OPTS** bit string. If the option is true, any calculated output below the low cutoff value is changed to zero. This is only useful for zero-based measurement devices, such as flow. The **PV** filter, whose time constant is **PV\_FTIME**, is applied to the **PV** and not the **FIELD\_VAL**.



Equations:

$FIELD\_VAL = 100 * (\text{channel value} - EU@0\%) / (EU@100\% - EU@0\%) [XD\_SCALE]$

Direct:  $PV = \text{channel value}$

Indirect:  $PV = (FIELD\_VAL/100) * (EU@100\% - EU@0\%) + EU@0\% [OUT\_SCALE]$

Ind Sqr Root:  $PV = \text{sqrt}(FIELD\_VAL/100) * (EU@100\% - EU@0\%) + EU@0\% [OUT\_SCALE]$

All input and output parameters contain a value component and a status component. There are four quality states, described below.

- ◆ **Good (cascade):** The value may be used in control.
- ◆ **Good (noncascade):** The quality of the value is good, but the value may have an associated alarm.
- ◆ **Uncertain:** The quality of the value is less than normal, but the value may be useful.
- ◆ **Bad:** The value is not useful.

In addition, each of the status qualities has a number of sub-status conditions. Examples include Sensor Failure, Configuration Error, Local Override, and Active Block Alarm. Finally, there are four limit states: Not Limited, Limited High, Limited Low, and Value is Constant.

The **STATUS\_OPTS** parameter provides instructions for how the block should respond to certain status conditions. The status conditions are:

- ◆ IFS if Bad IN
- ◆ IFS if Bad CAS\_IN
- ◆ Use Uncertain as Good
- ◆ Propagate Fail Forward
- ◆ Propagate Fail Backward
- ◆ Target to Manual if Bad In
- ◆ Uncertain if Limited
- ◆ Bad if Limited
- ◆ Uncertain if Man Mode
- ◆ Target to Next Permitted Mode if Bad **CAS\_IN**

## *Explanation of AI Block Parameters*

### **ALARM\_HYS**

This read/write parameter enables you to set the amount in % by which the PV must return within the alarm limits before the alarm condition clears. Alarm Hysteresis is expressed as a percent of the PV span.

### **CHANNEL**

This read/write parameter enables you to set the number of the logical hardware channel that is connected to the AI function block. This information defines the transducer to be used going from the physical world.

## **FIELD\_VAL**

This read only parameter shows the raw value of the field device in percent of the PV range, with a status reflecting the transducer condition, before signal characterization (**L\_TYPE**) or filtering (**PV\_FTME**). Transducer scaling (**XD\_SCALE**) is applied to the value from the channel to produce the **FIELD\_VAL** in percent.

## **GRANT\_DENY**

This read/write parameter is used to grant or deny access of host computers and local control panels to operating, tuning, and alarm parameters of the block.

## **HI\_ALM** and **HI\_HI\_ALM**

This read only parameter shows the status and time stamp associated with the high and high-high alarms respectively.

## **HI\_LIM** and **HI\_HI\_LIM**

This read/write parameter enables you to set the high and high-high alarm setpoints respectively in engineering units.

## **HI\_PRI** and **HI\_HI\_PRI**

This read/write parameter enables you to set the priority of the high and high-high alarms respectively.

## **IO\_OPTS**

This read/write parameter enables you to select options to alter input and output block processing. The options are:

- ◆ **Invert:** Indicates whether the discrete input value should be logically inverted before it is stored in the process variable. A discrete value of zero is considered to be a logical (0) and a nonzero discrete value is considered to be logical (1). For example, if Invert is selected, the logical NOT of a nonzero field value results in a zero discrete output, the logical NOT of a zero field value results in a discrete output value of one.
- ◆ **SP-PV Track in Man:** Permits the setpoint to track the process variable when the Target mode of the block is Man.
- ◆ **SP-PV Track in LO:** Permits the setpoint to track the process variable when the actual mode of the block is LO.
- ◆ **SP Track Retained Target:** Permits the setpoint to track the **RCAS** or **CAS** parameter based on the retained Target mode when the actual mode of the block is LO or Man. When SP-PV track options are enabled, then SP Track retained target has precedence in the selection of the value to track when the actual mode is Man and LO.
- ◆ **Increase to close:** Indicates whether the output value should be inverted before it is communicated to the I/O channel.
- ◆ **Fault State to Value:** Sets the output action when a fault occurs. (0=freeze, 1=go to preset value)
- ◆ **Use Fault State Value on Restart:** Uses the value of **FSTATE\_VAL** if the device is restarted. This latches an output block into the manual mode.
- ◆ **Target to Man if Fault State Activated:** Sets the Target mode to Man, thus losing the original target, if Fault State is activated.

- ◆ Use PV for **BKCAL\_OUT**: The **BKCAL\_OUT** value is normally the working SP. This option changes it to the PV.
- ◆ Low Cutoff: The AI Low Cutoff algorithm is enabled.

### **L\_TYPE**

This read/write parameter enables you to specify if the values passed by the transducer block to the AI block may be used directly (**Direct**) or if the value is in different units and must be converted linearly (**Indirect**), or with square root (**Ind Sqr Root**), using the input range defined by the transducer and the associated output range.

### **LO\_ALM** and **LO\_LO\_ALM**

This read only parameter shows the status and time stamp associated with the low and low-low alarms respectively.

### **LO\_LIM** and **LO\_LO\_LIM**

This read/write parameter enables you to set the low and low-low alarm setpoints respectively in engineering units.

### **LO\_PRI** and **LO\_LO\_PRI**

This read/write parameter enables you to set the priority of the low and low-low alarms respectively.

### **LOW\_CUT**

This read/write parameter enables you to set a low limit used in square root processing. A value of zero percent of scale is used in block processing if the transducer value falls below this limit, in % of scale. This feature is used to eliminate noise near zero for a flow sensor.

### **OUT**

This parameter shows primary analog value and status calculated by this function block or entered by the user.

### **OUT\_SCALE**

This read/write parameter enables you to set the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the **OUT** parameter and parameters which have the same scaling as **OUT**.

### **PV**

This read only parameter shows the value and status of the primary input value or calculated value on which the algorithm bases the output calculation. The units of the input must match the process variable units.

### **PV\_FTIME**

This read/write parameter enables you to set the time constant of a single exponential filter for the PV, in seconds.

### **SIMULATE**

This read/write parameter acts as a switch at the interface between the I/O function block and the associated transducer block or hardware channel.

For inputs, the transducer value and status come from the transducer block or input channel, and contain what will be sent to the input block if the switch is disabled. The simulate value and status are presented to the input block when the switch is enabled, and the transducer block or

input channel is ignored. The status can be used to simulate transducer faults. The transducer value and status is always written with the transducer data at each evaluation of the input function block.

For outputs, the simulate value and status become the **READBACK** value and status when the simulate switch is enabled, and the transducer block is ignored. The status can be used to simulate transducer faults. The transducer attribute value and status reflect the transducer readback value and status when simulation is enabled and the transducer maintains the last output and ignores the **OUT** of the output block.

It is necessary to show that a block has a simulated value, without touching the status or parameters that may be linked elsewhere. The block alarm parameter provides this visibility through the simulate active attribute.

When disabled, the simulation parameter should take on the value and status it would supply if enabled. It is not possible to provide a bumpless transfer from enabled to disabled. Therefore, it is not useful to enable simulation in an operating plant. The simulate is always initialize to disabled and is stored in dynamic memory.

### **STATUS\_OPTS**

This read/write parameter enables you to select options in the block processing of status. They are:

- ◆ **IFS if Bad IN:** Sets Initiate Fault State status in the **OUT** parameter if the status of the **IN** parameter is **Bad**.
- ◆ **IFS if Bad CAS\_IN:** Sets Initiate Fault State status in the **OUT** parameter if the status of the **CAS\_IN** parameter is **Bad**.
- ◆ **Use Uncertain as Good:** If the status of the **IN** parameter is **Uncertain**, treat it as **Good**. Otherwise, treats it as **Bad**.
- ◆ **Propagate Fail Forward:** Propagates the status to **OUT** without generating an alarm if the status from the sensor is Bad Device Failure or Bad Sensor Failure
- ◆ **Propagate Fail Backward:** If the status from the actuator is Bad Device Failure, Fault State Active, or Local Override Active, propagates this as Bad Device Failure or Good Cascade, Fault State Active, or Local Override to **BKCAL\_OUT** respectively without generating an alarm. The use of these substatus in **BKCAL\_OUT** is determined by this option. Through this option, you can determine whether alarming is done by the block or propagated upstream for alarming.
- ◆ **Target to Manual if Bad In:** Sets the Target mode to Man if the status of the **IN** parameter is **Bad**. This latches a PID Block into the Man state if the input ever goes bad.
- ◆ **Uncertain if Limited:** Sets the status of **OUT** to **Uncertain** if the measured or calculated value is limited.
- ◆ **Bad if Limited:** Sets the status of **OUT** to **Bad** if the sensor is at high or low limit.
- ◆ **Uncertain if Man Mode:** Sets the status of **OUT** to **Uncertain** if the actual mode of the block is Man.
- ◆ **Target to Next Permitted Mode if Bad CAS\_IN:** Sets the Target mode to the next permitted mode if the Target mode is **CAS** and the status of **CAS\_IN** is Bad. This

latches a control block into the next permitted mode if the **CAS\_IN** is being used in control and status goes bad.

**XD\_SCALE**

This read/write parameter enables you to set the high and low scale values, engineering units code, and the number of digits to the right of the decimal point. It is used with the value obtained from the transducer for a specified channel.

**PID Block**

The PID Block is key to many control schemes and is used almost universally. As long as an error exists, the PID function integrates the error, which moves the output in a direction to correct the error. PID blocks can be cascaded when the difference in process time constants of a primary and secondary process measurement makes it necessary or desirable.

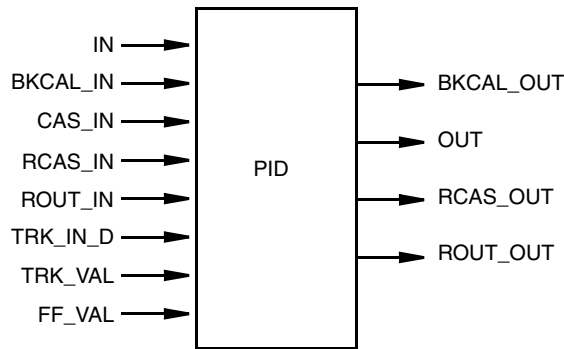


Figure 9. PID Block Parameter Summary

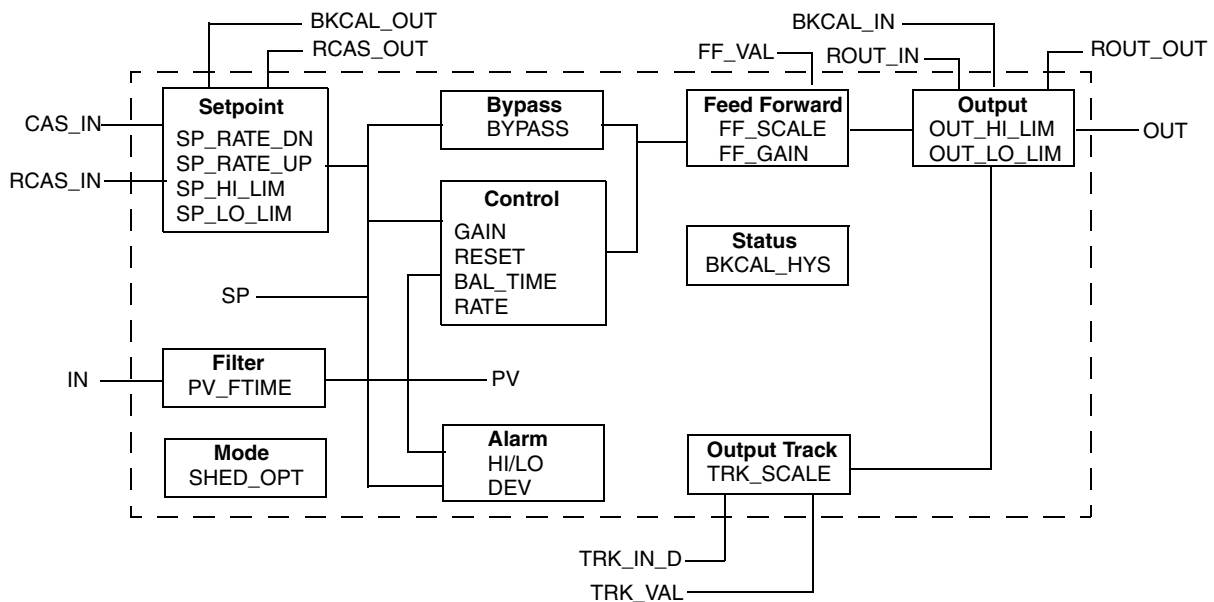


Figure 10. PID Block Schematic

## Description

The process value to be controlled is connected to the **IN** input. This value is passed through a filter whose time constant is **PV\_FTIME**. The value is then shown as the **PV**, which is used in conjunction with the **SP** in the PID algorithm. A PID does not integrate if the limit status of **IN** is constant. A full PV and DV alarm subfunction is provided. The **PV** has a status, although it is a Contained parameter. This status is a copy of the **IN** status unless **IN** is good and there is a PV or block alarm.

The full cascade SP subfunction is used with rate and absolute limits. There are additional control options which cause the **SP** value to track the **PV** value when the block is in an actual mode of **IMan**, **LO**, **Man**, or **Rout**. Limits do not cause SP-PV tracking.

A **BYPASS** switch may be available to the operator if the Bypass Enable control option is true. **BYPASS** is used in secondary cascade controllers that have a bad PV. The Bypass Enable option is necessary because not all cascade control schemes are stable if **BYPASS** is true. **BYPASS** can only be changed when the block mode is **Man** or **OOS**. While it is set, the value of **SP**, in percent of range, is passed directly to the target output, and the value of **OUT** is used for **BKCAL\_OUT**. When the mode is changed to **Cas**, the upstream block is requested to initialize to the value of **OUT**. When a block is in Cas mode, then on the transition out of **BYPASS**, the upstream block is requested to initialize to the PV value, regardless of the “Use PV for **BKCAL\_OUT**” option.

**GAIN**, **RESET**, and **RATE** are the tuning constants for the P, I, and D terms, respectively. **GAIN** is a dimensionless number. **RESET** and **RATE** are time constants expressed in seconds. There are existing controllers that are tuned by the inverse value of some or all of them, such as proportional band and repeats per minute. The human interface to these parameters should be able to display your preference. The Direct Analog control option, if true, causes the output to increase when the **PV** exceeds the **SP**. If false, the output decreases when the **PV** exceeds the **SP**. It makes the difference between positive and negative feedback, so it must be set properly and never changed while in automatic mode. The setting of the option must also be used in calculating the limit state for **BKCAL\_OUT**.

The output supports the feed-forward algorithm. The **FF\_VAL** input brings in an external value which is proportional to some disturbance in the current loop. The value is converted to percent of output span using the values of parameter **FF\_SCALE**. This value is multiplied by the **FF\_GAIN** and added to the target output of the PID algorithm. If the status of **FF\_VAL** is **Bad**, the last usable value is used because this prevents bumping the output. When the status returns to **Good**, the block adjusts its integral term to maintain the previous output.

## Explanation of PID Block Parameters

### ALARM\_HYS

This read/write parameter enables you to set the amount by which the PV must return within alarm limits before the alarm condition clears. It is expressed as a percent of the PV span.

### BAL\_TIME

This read/write parameter enables you to specify the time for the internal working value of bias or ratio to return to the operator set bias or ratio in seconds. It can be used to specify the time constant at which the integral term moves to obtain balance when the output is limited and the mode is Auto, Cas, or RCas.

**BKCAL\_HYS**

This read/write parameter enables you to specify the amount that the output must change away from its output limit before the limit status is turned off. It is expressed as a percent of span of the output.

**BKCAL\_IN**

This read only parameter shows the value and status from a lower block's **BKCAL\_OUT** that is used to prevent reset windup and to initialize the control loop.

**BKCAL\_OUT**

This read only parameter shows the value and status required by an upper block's **BKCAL\_IN** so that the upper block can prevent reset windup and provide bumpless transfer to closed loop control.

**BYPASS**

This read/write parameter enables you to bypass the normal control algorithm. When Bypass is set, the setpoint value (in percent) is directly transferred to the output. To prevent a bump on transfer to/from bypass, the setpoint is automatically initialized to the output value or process variable, respectively, and the path broken flag is set for one execution.

**CAS\_IN**

This parameter shows the remote setpoint value, which must come from another fieldbus block or a DCS block through a defined link.

**CONTROL\_OPTS**

This read/write parameter enables you to select from a list of options to alter the calculations done in a control block. The options are:

- ◆ Bypass Enable:
- ◆ **SP-PV Track in Man:** Permits the setpoint to track the process variable when the Target mode of the block is Man.
- ◆ **SP-PV Track in LO:** Permits the setpoint to track the process variable when the actual mode of the block is ROut.
- ◆ **SP-PV Track in LO or IMan:** Permits the setpoint to track the process variable when the actual mode of the block is LO or IMan.
- ◆ **SP Track Retained Target:** Permits the setpoint to track the **RCAS** or **CAS** parameter based on the retained Target mode when the actual mode of the block is IMan, LO, Man, or ROut. When SP-PV track options are enabled, then SP Track retained target has precedence in the selection of the value to track when the actual mode is Man, IMan, ROut, and LO.
- ◆ **Direct Acting:** Defines a relationship between a change in PV and corresponding change in output. When Direct is selected, an increase in PV results in an increase in the output.
- ◆ **Track Enable:** This enables the external tracking function. If true, the value in **TRK\_VAL** replaces the value of **OUT** if **TRK\_IN\_D** becomes true and the target mode is not Man.
- ◆ **Track in Manual:** This enables **TRK\_VAL** to replace the value of **OUT** when the Target mode is Man and **TRK\_IN\_D** is true. The actual mode is then LO.

- ◆ **Use PV for BKCAL\_OUT:** The **BKCAL\_OUT** and **RCAS\_OUT** values are normally the working SP. If this option is enabled, then the PV value is used after the cascade is closed.
- ◆ **Act on IR:** If this option is true, then when IR is received on **BKCAL\_IN**, the SP is adjusted within setpoint limits to provide bumpless transfer when the cascade is closed. If the setpoint required to provide bumpless transfer is outside the setpoint limits, then any difference added to provide bumpless transfer is removed in the **BAL\_TIME**.
- ◆ **Use BKCAL\_OUT with IN\_1:** Normally **BKCAL\_OUT** is associated with initialization of an upstream block which is providing **CAS\_IN**. If this option is set, then **BKCAL\_OUT** is associated with the upstream block providing IN\_1. This option may be used with the Ratio and Bias/Gain block to determine the value and status which must be provided in **BKCAL\_OUT** for proper initialization and handshaking.
- ◆ **Obey SP Limits if Cas or RCas:** Normally the setpoint is not restricted to the setpoint limits except when entered by a human interface device. However, if this option is selected, the setpoint is restricted to the setpoint absolute limits in the Cas and RCas modes.

#### **DV\_HI\_ALM and DV\_LO\_ALM**

This read only parameter shows the status and time stamp associated with the high and low deviation alarms respectively.

#### **DV\_HI\_LIM and DV\_LO\_LIM**

This read/write parameter enables you to set the high and low deviation alarm setpoints respectively in engineering units.

#### **DV\_HI\_PRI and DV\_LO\_PRI**

This read/write parameter enables you to set the priority of the high and low deviation alarms respectively.

#### **FF\_GAIN**

This read/write parameter enables you to enter the gain that the feed forward input is multiplied by before it is added to the calculated control output.

#### **FF\_VAL**

This parameter shows the feed forward value and status.

#### **FF\_SCALE**

This read/write parameter enables you to enter the feed forward input high and low scale values, engineering units code, and number of digits to the right of the decimal point.

#### **FIELD\_VAL**

This read only parameter shows the raw value of the field device in percent of the PV range, with a status reflecting the transducer condition, before signal characterization (**L\_TYPE**) or filtering (**PV\_FTIME**) feed forward value and status. Transducer scaling (**XD\_SCALE**) is applied to the value from the channel to produce the **FIELD\_VAL** in percent.



**GAIN**

This read/write parameter enables you to enter the dimensionless value used by the block algorithm in calculating the block output.

**GRANT\_DENY**

This read/write parameter is used to grant or deny access of host computers and local control panels to operating, tuning, and alarm parameters of the block.

**HI\_ALM and HI\_HI\_ALM**

This read only parameter shows the status and time stamp associated with the high and high-high alarms respectively.

**HI\_LIM and HI\_HI\_LIM**

This read/write parameter enables you to set the high and high-high alarm setpoints respectively in engineering units.

**HI\_PRI and HI\_HI\_PRI**

This read/write parameter enables you to set the priority of the high and high-high alarms respectively.

**IN**

This read only parameter shows the primary input value and status of the block. It is required for blocks that filter the input to get the PV.

**LO\_ALM and LO\_LO\_ALM**

This read only parameter shows the status and time stamp associated with the low and low-low alarms respectively.

**LO\_LIM and LO\_LO\_LIM**

This read/write parameter enables you to set the low and low-low alarm setpoints respectively in engineering units.

**LO\_PRI and LO\_LO\_PRI**

This read/write parameter enables you to set the priority of the low and low-low alarms respectively.

**OUT**

This parameter shows primary analog value and status calculated by this function block or entered by the user.

**OUT\_HI\_LIM and OUT\_LO\_LIM**

These read/write parameters enable you to enter the maximum and minimum output values.

**OUT\_SCALE**

This read/write parameter enables you to set the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the **OUT** parameter and parameters which have the same scaling as **OUT**.

## **PV**

This read only parameter shows the value and status of the primary input value or calculated value on which the algorithm bases the output calculation. The units of the input must match the process variable units.

## **PV\_FTIME**

This read/write parameter enables you to set the time constant of a single exponential filter for the PV, in seconds.

## **PV\_SCALE**

This read/write parameter enables you to enter the high and low scale values, engineering units code and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters which have the same scaling as the PV.

## **RATE**

This read/write parameter enables you to enter the derivative time constant in seconds.

## **RCAS\_IN**

This read only parameter shows the target setpoint and status provided by a supervisory Host to an analog control or output block.

## **RCAS\_OUT**

This read only parameter shows the block setpoint and status after ramping that is provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change.

## **RESET**

This read/write parameter enables you to enter the integral time constant in seconds per repeat.

## **ROUT\_IN**

This read only parameter shows the target output and status provided by a Host to the control block for use as the output (ROUT mode).

## **ROUT\_OUT**

This read only parameter shows the block output and status that is provided to a Host for back calculation in ROUT mode and to allow action to be taken under limited conditions or mode change.

## **SHED\_OPT**

This read/write parameter enables you to define the action to be taken on remote control device timeout.

## **SP\_HI\_LIM and SP\_LO\_LIM**

This read/write parameter enables you to set the highest and lowest setpoint operator entry that can be used for the block.

## **SP\_RATE\_DN and SP\_RATE\_UP**

This read/write parameter enables you to set the ramp rate at which downward (upward) setpoint changes are acted on in Auto mode in PV units per second. If the ramp is set to zero, then the setpoint is used immediately. For control blocks, rate limiting applies only in Auto mode. For output blocks, rate limiting applies in Auto, Cas, and RCas modes.

## **STATUS\_OPTS**

Refer to “STATUS\_OPTS” on page 16.

## **SV**

This read only parameter shows the value and status of the secondary value.

## **TRK\_IN\_D**

This read only parameter shows the discrete input used to initiate external tracking of the block output to the value specified by **TRK\_VAL**. If the status of **TRK\_IN\_D** is Bad, its last usable value is maintained and acted upon. If the device restarts, losing the last usable value, it is set to false.

## **TRK\_SCALE**

This read/write parameter enables you to set the high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with **TRK\_VAL**.

## **TRK\_VAL**

This read only parameter shows the input used as the track value when external tracking is enabled by **TRK\_IN\_D**. If the status of **TRK\_VAL** is Bad, the last usable value is used. If there is no last usable value, the present value of the output is used.

33 Commercial Street  
Foxboro, MA 02035-2099  
United States of America  
<http://www.foxboro.com>  
Inside U.S.: 1-866-746-6477  
Outside U.S.: 1-508-549-2424  
or contact your local Foxboro  
Representative.  
Facsimile: (508) 549-4492

Invensys and Foxboro are trademarks of Invensys plc, its subsidiaries, and affiliates.  
All other brand names may be trademarks of their respective owners.

Copyright 2005 Invensys Systems, Inc.  
All rights reserved

MB 100

Printed in U.S.A.

0605