Honeywell

FOUNDATION Fieldbus STT850 Temperature Transmitter User's Guide

34-TT-25-07
Revision 2.0
March 2016

Honeywell Process Solutions

Copyrights, Notices and Trademarks

Honeywell Process Solutions

1250 W Sam Houston Pkwy S

Houston, TX 77042

About This Document

This guide provides the details of programming Honeywell STT850 SmartLine Temperature Transmitters for applications involving FOUNDATION Fieldbus protocol. For installation, wiring, and maintenance information, refer to the *STT850 SmartLine Temperature Transmitter User Manual*.

The configuration of your transmitter depends on the mode of operation and the options selected for it with respect to operating controls, displays and mechanical installation.

An STT850 FF temperature transmitters can be digitally integrated with any FF compliant Host. Among Honeywell systems, it can be integrated with Experion PKS DCS and also use Field Device manager (FDM) for asset management and configuration.

Release Information

Document Name FOUNDATION Fieldbus STT850 Temperature Transmitter User's Guide	**Document ID**	**Release Number**	**Publication Date**
1st Release	34-TT-25-07	1.0	November 2014
CVD added, T/C Type C (W5W26) and RTD (Ni 120 & CU 10) added	34-TT-25-07	2.0	March 2016

References

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

STT850 SmartLine Temperature Transmitter Specification, # 34-TT-03-14

STT850 SmartLine Temperature Transmitter User Manual, # 34-TT-25-03

STT850 SmartLine Temperature Transmitter Quick Start Guide, # 34-TT-25-04

STT850 SmartLine Temperature Transmitter Safety Manual, # 34-TT-25-05

STT850 SmartLine Temperature Transmitter HART/DE Comms Option User Manual, Document # 34-TT-25-06

SmartLine Temperature Transmitter Pocket Configuratuion Guide, # 34-TT-00-01

SmartLine Anytime Tool (SAT) User's Guide, # 34-TT-25-12

Patent Notice

The Honeywell STT850 SmartLine Temperature Transmitter family is covered by one or more of the following U. S. Patents: 5,485,753; 5,811,690; 6,041,659; 6,055,633; 7,786,878; 8,073,098; and other patents pending.

Support and Contact Information

For Europe, Asia Pacific, North and South America contact details, refer to the back page of this manual or the appropriate Honeywell Solution Support web site:

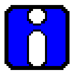
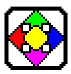
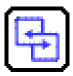
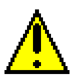
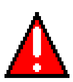
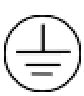
Honeywell Corporate	www.honeywell.com
Honeywell Process Solutions	https://www.honeywellprocess.com
Smartline Temperature Transmitters	https://www.honeywellprocess.com/en-US/explore/products/instrumentation/temperature-transmitters-and-sensors/smartline-temperature/Pages/default.aspx
Training Classes	https://www.honeywellprocess.com/en-US/training/Pages/default.aspx

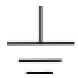
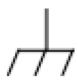
Telephone and Email Contacts

Area	Organization	Phone Number
United States and Canada	Honeywell Inc.	1-800-343-0228 Customer Service 1-800-423-9883 Global Technical Support
Global Email Support	Honeywell Process Solutions	ask-ssc@honeywell.com

Symbol Definitions

The following table lists those symbols used in this document to denote certain conditions.

Symbol	Definition
	**ATTENTION:** Identifies information that requires special consideration.
	**TIP:** Identifies advice or hints for the user, often in terms of performing a task.
	**REFERENCE -EXTERNAL:** Identifies an additional source of information outside of the bookset.
	**REFERENCE - INTERNAL:** Identifies an additional source of information within the bookset.
CAUTION	Indicates a situation which, if not avoided, may result in equipment or work (data) on the system being damaged or lost, or may result in the inability to properly operate the process.
	**CAUTION**: Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices. **CAUTION** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.
	**WARNING**: Indicates a potentially hazardous situation, which, if not avoided, could result in serious injury or death. **WARNING** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.
	**WARNING, Risk of electrical shock**: Potential shock hazard where HAZARDOUS LIVE voltages greater than 30 Vrms, 42.4 Vpeak, or 60 VDC may be accessible.
	**ESD HAZARD:** Danger of an electro-static discharge to which equipment may be sensitive. Observe precautions for handling electrostatic sensitive devices.
	**Protective Earth (PE) terminal**: Provided for connection of the protective earth (green or green/yellow) supply system conductor.
	**Functional earth terminal**: Used for non-safety purposes such as noise immunity improvement. NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national local electrical code requirements.

Symbol	Definition
	**Earth Ground**: **Functional earth connection.** NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.
	**Chassis Ground**: Identifies a connection to the chassis or frame of the equipment shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.

Terms and Acronyms

Term	Definition
Alarm	The detection of a block leaving a particular state and when it returns back to that state.
Analog Input (function block)	One of the standard function blocks define by the Foundation Fieldbus
Application	A software program that interacts with blocks, events and objects. One application may interface with other applications or contain more than one application.
AWG	American Wire Gauge
Block	A logical software unit that makes up one named copy of a block and the associated parameters its block type specifies. It can be a resource block, transducer block or a function block.
Configuration (of a system or device)	A step in system design: selecting functional units, assigning their locations and identifiers, and defining their interconnections.
CVD	Callendar Van Duesen is an equation that describe the relationship between resistance (R) and temperature (t) of platinum resistance thermometers (RTD)
Device	A physical entity capable of performing one or more specific functions. Examples include transmitters, actuators, controllers, operator interfaces.
Device Description	Description of FBAPs within a device. Files that describe the software objects in a device, such as function blocks and parameters. The DD binary are created by passing DD source files through a standard tool called a tokenizer.
Device Description Language	A standardized programming language (similar to C) used to write device description source files.
Device Tag	The Physical Device Tag of the device as specified in the Foundation Fieldbus specifications.
EEPROM	Electrically Erasable Programmable Read Only Memory
EMI	Electromagnetic Interference
Event	An instantaneous occurrence that is significant to scheduling block execution and to the operational (event) view of the application.
Field Device	A fieldbus-compatible device that contains and executes function blocks.
Foundation Fieldbus	Communications protocol for a digital, serial, two-way system which interconnects industrial field equipment such as sensors, actuators and controllers.
FTA	Field Termination Assembly
Function Block	An executable software object that performs a specific task, such as measurement or control, with inputs and outputs that connect to other function blocks in a standard way.
Function Block Application Process	The part of the device software that executes the blocks (function, transducer, or resource blocks).
Hz	Hertz

Term	Definition
Link Active Scheduler	A device which is responsible for keeping a link operational. The LAS executes the link schedule, circulates tokens, distributes time messages and probes for new devices.
Macrocycle	The least common multiple of all the loop times on a given link.
Manufacturer's Signal Processing	A term used to describe signal processing in a device that is not defined by FF specifications.
mV	Millivolts
Network Management	A part of the software and configuration data in a Foundation Fieldbus device that handles the management of the network.
Network Management Agent	Part of the device software that operates on network management objects.
Network Management Information Base	A collection of objects and parameters comprising configuration, performance and fault-related information for the communication system of a device.
Nm	Newton. Meters
NPT	National Pipe Thread
NVM	Non-Volatile Memory
Object Dictionary	Definitions and descriptions of network visible objects of a device. There are various object dictionaries within a device. The dictionaries contain objects and their associated parameters which support the application in which they are contained.
Objects	Entities within the FBAP, such as blocks, alert objects, trend objects, parameters, display lists, etc.
T	Temperature
TC	Thermocouple
Parameters	A value or variable which resides in block objects
PM	Process Manger
Proportional Integral Derivative control	A standard control algorithm. Also refers to a PID function block.
PV	Process Variable
PV1/PV2	Process Value 1/ Process Value 2
PWA	Printed Wiring Assembly
RFI	Radio Frequency Interference
RTD	Resistance Temperature Detector
Stack	The software component that implement the Foundation Fieldbus communications protocol specifications, including FMS, FAS, DLL, SM and NM.
Status	A coded value that qualifies dynamic variables (parameters) in function blocks. This value is usually passed along with the value from block to block. Status is fully defined in the FF FBAP specifications.
System Management	Provides services that coordinate the operation of various devices in a distributed fieldbus system.

Term	Definition
System Management Agent	Part of the device software that operates on system management objects.
System Management Information Base	A collection of objects and parameters comprising configuration and operational information used for control of system management operations.
TAC	Technical Assistance Center
URL1	Upper Range Limit of sensor 1
URL2	Upper Range Limit of sensor 2
US	Universal Station
Vac	Volts Alternating Current
Vdc	Volts Direct Current
Virtual Communication Relationship	A defined communication endpoint. Fieldbus communications can primarily only take place along an active communications "path" that consists of two VCR endpoints.
Virtual Field Device	A logical grouping of "user layer" functions. Function blocks are grouped into a VFD, and system and network management are grouped into a VFD. For example, to establish communications between a transducer block and a function block, a VCR must be defined at the transducer block and a VCR must be defined at the function block.

Contents

Tables

Figures

1. Introduction

1.1 About the STT850 FF Temperature Transmitter

The newly designed Honeywell STT850 is a smart temperature transmitter that has a wide range of additional features along with supporting the FOUNDATION™ Fieldbus (FF) communication protocol. The STT850 temperature transmitter with FF protocol provides a FOUNDATION™ Fieldbus interface to operate in a compatible distributed Fieldbus system. The transmitter includes FOUNDATION™ Fieldbus electronics for operating in a 31.25 Kbit/s Fieldbus network and can interoperate with any FOUNDATION™ Fieldbus registered device.

The Honeywell SmartLine STT850 is a high performance Temperature transmitter offering high accuracy and stability over a wide range of process and ambient temperatures. The STT850 Fieldbus device is fully tested and compliant with Honeywell Experion ® PKS providing the highest level of compatibility assurance and integration capabilities.

Integration with Honeywell's Experion PKS offers the following unique advantages through Smart Connection suite.

- **Transmitter messaging** – To enhance safety and productivity through clear identification and assignment of maintenance tasks in the local transmitter display
- **Maintenance mode indication** – To enhance Safety through System initiated command to identify that the device is available for maintenance
- **FDM Plant Area Views with Health summaries** – To reduce the time to identify, diagnose and fix device problems by providing an overview of device health based on user defined groups in the Honeywell Field Device Manager.

SmartLine easily meets the most demanding needs for temperature measurement applications. The device can also provide advanced calculation and control capability via the use of the optional PID, signal characterizer, arithmetic, input selector, and output splitter blocks. The best in class features of the transmitters are as follows:

- **Industry leading performance**
  - Digital Accuracy up to 0.10 Deg C for RTD
  - Stability up to 0.01% of URL per year for ten years
  - 125 mSec update time for single input models
  - 250 mSec update time for dual input models
- **Reliable measurement**
  - Built in Galvanic Isolation
  - Differential/Averaging/Redundant/Split Range measurements
  - Dual Compartment Housing
  - Sensor Break detection
  - Comprehensive on-board diagnostic capabilities
- **Lower Cost of Ownership**
  - Universal input
  - Dual sensor option
  - Multiple local display capabilities
  - Modular construction

1.2 STT850 major assembly and electronic housing components

The following illustrations depict the major assembly and electronic housing components.

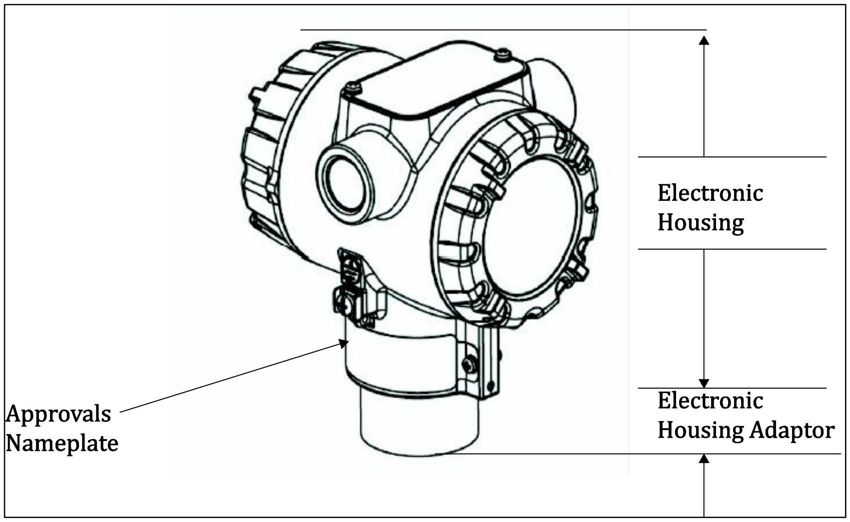


Figure 1: STT850 Major assemblies

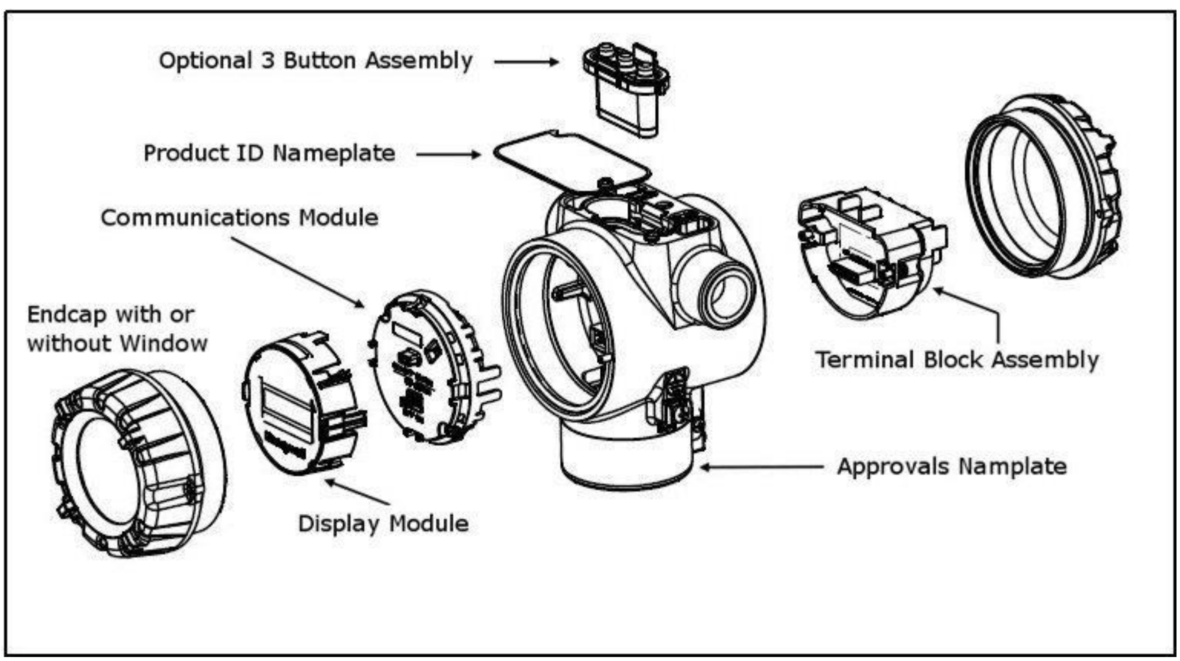


Figure 2: Electronic Housing components

1.3 Features of the transmitter

The transmitter is a configurable intelligent field device that acts as a temperature sensor, and is capable of performing control algorithms on process variables. The core functionalities of the field device include:

- Process Variable (PV) measurement
- Function Block Application Process (FBAP)
- Device diagnostics

The STT850 features standard fieldbus function blocks with manufacturer-specific additions for enhanced operation. The transmitter can function as a Link Active Scheduler (LAS) in a Fieldbus network.

It supports the following features:

- Link-master capability
- Supports the following standard function blocks apart from the Resource and Transducer blocks:
  - Analog Input block
  - Input Selector block
  - Signal Characterizer block
  - PID with auto tune block
  - Arithmetic block
  - Output splitter
  - Discrete input
- Function block instantiation is supported by the following blocks:
  - Analog Input block
  - PID with auto tune block
  - Discrete Input
  - Arithmetic block
- Supports the following Transducer blocks:
  - Temperature Transducer block
  - LCD Transducer block
  - Diagnostic Transducer block
- Supports class 3 type firmware download through commercial hosts.

DD and EDDL Features

The STT850 supports DD and EDD file formats, and the data is displayed using the EDDL features in the form of menus, graphs, charts, and pictures.

2. Getting started

2.1 Verifying the installation

Verifying transmitter installation tasks

After the transmitter is installed and powered up, you can verify communication between the transmitter and the field devices on the network. Table 1 outlines the steps for identifying and checking the transmitter on a Fieldbus network.

Table 1: Transmitter installation verification tasks

Task	Description	Comment
Verify device location	Check that the device is installed in the correct physical location.	
Verify device ID	Match the device ID with the physical location. The device serial number is the PROM ID which is stamped on the transmitter housing nameplate.	
Verify connection with host computer to device	On the operator interface, check and make sure communications are established with the device on the Fieldbus network.	
Verify or assign Device Tag and address **ATTENTION** The transmitter is shipped at a temporary (248) address. This will enable FOUNDATION Fieldbus host system to automatically recognize the device and move it to a permanent address.	Check that the Device Tag and node address are set. If not, assign the Device Tag and the correct node address. The Device Tag and address can be set and viewed using the Fieldbus device configurator application. Use a Device Tag name (up to 16 characters) that does not contain spaces.	
Configure device	Using a Fieldbus configuration program, create a function block application as part of the device configuration and process control strategy.	
Verify device operation	Bring the network online, verify operation, tune loops, and so on.	

2.2 Verifying communication with the transmitter

On the operator interface, establish communication with the device on the Fieldbus network. If the device is not visible on the network, verify that the device has been installed properly.

Identify the transmitter

Verify the Device ID of the transmitter by checking the device parameters. The parameters contain the following information:

- Transmitter type (temperature transmitter, pressure transmitter, flow transmitter and remote meter)
- Device Tag (tag description of the transmitter)
- Sensor serial number
- Firmware revision level (revision level of the firmware elements)

Check the transmitter parameters listed in Table 2 and note down the values to identify the transmitter.

ATTENTION

It is recommended to verify the correct version of the Device Description file is present on the host computer. This helps in getting the correct parameter names and its corresponding descriptions, while viewing the device parameters.

Table 2: Transmitter parameters

Parameter	To verify
Resource block **DEV_TYPE**	That the transmitter is of the proper device type. For all the STT850 type temperature transmitters, the value is 0006.
Device Tag (Physical device tag name of the transmitter) **ATTENTION** The Device Tag name can be set and viewed using the Fieldbus device configurator application. Use a device tag name (up to sixteen characters) that does not contain spaces.	The Device Tag is correct. Device Tag name ___________________
Resource Block **SERIAL_NO**	This is the serial number of the FF Transmitter which is obtained from the Sensor module. Check that the module has a valid serial number.
Resource Block **SOFTWARE_REV**	This is the Software revision of the Communication board. This may be checked when instructed by Honeywell TAC for troubleshooting.

2.3 Establishing communication with host systems

The transmitter establishes communication with the host systems using DD or DTM.

Device Description (DD)

DD is a binary file that provides the definition for parameters in the FBAP of the transmitter. For example, DD refers to the function blocks that a transmitter contains, and the corresponding parameters in the blocks that are critical to the interoperability of Fieldbus devices. They define the data required to establish communications between different Fieldbus devices from multiple vendors with control system hosts. The DD provides an extended description of each object in the Virtual Field Device (VFD).

The Fieldbus Foundation provides the DD for all registered devices on its website, www.fieldbus.org.

Enhanced Device Description (EDD)

There are two types of EDDs are available, namely .ff5/.sy5 and .ffo/sym. The.ffo/.sym binary files are generated for the legacy hosts to load the device DD that is generated using latest tokenizer. Few constructs like Images that are supported in .ff5/.sy5 binaries, are not supported in .ffo/.sym binary files.

Device Type Manager (DTM)

DTM is similar to a device driver that enables usage of devices in all the asset management and device configuration software like FDM or PACTware, with the help of the FDT-DTM technology.

The DTM has the following primary functions:

- Provides a graphic user interface for device configuration.
- Provides device configuration, calibration, and management features for the particular device.

DTM provides functions for accessing device parameters, configuring and operating the devices, calibrating, and diagnosing problems.

3. STT850 FF Configuration

3.1 Importing the STT850 FF Device Description (DD) files

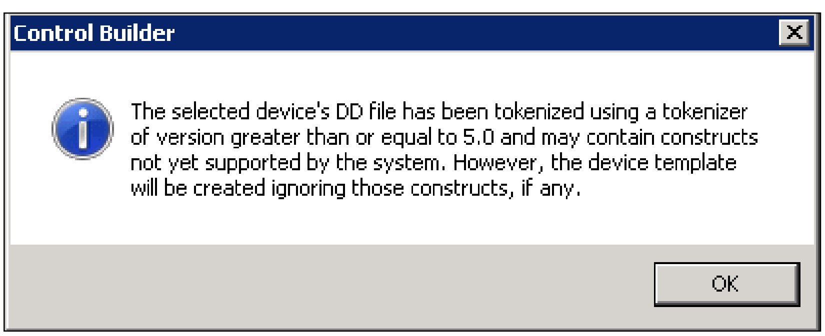
Importing the DD to Experion PKS

ATTENTION

Experion release compatibility

Experion Release	DD Compatibility
431.1	Yes
430.3	Yes
410.7	Yes

The steps in the following procedure are specific to Experion only.

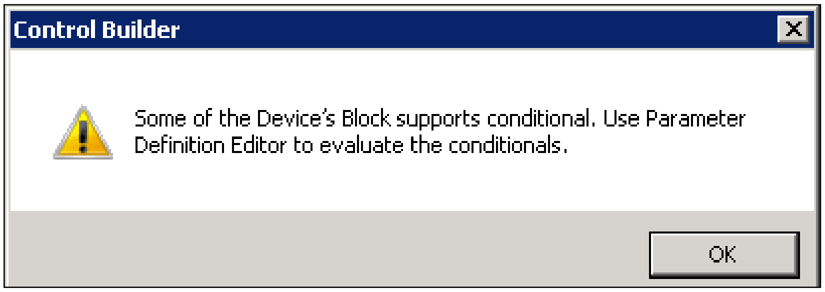
Step	Action
1	From the **Control builder** main screen, click **Fieldbus Device Description Import** . OR Select **File > New > Type >Fieldbus Device**
2	You can Import the **DD** using one of the following steps: • Choose **Browse**  to locate the folder where you have stored the DD file. • Select the required folder, and click **OK**. • Select the **DD** from the **Device List**, and click **OK**.
2	The following dialog box appears,  Click **OK**.

3 Type the **Device Type Name**, and then click **Save As**.

ATTENTION

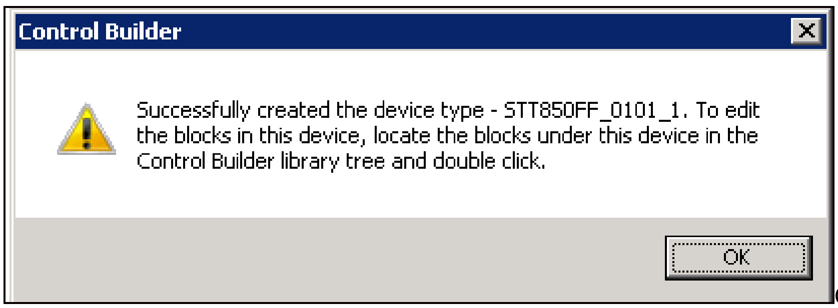
In some versions of Experion, the user must select the capability level 1 for All Function Blocks.

4 The following dialog box appears,



Click **OK**.

5 The following dialog box appears,



Click **OK**.

ATTENTION

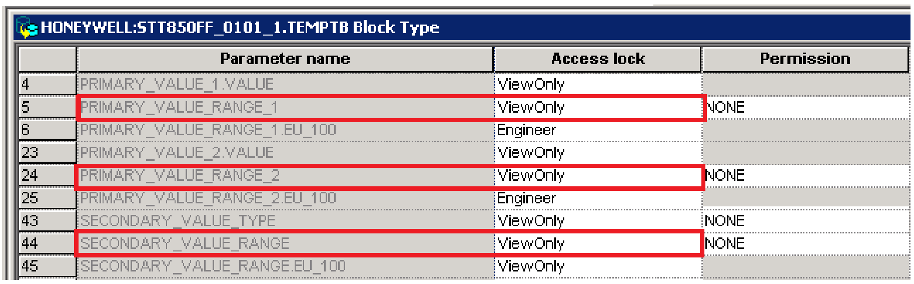
The device type – STT850FF_0101_1 is used as an example.

6 The device is created in the **Library-Containment** window under the folder named **Honeywell**.

ATTENTION

When using Experion PKS, before commissioning the device, check the Access Lock attribute value of the following parameters: **PRIMARY_VALUE_RANGE_1, PRIMARY_VALUE_RANGE_2,** and **SECONDARY_VALUE_RANGE**.
Set the Access Lock attribute value to '**Engineer'** if it is **'ViewOnly'** so in the Parameter Definition Editor (PDE) of Temperature Transducer Block (TEMPTB). PDE can be accessed by double clicking the TEMPTB block under the newly imported DD template in Library View.

During Device Load (Step 11) we need to select Full Load for access lock attribute to get reflected



HONEYWELL:STT850FF_0101_1.TEMPTB Block Type

	Parameter name	Access lock	Permission
4	PRIMARY_VALUE_1.VALUE	ViewOnly	
5	PRIMARY_VALUE_RANGE_1	ViewOnly	NONE
6	PRIMARY_VALUE_RANGE_1.EU_100	Engineer	
23	PRIMARY_VALUE_2.VALUE	ViewOnly	
24	PRIMARY_VALUE_RANGE_2	ViewOnly	NONE
25	PRIMARY_VALUE_RANGE_2.EU_100	Engineer	
43	SECONDARY_VALUE_TYPE	ViewOnly	NONE
44	SECONDARY_VALUE_RANGE	ViewOnly	NONE
45	SECONDARY_VALUE_RANGE.EU_100	ViewOnly	

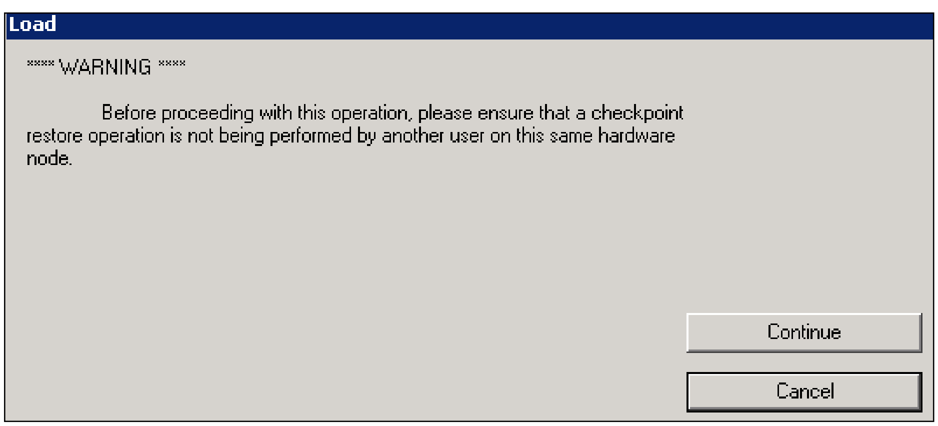
7 From the **Library-Containment** window, drag and drop the device into the corresponding FF link on the **Project-Assignment** window.

8 You are prompted to name the new function block. If you want to change the name in the destination column, type the new name or if you want to use the default name, click **Finish**.

The device is added on the FF link on the **Project-Assignment** window.

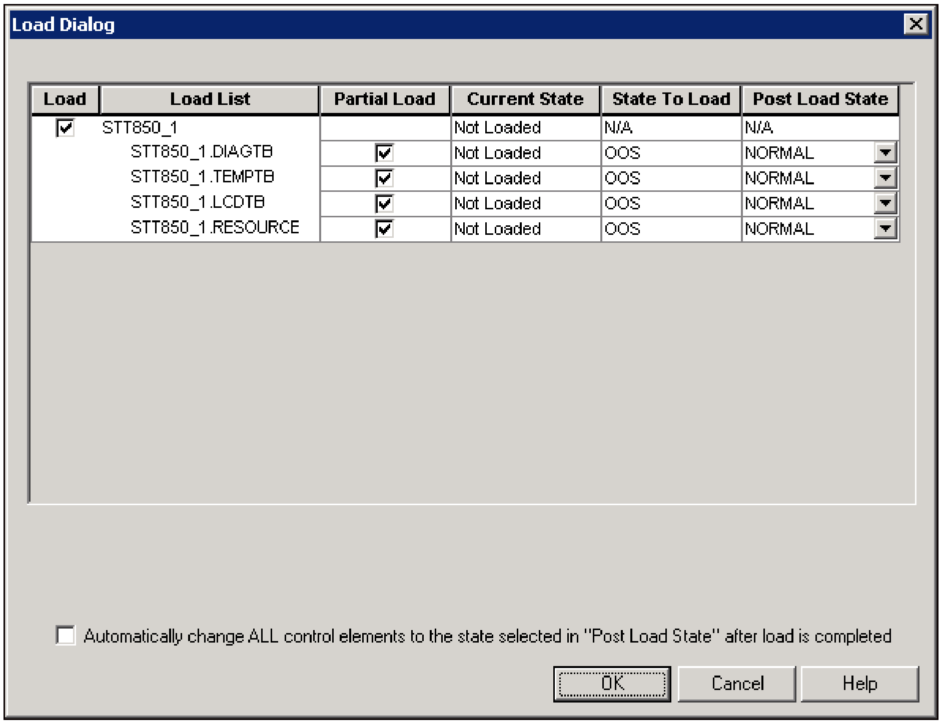
9 Right-click the new device and then click **Load**.

10 The following **WARNING** appears.



Click **Continue**.

11 The following dialog box appears,



Select the **Automatically change ALL control elements to the state selected in "Post Load State" after load is completed** checkbox and and uncheck tick mark for TEMPTB in partial load column, then click **OK**,

12 On the **Monitoring-Assignment** window, you can notice that device on the **Project-Assignment** window has been loaded to the corresponding FF link.

13 Right-click the device, and then click **Activate >> Selected Item(s) and Content(s)**. The device is commissioned.

ATTENTION

Note that after importing the DD, you have to create control strategies.

Control strategy

A control strategy is an organized approach to define a specific process using detailed information to:

- create control modules in an associated controlled environment
- configure function blocks to enable control applications, and
- runs in a control software infrastructure

To build a control strategy, a Control Module (CM) must be created where function blocks are inserted and connected with other function blocks.

Creating control strategy

For information on creating control strategy, refer to the corresponding DCS document.

ATTENTION

When control strategy is loaded by deselecting the partial download option in the Experion, parameter check errors will appear. Ignore the errors and continue the loading of control strategy.

3.2 Device replacement

Device replacement is a common plant operation, where an old or defective device is replaced with a functional device. However, the new device that is used may not be from the same manufacturer or may not have the same device type and revision as the device being replaced. The Honeywell Experion PKS DCS gives the user a simple and easy procedure to replace FF devices called ‘Unlike Device Replacement’. This procedure can be used in situations like replacing a non Honeywell FF device with a Honeywell FF device like STT850 FF Temperature transmitter in the Experion system. The Unlike device replacement report option in the control builder menu can be selected after clicking on the failed device in the monitoring (On-line) side. This report contains the steps to perform the device replacement procedure. The user can refer to Knowledge Builder in Experion PKS for more detailed steps.

3.3 Configuring the function block application process

About the Function Block Application Process (FBAP)

The transmitter has one resource block, three transducer blocks, and seven function blocks respectively. The DD-View feature supports all the blocks. The FBAP provides the block related information in a much more organized way. The FBAP defines blocks to represent different types of application functions.

In addition, the blocks have a static revision parameter. The revision level of the static data is associated with the function block. To support tracking changes in static parameter attributes, the associated block's static revision parameter is incremented each time a static parameter attribute value is changed. In addition, the associated block's static revision parameter is incremented, if a static parameter attribute is written but the value is not changed.

The FBAP supports two types of alarms: block alarms and process alarms. A block alarm is generated whenever the **BLOCK_ERR** has an error bit set. The types of block error for the AI block are shown in Table 3. The following alarms are supported by each function block:

Block Alarms

Table 3: Bit mapping of the BLOCK_ERR

Block_ERR Bit	Block Alarms	Description
0	Other	Least significant bit (LSB). **NOTE:** It is not supported by the transmitter.
1	Block Configuration error	A feature in **FEATURES_SEL** is set that is not supported by features or an execution cycle in **CYCLE_SEL** is set that is not supported by **CYCLE_TYPE**.
2	Link Configuration error	If the link is not configured properly.
3	Simulation Active	The jumper or switch that enables simulation within the resource is ON. The individual I/O function blocks disable the simulation.
4	Local Override	The block output is being set to track the value of the track input parameter. **NOTE:** It is not supported by the transmitter.
5	Device Fault State Set	If the Device Fault State condition is True. **NOTE:** It is not supported by the transmitter.
6	Device Needs Maintenance Soon	A diagnostic algorithm has found a warning condition. The NV memory is approaching the maximum number of reliable writes. **NOTE:** It is not supported by the transmitter.

Block_ERR Bit	Block Alarms	Description
7	Input Failure	When a sensor failure (open thermocouple) or sensor conversion not accurate.
8	Output Failure	Output Failure detected by this block/back calculation input has a status of Bad or Device Failure. **NOTE:** It is not supported by the transmitter.
9	Memory Failure	A diagnostic algorithm has found a failure in memory (includes all types) and the device is still able to communicate that condition.
10	Lost Static data	If the object's static data is Bad, then the object's database is set to its default values.
11	Lost NV data	The NV and static parameters are saved periodically. This alarm occurs, if new data was supposed to be saved to NV at the next NV write cycle, but prevented the write due to power failure.
12	Readback Check failed	This indicates the readback of the actual continuous valve or other actuator position in transducer units has failed.
13	Device needs maintenance now	A diagnostic algorithm has found an invalid condition, but the device is still able to operate and communicate. The NV memory has reached the maximum number of reliable writes. **NOTE:** It is not supported by the transmitter.
14	Power-up	The resource is performing its first normal execution, after power was applied to the device. It is not an error but generates an alarm that says that normal operation was interrupted and is now being restored. **NOTE:** It is not supported by the transmitter.
15	Out-of-Service	The actual mode is OOS. No control function blocks are being processed.

Process Alarms

A set of alarms that indicates a process variable has exceeded a certain threshold. Process Alarm detection is based on the **OUT** value. The alarm limits can be configured for the following standard alarms:

- High (HI_LIM)
- High High (HI_HI_LIM)
- Deviation High Limit (DEV_HI_LIM)
- Deviation Low Limit (DEV_LO_LIM)
- Low (LO_LIM)
- Low Low (LO_LO_LIM)

When the value **OUT** oscillates, **ALARM_HYS** is used to avoid alarm triggering. The priority of each alarm is set by the following parameters:

- HI_PRI
- HI_HI_PRI
- DV_HI_PRI
- DV_LO_PRI
- LO_PRI
- LO_LO_PRI

The following is the order of priority for alarms.

Table 4: Priority for Alarms

Priority	Description
0	To disable the triggered alarm, the priority of an alarm condition is changed to 0.
1	Alarm condition with a priority 1 is reported to the system, but not reported as an event and alarm
2	Alarm condition with priority of 2 is reported to the system and event, but not reported as an alarm.
3-7	Alarm conditions of priority 3 to 7 are reported as advisory alarms.
8-15	Alarm conditions of priority 8 to 15 are reported as critical alarms.

ATTENTION

Process alarms are not supported by all blocks.

3.4 Resource block

The Resource block is used to describe characteristics of the Fieldbus device such as the device name, manufacturer, and serial number. The block does not contain any input or output parameters. The block contains data that is specific to the hardware associated with the resource. The resource block monitors and controls the general operation of the device hardware. For example, if the resource block is in out of service mode, it affects all the other blocks. The **ITK_VER** parameter is used to identify the version of the Interoperability Tester. The transmitter's Revision and Versions, and Model Number can be obtained by executing the methods available in the resource block.

The block modes are used to control major states of the resource:

- The OOS mode stops all function block execution.
- The user selects the desired mode as the target. Current mode of the block is shown as the Actual mode.
- The AUTO mode allows normal operation of the resource.

Configuring the Resource block

The Resource block supports scalar input and discrete input as **HARD_TYPES**. This parameter is a read-only bit string that indicates the types of hardware that are available for this resource. The **RS_STATE** parameter contains the operational state of the Function Block Application for the data containing that resource block.

RESTART

The **RESTART** parameter allows degrees of initialization of the resource.

Restart	Operation
Run (1)	The passive state of the parameter.
Restart resource (2)	Discards unnecessary alarms, and also discards the resource dynamic values.
Restart with defaults (3)	Resets all configurable function block application objects to their initial value, which is their value before any configuration is done.
Restart processor (4)	Provides a way to press the reset button on the processor associated with the resource.

Execution

CYCLE TYPE

The parameter **CYCLE_TYPE** is a bit string that defines the types of cycles that are available for the resource and supports scheduled and block execution. **CYCLE_SEL** allows the person doing the configuration to indicate that one or more of these execution types can be used by the device. **MIN_CYCLE_T** is the minimum time to execute a cycle; the minimum cycle time supported is 100 ms.

MEMORY

MEMORY_SIZE is the size of the resource for configuration of function blocks; it is represented in kilobytes. **SHED_RCAS** and **SHED_ROUT** set the time limit for loss of communication from a remote device. These constants are used by each function block and are configurable values.

MAX NOTIFY

The **MAX_NOTIFY** parameter value is the maximum number of alert reports that this resource can send without getting a confirmation, and to control alert flooding, adjust the **LIM_NOTIFY** parameter to a lower value. If **LIM_NOTIFY** is set to zero, no alerts are reported. The **CONFIRM_TIME** parameter is the time for the resource to wait for confirmation of receipt of a report before trying again.

FEATURES

The bit strings **FEATURES** and **FEATURE_SEL** determine optional behaviour of the resource. **FEATURES** bit string defines the available features; it is read-only. **FEATURE_SEL** is used to turn on an available feature by configuration.

Reports

If the Reports option is set in the Features bit strings, the transmitter actively sends alerts to host/master. If it is not set, the host/master must poll for alerts.

SOFT W LOCK and HARD W LOCK

There are two types of write locks: Hardware write lock and Software write lock. The software write lock is used to lock the device. The software write lock does not need a jumper. A hardware write lock is provided with a jumper in the device to perform the write lock operation.

If the **WRITE_LOCK** parameter is set, it prevents any external change to the static or non-volatile database in the Function Block Application of the resource. Block connections and calculation results proceeds normally but the configuration is locked. A hard write lock is provided by a jumper in the device as indicated in the **FEATURES** bit string. Clearing **WRITE_LOCK** generates the discrete alert **WRITE_ALM** at the **WRITE_PRI** priority.

If the soft write lock bit is not set as True in the features bit strings, the writing to the parameter **WRITE_LOCK** is rejected by the device. For devices that support hard write lock and have the associated **FEATURE_SEL** attribute enabled, the parameter **WRITE_LOCK** is only an indicator of the state of write-lock. The writing to **WRITE_LOCK** is rejected by the device.

Software write lock

To activate write lock, the soft write lock supported bit in **FEATURE_SEL** must be set, and then set the **WRITE_LOCK** to locked. To deactivate write lock, set the **WRITE_LOCK** to unlocked.

Refer to Section 6.6 on how to perform the software write lock

Hardware write lock

To activate write lock, the hard write lock supported bit in **FEATURE_SEL** must be set, and additionally the write lock jumper must be in the correct position as determined by the manufacturer. When this is detected by the device, **WRITE_LOCK** is set to locked. If hard write lock is enabled in **FEATURE_SEL,** the configured value of soft write lock has no impact on device operation. To deactivate write lock, the jumper must be changed as **FEATURE_SEL** is not writeable during write lock. Once the device detects the change in jumper position, the write-lock is disabled and **WRITE_LOCK** is set to 1.

Refer to Section 6.6 on how to perform the software write lock

Install Date

When the device is connected to the master/host, the time at which the device is powered up is taken as the install date. It is a read-only parameter.

Field Diagnostics

The Resource block acts as a coordinator for alarms. There are four alarm parameters: Fail alarm, Offspec alarm, Maintenance alarm, and Check alarm. It contains information of device errors that are detected by the transmitter. Based on the error detected, the device provides the recommended actions; it is a read only parameter. It displays the recommended action text for the reported alarms.

Table 5: Diagnostic Definitions

Name	Description
Maintenance	Although the output signal is valid, the wear reserve is nearly exhausted or a function is soon restricted due to operational conditions. For example, build-up of deposits.
Off Specification	Indicates if the device is operating outside its specified range or internal diagnostics indicate deviations from measured or set values due to internal problems in the device or process characteristics.
Check Function	Output signal temporarily invalid due to on-going work on the device.
Failed	Output signal invalid due to malfunction in the field device or its peripherals.

FAILED_ALARMS

Failed alarms indicate a failure within a device that makes the device or some part of the device non-operational. This implies that the device needs repair and must be fixed immediately.

- **FAILED_MAPPED** parameter contains a list of failures in the device which makes the device non-operational that causes an alarm. There are five parameters mapped by default with **FAILED_MAPPED**: Sensor Board Fault, input 1 Fault, Input 2 Fault, Communication Board Fault, and Sensor Communication Fault.
- **FAILED_MASK** parameter masks any of the failed conditions listed in **FAILED_MAPPED**. A bit on means that the condition is masked out from alarming and is not reported.
- **FAILED_PRI** parameter designates the alarming priority of **FAILED_ALM**. The default is 0.
- **FAILED_ACTIVE** parameter displays the alarms that are active.
- **FAILED_ALM** parameter indicates a failure within a device which makes the device non-operational.

MAINT_ALARMS

A maintenance alarm indicates either the device or some part of the device needs maintenance. If the condition is ignored, the device eventually fails.

- **MAINT_MAPPED** parameter contains a list of conditions indicating either the device or some part of the device needs maintenance soon. If the condition is ignored, the device eventually fails. The following are the seven parameters mapped by default with **MAINT_MAPPED**:
    - Excess Calibration 1 Correct
    - Excess Calibration 2 Correct
    - Input 1 Out of Range
    - Input 2 Out of Range
    - Sensor Board Over Temperature
    - Cold Junction Temperature Out of Range
    - Communication Board Over Temperature
- **MAINT_MASK** parameter masks any of the failed conditions listed in **MAINT_MAPPED**. A bit on means that the condition is masked out from alarming and is not reported.
- **MAINT_PRI** designates the alarming priority of the **MAINT_ALM**. The default is 0.
- **MAINT_ACTIVE** parameter displays the alarms that are active.
- **MAINT_ALM** parameter indicates that the device needs maintenance. If the condition is ignored, the device fails.

CHECK_ALARMS

It indicates that the output signal is temporarily invalid due to on-going work on the device.

- **CHECK_MAPPED** parameter contains a list of informative conditions that do not have a direct impact on the device's primary functions.
- **CHECK_MASK** parameter masks any of the failed conditions listed in **CHECK_MAPPED**. A bit on means the condition is masked out from alarming and is not reported.
- **CHECK_PRI** parameter designates the alarming priority of the **CHECK_ALM**. The default is 0.
- **CHECK_ACTIVE** parameter displays the check alarms that are active.
- **CHECK_ALM** parameter indicates check alarms. These conditions do not have a direct impact on the process or device integrity.

OFFSPEC_ALARMS

Indicates if the device is operating outside its specified range or internal diagnostics indicates deviations from measured or set values due to internal problems in the device or process characteristics.

- **OFFSPEC_MAPPED** parameter contains a list of informative conditions that do not have a direct impact on the device's primary functions. Following are the OFFSPEC_MAPPED conditions:
    - Input 1 Health Warning
    - Input 2 Health Warning
    - No Factory Calibration
    - Input 1 Open
    - Input 2 Open
    - Input 1 TB5 Open (RTD or Ohms Sensors Only)
    - Input 1 TB6 Open (RTD or Ohms Sensors Only)
    - Input 1 TB7* Open (RTD or Ohms Sensors Only)
    - Input 2 TB7* Open (RTD or Ohms Sensors Only)
    - Input 2 TB8 Open (RTD or Ohms Sensors Only)
    - Input 2 TB9 Open (RTD or Ohms Sensors Only)
    - Unreliable Sensor Communications
    - Excess Delta Alert

*The TB7 bit is split into 2 bits to indicate whether it is open for Input 1 or input 2. TB7 terminal is common to both inputs in case of RTD and ohm sensor types.

- **OFFSPEC_MASK** parameter masks any of the failed conditions listed in **OFFSPEC_MAPPED.** A bit on means the condition is masked out from alarming and is not reported.
- **OFFSPEC_PRI** parameter designates the alarming priority of the **OFFSPEC_ALM**. The default is 0.
- **OFFSPEC_ACTIVE** parameter displays the offspec alarms that are active.
- **OFFSPEC_ALM** parameter indicates offspec alarms. These conditions do not have a direct impact on the process or device integrity.

RECOMMENDED_ACTION

The **RECOMMENDED_ACTION** parameter displays a text string that give a recommended course of action to take based on which type and which specific event of the alarms is active.

FD_SIMULATE

When simulation is enabled the Field Diagnostics conditions are taken from the Diagnostic Simulate Value, or else the conditions are taken from Diagnostic Value, and the **RECOMMENDED_ACTION** parameter displays the text as ‘Simulation Active’.

ATTENTION

Note that **FD_SIMULATE** can be enabled only if the simulation jumper is enabled in the device. For more information refer section 6.5

MAINTENANCE_MODE

It indicates if the device is available for maintenance. When the resource block is in AUTO mode, **MAINTENANCE_MODE** parameter displays the text as 'Chk with Oper' i.e., the device is in process and is not available for maintenance. When the resource block is in OOS mode, **MAINTENANCE_MODE** parameter displays the text as ‘Avail for Maint' i.e., the device is out of process and is available for maintenance. The same text is displayed in the advanced display.

'Chk with Oper'- Check with operator to determine availability.

‘Avail for Maint'- The device is available for maintenance.

SERIAL_NO

The **SERIAL_NO** parameter shows the device serial number as obtained from the Sensor module.

COMM_SERIAL_NO

The **COMM_SERIAL_NO** parameter is the serial number of the Communication board.

Parameter List

Table 6: Resource block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	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.
RS_STATE	Indicates the State of the function block application state machine.
TEST_RW	Read/write test parameter is used only for conformance testing.
DD_RESOURCE	String identifying the tag of the resource, which contains the Device Description for the resource.
MANUFAC_ID	Manufacturer identification number is used by an interface device to locate the DD file for the resource.
DEV_TYPE	Manufacturer model number associated with the resource. It is used by interface devices to locate the DD file for the resource.
DEV_REV	Manufacturer revision number associated with the resource. It is used by an interface device to locate the DD file for the resource.
CAPABILITY_LEV	The Capability Level of the Device.
DD_REV	Revision of the DD associated with the resource. It is used by the interface device to locate the DD file for the resource.

Parameter	Description
GRANT_DENY	Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.
HARD_TYPES	The types of hardware available as channel numbers. The supported hardware types are scalar input and discrete input.
RESTART	Allows a manual restart to be initiated.
FEATURES	Used to show supported resource block options. The supported features are: REPORT, SOFT_WRITE_LOCK, HARD_WRITE_LOCK, and MULTI_BIT_ALARM.
FEATURE_SEL	Used to select resource block FEATURE_SEL options
CYCLE_TYPE	Identifies the block execution methods available for this resource. The supported cycle types are: Scheduled and Block Execution.
CYCLE_SEL	Used to select the block execution method for this resource.
MIN_CYCLE_T	Time duration of the shortest cycle interval of which the resource is capable.
MEMORY_SIZE	Available configuration memory in the empty resource. It must be checked before starting a download.
NV_CYCLE_T	Minimum time interval specified by the manufacturer for writing copies of NV parameters to non-volatile memory. Zero implies it is never automatically copied. At the end of NV_CYCLE_T, only those parameters that have changed need to be updated in NVRAM.
FREE_SPACE	Percent of memory available for further configuration. Zero in preconfigured resource.
FREE_TIME	Percent of the block processing time that is free to process additional blocks.
SHED_RCAS	Time duration at which to give up on computer writes to function block RCas locations. Shed from RCas does not happen, if SHED_RCAS = 0.
SHED_ROUT	Time duration at which to give up on computer writes to function block ROut locations. Shed from Rout does not happen, if SHED_ROUT = 0.
FAULT_STATE	Condition set by loss of communication to an output block, fault promoted to an output block or a physical contact. When Fault State condition is set, output function blocks perform their FSTATE actions.
SET_FSTATE	Allows the Fault State condition to be manually initiated by selecting Set.
CLR_FSTATE	Writing a Clear to this parameter removes the device fault state if the field condition, if any has cleared.
MAX_NOTIFY	Maximum numbers of unconfirmed notify messages possible.
LIM_NOTIFY	Maximum numbers of unconfirmed alert notify messages allowed.
CONFIRM_ TIME	The time the resource waits for confirmation of receipt of a report before trying again. Retry does not happen when CONFIRM_TIME=0.
WRITE_LOCK	If set, no writes from anywhere are allowed, except to clear WRITE_LOCK. Block inputs continues to be updated.

Parameter	Description
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for configuration, hardware, and 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. When the Unreported status is cleared by the alert reporting task, another block alert is reported without clearing the Active status, if the subcode has changed.
ALARM_SUM	The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.
ACK_OPTION	Selection of whether alarms associated with the block is automatically acknowledged.
WRITE_PRI	Priority of the alarm generated by clearing the write lock.
WRITE_ALM	This alert is generated if the write lock parameter is cleared.
ITK_VER	Major revision number of the interoperability test case used in certifying this device as interoperable. The format and range are controlled by the Fieldbus Foundation. The current ITK version is 6.1.1.
FD_VER	A parameter equal to the value of the major version of the Field Diagnostics specification that the device is designed for.
FD_RECOMMEN_ACT	Enumerated list of recommended actions displayed with a device alert.
FD_FAIL_PRI	Designates the alarming priority of the FAIL_ALM. The valid range is 0-15.
FD_FAIL_MAP	Mapped FAIL_ALM alarm conditions, and corresponds bit for bit to the FAIL_ACTIVE. A bit on means that the corresponding alarm condition is Mapped and it is detected. A bit off means the corresponding alarm condition is disabled and is not detected.
FD_FAIL_MASK	Mask of FAIL_ALM. It corresponds to the bit of bit to FAIL_ACTIVE. A bit on means that the condition is masked out from alarming.
FD_FAIL_ACTIVE	Enumerated list of failure conditions within a device.
FD_FAIL_ALM	Alarm indicating a failure within a device which makes the device non-operational.
FD_MAINT_PRI	Designates the alarming priority of the MAINT_ALM. The valid range is 0-15.
FD_MAINT_MAP	Mapped MAINT_ALM alarm conditions and corresponds bit for bit to the MAINT_ACTIVE. A bit on means that the corresponding alarm condition is Mapped and is not detected. A bit off means the corresponding alarm condition is disabled and is not detected.
FD_MAINT_MASK	Mask of MAINT_ALM. It corresponds to the bit of bit to MAINT_ACTIVE. A bit on means that the condition is masked out from alarming.
FD_MAINT_ACTIVE	Enumerated list of maintenance conditions within a device.
FD_MAINT_ALM	Alarm indicating the device needs maintenance soon. If the condition is ignored, the device eventually fails.

Parameter	Description
FD_OFFSPEC_PRI	Designates the alarming priority of the OFFSPEC_ALM. The valid range is 0-15.
FD_OFFSPEC_MAP	Mapped OFFSPEC_ALM alarm conditions. Corresponds bit for bit to the OFFSPEC_ACTIVE. A bit on implies that the corresponding alarm condition is Mapped and detected. A bit off means the corresponding alarm condition is disabled and is not detected.
FD_OFFSPEC_MASK	Mask of OFFSPEC_ALM. It corresponds to the bit of bit to OFFSPEC_ACTIVE. A bit on implies that the condition is masked out from alarming.
FD_OFFSPEC_ACTIVE	Enumerated list of offspec conditions within a device.
FD_OFFSPEC_ALM	Alarm indicating offspec alarms. These conditions do not have a direct impact on the process or device integrity.
FD_CHECK_PRI	Designates the alarming priority of the CHECK_ALM. The valid range is 0-15.
FD_CHECK_MAP	Mapped CHECK_ALM alarm conditions. Corresponds bit for bit to the CHECK_ACTIVE. A bit on means that the corresponding alarm condition is Mapped and is detected. A bit off means the corresponding alarm condition is disabled and is not detected.
FD_CHECK_MASK	Mask of CHECK_ALM. It corresponds to the bit of bit to CHECK_ACTIVE. A bit on means that the condition is masked out from alarming.
FD_CHECK_ACTIVE	Enumerated list of check conditions within a device.
FD_CHECK_ALM	Alarm indicating check alarms. These conditions do not have a direct impact on the process or device integrity.
FD_SIMULATE	When simulation is enabled, the Field Diagnostics conditions are taken from Diagnostic Simulate Value, or else the conditions are taken from Diagnostic Value.
HARDWARE_REV	The hardware revision number of the communications module.
SOFTWARE_REV	The software revision number of the communications module.
COMPATIBILITY_REV	The compatibility revision number of the communications module.
MODEL_KEY	The key number of STT850 temperature transmitter (Example: STT850).
MOD_PART_1	First part of the Material of Construction Information.
MOD_PART_2	Second part of the Material of Construction Information.
MOD_PART_3	Third part of the Material of Construction Information.
MOD_PART_4	Fourth part of the Material of Construction Information.
HW_SIMULATE_JUMPER_STATE	State of Hardware Simulation Jumper (Enabled / Disabled).
INSTALL_DATE	The date and time when the device is installed in the field. The date and time is directly acquired from the FF Host.

Parameter	Description
MAINTENANCE_MODE	It indicates whether device is ready for maintenance.'Chk with Oper'- Check with operator to determine availability. 'Avail for Maint'- The device is available for maintenance.
SERIAL_NO	Serial number of the device.
COMM_SERIAL_NO	Serial Number of the Communication Module.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms (see section 3.2), and added to it, a discrete alarm for write lock.

3.5 Temperature Transducer block

The Temperature Transducer block has all the parameters and functions required to measure and calculate the temperature. The values that are measured and calculated by the transducer block are available as output values and are called as “channels”. The measured values can be read cyclically from function blocks.

Two variants of STT850 Temperature transmitter are available: Single input sensor and Dual input sensor. The temperature process variables from the two sensor inputs are available as the two Primary Values of the temperature block. In addition to the measured temperature values, calculated and derived temperatures are also available as described in the following sections. The temperature block has cold junction temperature parameters for compensation as well as for electronics housing temperature.

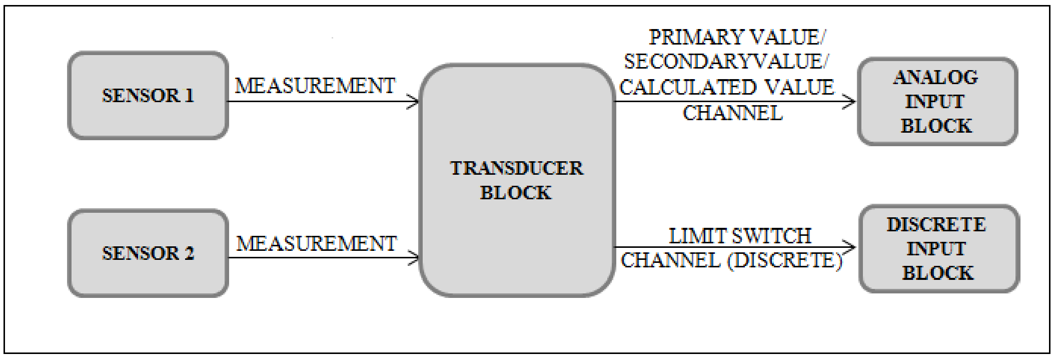


Figure 3: Temperature Transducer Block

Execution

The primary measurement is represented by **PRIMARY_VALUE_TYPE_1** and **PRIMARY_VALUE_TYPE_2** parameters. The block computes its output using primary sensor data and parameters. The calculation is modeled as shown in Figure 3.

The Temperature Transducer block supports the following process variables:

- Primary Value 1
- Primary Value 2
- Secondary Value/Cold Junction Temperature
- Electronics Temperature
- Differential Temperature
- Average Temperature
- Redundant Temperature
- Split Range Temperature

PRIMARY_VALUE_1 and **PRIMARY_VALUE_2** are the values and statuses of the measured temperatures at input 1 and 2 respectively. **PRIMARY_VALUE_RANGE_1** and **PRIMARY_VALUE_RANGE_2** include the limits, units, and the decimal point position (number of significant digits to the right of the point) of the **PRIMARY_VALUE_1**, and **PRIMARY_VALUE_2** respectively. The status of PV1 or PV2 goes bad when the input measured from Sensor 1 or Sensor 2 is invalid; for example, under circumstances when **Enable Break Detect** is set and the sensor wires get disconnected. When the units of the **PRIMARY_VALUE_1** and **PRIMARY_VALUE_2** are changed, their limits also automatically change. The engineering units must match the units selected in the parameter **XD_SCALE** of the Analog Input block that reads the channel with this value.

SECONDARY_VALUE is the value and status of the Cold Junction Temperature. **SECONDARY_VALUE_RANGE** is the limit of the **SECONDARY_VALUE,** and it is a read-only parameter. The units of the **SECONDARY_VALUE** can be changed as desired (changing the units of the value automatically changes the limits). The engineering units must match the units selected in the parameter **XD_SCALE** of the Analog Input block that reads the channel with this value.

Electronics housing temperature

EL_TEMPERATURE is the value and status of the temperature of the Electronic housing. The units can be changed as desired. The engineering units must match the units selected in the parameter **XD_SCALE** of the Analog Input block that reads the channel with this value.

Temperature Calculation

The calculated values of Differential temperature, Average temperature, Redundant temperature and Split Range temperature are only available for dual input variants of the transmitter.

Differential Temperature

The Differential temperature is the difference between PV1 and PV2.

$$Differential\ Temperature = PV1 - PV2$$

The status of the Differential temperature goes bad when either PV1 or PV2 goes bad.

The default Differential temperature ranges from $(LRL1 - URL2)$ to $(URL1 - LRL2)$.

Average Temperature

The Average temperature is the average of PV1 and PV2.

$$Average\ Temperature = \frac{PV1 + PV2}{2}$$

When the **Average: Enable Redundant** mode bit is disabled, if either PV1 or PV2 goes bad the status of the Average temperature also goes bad.

With the **Average: Enable Redundant** bit enabled, if one of the PV goes bad then the Average temperature is the value of the other PV. If both PV1 and PV2 go bad the status of the Average temperature goes bad.

The default Average temperature ranges from the greater value of LRL1 and LRL2 to the lower value of URL1 and URL2.

Redundant Temperature

In Redundant Temperature mode, both sensors are connected to the same process. By default, the value measured by Sensor 1 (PV1) is represented as the redundant value. If PV1 goes bad, the redundant value becomes the value measured by Sensor 2 (PV2).
Once Redundant value switches to PV2, it switches back to PV1 and recovers back to good status. When Redundant: Switch to Sensor1 when Sensor 2 Bad option is enabled, it will switch back to PV1 only after PV2 goes bad.

The default Redundant temperature ranges from the greater value of LRL1 and LRL2 to the lower value of URL1 and URL2.

Split Range Temperature

In split range temperature mode, a temperature range is split into two ranges: lower value of the temperature range to Mid Range Value (MRV) and MRV to the higher value of the temperature range.

Split range temperature value is the value PV1 measured by sensor 1 as long as PV1 is below MRV. Once PV1 exceeds MRV, the split range temperature value becomes the value measured by Sensor2 (PV2).

The default range of Split Range temperature ranges from LRL1 to URL2.

The default range of MRV ranges from 0 to $\frac{LRL1+URL2}{2}$.

The equation for MRV is as follows:

$$MRV = \frac{LRL1 + URL2}{2}, \quad when \frac{LRL1 + URL2}{2} \leq URL1$$

$$MRV = URL1, \qquad when \frac{LRL1 + URL2}{2} > URL1$$

By default, Hysteresis is only applied on PV2 to PV1 transition. It can also be enabled for PV1 to PV2 transition using **Split Range: Hysteresis on Sensor 1 to 2 transition** in **ENABLE_OPT**.

The value of the Split Range temperature with **Split Range: Hysteresis on Sensor 1 to 2 transition** option not enabled is as follows:

$$Split\ Range\ Temperature = PV1, \quad when\ PV1 \leq (MRV - Hysteresis)$$

$$Split\ Range\ Temperature = PV2, \quad when\ PV1 > MRV$$

During transition of PV from PV1 to PV2, the value of the Split Range temperature remains PV1 for all values of PV1 such that MRV-Hysteresis <PV1< MRV.

During transition of PV from PV2 to PV1, the value of the Split Range temperature remains PV2 for all values of PV2 such that MRV-Hysteresis <PV2< MRV.

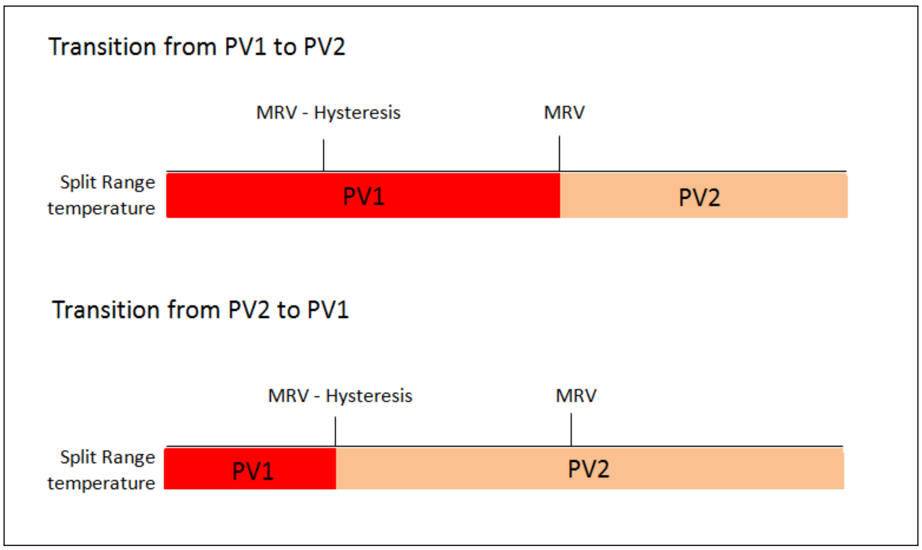


Figure 4: Transition of Split Range Temperature with 'Split Range: Hysteresis on Sensor 1 to 2 transition' option not enabled

The Split Range temperature value with **Split Range: Hysteresis on Sensor 1 to 2 transition** enabled is as follows:

$$Split\ Range\ Temperature = PV1, \quad when\ PV1 < (MRV - Hysteresis)$$

$$Split\ Range\ Temperature = PV2, \quad when\ PV1 > (MRV + Hysteresis)$$

During transition of PV from PV1 to PV2, the value of the Split Range temperature remains PV1 for all values of PV1 such that: (MRV-Hysteresis) < PV1 < (MRV + Hysteresis).

During transition of PV from PV2 to PV1, the value of the Split Range temperature remains PV2 for all values of PV2 such that: (MRV-Hysteresis) < PV2 < (MRV + Hysteresis).

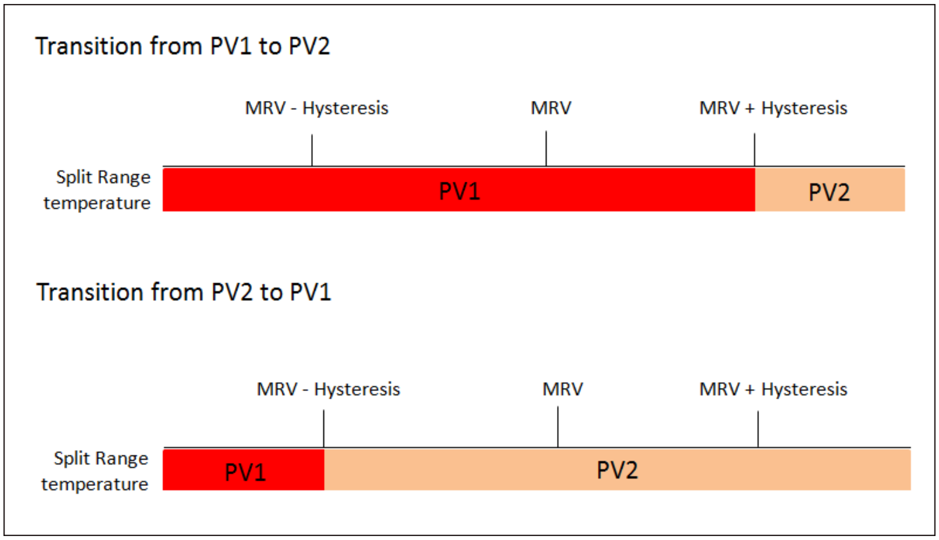


Figure 5: Transition of Split Range Temperature with 'Split Range: Hysteresis on Sensor 1 to 2 transition' option enabled

The parameter **ENABLE_OPT** has bit options to configure the inputs to the temperature block. The bit options and their functions are as follows:

- **Enable Latch:** When a PV goes bad, the Bad status of that PV stays latched. Once the PV becomes good, the status changes to good only if the latched status is manually cleared using the Clear latch method.
- **Enable Excess Delta Detect:** Sets a Field Diagnostics Alert when **TEMP_DELTA_VALUE**, the absolute difference between PV1 and PV2, crosses the configurable **EXCESS_DELTA_LIMIT**.
- **Enable Break Detect:** Identifies any break in Sensor 1 or 2 connections and causes the corresponding PV status to become 'Bad' sensor failure when a break is detected.
- **Enable Match PV:** Matches PV2 to PV1, where PV1 is the reference value.
- **Enable Differential Temperature:** Enables Differential Temperature value.
- **Enable Average Temperature:** Enables Average Temperature value.
- **Enable Redundant Value:** PV2 is the input when PV1 becomes non-functional.
- **Enable Split Range:** Enables Split Range Temperature value.
- **Average: Enable Redundant mode:** In the Average Temperature mode, when this bit is enabled and one of the PVs go bad, the other good PV is the Average temperature value.

- **Split Range: Hysteresis on Sensor 1 to 2 transition:** In Split Range Mode when this bit is enabled, the output of the temperature block transits from PV1 to PV2 only when value of PV1 exceeds the sum of MRV and the predefined hysteresis value. Similarly, the output of the temperature block transits from PV2 to PV1 only when value of PV2 decreases below the difference between MRV and hysteresis.

- **Redundant: Switch to Sensor1 when Sensor 2 Bad:** If the Redundant value is enabled and is currently PV2, with this option enabled, the Redundant value switches back to PV1 if PV2 goes bad and PV1 is good at that time.

ATTENTION

- Note that the status and value of PV from the temperature block can also be viewed in AI block.

Limit Switch

Limit switches are temperature based switches. There are four limit switches. Each limit switch has the following parameters:

LIM_SW_VAL: The discrete value and status of the limit switches. OFF is represented by 0 and ON by 1. LIM_SW_VAL of all the four switches are represented as: **LIM_SW_VAL_1, LIM_SW_VAL_2**, **LIM_SW_VAL_3**, and **LIM_SW_VAL_4**.

LIM_SW_PV_SRC: The temperature value which is used as the source for the limit switches. Following are the possible values that can be used as source for the limit switches: Average Value, Differential Value, Primary Value 1, Primary Value 2, Redundant Value, Secondary Value, and Split Range Value. The LIM_SW_PV_SRC for all the four limit switches are represented as: **LIM_SW_PV_SRC_1, LIM_SW_PV_SRC_2, LIM_SW_PV_SRC_3,** and **LIM_SW_PV_SRC_4**.

LIM_SW_SP: The setpoint for limit switches. The setpoint acts as the threshold for Limit Switch to switch to ON/OFF. The LIM_SW_SP for all the four limit switches are represented as: **LIM_SW_SP_1, LIM_SW_SP_2, LIM_SW_SP_3,** and **LIM_SW_SP_4**.

LIM_SW_DIR: The direction of action for limit switches. Two types of actions are possible: HI_LIMIT and LO_LIMIT. For HI_LIMIT, the limit switch turns ON when the PV source goes above the **LIM_SW_SP**. For LO_LIMIT, the limit switch turns ON when the PV source goes below the **LIM_SW_SP**. The LIM_SW_DIR for all the four limit switches are represented as: **LIM_SW_DIR_1, LIM_SW_DIR_2, LIM_SW_DIR_3,** and **LIM_SW_DIR_4.**

LIM_SW_HYST: Hysteresis value applied for the corresponding limit switch calculation. The **LIM_SW_HYST** for all the four limit switches are represented as: **LIM_SW_HYST_1, LIM_SW_HYST_2, LIM_SW_HYST_3, and LIM_SW_HYST_4.**

The combination of Set Point temperature, Limit Switch Direction and Hysteresis determines the temperature at which the transition from On to Off and vice-versa must occur.

If the High_Limit option is enabled for Limit Switch Direction, then the following occurs:

- Transition occurs from OFF to ON when process value goes above setpoint temperature.
- Transition occurs from ON to OFF when the process value goes below (Set Point temperature – Hysteresis).

If the Low_Limit option is enabled for Limit Switch Direction, then the following occurs

- Transition occurs from OFF to ON when process value goes below setpoint temperature.
- Transition occurs from ON to OFF when the process value goes above (Set Point temperature + Hysteresis).

LIM_SW_UNIT: The units of limit switch parameters: **LIM_SW_SP_n** and **LIM_SW_HYST_n**. The LIM_SW_UNIT for all the four limit switches are represented as: **LIM_SW_UNIT_1, LIM_SW_UNIT_2, LIM_SW_UNIT_3,** and **LIM_SW_UNIT_4,** represents the unit of the value selected for the **LIM_SW_PV_SRC_n** and it is a read-only parameter.

Calibration

Calibration is performed in order to get the precise value. If there is any offset in value it can be eliminated by performing calibration. For example, for zero temperature, if the sensor indicates a value of 0.1, then it can be made zero by performing calibration. Hence, when zero temperature is measured next time, it indicates exactly zero.

The following six types of sensor calibration actions can be performed on STT850:

- For Sensor 1
  - **Correct Cal Lo #1**
  - **Correct Cal Hi #1**
  - **Correct Reset #1**
- For Sensor 2
  - **Correct Cal Lo #2**
  - **Correct Cal Hi #2**
  - **Correct Reset #2**

ATTENTION

Note that calibration is performed only for the primary variables.

Calibration Parameters

Correct Cal Lo #1 and **Correct Cal Lo #2** are used to correct the Lower range values. For Calibration Lo, use **CAL_POINT_LO_1** and **CAL_POINT_LO_2** for the respective sensors.

Correct Cal Hi #1 and **Correct Cal Hi #2** are used to correct the Upper range values. For Calibration Hi, use **CAL_POINT_HI_1** and **CAL_POINT_ HI_2** for the respective sensors.

CAL_POINT_HI_1 and **CAL_POINT_HI_2** are the upper calibrated values. For Sensor 1, the value **CAL_POINT_HI_1** must be at least **CAL_MIN_SPAN_1** away from **CAL_POINT_LO_1** and at or below the high range value of **SENSOR_RANGE_1**. Similarly, for Sensor 2 the value **CAL_POINT_HI_2** must be at least **CAL_MIN_SPAN_2** away from **CAL_POINT_LO_2** and at or below the high range value of **SENSOR_RANGE_2**.

CAL_POINT_LO_1 and **CAL_POINT_LO_2** are the lower calibrated values. For Sensor 1, the value **CAL_POINT_LO_1** must be at least **CAL_MIN_SPAN_1** away from **CAL_POINT_HI_1**, and at or above the low range value of **SENSOR_RANGE_1** . Similarly, for Sensor 2 the value **CAL_POINT_LO_2** must be at least **CAL_MIN_SPAN_2** away from **CAL_POINT_HI_2**, and at or above the low range value of **SENSOR_RANGE_2**.

CAL_MIN_SPAN_1 is the absolute minimum span between **CAL_POINT_HI_1** and **CAL_POINT_LO_1**. Similarly, **CAL_MIN_SPAN_2** is the absolute minimum span between **CAL_POINT_HI_2** and **CAL_POINT_LO_2**.

CAL_VALUE_1 shows the **PRIMARY_VALUE_1** in the units defined by **CAL_UNIT_1**. Similarly, **CAL_VALUE_2** shows the **PRIMARY_VALUE_2** in the units defined by **CAL_UNIT_2**.

CAL_UNIT_1 and **CAL_UNIT_2** are the engineering units to be used when calibrating the respective sensors.

SENSOR_RANGE_1 and **SENSOR_RANGE_2** are the high and low range limit values, the engineering units code, and the number of significant digits to the right of the decimal point of the respective sensors.

SENSOR_SN_1 and **SENSOR_SN_2** show serial number of the respective sensors.

SENSOR_CAL_METHOD_1 and **SENSOR_CAL_METHOD_2** are the last calibration methods of the respective sensors.

SENSOR_CAL_LOC_1 and **SENSOR_CAL_LOC_2** are the last calibration locations of the respective sensors.

SENSOR_CAL_DATE_1 and **SENSOR_CAL_DATE_2** are the last calibration dates of the respective sensors.

SENSOR_CAL_WHO_1 and **SENSOR_CAL_WHO_2** identify the person that last calibrated the respective sensors.

Calibration Action

To perform a Calibration Action, **CAL_ACTION_1** and **CAL_ACTION_2** are selected for Sensors 1 and 2 respectively. The value for these parameters can be set to any of the following:

- **Prepare for Calibration**
- **Correct Cal Lo #1**
- **Correct Cal Hi #1**
- **Correct Reset #1**
- **Correct Cal Lo #2**
- **Correct Cal Hi #2**
- **Correct Reset #2**

Calibration Sequence

The “n” at suffix of the parameter names can have the value 1 or 2 depending on the respective sensor. To calibrate the sensors of STT850, perform the following steps:

1. Select **CAL_ACTION_n** to prepare for calibration. In case of **Correct Reset** #**1** and #**2**, skip Steps 2 to 4.
2. Set **CAL_UNIT_n** to the desired calibration units.
3. Set **CAL_POINT_LO_n** or **CAL_POINT_HI_n** to the desired LO or HI calibration points. In case of LRV correct, set **CAL_POINT_LO_n** and in case of HRV correct, set **CAL_POINT_HI_n**.
4. Set the temperature input value to the value set in Step 3 using a precise source such as, certified calibrator when **CAL_VALUE_n** shows the **PV_n** in **CAL_UNITS_n** units. **CAL_VALUE_n** shows the measured value before calibration at this stage.
5. Update **SENSOR_CAL_LOC_n**, **SENSOR_CAL_DATE_n**, **SENSOR_CAL_WHO_n** to reflect the location, date and User Identity for the current calibration.
6. Set **CAL_ACTION_n** to the desired calibration type.
7. Check **CAL_STATUS_n** to see the status of the calibration action performed.
   Based on the input provided, the status may be either of the following:
   - **SUCCESS**
     In case of success, proceed to Step 8.
   - **ERROR**

     In case of error, the calibration has failed. The user may have to repeat the calibration by entering correct values.
8. Check **CAL_VALUE_n** to see the value after calibration.

The STT850 device has six methods corresponding to each calibration type. The methods guide the user through the calibration steps described above and provide an improved user experience.

Calibration diagnostics

The transducer block keeps track of calibration history for each of the calibration type: LRV Correct, URV Correct and Corrects Reset for Sensors 1 and 2. The parameters used to track the calibration history are as follows:

- **CURR_TIME_DATE:** Represents the date and time when the latest calibration was performed.
- **LAST_TIME_DATE:** Represents the date and time of calibration done prior to the current calibration.
- **PREV_TIME_DATE:** Represents the date and time of calibration done prior to the last calibration.

Using **UPLOAD_CALIB_DATA**, the calibration type is selected to update the Current, Last and Previous calibration timestamps in the corresponding parameters.

Sensors

SENSOR_TYPE_1 and **SENSOR_TYPE_2** are the types of respective sensors. Various types of sensors are available and are as follows:

- Ohms (0 to 500 ohms)
- Ohms (0 to 2000 ohms)
- Ohms (0 to 3000 ohms)
- mV (-100 to 1200 mV)
- mV1(-20 to 125 mV)
- PT1000_A_385(IEC 751)
- PT500_A_385 (IEC 751)
- PT100_A_385(IEC 751)
- PT200_A_385(IEC 751)
- PT25_A_385(IEC 751)
- RTD_NI120 (Edison #7)
- RTD_CU10 (Edison #15)
- T/C Type B (IEC 584-1, and NIST 175)
- T/C Type C (ASTM E 988-96) (W5W26)
- T/C Type E (IEC 584-1, and NIST 175)
- T/C Type J (IEC 584-1, and NIST 175)
- T/C Type K (IEC 584-1, and NIST 175)
- T/C Type N (IEC 584-1, and NIST 175)
- T/C Type R (IEC 584-1, and NIST 175)
- T/C Type S (IEC 584-1, and NIST 175)
- T/C Type T (IEC 584-1, and NIST 175)

SENSOR_CONNECTION_1 and **SENSOR_CONNECTION_2** specify the wire count used for the respective sensors. The possible values are: 2-Wire, 3-Wire and 4-Wire. This selection can be changed only for RTDs and is always 2-Wire in case of TC.

SENSOR_1_INFO and **SENSOR_2_INFO** are user editable fields of the respective sensors. The user editable field is of 32 characters and can be used to store user desired information; such as, model or type of sensor.

SENS_RESIS_1 and **SENS_RESIS_2** specify the resistance of the respective sensors.

SENSOR_OPTS indicates if Callender van-Dusen Coefficient(CVD) support for RTDs is available in the sensor firmware. This feature is supported for Pt100, Pt200, Pt500 and Pt1000 RTDs and not available for Pt25.

CVD_ENABLE_1 and **CVD_ENABLE_2** can be used to enable or disable the CVD feature for the respective sensors.

CVD_COEFF_1 and **CVD_COEFF_2** are the CVD Coefficients for RTDs. These consist of the R0(Resistance at 0°C),Alpha, Delta and Beta CVD parameters for the respective sensors.

CVD_UCAL_HI_LIM_1, CVD_UCAL_LO_LIM_1, CVD_UCAL_HI_LIM_2 and **CVD_UCAL_LO_LIM_2** are the CVD Calibration high and low points (in Ohms) for the respective sensors.

ATTENTION

Changing the RTD type (Pt100, Pt200, Pt500, Pt1000) will automatically disable CVD and set the CVD coefficients to their default values for the RTD type selected.

Sensor Methods

There are three sensor methods available and are as follows:

Clear latch: This method clears the latched states of PV.

Sensor Reset: This method resets the Sensor Board. The method can be run only in OOS mode. It also clears the latched states or faults.

Redundant: Switch to Sensor 1: When Redundant temperature value is enabled, this method can be used to switch from Sensor 2 to Sensor 1 if PV1 is good. This method is useful when 'Redundant: Switch to Sensor1 when Sensor 2 Bad' option is enabled and the user wants to switch back to Sensor 1 after it has recovered.

Parameter List

Table 7: Temperature Transducer block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	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.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for all configuration, hardware, and 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. After 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.
TRANSDUCER_DIRCTORY	A directory that specifies the number and starting indices of the transducers in the transducer block.
TRANSDUCER_TYPE	Identifies the transducer that follows.
TRASDUCER_TYPE_VER	The version number of the transducer that follows.
XD_ERROR	Provides additional error codes related to transducer blocks.
COLLECTION_DIRECTORY	A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer block.
CAL_POINT_HI_1	The highest calibrated value of sensor 1.
CAL_POINT_LO_1	The lowest calibrated value of sensor 1.
CAL_MIN_SPAN_1	The minimum calibration span value allowed for sensor1. This minimum span information is necessary to ensure that when calibration is done, the two calibrated points are not too close together.
CAL_VALUE_1	PRIMARY_VALUE_1 shown in selected CAL_UNIT_1 to aid in calibration

Parameter	Description
XD_OPTS	Transducer options. Two options are supported: Input Status BAD in MAN and Input Status uncertain in MAN. These are used to determine the status of PV when the actual mode of the block is MAN.
SENSOR_CAL_LOC_1	The location of the last sensor calibration of sensor 1. This describes the physical location at which the calibration was performed.
SENSOR_CAL_DATE_1	The date of the last sensor calibration of sensor 1. It shows the calibration of that part of the sensor that is usually wetted by the process.
SENSOR_CAL_WHO_1	The name of the person who did the last sensor calibration for sensor1.
CAL_POINT_HI_2	The highest calibrated value of sensor 2.
CAL_POINT_LO_2	The lowest calibrated value of sensor 2.
CAL_MIN_SPAN_2	The minimum calibration span value allowed for sensor 2. This minimum span information is necessary to ensure that when calibration is done, the two calibrated points are not too close together.
SENSOR_CAL_LOC_2	The location of the last sensor calibration of sensor 2. This describes the physical location at which the calibration was performed.
SENSOR_CAL_DATE_2	The date of the last sensor calibration of sensor 2. It shows the calibration of that part of the sensor that is usually wetted by the process.
SENSOR_CAL_WHO_2	The name of the person who did the last sensor calibration for sensor 2.
SECONDARY_VALUE	The secondary value that is related to the sensor.
PRIMARY_VALUE_1	The measured value and status available to the function block from sensor 1.
PRIMARY_VALUE_TYPE_1	The type of measurement represented by the primary value.
PRIMARY_VALUE_RANGE_1	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the final value.
CAL_UNIT_1	The Device Description engineering units code index for the calibration values for Sensor 1.
SENSOR_TYPE_1	The type of sensor connected to input 1 of the transducer block.
SENSOR_RANGE_1	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point for sensor 1.
SENSOR_SN_1	The serial number of sensor 1.
SENSOR_CAL_METHOD_1	The method of last sensor calibration for sensor 1.

Parameter	Description
SENSOR_CONNECTION_1	The wire count used in for sensor 1. Possible values are: 2-Wire, 3-Wire and 4-Wire. This selection can be changed only for RTDs and is always 2-Wire in case of TC
PRIMARY_VALUE_TYPE_2	The measured value and status available to the function block from sensor 2.
PRIMARY_VALUE_2	The measured value and status available to the function block from sensor 2.
PRIMARY_VALUE_RANGE_2	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the final value.
CAL_VALUE_2	PRIMARY_VALUE_2 shown in selected CAL_UNIT_2 to aid in calibration.
CAL_UNIT_2	The Device Description engineering units code index for the calibration values for Sensor 1.
SENSOR_TYPE_2	The type of sensor connected to input 2 of the transducer block.
SENSOR_RANGE_2	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point for sensor 2.
SENSOR_SN_2	The serial number of sensor 2.
SENSOR_CAL_METHOD_2	The method of last sensor calibration for sensor 1.
SENSOR_CONNECTION_2	The wire count used in for sensor 2. Possible values are: 2-Wire, 3-Wire and 4-Wire. This selection can be changed only for RTDs and is always 2-Wire in case of TC.
NUM_OF_INPUTS	This parameter defines the number of sensors that can be connected to the transmitter. The value can be 1 or 2 depending on the transmitter variant in use.
SECONDARY_VALUE_TYPE	This is a read only parameter indicating secondary value type. In the temperature transmitter, the Cold Junction temperature is shown as the secondary value.
SECONDARY_VALUE_RANGE	The range of the secondary value.
CJC_TYPE_PARAM	The Cold junction compensation type to be used: Internal, Fixed or External. External can be selected to use an external PT100 RTD as a CJ sensor connected to input 2 for compensating the TC connected to input 1.
FIXED_CJC_VALUE	The Fixed value to be used for Cold junction compensation. This value is used only when CJC_TYPE_PARAM is chosen to be fixed.
BIAS1	The temperature bias applied to the measurement at input1.
LEAD_WIRE_RESISTANCE_1	The lead wire resistance for input 1. Used only for RTDs.
SENS1_INSTALL_DATE	The installation date of sensor 1. It is updated manually by the user.

Parameter	Description
BIAS2	The temperature bias applied to the measurement at input 2.
LEAD_WIRE_RESISTANCE_2	The lead wire resistance for input 2. Used only for RTDs.
SENS2_INSTALL_DATE	The installation date of sensor 2. It is updated manually by the user.
SENSOR_1_INFO	User editable field of 32 characters for storing user desired information on sensor 1 such as, model or type of sensor.
SENSOR_2_INFO	User editable field of 32 characters for storing user desired information on sensor 2 such as, model or type of sensor.
ENABLE_OPT	Used to select various options that affect the calculated temperature values. See options of ENABLE_OPT for more details.
DIFFERENTIAL_VALUE	A calculated process variable representing the difference in measured temperature between sensor1 and sensor2.
DIFFERENTIAL_VALUE_RANGE	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the DIFFERENTIAL_VALUE.
AVERAGE_VALUE	A calculated process variable representing the average of temperature measured by sensor 1 and sensor 2.
AVERAGE_VALUE_RANGE	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the AVERAGE_VALUE
AVG_SENS_USED	The sensor used in determining AVERAGE_VALUE. Possible options are: sensor 1, sensor 2 or Both.
REDUNDANT_VALUE	A process variable which represents the temperature measured in a redundant setup with sensors 1 and 2 connected to the same process.
REDUNDANT_VALUE_RANGE	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the REDUNDANT_VALUE.
RED_SENS_USED	The sensor used in determining REDUNDANT_VALUE. Possible options are: sensor 1, sensor 2.
SPLIT_RANGE_TEMP_VALUE	A process variable which represents the temperature measured in Split Range mode.
SPLIT_RANGE_TEMP_VALUE_RANGE	The high and low range limit values, the engineering units code, and the number of digits to the right of the decimal point to be used to display the SPLIT_RANGE_TEMP_VALUE
SPL_RNG_SENS_USED	The sensor used in determining SPLIT_RANGE_TEMP_VALUE. Possible options are: Sensor 1, Sensor 2
MRV_PARAM	The Mid Range Value which determines the Sensor used in Split Range.
SPL_RNG_HYSTERESIS	Hysteresis applied in deciding to switch the Sensor used in Split Range.

Parameter	Description
BMPLSS_TRF_DAMPING	For Redundant and Split Range PVs, specifies the damping value that is applied when switching between sensors. Range is from 0 to 99.9 sec
EL_TEMPERATURE	The value and status of the measured temperature inside the electronics housing.
EL_TEMP_UNIT	The engineering unit code used to display the Electronics Temperature.
SENS_RESIS_1	The resistance of sensor 1 input. Currently, available only for RTDs.
SENS_RESIS_2	The resistance of sensor 2 input. Currently, available only for RTDs.
HARD_REV	Hardware revision of the sensor board.
FIRM_REV	Firmware revision of the sensor board.
CHAR_DATE	Characterization date of the sensor board.
CAL_ACTION_1	Used by the calibration methods to initiate a calibration on a device.
CAL_ACTION_2	Used by the calibration methods to initiate a calibration on a device.
UPLOAD_CALIB_DATA	Selection of appropriate calibration, updates the current, last and previous calibration dates performed on the device.
CAL_STATUS_1	The current status of the last performed calibration.
CAL_STATUS_2	The current status of the last performed calibration.
CURR_TIME_DATE	Represents the date and time of the last performed calibration. This parameter is to be used in conjunction with the Upload Cal History.
LAST_TIME_DATE	Represents the date and time of calibration performed prior to the current calibration.
PREV_TIME_DATE	Represents the date and time of calibration performed prior to the last calibration.
EXCESS_DELTA_LIMIT	The temperature limit above which Excess Delta alert is asserted.
TEMP_DELTA_VALUE	Absolute temperature difference between sensor 1 and sensor 2.
LIM_SW_VAL_1	The discrete value and status of Limit Switch 1. Off is represented by 0 and On by 1.
LIM_SW_PV_SRC_1	The temperature value which is used as the source for the Limit Switch 1. Possible values are: Average Value, Differential Value, Primary Value 1, Primary Value 2, Redundant Value, Secondary Value, Split Range Value
LIM_SW_SP_1	The setpoint for Limit Switch 1. The setpoint acts as the threshold for Limit Switch to switch to On/Off.

Parameter	Description
LIM_SW_DIR_1	The direction of action for Limit Switch1. Two types of actions are possible: HI_LIMIT and LO_LIMIT. In HI_LIMIT the limit switch turns On when the PV Source goes above the setpoint and for LO_LIMIT when it goes below the SET_POINT.
LIM_SW_HYST_1	Hysteresis value applied in Limit Switch 1 calculation.
LIM_SW_UNIT_1	The units for Limit Switch 1 parameters: LIM_SW_SP_1 and LIM_SW_HYST_1.
LIM_SW_VAL_2	The discrete value and status of Limit Switch 2. Off is represented by 0 and On by 1.
LIM_SW_PV_SRC_2	The temperature value which is used as the source for the Limit Switch 2. Possible values are: Average Value, Differential Value, Primary Value 1, Primary Value 2, Redundant Value, Secondary Value, Split Range Value
LIM_SW_SP_2	The setpoint for Limit Switch 2. The setpoint acts as the threshold for Limit Switch to switch to On/Off.
LIM_SW_DIR_2	The direction of action for Limit Switch2. Two types of actions are possible: HI_LIMIT and LO_LIMIT. In HI_LIMIT the limit switch turns On when the PV Source goes above the setpoint and for LO_LIMIT when it goes below the SET_POINT.
LIM_SW_HYST_2	Hysteresis value applied in Limit Switch 2 calculation.
LIM_SW_UNIT_2	The units for Limit Switch 2 parameters: LIM_SW_SP_2 and LIM_SW_HYST_2.
LIM_SW_VAL_3	The discrete value and status of Limit Switch 3. Off is represented by 0 and On by 1.
LIM_SW_PV_SRC_3	The temperature value which is used as the source for the Limit Switch 3. Possible values are: Average Value, Differential Value, Primary Value 1, Primary Value 2, Redundant Value, Secondary Value, Split Range Value
LIM_SW_SP_3	The setpoint for Limit Switch 3. The setpoint acts as the threshold for Limit Switch to switch to On/Off.
LIM_SW_DIR_3	The direction of action for Limit Switch 3. Two types of actions are possible: HI_LIMIT and LO_LIMIT. In HI_LIMIT the limit switch turns On when the PV Source goes above the setpoint and for LO_LIMIT when it goes below the SET_POINT.
LIM_SW_HYST_3	Hysteresis value applied in Limit Switch 3 calculation.
LIM_SW_UNIT_3	The units for Limit Switch 2 parameters: LIM_SW_SP_2 and LIM_SW_HYST_3.
LIM_SW_VAL_4	The discrete value and status of Limit Switch 4. Off is represented by 0 and On by 1.
LIM_SW_PV_SRC_4	The temperature value which is used as the source for the Limit Switch 4. Possible values are: Average Value, Differential Value, Primary Value 1, Primary Value 2, Redundant Value, Secondary Value, Split Range Value

Parameter	Description
LIM_SW_SP_4	The setpoint for Limit Switch 4. The setpoint acts as the threshold for Limit Switch to switch to On/Off.
LIM_SW_DIR_4	The direction of action for Limit Switch 4. Two types of actions are possible: HI_LIMIT and LO_LIMIT. In HI_LIMIT the limit switch turns On when the PV Source goes above the setpoint and for LO_LIMIT when it goes below the SET_POINT.
LIM_SW_HYST_4	Hysteresis value applied in Limit Switch 4 calculation.
LIM_SW_UNIT_4	The units for Limit Switch 2 parameters: LIM_SW_SP_2 and LIM_SW_HYST_4.
HON_RES_3	Honeywell Reserved Parameter
SENSOR_OPTS	Indicates if Callender van-Dusen Coefficient support for RTDs is available in the sensor firmware.
CVD_ENABLE_1	Used to enable or disable Callendar - van Dusen RTD coefficients for Sensor 1
CVD_ENABLE_2	Used to enable or disable Callendar - van Dusen RTD coefficients for Sensor 2
CVD_COEFF_R0.1	CVD coefficient for resistance at 0°C for Sensor 1
CVD_COEFF_ALPHA_1	Alpha CVD coefficient for Sensor 1
CVD_COEFF_DELTA_1	Delta CVD coefficient for Sensor 1
CVD_COEFF_BETA_1	Beta CVD coefficient for Sensor 1
CVD_COEFF_R0.2	CVD coefficient for resistance at 0°C for Sensor 2
CVD_COEFF_ALPHA_2	Alpha CVD coefficient for Sensor 2
CVD_COEFF_DELTA_2	Delta CVD coefficient for Sensor 2
CVD_COEFF_BETA_2	Beta CVD coefficient for Sensor 2
CVD_UCAL_HI_LIM_1	CVD Calibration high point for Sensor 1 (Ohms)
CVD_UCAL_LO_LIM_1	CVD Calibration low point for Sensor 1 (Ohms)
CVD_UCAL_HI_LIM_2	CVD Calibration high point for Sensor 2 (Ohms)
CVD_UCAL_LO_LIM_2	CVD Calibration low point for Sensor 2 (Ohms)

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS mode (Out of Service)
Alarm Types	The block supports standard block alarms (see section 3.2).

3.6 Diagnostic Transducer block

The Diagnostics Transducer block is used to monitor or track Process Variables (PV) of the device. The block supports several types of diagnostics: Process Variables (Primary Value 1, Primary Value 2, Cold Junction Temperature, Sensor Core Temperature, CT-CJ Delta), and Transmitter Electronics.

ATTENTION

The PVs and core temperature diagnostics are tracked in six ways:

- Maximum in lifetime
- Minimum in lifetime
- Accumulated time above a limit
- Accumulated time below a limit
- Time stamp of last transition above a limit
- Time stamp of last transition below a limit

Execution

The block has Sensor and Device diagnostics. The block is executed as follows:

Sensor

Sensor Diagnostics

Sensor Diagnostics has two components **Sensor Diagnostics** and **Sensor Voltage diagnostics**. These are updated based on the selection in the **Upload Track Data.**

The device updates the PV tracking data of sensor module and processes the rest of the diagnostic data such as electronic temperature, Time-in-Service, etc.

PV diagnostics and Sensor Core temperature diagnostics are displayed in the **Sensor Diagnostics** record based on the selection in the Upload Track Data. **Maximum Value** and **Minimum Value** are the maximum and minimum values reached during the life time of the device. **Process Variable Unit** is the engineering unit of the process variable currently under use. The **Over Range Counter** is the accumulation of minutes that device's PV (or Sensor Core temperature) has been above the value of maximum specification limit less 10% of range. (Example: for temperature ranging from -200 to 850 Deg C, 850 Deg C – 105 Deg C = 745 Deg C). The **Over Range Date** is the date and time when the PV (or Sensor Core temperature) last passed above the value of maximum specification limit less 10% of range. (Example: for temperature ranging from -200 to 850 Deg C, 850 Deg C – 105 Deg C = 745 Deg C). **Under Range Counter** is the accumulation of minutes that device's PV (or Sensor Core temperature) has been below the value of minimum specification limit plus 10% of range. (Example: for temperature ranging from -200 to 850°C, -200Deg C + 105Deg C = -95 Deg C). **Under Range Date** is the date and time when the PV (or Sensor Core temperature) temperature last passed below the value of minimum specification limit plus 10% of range. (Example: for temperature ranging from -200 to 850 Deg C, -200Deg C + 105Deg C = -95 Deg C).

The **Sensor Voltage Diagnostics** are also updated based on the selection in the Upload Track Data. **Max AVDD** and **Min AVDD** are the maximum and minimum values of the VDD recorded by the sensor module in its life time and **Max AVDD Time stamp** and **Min AVDD Time stamp** are the corresponding time stamps at maximum and minimum values.

Sensor General Diagnostics

ATTENTION

Note that the sensor stress monitor is different from the device stress monitor. The stress monitor of the sensor depends on other parameters such as sensor core temperature.

Stress monitor

It is the amount of time the device has been used under stressful conditions. For example, say temperature of the sensor. The stress monitor is based on the temperature of the sensor. The Stress monitor is calculated in percentage.

$$Stress\ monitor\ of\ sensor = \frac{Time\ spent\ in\ stressful\ condition}{Timing\ service\ of\ the\ device}$$

Time in Service

It is the amount of time the sensor is in operation and is shown in minutes.

Service Life

Service life indicates the amount of service life that has been consumed by the sensor. Service Life is dependent on the temperature of the sensor only. The Service life is calculated in percentage.

Device Diagnostics

Electronic Temperature Diagnostics

Similar to the sensor diagnostics, the **Electronic Temperature Diagnostics** are also tracked in 6 ways: **Max Electronic Temperature** and **Min Electronic Temperature** are the maximum and minimum electronics temperatures recorded by the communication module electronics in its life time and **Electronic Temperature Unit** is the engineering unit currently under use which is set in the temperature transducer basic block. **ET Over Range CTR** is the accumulation of minutes the device is above the maximum specification limit less 10% of range (for the range of -40 Deg C to 85 Deg C it is 85Deg C – 12.5Deg C = 72.5 Deg C) and **ET Over Range Date** is the date and time when the electronic temperature last passed above the maximum specification limit less 10% of the range. The **ET Under Range CTR** is the accumulation of minutes the device is below the minimum specification limit plus 10% of range (for the range of -40 Deg C to 85 Deg C it is -40Deg C + 12.5Deg C = -27.5 Deg C) and **ET Under Range Date** is the date and time when the electronic temperature last passed above the maximum specification limit less 10% of the range.

Sensor Detailed Status

SENSOR_DETAILED_STATUS parameter indicates the various status bits set by the sensor. Table 8 shows the various possible bits that could be set.

Table 8: Sensor Detailed Status

Critical Status	Non-Critical Status 1	Non-Critical Status 1
Input 1 Fault	Sensor Over Temperature	Sensor 1 Health Warning
Input 2 Fault	No Factory Calibration	Input 2 Over Range
Suspect Input Fault	Input 1 Under Range	Input 2 Under Range
Characterization/ Calibration Table CRC Fault	Input 1 Over Range	Excess LRV 2 Correct
NVM Fault	Excessive LRV 1 Correct	Excess Span 2 Correct
RAM Fault	Excess Span 1 Correct	User Correct Active
Flash CRC Fault	CJ Out of Range	CJ CT Delta Temp Warning
Flow Control Fault	Unused 1	Sensor 2 Health Warning
Miscellaneous Status 1	**Miscellaneous Status 2**	**Miscellaneous Status 3**
Supply Volts Fault	Break Detect Active	Input 2 TB8 Open
ADC 0 Range Fault	Match PV Active	Input 2 Open
ADC 1 Range Fault	Input 1 TB 6 Open	Characterization Table Fault
ADC Reference Fault	Input 1 Open	Calibration Table Fault
EE Update Active	Latching Active	Sensor in Production Mode
EE Updates Blocked	Input 1 TB 5 Open	Input 2 TB 9 Open
Communication Request Board Message CRC Fault	Unused 2	Input 1 TB 7 Open (RTD)
Low Current Mode	Unused 3	Input 2 TB 7 Open (RTD)

Time in Service

This is the amount of time the device is in operation and is displayed in minutes.

Service Life

This is the average service life of the device under ideal conditions is 27.3 years. But, the service life varies depending on external factors such as temperature. Service life indicates the amount of service life that has been used by the device. Service Life is dependent on the temperature of the device only. The Service life is calculated in percentage.

Stress monitor

This is the amount of time the device has been used under stressful conditions. For example, say temperature of the device. The stress monitor is based on the temperature of the device. The Stress monitor is calculated in percentage.

$$Stress\ monitor\ = \frac{Amount\ of\ time\ the\ device\ was\ under\ stressful\ conditions}{Time\ in\ service\ of\ the\ device}$$

Power Cycle Track

The power cycle track gives diagnostics related to the power up information of the device. The **Power Cycles** is the number of power ups experienced by the device after leaving factory. The **Last Power Up Cycle time** is the date and time of the last power up.

Operating Voltage Track

The statistics data for the supply voltage are tracked in the Operating voltage track. **Supply Voltage** is the current value of the voltage at the device input terminals. The status of the supply voltage whether it is normal or below operating value is indicated in **the Status of Current Voltage** parameter. **Minimum Voltage** is the value of the least voltage experienced by the device at the input terminals in its life time. **Last Minimum Voltage Time** is the date and time of the last minimum voltage experienced by the device. The **Minimum Voltage** can be reset by using the **Reset Minimum Voltage** parameter.

Parameter List

Table 9: Diagnostic Transducer block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	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.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for all configuration, hardware, and 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 may be reported without clearing the Active status, if the subcode has changed.
EL_TEMP_DIAGNOSTIC	Electronic Temperature Diagnostic parameters.
SENSOR_DIAGNOSTICS	Sensor Diagnostics parameters.
POWER_TRAC	Power Up Track Data.
OP_VOLTAGE	Operating Voltage.
TIME_IN_SERVICE	Summation of time in minutes that power has been applied to the device since leaving the factory.
SERVICE_LIFE	It is the elapsed Service life of device in percentage.
STRESS_MONITOR	It monitors various diagnostic parameters which are then input into an algorithm to calculate an estimated percent of time that the transmitter has spent in stressful conditions.
SENS_VOLT_DIAGNOSTICS	Sensor Voltage Diagnostics.
SENS_GEN_DIAGNOSTICS	Sensor General Diagnostics.
UPLOAD_TRACK_DATA	Process Variable selection for uploading the Track data from sensor device.
HON_RES_1	Reserved for Honeywell use only.
HON_RES_2	Reserved for Honeywell use only.
CJ_CT_DELTA	The difference between Sensor Core Temperature and Cold Junction Temperature.
SENSOR_DETAILED_STATUS	Six bytes whose constituent bits represent the various status conditions set by the Sensor.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms (see section 3.2).

3.7 LCD Transducer block

The LCD Transducer block supports Basic and Advanced Display. The block is used to configure the basic or advanced display connected to the STT850 transmitter. The block stores the LCD configurations, and sends these values to the Display while the transmitter is powered up or restarted. The STT850 device supports upto eight LCD screens.

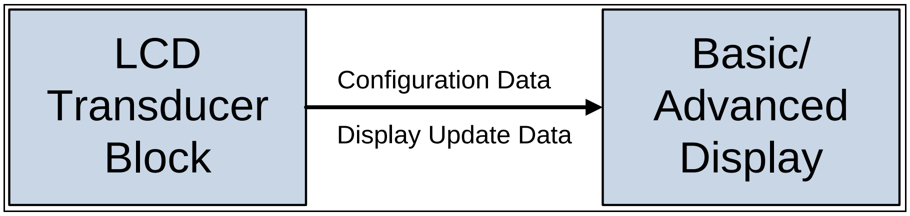


Figure 6: LCD Transducer Block

ATTENTION

The initial configuration of LCD transmitter is configured to show four screens with Primary Value 1, Primary Value 2, Secondary Value, and Electronics temperature.

The Display shows the available set of process variables, and all function block inputs/outputs. In addition, the block reports the current device status and errors. If a function block parameter which is not currently a part of the control strategy is selected, an error appears in the display.

Execution

Basic Display

The PV value is user-configurable. This field has 8 characters. The maximum allowable numeric value is 9999999 or -999999. If fractional decimals are configured, the fractional positions are dropped, as desired. If the PV value exceeds the above limits, it is divided by 1000 and “K” is appended to the result, allowing a maximum value with multiplier of 999999K or -99999K. This field is user-configurable. This field has 8 characters.

Advanced Display

The Advanced Display provides three formats, and describes the field in each of the three Advanced Display formats namely, PV, Bar Graph, and PV Trend. Essentially, all three formats provide the same information, but with the following differences:

- **PV**

  It is a user configurable display, and it shows the configured PV.

- **Bar Graph**

  It is a user configurable 126 segment Bar Graph with range settings. The Bar Graph displays the current value of the configured PV.

- **PV Trend**

  It is a user-configurable display period from one hour to 24 hours. The chart displays minimum, maximum, and average of the configured PV over the selected trend period.

The LCD Transducer block supports configuration of up to eight LCD screens on the Advanced and Basic displays. By default, the Display has a screen configured with default settings. The Basic Display does not support advanced features such as Transmitter Messaging, Bar Graph, and Trends, while the Advanced Display supports all these advanced features.

Transmitter Messaging

The transmitter messaging is a feature that allows message typed through host up to 64 alphanumeric characters) which is sent to the Local Display of the transmitter. The message is shown on the Display interspersed with the configured screens.

Clear Message

To stop displaying the message, select the Clear Message method. After selecting this option, the device clears the entered Message and it is not shown in the Display.

Table 10 lists the allowed parameters that can be configured using the LCD block.

Table 10 LCD parameters

Block	FF Parameter
TEMPTB	PRIMARY_VALUE_1
	PRIMARY_VALUE_2(Dual Input only)
	SECONDARY_VALUE
	EL_TEMPERATURE
	DIFFERENTIAL_VALUE (Dual Input only)
	AVERAGE_VALUE (Dual Input only)
	REDUNDANT_VALUE (DUAL Input only)
	SPLIT_RANGE_TEMP_VALUE (Dual Input only)
	TEMP_DELTA_VALUE(DUAL Input only)
	SENS_RESISTANCE_1
	SENS_RESISTANCE_2(DUAL Input only)
ANALOG INPUT BLOCK	PV
	OUT
	FIELD_VAL
ARITH	IN
	IN_LO
	IN_1
	IN_2
	IN_3
ISEL	OUT
	IN_1
	IN_2
	IN_3
	IN_4
PID BLOCK (PID)	SP
	PV
	OUT

Block	FF Parameter
	IN
	CAS_IN
	BKCAL_IN
	BKCAL_OUT
	RCAS_IN
	ROUT_IN
	RCAS_OUT
	ROUT_OUT
	FF_VAL
	TRK_VAL
SIGNAL CHARACTERIZER BLOCK	OUT_1
	OUT_2
	IN_1
	IN_2
OUTPUT SPLITTER BLOCK	CAS_IN
	BKCAL_IN_1
	BKCAL_IN_2
	BKCAL_OUT
	OUT_1
	OUT_2
DISCRETE INPUT BLOCK	PV_D
	FIELD_VAL_D
	OUT_D

Parameters List

Table 11: LCD Transducer block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	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.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for all configuration, hardware, and 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 may be reported without clearing the Active status, if the subcode has changed.
DISP_SEQ_TIME	Periodic rotation time of the display screens in seconds. Range 3-30 sec.
LANGUAGE	Language selection for the Display. Supported Languages: English, French, German, Spanish, Turkish, Italian, Chinese, Japanese and Russian.
DISPLAY_TYPE	Type of Display Connected. Possible Values: No Display Connected, Basic Display, Advanced Display.
LCD_CONTRAST	Contrast of the LCD screen can be controlled by this parameter. Its range is 1-9.
DISP_FW_VER	Version Number of Display Firmware.
BLOCK_TYPE	Block type selection for screen process variable. The BLOCK_TYPE is present in all the eight screens: BLOCK_TYPE_1, BLOCK_TYPE_2, BLOCK_TYPE_3, BLOCK_TYPE_4, BLOCK_TYPE_5, BLOCK_TYPE_6, BLOCK_TYPE_7 and, BLOCK_TYPE_8.
PARAM_INDEX	Parameter selection for screen process variable. Parameters need to be chosen based on Block type. The PARAM_INDEX is present in all the eight screens: PARAM_INDEX_1, PARAM_INDEX_2, PARAM_INDEX_3, PARAM_INDEX_4, PARAM_INDEX_5, PARAM_INDEX_6, PARAM_INDEX_7 and, PARAM_INDEX_8.

Parameter	Description
UNIT_TYPES	Unit selection for screen process variable. Appropriate units need to be selected based on the configured parameter. If desired units are not present, 'custom' may be selected. The UNIT_TYPES is present in all the eight screens: UNIT_TYPES_1, UNIT_TYPES_2, UNIT_TYPES_3, UNIT_TYPES_4, UNIT_TYPES_5, UNIT_TYPES_6, UNIT_TYPES_7 and, UNIT_TYPES_8.
CUSTOM_UNIT	Character string to represent custom units. This value is used when Unit type of 'custom' is selected. Size: 8 Characters. The CUSTOM_UNIT is present in all the eight screens: CUSTOM_UNIT_1, CUSTOM_UNIT_2, CUSTOM_UNIT_3, CUSTOM_UNIT_4, CUSTOM_UNIT_5, CUSTOM_UNIT_6, CUSTOM_UNIT_7 and CUSTOM_UNIT_7.
CUSTOM_TAG	Tag to be displayed for the screen. Length: 14 Characters. The CUSTOM_TAG is present in all the eight screens: CUSTOM_TAG_1, CUSTOM_TAG_2, CUSTOM_TAG_3, CUSTOM_TAG_4, CUSTOM_TAG_5, CUSTOM_TAG_6, CUSTOM_TAG_7 and CUSTOM_TAG_8.
DISPLAY_TEMPLATE	Represents the display screen template. Possible Values: a) PV : PV value is displayed b) PV and Trend : PV value followed by a Trend is shown on the display c) PV and Bargraph : PV value followed by a Bargraph is shown on the display d) None: Screen will not be seen. The DISPLAY_TEMPLATE is present in all the eight screens: DISPLAY_TEMPLATE_1, DISPLAY_TEMPLATE_2, DISPLAY_TEMPLATE_3, DISPLAY_TEMPLATE_4, DISPLAY_TEMPLATE_5, DISPLAY_TEMPLATE_6, DISPLAY_TEMPLATE_7 and DISPLAY_TEMPLATE_8.
DECIMALS	Number of digits to display after the decimal point. Range: 0 - 3. DECIMALS are present in all the eight screens: DECIMALS_1, DECIMALS_2, DECIMALS_3, DECIMALS_4, DECIMALS_5, DECIMALS_6, DECIMALS_7 and DECIMALS_8.
PV_LOLIM	Display Low Limit (Trend, Bar, Custom PV scaling, usually equal to LRV). The PV_LOLIM is present in all the eight screens: PV_LOLIM_1, PV_LOLIM_2, PV_LOLIM_3, PV_LOLIM_4, PV_LOLIM_5, PV_LOLIM_6, PV_LOLIM_7 and PV_LOLIM_8.
PV_HILIM	Display High Limit (Trend, Bar, Custom PV scaling, usually equal to URV). The PV_HILIM is present in all the eight screens: PV_HILIM_1, PV_HILIM_2, PV_HILIM_3, PV_HILIM_4, PV_HILIM_5, PV_HILIM_6, PV_HILIM_7 and PV_HILIM_8.
TREND_DURATION	Duration of a trend screen in hours. Its valid range is 1-999. The TREND_DURATION is present in all the eight screens: TREND_DURATION_1, TREND_DURATION_2, TREND_DURATION_3, TREND_DURATION_4, TREND_DURATION_5, TREND_DURATION_6, TREND_DURATION_7 and TREND_DURATION_8.

Parameter	Description
DISPLAY_MESSAGE	A message with a maximum of 64 characters that appears on the advanced display of the transmitter.
PREF_UNITS	These Units are used in the displays to show Temperature related parameters in Menus such as, LRL, URL, and MRV.
ROTATE_ENABLE	Parameter to Enable or Disable screen rotation.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms (see section 3.2).

3.8 Analog Input block

The Analog Input (AI) block takes the transducer's input data, selected by channel number, and makes it available to other function blocks at its output. The variables to be used by the block are defined through the available channels: Primary Value 1, Primary Value 2, Cold Junction Temperature, Electronics Housing Temperature, Differential Temperature, Average Temperature, Redundant Temperature and Split Range Temperature.

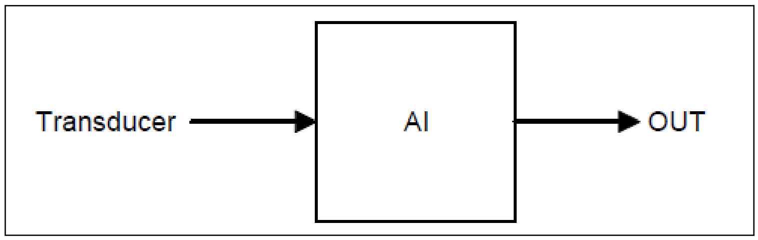


Figure 7: Analog Input Block

Execution

Transmitter Output Signal and Status

Viewing certain parameters, their values and status in the transmitter and understanding their relationship to each other are helpful in understanding transmitter output signal and status. The following paragraphs and tables describe transducer and AI block parameters which directly determine the way the transmitter output is presented.

Temperature Sensor Signal

In Transducer block, the temperature signal is represented as **PRIMARY_VALUE_1** or **PRIMARY_VALUE_2** or as one of the calculated temperature values. This temperature signal uses the elements in **PRIMARY_VALUE_RANGE_1** or **PRIMARY_VALUE_RANGE_2** or one of the calculated temperature value range to determine the engineering units, the decimal places for the display and also the high and low scale of the value. This temperature signal becomes the PV value in the AI block, and uses the elements of **OUT_SCALE** in determining the units, decimal places and also the high and low scale values of PV. The temperature signal leaves the AI block as **OUT** value, which also uses the elements of **OUT_SCALE**.

The Transducer scaling (**XD_SCALE**) is applied to the value from the channel to produce the **FIELD_VAL** in percent. The **XD_SCALE** unit's code must match the channel unit's code or be supported by the device if this is not the case the block remains in OOS mode, after being configured.

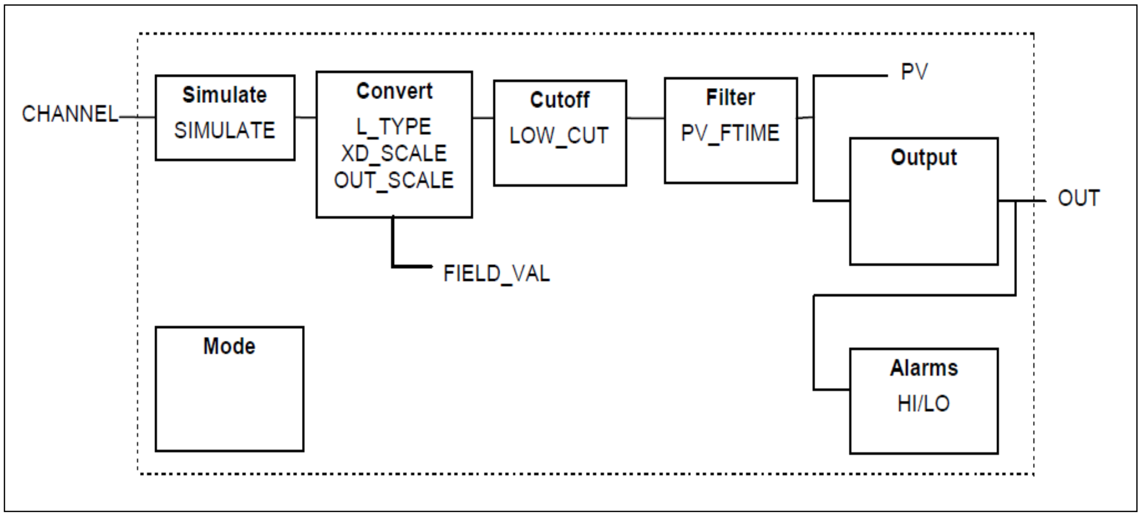


Figure 8: Analog Input Block Schematic Diagram

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 block places the value in **OUT** if the mode is AUTO. If MAN mode is allowed, 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 cut-off**" option in the **IO_OPTS** bit string. If the option bit is set as True, any calculated output below the low cut-off value changes 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*(channel value - EU@0%) / (EU@100% - EU@0%) [XD_SCALE]

Direct: PV = channel value

Indirect: PV = (FIELD_VAL/100) * (EU@100% - EU@0%) + EU@0% [OUT_SCALE]

Ind Sqr Root: PV = sqrt (FIELD_VAL/100) * (EU@100% - EU@0%) + EU@0% [OUT_SCALE]

XD_SCALE Range

In the AI block, **XD_SCALE** values are used when **L_TYPE** is set to Indirect which converts the signal to other units. The high and low scale values of **XD_SCALE** (**EU_100** and **EU_0**) define the range over which the **AI OUT** shows the status as Good.

- When **L_TYPE** is set to either Indirect or Direct, **XD_SCALE** units must match the transducer units.
- When **L_TYPE** is set to Direct, it is recommended that **XD_SCALE** and **OUT_SCALE** must contain the same values.

PV Value

The AI block PV value is determined based on the selected transducer channel's **PRIMARY_VALUE**.

AI OUT

AI in Manual Mode

When the AI block is in manual mode, **OUT** can be written as a fixed value between **-**10% and +110% of the **OUT_SCALE** range. **OUT** values between 0 and 100% shows a status of Good. **OUT** values outside the range shows a status of Uncertain. The "limit" field is marked as Constant for all values. PV shows the live temperature signal in manual mode.

AI in AUTO Mode

L_TYPE determines whether the signal is taken directly from the transducer block and passed to the AI block output (**L_TYPE** = Direct) or converted into different units before it is passed to the AI block output (**L_TYPE** = Indirect or Ind Sqr Root). **OUT_SCALE** determines the units' conversion of the signal presented to the output.

- When **L_TYPE** equals Direct, **OUT** is the same as the value passed from the transducer block.
- When **L_TYPE** is Indirect, the **PRIMARY_VALUE** is converted to **XD_SCALE** and that value is set equal to **OUT** (**FIELD_VAL** = %). The **OUT** in % is re-ranged to a value using the **OUT_SCALE**.

OUT status

The following table provides the resulting status of AI block **OUT** for a given status of **PRIMARY_VALUE** in the transducer block.

If . . .	Then . . .
PRIMARY_VALUE status = Good::[alarm status]:Not Limited	**OUT** value is tested against **OUT_SCALE** range values: If **OUT** value is within the **OUT_SCALE** range, then **OUT** status = Good Non Cascade::[alarm status]:Not Limited If **OUT** exceeds **OUT_SCALE** range, then **OUT** status = Uncertain:: Engineering Units Range Violation:& High or Low Limited
PRIMARY_VALUE status = Uncertain	**OUT** status = Uncertain
2nd field in the **PRIMARY_VALUE** status = Non Specific	**OUT** status = Non Specific
PRIMARY_VALUE status = High or Low	**OUT** status = High or Low

Parameters List

Table 12: Analog Input block parameters

Parameter	Description
ST_REV	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.
TAG_DESC	The user description of the application of the block.
STRATEGY	It is used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, and so on.
MODE_BLK	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
BLOCK_ERR	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.
PV	The process variable used in block execution.
OUT	The block output value and status.
SIMULATE	A group of data that contains the current transducer value and status, the simulated transducer value and status, and the enable/disable bit.
XD_SCALE	Elements used to display the value obtained from the transducer block. The elements are: • High and low scale values (EU_100 and EU_0). • Engineering units to display the value (UNITS_INDEX). • Decimal places to display the value (DECIMAL).
OUT_SCALE	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT.
GRANT_DENY	Normally, the operator has permission to write to parameter values, but Program or Local remove that permission and give it to the host controller or a local control panel.
IO_OPTS	Allows the selection of input/output options used to alter the PV. Low cutoff enabled is the only selectable option.
STATUS_OPTS	Helps select options for status handling and processing. The supported status options for the AI block are Propagate Fault Forward Uncertain, if Limited Bad, if Limited and Uncertain if MAN mode.
CHANNEL	The CHANNEL value is used to select the measurement value. Configure the CHANNEL parameter before configuring the XD_SCALE parameter.

Parameter	Description
L_TYPE	The state (Direct or Indirect) values that are passed from the transducer block to the AI block. When L_TYPE = Direct, the values are passed directly from the transducer block to the AI block. (No units conversion.) When L_TYPE = Indirect, values from the transducer block are in different units, and must be converted either linearly (Indirect) or in square root (Ind Sqr Root) using the range defined by the transducer and the OUT_SCALE range.
LOW_CUT	If percentage value of transducer input fails below this, PV = 0.
PV_FTIME	The time constant of the first-order PV filter. It is the time required for a 63% change in the IN value.
FIELD_VAL	The value and status from the transducer block or from the simulated input when simulation is enabled.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The block alarm is used for all configuration, hardware, and 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 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.
ALARM_SUM	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 sets 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.
ACK_OPTION	Used to set AUTO acknowledgment of alarms.
ALARM_HYS	The amount the alarm value must return within the alarm limit before the associated active alarm condition clears.
HI_HI_PRI	The priority of the HI HI alarm.
HI_HI_LIM	The setting for the alarm limit used to detect the HI HI alarm condition.
HI_PRI	The priority of the HI alarm.
HI_LIM	The setting for the alarm limit used to detect the HI alarm condition.
LO_PRI	The priority of the LO alarm.
LO_LIM	The setting for the alarm limit used to detect the LO alarm condition.
LO_LO_PRI	The priority of the LO LO alarm.
LO_LO_LIM	The setting for the alarm limit used to detect the LO LO alarm condition.
HI_HI_ALM	The HI HI alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.
HI_ALM	The HI alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.
LO_ALM	The LO alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.

Parameter	Description
LO_LO_ALM	The LO LO alarm data, which includes a value of the alarm, a timestamp of occurrence and the state of the alarm.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms (see section 3.2). Additionally it supports, standard **HI_HI, HI, LO**, and **LO_LO** alarms applied to **OUT**.
Status Handling	**Uncertain** - **EU Range Violation** status is always set if the **OUT** value exceeds the **OUT_SCALE** range and no worse condition exists. The following options from **STATUS_OPTS** apply, where Limited refers to the sensor limits: • **Propagate Fault Forward** If the status from the sensor is Bad, Device failure or Bad, Sensor failure, propagate it to **OUT** without generating an alarm. The use of these sub-status in **OUT** is determined by this option. Through this option, the user may determine whether alarming (sending of an alert) is done by the block or propagated downstream for alarming. • **Uncertain, if Limited** Set the output status of the Analog Input block to uncertain if the measured or calculated value is limited. • **Bad if Limited** Set the output status to Bad if the sensor is violating a high or low limit. • **Uncertain if MAN Mode** Set the output status of the Analog Input block to uncertain if the actual mode of the block is MAN.

3.9 Proportional Integral Derivative (PID) block with auto tune

The PID block is the key to many control schemes, and it is commonly used. The PID function integrates the errors. If there is difference in process time constants of a primary process and secondary process measurement, then the block can be cascaded if required. Auto tuning is a feature that tunes the PID constants as per the process automatically.

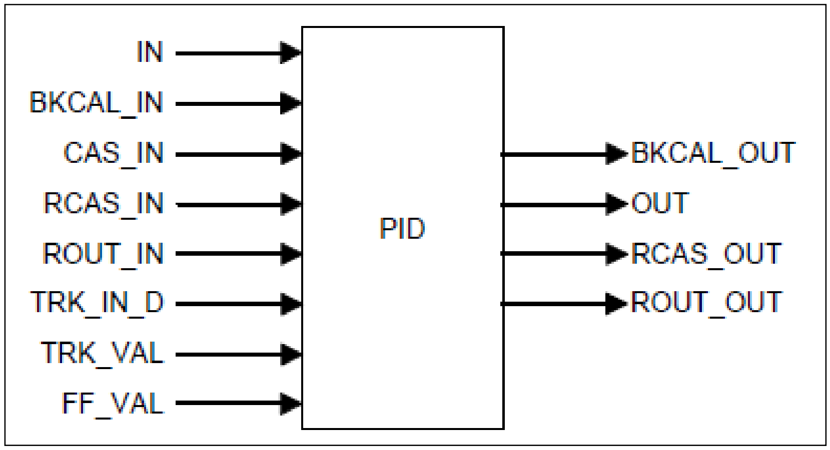


Figure 9: PID Block

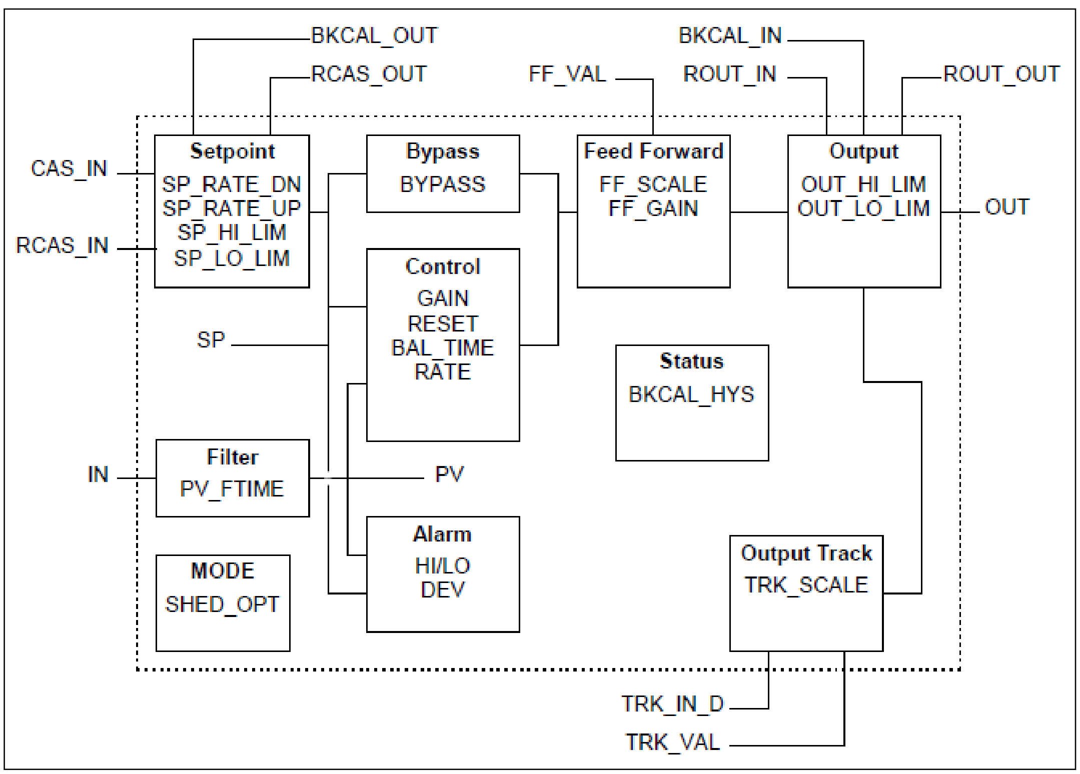


Figure 10: PID Block Schematic Diagram

Execution

The Process Variable to be controlled is connected to the **IN** input. The value is passed through a filter, and its 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** input is constant, or if further control action based on the PID error proceeds **IN** input further towards its active status limit. A full PV and DV alarm sub-function is provided. The PV has a status, although it is a contained parameter. This status is a copy of **IN**'s status, unless **IN** is Good and there is a PV or block alarm.

The full cascade **SP** sub-function is used with rate and absolute limits. The block has additional control options which cause the **SP** value to track the PV value. The **SP** value tracks the PV value while the block is in Actual mode of IMan, LO, or ROut, or when the target mode of the block is MAN.

The block provides a switch for **BYPASS**, which is available to the operator if the Bypass Enable control option is set as True. **BYPASS** can be used in secondary cascade controllers that have a Bad PV. The **BYPASS Enable** option is required, so if **BYPASS** is set as True, not all cascade control schemes are stable. **BYPASS** can only be changed when the block mode is in MAN or OOS mode. When **BYPASS** 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, on the transition out of BYPASS, the upstream block is requested to initialize to the PV value, irrespective of the "**Use PV for BKCAL_OUT**" option.

GAIN, **RESET**, and **RATE** are the tuning constants for the **P**, **I**, and **D** terms, respectively. The block provides 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 must be able to display the user's preference.

BAL_TIME parameter can be used to set the rate at which the **I** term moves towards balancing the difference between the previous integral term and the limited output. The **Direct Acting** control option, if set as True, causes the output to increase when the PV exceeds the **SP**. If set as False, the output decreases when the PV exceeds the **SP**. The **Direct Acting** control option must be set carefully, as it can cause a difference between positive and negative feedback.

ATTENTION

The **Direct Acting** control option can never be changed while in AUTO 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 control 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 as this prevents bumping the output. When the status returns to Good, the block adjusts its integral term to maintain the previous output. The output supports the track algorithm. The block provides an option to use either the **SP** value after limiting or the PV value for the **BKCAL_OUT** value.

PID Control block is an algorithm that produces an output signal in response to the measured variable and the setpoint. The PID block allows you to choose either a standard PID control equation (Ideal) or a robust PID equation defined by Honeywell. This selection is defined in the PID_FORM parameter.

The output has three terms, namely Proportional, Integral, and Derivative. The output is adjusted by tuning constants. There are three tuning constants in the ideal PID equation. The robust PID uses four tuning constants:

1. **GAIN** is the tuning constant of the Proportional term.
2. **RESET** is the tuning constant of the Integral.
3. **RATE** is the tuning constant of the Derivative. **RATE** is usually modified by a lag, which is set at some fixed ratio higher than the rate time, to create a rate gain. There is no lag with the rate in this implementation.
4. **OUT_LAG** is the fourth tuning constant used in the robust PID; it adds roll off to the output response. The action is similar to PID with rate gain.

PID Ideal and PID Robust

The ideal equation is a parallel or non-interacting implementation of PID control using three tuning constants. It automatically fixes **OUT_LAG** to 16 times the **RATE** time constant. This produces response characteristics equivalent to the algorithms used in TPS products.

The robust equation is the same parallel implementation of ideal PID control but allows the engineer to set the **OUT_LAG** and effectively change the rate gain.

ALGO_TYPE is a configuration parameter that contains one of three selected algorithm types, A, B, or C.

Where:

- A **-** **RATE**, **GAIN** and **RESET** all act on the error between setpoint and measured variable.
- B **-** **RATE** acts on the measured variable only, **GAIN** and **RESET** use the error.
- C **-** **RATE** and **GAIN** act on the measured variable only, and **RESET** uses the error.

PID Tuning Parameters

Table 13 lists the valid ranges for the tuning parameters for the PID block. Note that **OUT_LAG** parameter is not configurable when Ideal PID is selected (**PID_FORM** = 1) and can be configured when Robust PID is selected (**PID_FORM** = 2).

The values given for these tuning parameters are valid under the following conditions:

- The values assume that the minimum configurable PID function block execution period (T_s) is 0.125 seconds.
- Algorithm typesetting (A, B, or C) has no effect on the validation of these tuning parameters.

The PID function block rejects all values outside the following ranges:

Table 13: PID Tuning parameters

Parameter	Initial Value	Minimum Value	Maximum Value	Comment
PV_FTIME	0	0	200	Units: seconds.
GAIN	0	.004	250	
GAIN_NLIN	0	.004	250	
RATE (sec.)	0	32 • T_s	7500	The value of ZERO is permitted to turn off rate action.
RESET (sec.)	+INF	2 • T_s	7500	The value of +INF is permitted to turn off reset action. (Some versions of NI configurator program cannot set +/-INF).
OUT_LAG Ideal PID	N/A	N/A	N/A	Fixed for Ideal PID form - not configurable.
Robust PID	0	2 • T_s	7500	Zero permitted which implies no output lag.
BAL_TIME	0	N/A	N/A	Not used in Honeywell Implementation.

Auto tuning

Cycle tuning

The PID block supports the Cycle tuning algorithm. In Cycle tuning, the tuning parameter values are derived from the process response to the resultant action of causing the PV to oscillate about a **SP** value. The tuning method uses the measured ultimate gain and period to produce tuning parameter values, by using the relationship developed by Ziegler Nichols equations. Cycle tuning does not distinguish between process lags and always results in gain based on PV amplitude, and calculates the values of Reset and Rate based on time of the **SP** crossings using a fixed ratio of 4 to 1. Initially, this method does not require a stable process. Cycle tuning is applicable to Three Position Step control, and is used for integrating process.

Auto tuning procedure

There are nine parameters applicable for auto tuning: **AT_TYPE**, **TUNING_CRITERIA, TUNE_REQ, ATI, AT_MODE, AT_ERR, AT_GAIN, AT_RESET, and AT_RATE.**

AT_Type

There are two types of selections, namely Disable and Cycle Tune. When Disable is selected, **AT_MODE** becomes inactive. When Cycle Tune is selected, **AT_MODE** becomes AT Ready.

TUNING_CRITERIA

There are two types of tuning criteria available for selection: Normal and Fast.

- **NORMAL -** Conservative tuning designed to reduce overshoot as compared to FAST.
- **FAST -** Aggressive tuning designed to provide quarter-dampened response.

TUNE_REQ

TUNE_REQ can be turned ON only in the following modes, namely AUTO, CAS, RCAS, and ROUT. The ATI value becomes 1, and **AT_ERROR** shows the status as Run, this shows that auto tuning is in progress.

If **AT_ERROR** shows **OK**, auto tuning is successful. **AT_GAIN**, **AT_REST**, **AT_RATE** gets updated automatically and same values are copied to **GAIN**, **RESET** and **RATE** respectively.

Parameter list

Table 14: PID block parameters

Parameter	Description
ST_REV	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.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
MODE_BLK	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
BLOCK_ERR	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.
PV	The process variable used in block execution.
SP	It is the target block setpoint value. It is the result of setpoint limiting and setpoint rate of change limiting.
OUT	The block input value and status.
PV_SCALE	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with PV.
OUT_SCALE	The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with OUT.
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.
CONTROL_OPTS	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, Direct Acting, SP Track retain, SP-PV Track Out, Restrict SP to limits in CAS and RCAS, No output limits in MAN.
STATUS_OPTS	It helps to select options for status handling and processing. The supported status option for the PID block is Target to Manual if Bad IN. IFS if Bad IN, IFS if Bad CAS_IN, Use Uncertain as Good, Target to next permitted mode if Bad CAS_IN, Target to MAN if Bad TRK_IN_D and IFS if Bad TRK_IN_D.
IN	The connection for the PV input from another block.
PV_FTIME	The time constant of the first-order PV filter. It is the time required for a 63 percent change in the IN value.

Parameter	Description
BYPASS	Used to override the calculation of the block. When enabled, the SP is sent directly to the output.
CAS_IN	The remote setpoint value from another block.
SP_RATE_DN	Ramp rate for downward SP changes. When the ramp rate is set to zero, the SP is used immediately.
SP-RATE_UP	Ramp rate for upward SP changes. When the ramp rate is set to zero, the SP is used immediately.
SP_HI_LIM	The highest SP value allowed.
SP_LO_LIM	The lowest SP value allowed.
GAIN	The proportional gain value. This value cannot = 0.
RESET	The integral action time constant.
BAL_TIME	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 moves to obtain balance when the output is limited and the mode is AUTO, CAS, or RCAS.
RATE	The derivative action time constant.
BKCAL_IN	The analog input value and status from another block's BKCAL_OUT output that is used for backward output tracking for bump less transfer and to pass limit status.
OUT_HI_LIM	The maximum output value allowed.
OUT-LO_LIM	The minimum output value allowed
BKCAL_HYS	The amount the output value must change away from its output limit before limit status is turned off.
BKCAL_OUT	The value and status required by the BKCAL_IN input of another block to prevent reset windup and to provide bump less transfer of closed loop control.
RCAS_IN	Target setpoint and status that is provided by a supervisory host. Used when mode is RCAS.
ROUT_IN	Target output and status that is provided by a supervisory host. Used when mode is ROUT.
SHED_OPT	Defines action to be taken on remote control device timeout.
RCAS_OUT	Block setpoint and status after ramping, filtering, and limiting that are 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.
ROUT_OUT	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.
TRK_SCALE	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_IN_D	Discrete input that initiates external tracking.

Parameter	Description
TRK_VAL	The value (after scaling from TRK_SCALE to OUT_SCALE) APPLIED to OUT in LO mode.
FF_VAL	The feedforward control input value and status.
FF_SCALE	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_GAIN	The feedforward gain value. FF_VAL is multiplied by FF_GAIN before it is added to the calculated control output.
UPDATE_EVT	This alert is generated by any changes to the static data.
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 sets 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.
ALARM_SUM	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 sets 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.
ACK_OPTION	Used to set auto acknowledgment of alarms.
ALARM_HYS	The amount the alarm value must return to within the alarm limit before the associated active alarm condition clears.
HI_HI_PRI	The priority of the HI HI Alarm.
HI_HI_LIM	The setting for the alarm limit used to detect the HI HI alarm condition.
HI_PRI	The priority of the HI alarm.
HI_LIM	The setting for the alarm limit used to detect the HI alarm condition.
LO_PRI	The priority of the LO alarm.
LO_LIM	The setting for the alarm limit used to detect the LO alarm condition.
LO_LO_PRI	The priority of the LO LO alarm.
LO_LO_LIM	The setting for the alarm limit used to detect the LO LO alarm condition.
DV_HI_PRI	The priority of the deviation high alarm.
DV_HI_LIM	The setting for the alarm limit used to detect the deviation high alarm condition.
DV_LO_PRI	The priority of the deviation low alarm.
DV_LO_LIM	The setting for the alarm limit use to detect the deviation low alarm condition.

Parameter	Description
HI_HI_ALM	The HI HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
HI_ALM	The HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
LO_ALM	The LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
LO_LO_ALM	The LO LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
DV_HI_ALM	The DV HI alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
DV_LO_ALM	The DV LO alarm data, which includes a value of the alarm, a timestamp of occurrence, and the state of the alarm.
PID_FORM	Configuration parameter specifies the IDEAL or ROBUST PID equation to be used: • IDEAL PID (default): Non-interactive form of a three mode control equation that provides Proportional, Integral and Derivative control action. Linear and non-linear gain parameters are available. • ROBUST PID: The same as Ideal PID. Additionally, the equation supports a user-configurable lag filter applied to calculated output value. (See OUT_LAG parameter.) Linear and non-linear gain parameters are available.
ALGO_TYPE	Configuration parameter specifies algorithm type which can be A, B, or C: • Type “A” equation where Proportional, Integral and Derivative act on ERROR. • Type “B” equation where Proportional and Integral act on ERROR and Derivative acts on PV. • Type “C” equation where Integral acts on ERROR and Proportional and Derivative act on PV.
OUT_LAG	Time constant of single exponential LAG filter applied to the OUT parameter (primary output). Units (in seconds). For Ideal PID equation the lag filter is fixed at 1/16 and is not configurable.
GAIN_NLIN	Dimensionless gain factor. When the gain factor is multiplied by absolute value of the error and added to the linear GAIN, the result is a gain response which is proportional to the deviation. The default value is zero resulting in no response due to non-linear gain action.
GAIN_COMP	The composite gain quantity including both linear and non-linear gain parameters. It is a read only parameter.
ERROR_ABS	Absolute value of the difference between PV and working setpoint. Read only parameter.
WSP	Working setpoint. This is the setpoint value after absolute and rate limits have been applied. Deviation alarms are computed on this value. It is a read only parameter.
BLOCK_TEST	Test parameter to determine if the block is functioning correctly.

Parameter	Description
AT_TYPE	Auto Tune Selection supports two types: Disable, Cycle Tune.
TUNING_CRITERIA	Tuning Criteria supports two types: Normal, Fast.
TUNE_REQ	Tuning Request performs auto tuning process.
ATI	Auto Tune Indicator indicates Auto tune ON/OFF.
AT_MODE	Auto Tune Mode supports two options: AT Ready, Inactive • AT Ready indicates block is ready for auto tune • Inactive indicates auto tuning is disabled.
AT_ERROR	Auto Tune Error supports the following errors: Abort, Not ready, OK, and Run.
AT_GAIN	Auto tuned Gain.
AT_RESET	Auto tuned Reset.
AT_RATE	Auto tuned Rate.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS (Out of Service) • IMan • Cas • RCas • ROut • LO
Alarm Types	The block supports standard block alarms (see section 3.2), in addition to it standard **HI_HI, HI, DV_HI, DV_LO, LO**, and **LO_LO** alarms applied to PV.
Status Handling	Standard, in addition to the following things for the control selector. If Not selected is received at **BKCAL_IN**, the PID algorithm must make necessary adjustments to prevent windup.

3.10 Input Selector block

The Input Selector block performs maximum, minimum, middle, average and ‘first good’ input selection. The Input Selector block provides selection of up to four inputs and generates an output based on the selected type of input. The block normally receives its inputs from AI blocks, and provides a combination of parameter configuration options. The block functions as a rotary position switch, or a validated priority selection based on the use of the **first good** parameter and the **disable_n** parameter. As a switch, the block receives switching information from either the connected inputs or from an operator input. The block supports signal status propagation.

The block is used to provide control input selection in the forward path only, and hence no back calculation support is provided. **SELECTED** indicates which input has been selected or the number of inputs selected by the algorithm. The block does not support process alarms.

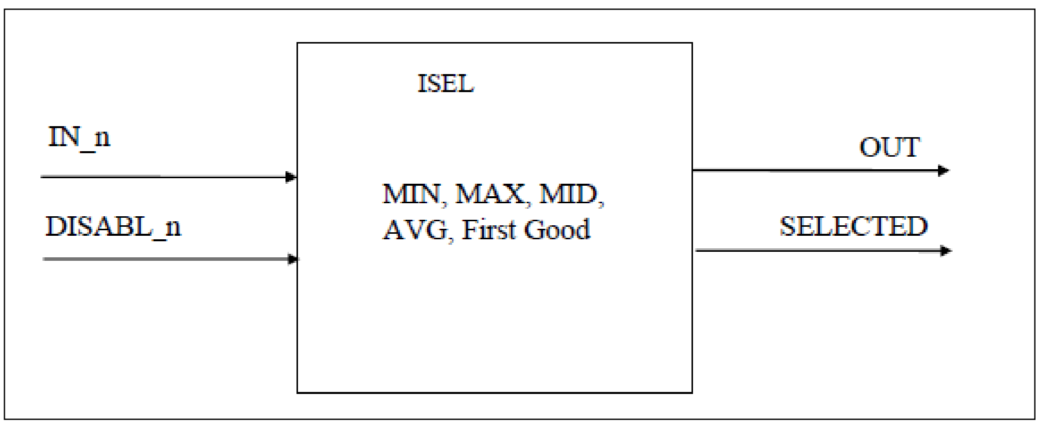


Figure 11: Input Selector Block

Execution

Input processing

If **DISABLE_n** is True, the corresponding input **IN_n** is discarded. If there are no inputs left, or if there are inputs fewer than **MIN_GOOD** inputs, then the value of **SELECTED** becomes zero.

Selection Processing

- If **OP_SELECT** is non-zero, the **OP_SELECT** value determines the selected input, irrespective of the **SELECT_TYPE** selection. The value of **SELECTED** is the number of the input used.
- If **SELECT_TYPE** is 'First Good', it transfers the value of the first remaining input to the output of the block. The value of **SELECTED** is the number of the input used.

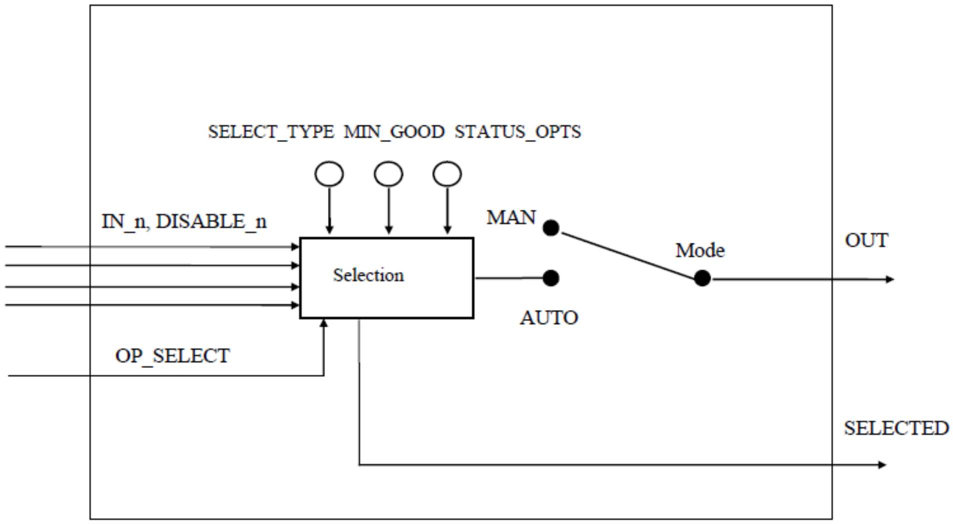


Figure 12: Input Selector Schematic Diagram

- If **SELECT_TYPE** is Minimum, it transfers the lowest value to the output of the block. The value of **SELECTED** is the number of the input with the lowest value.
- If **SELECT_TYPE** is Maximum, it transfers the highest value to the output of the block. The value of **SELECTED** is the number of the input with the highest value.
- If **SELECT_TYPE** is Middle, if there are 3 or 4 values, the highest and lowest value is discarded. The average of the remaining two values is computed, and the value is transferred to the output of the block. The value of **SELECTED** becomes zero if an average is used, else the value of **SELECTED** is the number of the input with the middle value.
- If **SELECT_TYPE** is Average, it computes the average of the remaining inputs and transfers the value to the output of the block. The value of **SELECTED** is the number of inputs used in the average.

Parameters List

Table 15: Input Selector block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block. The revision value increments each time a static parameter value in the block is changed.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
MODE_BLK	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
BLOCK_ERR	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	The block output value and status.
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 device.
STATUS_OPTIONS	It helps 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	The block input value and status.
IN_2	The block input value and status.
IN_3	The block input value and status.
IN_4	The block input value and status.
DISABLE_1	Parameter to switch off the input from being used. 0 - On, 1 - Off.
DISABLE_2	Parameter to switch off the input from being used. 0 - On, 1 - Off.
DISABLE_3	Parameter to switch off the input from being used. 0 - On, 1 - Off.
DISABLE_4	Parameter to switch off the input from being used. 0 - On, 1 - Off.
SELECT_TYPE	Determines the selector action: First Good, Minimum, Maximum, Middle, and Average.
MIN_GOOD	The minimum number of inputs which are "Good" is less than the value of MIN_GOOD then set the OUT status to "Bad".
SELECTED	The integer indicating the selected input number.
OP_SELECT	An operator settable parameter to force a given input to be used.

Parameter	Description
UPDATE_EVT	This alert is generated by any change to the static data.
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 sets 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.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms, (see section 3.2).
Status Handling	During normal operations, the value and status of the selected input is shown by **OUT**. If the number of inputs with Good status is fewer than **MIN_GOOD**, then the output status is Bad. The **SELECTED** output status is Good (NC), until the block is out of service. The block supports two status option: • **Uncertain as Good**: If the selected input status is Uncertain, set the **OUT** status as Good. • **Uncertain, if in Manual mode**: If the block is set to Manual mode, the status of the Output is set to Uncertain.

3.11 Arithmetic block

The Arithmetic block is designed for using popular measurement math functions easily. The math algorithm is selected by name and the type of function to be performed. The block is used for calculating measurements from a combination of signals from the sensors. The block must not be used in a control path. The block does not support process alarms.

The Arithmetic block supports the following functions:

- Flow compensation, linear
- Flow compensation, square root
- Flow compensation, approximate
- BTU flow
- Traditional Multiply Divide
- Average
- Traditional Summer
- Fourth order polynomial
- Simple HTG compensated level
- Fourth order Polynomial Based on PV

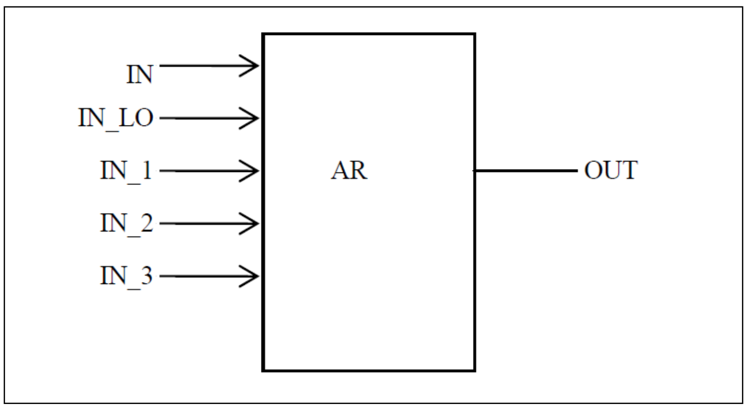


Figure 13: Arithmetic Block

Execution

The block has five inputs, namely **IN, IN_LO, IN_1, IN_2,** and **IN_3**. The first two inputs (**IN, IN_LO**) are designed for a range extension function that results in a Process Variable (PV), with the status indicating the input in use.

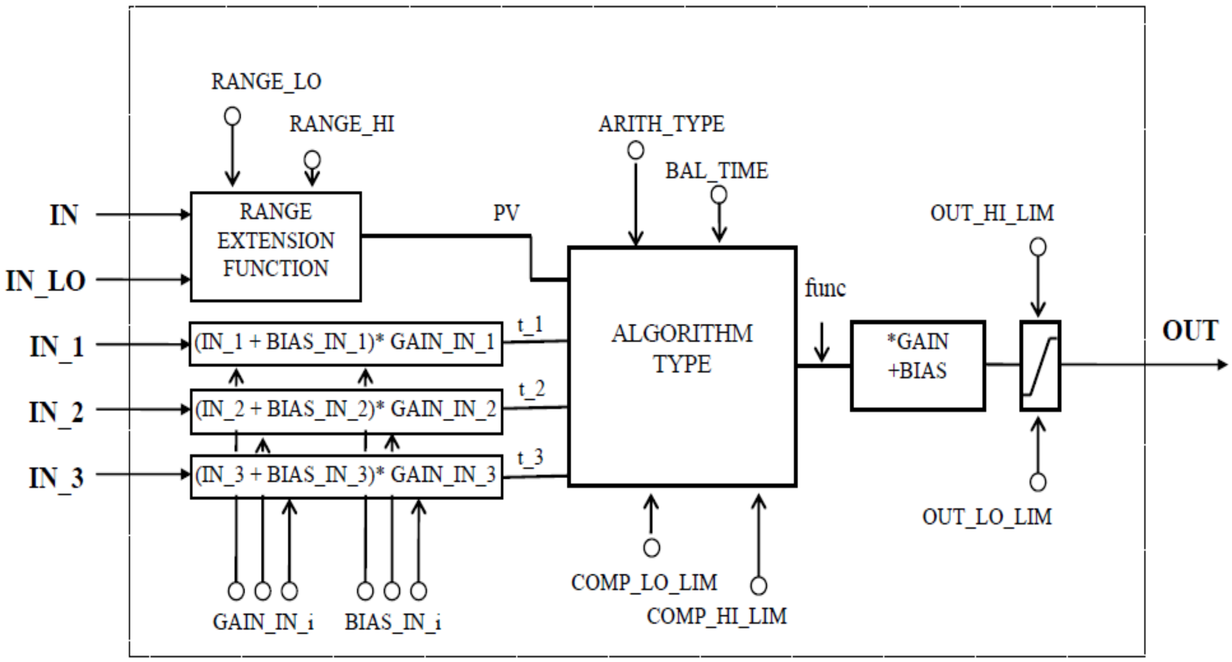


Figure 14: Arithmetic Schematic Diagram

The remaining three inputs (**IN_1, IN_2, and IN_3**) are combined with the PV in a selection of four term math functions. To ensure that the PV enters the equation with the right units, the inputs used to form the PV must come from devices with the desired engineering units. Each additional input has a bias constant and gain constant. To correct Absolute Pressure, use the bias constant, and to normalize terms within a square root function, use the gain constant.

Calculation of PV

The range extension function has a graduated transfer controlled by two constants referenced to IN. An internal value, g, is zero for IN less than **RANGE_LO**. It is one when IN is greater than **RANGE_HI**. It is interpolated from zero to one over the range of **RANGE_LO** to **RANGE_HI**. The equation for PV follows:

$$PV = g \times IN + (1 - g) \times IN\_LO$$

If the status of **IN_LO** is not usable and **IN** is usable and greater than **RANGE_LO**, then g is set to one. If the status of **IN** is unusable, and **IN_LO** is usable and less than **RANGE_HI**, then g is set to zero.

For three auxiliary inputs, six constants are used, and each input has a **BIAS_IN_i** and a **GAIN_IN_i**. The output has a **BIAS** and a **GAIN** static constant. For the inputs, the bias is added, and the gain is applied to the sum. The result is an internal value called **t_i** in the function equations. The equation for each auxiliary input is the following:

$$t\_i = (IN\_i + BIAS\_IN\_i) \times GAIN\_IN\_i$$

If an auxiliary input is unstable, to assure smooth degradation, the flow compensation functions have limits on the amount of compensation applied to the PV. The internal limited value is f.

The following function types are supported:

1. Flow compensation, linear. Used for density compensation of volume flow.

$$func = f \times PV$$

$$f = \frac{(t\_1)}{(t\_2)} \times [limited]$$

2. Flow compensation, square root. Usually, **IN_1** is pressure, **IN_2** temperature, and IN_3 is the compressibility factor Z.

$$func = f \times PV$$

$$f = \sqrt{\frac{\frac{(t\_1)}{(t\_2)}}{(t\_3)}} \times [limited]$$

3. Flow compensation, approximate. Both **IN_2** and **IN_3** would be connected to the same temperature.

$$func = f \times PV$$

$$f = \sqrt{(t\_1) \times (t\_2) \times (t\_3) \times (t\_3)} \times [limited]$$

4. BTU flow, where **IN_1** is inlet temperature, and **IN_2** the outlet temperature.

$$func = f \times PV$$

$$f = (t\_1 - t\_2) \times [limited]$$

5. Traditional Multiply Divide

$$func = f \times PV$$

$$f = \frac{(t\_1)}{(t\_2)} + (t\_3) \times [limited]$$

6. Average

$$func = \frac{(PV + (t\_1) + (t\_2) + (t\_3))}{f}$$

f = number of inputs used in computation (unusable inputs are not used).

7. Traditional Summer

$$func = PV + (t\_1) + (t\_2) + (t\_3)$$

8. Fourth order polynomial. All inputs except **IN_LO** (not used) are linked together.

$$func = \mathrm{PV} + (t\_1)^2 + (t\_2)^3 + (t\_3)^4$$

9. Simple HTG compensated level, where PV is the tank base pressure, IN_1 is the top pressure, IN_2 is the density correction pressure, and GAIN is the height of the density tap.

$$func = \frac{(PV - (t\_1))}{(PV - (t\_2))}$$

10. Fourth order polynomial based on PV

$$func = \mathrm{PV} + \ \mathrm{GAIN\_IN\_1} \times (PV)^2 + \ \mathrm{GAIN\_IN\_2} \times (PV)^3 + \mathrm{GAIN\_IN\_3} \times (PV)^4$$

After the value of **func** is calculated, it is multiplied by **GAIN**, and then **BIAS** is added to the result. Then, the high and low output limits are applied as per configured range scaling, and **PRE_OUT** is updated with the calculated value. If the mode is AUTO, **PRE_OUT** is copied to **OUT**.

Parameter List

Table 16: Arithmetic block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block. The revision value increments each time a static parameter value in the block is changed.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks. This data is not checked of processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
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.
BLOCK_ERR	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.
PV	It calculates the proportions of IN and IN_LO to for PV.
OUT	The analog output value and status.
PRE_OUT	Displays what would be the OUT value if the mode is AUTO or lower.
PV_SCALE	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.
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.

Parameter	Description
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)
INPUT_OPTIONS	Option bit string for handling the status of the auxiliary inputs.
IN	The block input value and status.
IN_LO	Input of the low range transmitter, in a range extension application.
IN_1	The first block input value and status.
IN_2	The second block input value and status.
IN_3	The third block input value and status.
RANGE_HI	Constant value above which the range extension has switch to the high range transmitter.
RANGE_LO	Constant value below which the range extension has switch to the high range transmitter.
BIAS_IN_1	The bias value for IN_1.
GAIN_IN_1	The proportional gain (multiplier) value for IN_1.
BIAS_IN_2	The bias value for IN_2.
GAIN_IN_2	The proportional gain (multiplier) value for IN_2.
BIAS_IN_3	The bias value for IN_3.
GAIN_IN_3	The proportional gain (multiplier) value for IN_3.
COMP_HI_LIM	Determines the high limit of the compensation input.
COMP_LO_LIM	Determines the low limit of the compensation input.
ARITH_TYPE	The set of 9 arithmetic functions applied as compensation to or augmentation of the range extended input.
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.
BIAS	The bias value is used to calculate the output.
GAIN	The gain value is used to calculate the output.
OUT_HI_LIM	The maximum output value allowed.
OUT_LO_LIM	The minimum output value allowed.
UPDATE_EVT	This alert is generated by any changes to the static data.
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 sets 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.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms, (see section 3.2).
Status Handling	The **INPUT_OPTS** bit string controls the use of auxiliary inputs with less than Good status. The status of unused inputs is ignored. The status of the output is the worst of the inputs used in the calculation after applying **INPUT_OPTS**.

3.12 Signal Characterizer block

The Signal Characterizer block describes the input/output relationship for any type of function. The block has two paths, each with an output that is a non-linear function of the corresponding input. The non-linear function is configured based on a single look-up table with 21 arbitrary x-y pairs. To use the block in a control or process signal path, the status of an input is provided to the corresponding output. To use the backward control path, the block provides an option to swap the axes of the function.

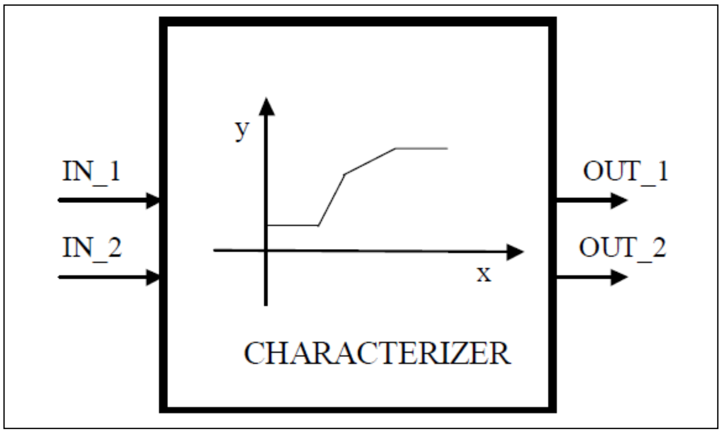


Figure 15: Signal Characterizer Block

The block calculates **OUT_1** from **IN_1** and **OUT_2** from **IN_2** using a curve given by the co-ordinates:

[x1; y1], [x2; y2] ... [x21; y21]

Where,

- x is the Input, and
- y is the Output.

The x-coordinates are given in engineering units of **X_RANGE**. The y-coordinates are given in engineering units of **Y_RANGE.**

Execution

Figure 15 describes the components of the block. The output value is calculated by linear interpolation between two points enclosing the input value. **OUT_1** is associated to **IN_1** and **OUT_2** to **IN_2** by the same curve, but there is no association between **IN_1** and **IN_2** or between **OUT_1** and **OUT_2**.

To derive the output value that corresponds to the input, use the following formula,

$$y = mx + c$$

Where,

- m is the slope of the line.
- c is the y-intercept of the line

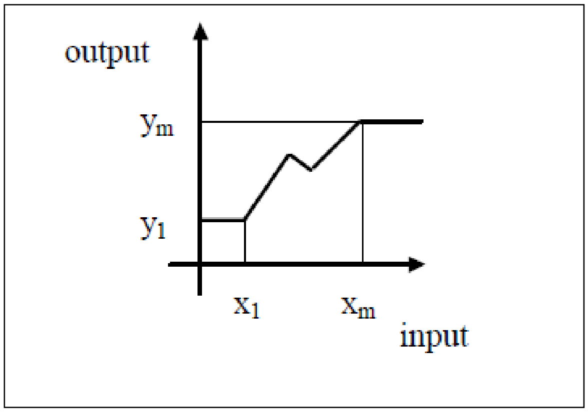


Figure 16: Signal Characterizer Curve

The values of x must increase sequentially for interpolation to be applicable. If not, a configuration error is set in **BLOCK_ERR**, and the **Actual** mode of the block goes to **Out of Service** mode.

If the curve has m points, m<21, the non-configured points, [xm+1; ym+1], [xm+2; ym+2], ... [x21; y21] is set to +INFINITY to mark them as unused.

Since x1 is the smallest specified value for the input and x_m is the largest, the output is at y1 when the input is smaller than x1, and the output is at y_m when the input is larger than x_m. Since the ends of the y curve act as limits, the **OUT** status is shown when either limit is active.

Backward Control path

A reverse function swaps the interpretation of **IN_2** and **OUT_2** that provides a way to do reverse calculation using the same curve. If the parameter **SWAP_2** is set to True, the block provides:

IN_1 = x and **OUT_1** = y while **IN_2** = y and **OUT_2** = x

If the function is not sequential in y and SWAP_2 is True, **BLOCK_ERR** indicates a configuration error, and the **Actual** mode goes to **Out of Service** mode for x. A function is said to be sequential when y values always increase or decrease when x values increase.

If **SWAP_2** = False, **IN_1** and **IN_2** have the same engineering units defined in **X_RANGE** and **OUT_1** and **OUT_2** use the units defined in **Y_RANGE**.

If **SWAP_2** = True, **OUT _1** and **IN_2** have **Y_RANGE** and **OUT_2** and **IN_1** have **X_RANGE**.

Parameter list

Table 17: Signal Characterizer block parameters

Parameter	Description
ST_REV	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.
TAG_DESC	The use description of the intended application of the block.
STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
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
BLOCK_ERR	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_1	The block output value and status.
OUT_2	The block output value and status.
X_RANGE	The display scaling of the variable corresponding to the x-axis for display. It has no effect on the block.
Y_RANGE	The display scaling of the variable corresponding to the y-axis for display. It has no effect on the block.
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)
IN_1	The block input value and status.
IN_2	The block input value and status.
SWAP_2	Changes the algorithm in such a way that IN_2 corresponds to “y” and OUT_2 to “x”.
CURVE_X	Curve input points. The “x” points of the curve are defined by an array of 21 points.
CURVE_Y	Curve input points. The “y” points of the curve are defined by an array of 21 points.
UPDATE_EVT	This alert is generated by any changes to the static data.
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 sets 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.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS (Out of Service).
Alarm Types	The block supports standard block alarms, (see section 3.2).
Status Handling	**OUT_1** shows the status of **IN_1** and **OUT_2** shows the status of **IN_2**. The sub-status is also passed to the outputs. If one of the curve limits is reached or the input is limited, the appropriate limit must be indicated in the output sub-status. Limits shall be reversed if the curve slope is negative. If **SWAP_2** is set, cascade initialization is controlled by the lower block. When this block is in OOS mode, the cascade to both the lower and upper blocks is broken by Bad status at the outputs. When the block goes to AUTO mode, the lower block can begin cascade initialization with status values that pass through this block to the upper block. The output status signals from the upper block pass through this block to the lower block. The block does not use **STATUS_OPTS**.

3.13 Output Splitter block

The output splitter block drives two control output signals from a single input signal. Each output is a linear function of a fraction of the input signal. The same linear function when used in reverse provides the back calculation support. For different combinations of input and output conditions, a decision table supports cascade initialization. This block finds application in split ranging or sequencing of multiple valve. In a typical split range application, when the splitter input is 50% both the output valves remain closed. One of the valves opens proportionately to full as the input drops to 0% and the other valve opens proportionately as the input rises above 50%. In a typical sequencing application, both the valves are closed at 0% input. One of the valves opens proportionately to full as the input rises to 50%, while the other stays shut. The second valve opens as the input rises above 50%, and the first valve may remain open or shut off quickly. As this block is in the control path, it has the ability to pass limit and cascade initialization information back to the upstream block.

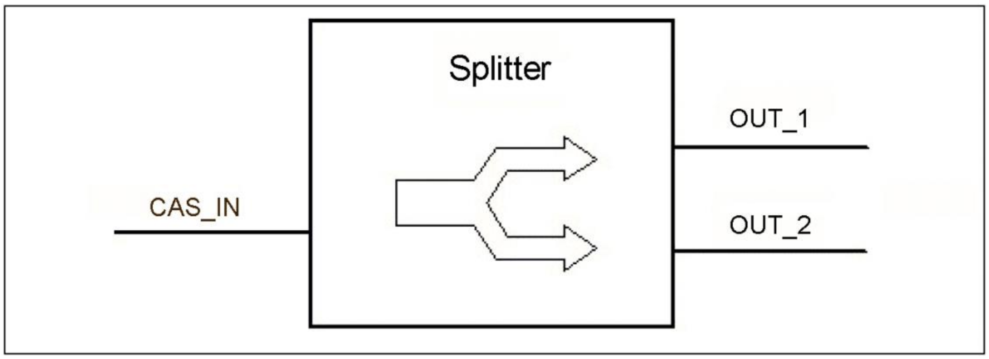


Figure 17: Output Splitter Block

Execution

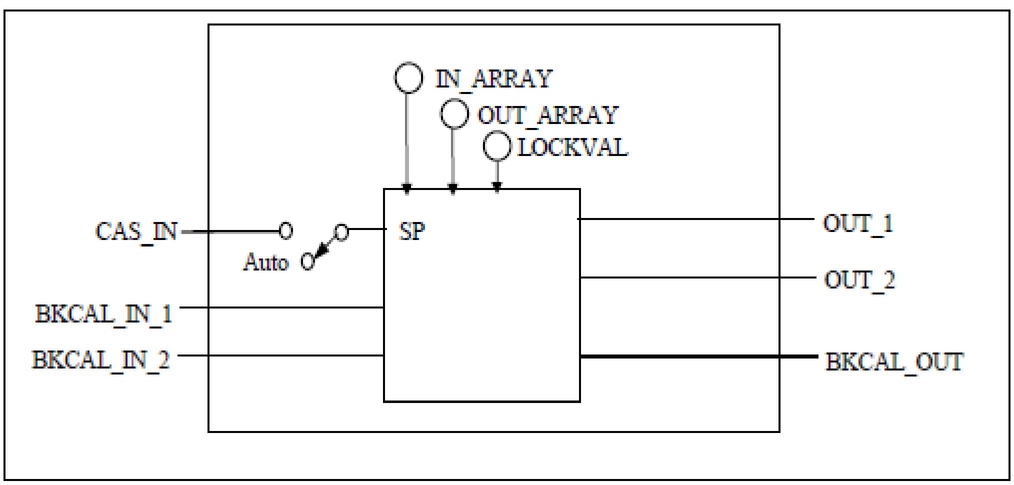


Figure 18: Output Splitter Schematic

The relationship of each output to the input may be defined by a line. Each line may be defined by its endpoints. Examples of graphical representations of OUT_1 and OUT_2 vs. SP are shown below for a split range and a sequencing application.

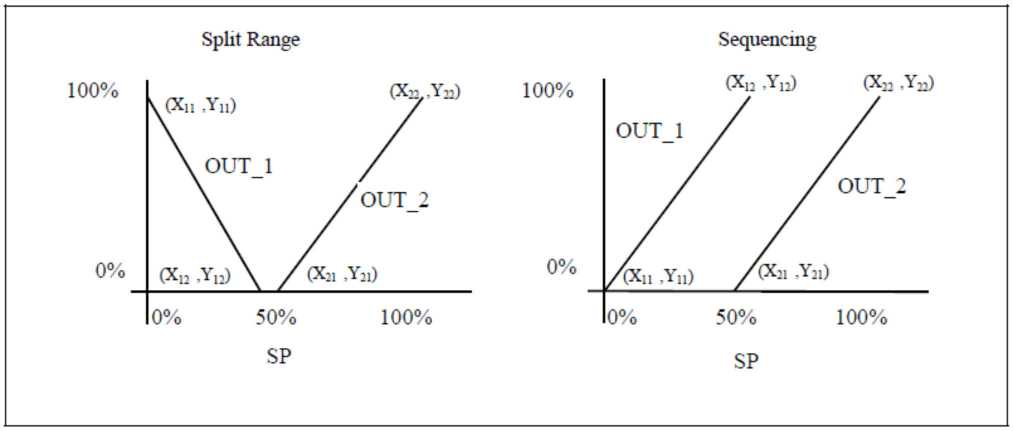


Figure 19: Split Range and Sequence Operation

The examples shown do not show the full range of possibilities. The lines could overlap like an X, or both start from the origin but have different slopes. The endpoints do not have to lie within 0-100%. Limits in the external blocks may affect the useful range of a line. Units of percent are used in the examples because the common application of this block is to valves, but any units may be used to suit the application.

The following parameters may be used to specify the output splitter operation:

X11, Y11, X12, Y12

X21, Y21, X22, Y22

Where XnJ is the value of SP associated with OUT_n and Xn1 and Xn2 refer to the 1st and 2nd coordinates of the nth curve respectively. YnJ is the value of OUT_n and Yn1 and Yn2 refer to the 1st and 2nd coordinates of the nth curve respectively.

IN_ARRAY

Index	Coordinate
1	X_{11} – Start value of SP for the OUT_1 line.(X_{11}<X_{12})
2	X_{12} – End value of SP for the OUT_1 line.(X_{11} <X_{12})
3	X_{21} – Start value of SP for the OUT_1 line.(X_{21} <X_{22})
4	X_{22} – Start value of SP for the OUT_1 line.(X_{21} <X_{22})

OUT_ARRAY

Index	Coordinate
1	Y_{11} – Value of OUT_1 at X_{11}
2	Y_{12} – Value of OUT_1 at X_{12}
3	Y_{21} – Value of OUT_2 at X_{21}
4	Y_{22} – Value of OUT_2 at X_{22}

By specifying the coordinates as shown above, the endpoints of the lines are defined. The contents of the respective X's are held in the IN_ARRAY parameter and the contents of the respective Y's are held in the OUT_ARRAY parameter. If a set of points are specified such are held in the IN_ARRAY parameter and the contents of the respective Y's are held in the OUT_ARRAY parameter. If a set of points are specified such that a region of the input range is not specified, then the corresponding OUT_n may be set to the closest endpoint of the input value, either high or low, when the specified region is exceeded.

A configuration error shall be set in BLOCK_ERR and the actual mode of the block shall go to Out of Service if the X values have any of the following conditions:
X21 < X11, X12 <= X11, X22 <= X21.
The parameter LOCKVAL provides an option to specify whether OUT_1 remains at its ending level when control is switched to OUT_2, or goes to Y11. If LOCKVAL is "LOCK", OUT_1 remains at its ending value when X is greater than X12. If LOCKVAL is "NO LOCK", then OUT_1 goes to Y11 when X is greater than X12. Some hysteresis in the switching point may be required because the output may change by a full stroke of the valve. HYSTVAL contains the amount of hysteresis. If X <= X12-HYSTVAL, OUT_1 may be determined by the calculated y value. If X12-HYSTVAL < X < X12 and X has not reached X12 since it was less than X12-HYSTVAL, OUT_1 may be determined by the calculated y value. If X transitioned from a value > X12 to a value where X12-HYSTVAL < X < X12, then the value of OUT_1 is determined by the LOCKVAL setting. If X12 < X, OUT_1 may be determined by the LOCKVAL setting.
In the following example LOCKVAL ="LOCK":

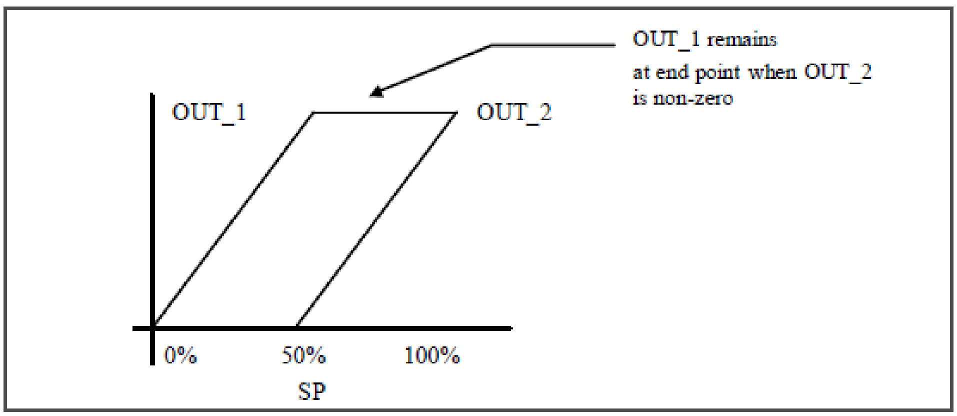


Figure 20: OUT with LOCKVAL"LOCK"

In this example LOCKVAL= "NOLOCK"

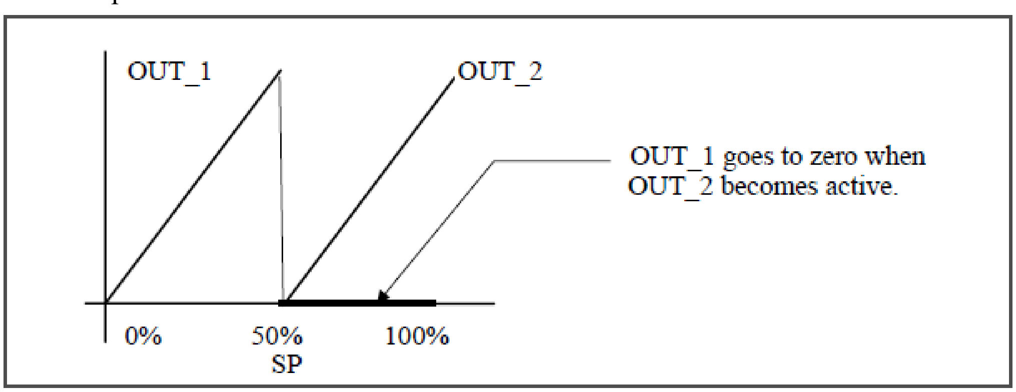


Figure 21: OUT with LOCKVAL "NO LOCK"

Parameter list

Table 18: Output Splitter block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	Reflects the error status of the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown.
SP	It is the target block setpoint value. It is the result of setpoint limiting and setpoint rate of change limiting.
OUT_1	The value and status of out_1 of the block.
OUT_2	The value and status of out_2 of the block.
OUT_1_RANGE	The maximum value range of out_1 of the block.
OUT_2_RANGE	The maximum value range of out_1 of the block.
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.
STATUS_OPTS	Helps select options for status handling and processing. The supported status options for the OS block are ‘IFS if Bad CAS_IN’ and ‘Target to next permitted mode if BAD CAS_IN’.
CAS_IN	The remote setpoint value from another block.
BKCAL_OUT	The value and status required by the BKCAL_IN input of another block to prevent reset windup and to provide bump less transfer of closed loop control.
IN_ARRAY	An array which contains the values of the input or X variables.
OUT_ARRAY	An array which contains the values of the output or Y variables.
LOCKVAL	Flag for holding the first output at current value when the other output is non-zero.
BKCAL_IN_1	The analog input value and status from another block’s BKCAL_OUT output that is used for backward output tracking for bump less transfer and to pass limit status.

Parameter	Description
BKCAL_IN_2	The analog input value and status from another block's BKCAL_OUT output that is used for backward output tracking for bump less transfer and to pass limit status.
BAL_TIME	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 moves to obtain balance when the output is limited and the mode is AUTO, CAS, or RCAS.
HYSTVAL	Specifies the Hysteresis value.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for configuration, hardware, and 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. When the Unreported status is cleared by the alert reporting task, another block alert is reported without clearing the Active status, if the subcode has changed.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • IMAN (Manual) • OOS (Out of Service) • Cas
Alarm Types	Standard block alarm
Status Handling	Sub-status values received at CAS_IN shall be passed to both outputs, except for those used in the cascade handshake. An IFS shall go to both outputs. The status option IFS if Bad CAS_IN is available. The splitter block shall propagate the BKCAL_IN status of Bad, Device failure or Good Cascade, Fault State Active or Local Override only if the statuses of both BKCAL_IN's contain a propagated fault status.

3.14 Discrete Input block

The Discrete Input (DI) function block processes a 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 block takes the transducer's input data from any one of the channels based on the channel parameters. There are four channel parameters: Limit Switch 1, Limit Switch 2, Limit Switch 3, and Limit Switch 4.

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

The block is normally used in Automatic (Auto) mode. In this mode, the process variable (PV_D) is copied to the output (OUT_D). The block can also be changed to Manual (Man) mode 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, enable simulation. When simulation is enabled, you can supply the measurement value manually through the SIMULATE_D parameter.

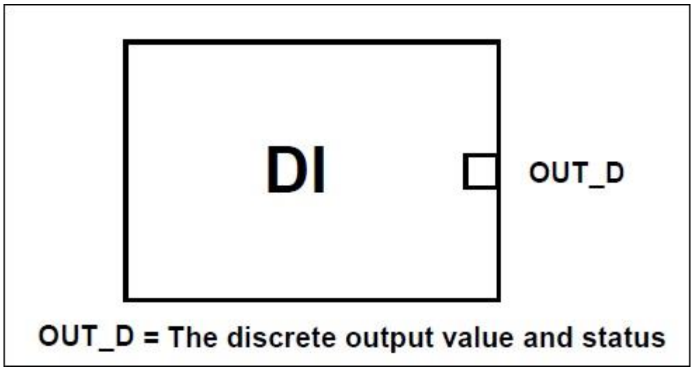


Figure 22: Discrete Input Block

Schematic

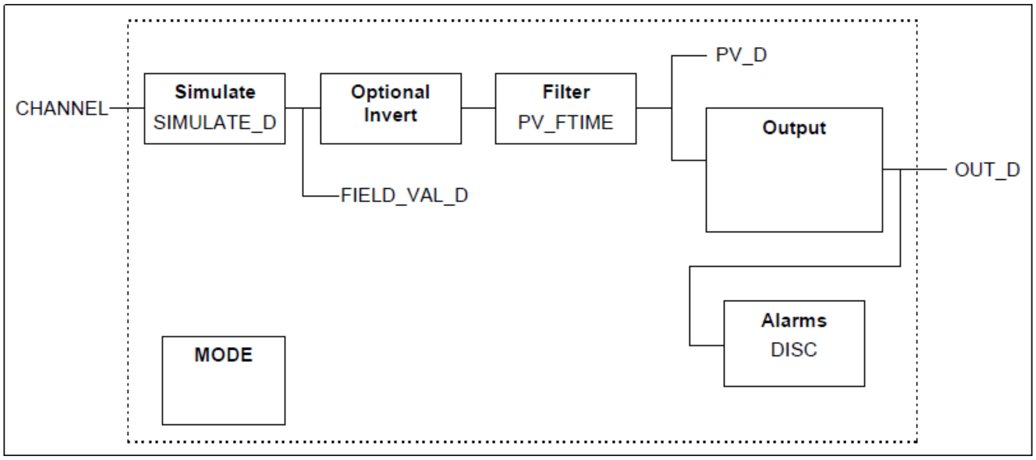


Figure 23: Discrete Input Block Schematic Diagram

Description

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

Parameters List

Table 19: Discrete Input block parameters

Parameter	Description
ST_REV	The revision level of the static data associated with the function block.
TAG_DESC	The user description of the application of the block.
STRATEGY	Used to identify grouping of blocks.
ALERT_KEY	The identification number of the plant unit.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	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.
PV_D	Process variable of the Discrete Input block.
OUT_D	Output variable of the Discrete Input block.
SIMULATE_D	A group of data for Discrete Input block that contains the current transducer value and status, the simulated transducer value and status, and the enable/disable bit.
XD_STATE	Index to the text describing the states of a discrete for the value obtained from the transducer.
OUT_STATE	Index to the text describing the states of a discrete output.
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.
IO_OPTS	Allows the selection of input/output options used to alter the PV. Low cutoff enabled is the only selectable option.
STATUS_OPTS	Helps select options for status handling and processing. The supported status options for the DI block are Propagate Fault Forward and Uncertain if Man mode

Parameter	Description
CHANNEL	The CHANNEL value is used to select the measurement value. Configure the CHANNEL parameter before configuring the XD_SCALE parameter.
PV_FTIME	Time constant of a single exponential filter for the PV, in seconds.
FIELD_VAL_D	Raw value of the field device discrete input, with a status reflecting the Transducer condition.
UPDATE_EVT	This alert is generated by any change to the static data.
BLOCK_ALM	The BLOCK_ALM is used for configuration, hardware, and 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. When the Unreported status is cleared by the alert reporting task, another block alert is reported without clearing the Active status, if the subcode has changed.
ALARM_SUM	The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.
ACK_OPTION	Used to set AUTO acknowledgment of alarms.
DISC_PRI	Priority of the discrete alarm.
DISC_LIM	State of discrete input which will generate an alarm.
DISC_ALM	The status and time stamp associated with the discrete alarm.

Attributes

Supported Modes	The block supports the following modes: • AUTO (Automatic) • MAN (Manual) • OOS mode (Out of Service)
Alarm Types	Standard block alarm plus standard discrete alarm applied to OUT_D.
Status Handling	The following options from STATUS_OPTS apply: • Propagate Fault Forward • Uncertain if Man mode

3.15 Configuring the transmitter using Field Device Manager system

The transmitter can be configured through Field Device Manager (FDM), by using DTM for releases R410 and R430 and using DD as well as DTM for release R440. For more information, refer the FDM manuals #EP-FDM-11410, #EP-FDM-11430 and #EP-FDM-11440 for the corresponding releases.

4. STT850 FF operation

4.1 Operational considerations

There are a number of considerations that must be noted when configuring a transmitter to operate in a fieldbus network.

LAS Capability

The transmitter is capable of operating as the Link Active Scheduler (LAS). The LAS is a fieldbus feature which controls traffic on the network, such as controlling token-rotation and coordinating data publishing. This fieldbus function is active in only one device at any given time on a network. Devices which can be designated as the LAS may be an operator station or a field device. The transmitter can be designated as LAS, in the event of a failure of the primary LAS, control in the field could continue.

ATTENTION

Note that the transmitter can be used only as “backup” LAS.

Special Non-volatile parameters and NVM Wear-out

All function block parameters designated as Non-Volatile (N) in the FF specifications are updated to non-volatile memory (NVM) on a periodic basis. **NV_CYCLE_T** parameter in the resource block specifies this update interval.

To provide predictable restart behavior in the transmitter, the following Non-Volatile parameters are updated to NVM each time they are written over the fieldbus.

- **MODE.TARGET** for all blocks
- **SP.VALUE** for the PID block

Since these are user-written parameters, these additional updates to NVM contribute negligibly to NVM wear out. However, users are cautioned to not construct control configurations where the above parameters are written continuously (via a computer application for example) or at rates greater than the **NV_CYCLE_T** interval. This consideration helps to minimize the possibility of NVM wear-out.

In the case of MODE this must not be a problem. When users wish to provide set-points to the PID block via a computer application, users should use RCAS mode with its corresponding setpoint value **RCAS_IN**. **RCAS_IN** is updated only at the **NV_CYCLE_T** update rate and this mode supports full shedding functionality and PID initialization necessary for a robust application.

Mode Restricted Writes to Parameters

Some block parameters have restrictions on having write access to them. These are specified in the FF specifications. Writing to certain function block parameters are restricted based on the block’s Target and/ or Actual mode.

4.2 Configuration of the transmitter using Handheld (HH)

Figure 24 graphically represents the connection of the transmitter to the handheld. Each transmitter includes a configuration database that stores its operating characteristics in a non-volatile memory. The handheld is used to establish and/or change selected operating parameters in a Transmitter database. The process of viewing and/or changing database parameters is called configuration.

Configuration can be accomplished both online and offline with the Transmitter powered up and connected to the handheld. The online configuration immediately changes the Transmitter operating parameters. For offline configuration, Transmitter operating characteristics are entered into the handheld memory for subsequent downloading to a Transmitter.

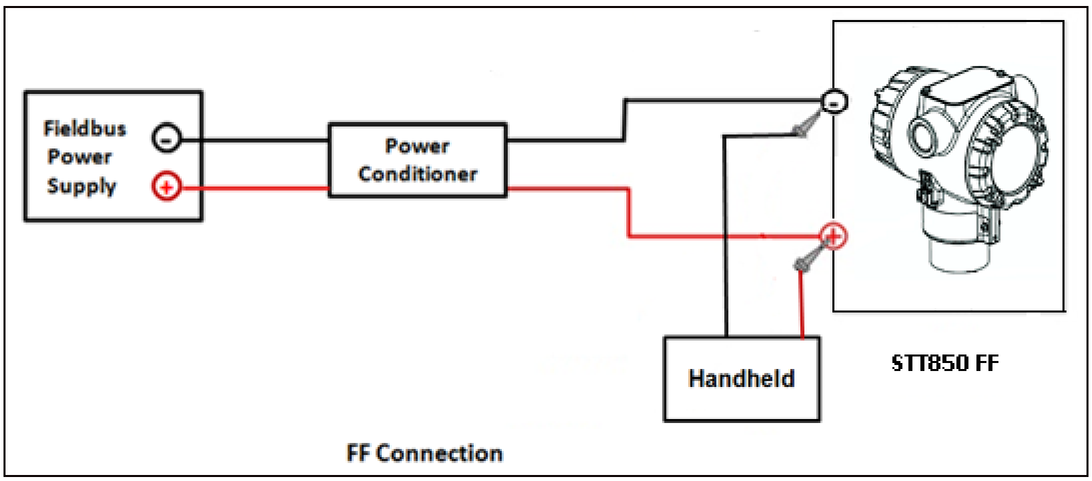


Figure 24: Connecting the transmitter to the handheld

4.3 Performing block instantiation

About block instantiation

A block instance is a copy of an available block in the device, say for example AI block. There are totally 10 permanent blocks, and only four blocks support instantiation in a device. The four blocks that support instantiation are Analog Input block, Discrete Input block, Arithmetic block and PID block. Four instances of the Analog Input block, two instances of the Discrete Input block and Arithmetic block, and one instance of the PID block can be instantiated. A block can be instantiated or deleted.

Before block instantiation, the device checks whether the particular block is supported, and if there is sufficient memory to store the parameters. After Instantiation, the instantiated block must be loaded into the device, and then the strategies can be created.

Block instantiation using Experion PKS

The following are the steps for performing block instantiation using Experion PKS.

Step	Action
1	From the DD at the **Library-Containment** window, select an instantiation block from the supported blocks, that is Analog Input block, or Input Selector block, or Signal Characterizer block.
2	Drag and drop the required instantiation block into the device on the **Project-Assignment** window.
3	After adding the instantiation block into the device in the **Project-Assignment** window, select the device.
4	Right-click the device, and click **Load**. The instantiated block is loaded into the device.

5. STT850 FF maintenance

5.1 Replacing the Local Display and Electronic Assembly

For more information about Local Display and Electronic Assembly, refer the STT850 SmartLine Temperature Transmitter User Manual, #34-TT-25-03.

5.2 Downloading the firmware

About firmware download feature

The download class indicates how the device operation is affected by the download process. There are three types of download classes (1, 2 &3). The transmitter supports only one type of download class as per FOUNDATION Fieldbus specifications. STT850 device FF variant supports download type Class 3 only. A class-3 firmware download is performed, irrespective of whether the device is ON /OFF process.

Class 3

When class 3 download is performed the device prepares for the download and goes out of the link as the memory of the device is re-written with the new firmware. After the restart of the device, the device comes back to the link automatically. However, the device retains the following credentials,

- Retains its original Device ID
- Retains only its System Management **VFD** in its **VFD_LIST**
- Retains its Node Address and **PD** Tag (only when the same firmware version is reloaded)
- Retains its management **VCR** to provide access to the SMIB.

Recommendations

If firmware upgrade is required for a large number of STT850 devices, the following are the guidelines,

1. **Diagnostics must be backed-up before initiating the firmware update.** The communication board diagnostics are initialized to zero if backup is not performed before initiating the firmware update. The backup diagnostics method is available in the Diagnostics transducer block.
2. **Only one device firmware download is allowed in a given H1 Link**:
   Firmware download to multiple devices must happen one after another in the same link. However, parallel downloads can be performed to devices on different H1 links.
3. **Download firmware to one device type at a time in a H1 link:**
   This reduces the chance for unknown interactions between devices to cause link issues or download failures.
4. **Reduce usage of DTM through tools like FDM in the H1 link:**
   This reduces the traffic on the link and therefore reduces the time required for the download to complete.
5. **Parallel Firmware downloads from single Control Builder**
   Firmware downloads to a single FIM should be done from single Control Builder instance. This reduces the chance of initiating multiple downloads to the same H1 link from different users.
6. **FF segment** design (the choice of devices to connect to a FF segment) must consider the maximum current draw of those devices, as well as the potential for inrush current during power-up.

   For reference, the STT850 provides the following:

   - Max current draw (observed during firmware download): 28 mA
   - Normal quiescent current: 18 mA
   - Inrush when powered on: 28 mA

Downloading the File

The firmware file to be downloaded is called as Gendomain file and have the file extension .ffd.

File Name

The file name is constructed as follows:

“Manufacturer ID” + “_” + “Device Type” + “_” + “Domain Name” ” + “_” +“Software Name” + “_” +“Software Revision” + “.” + “ffd”, where:

- **Manufacturer ID** is represented as six hexadecimal digits (leading and trailing zeroes are included).
- **Device Family** is represented as four hexadecimal digits (leading and trailing zeroes are included). For Multidomain devices, Device Family is replaced by Multidomain Family.
- **Device Type** is represented as four hexadecimal digits (leading and trailing zeroes are included).
- Leading "0"s are not suppressed for **Manufacturer ID** and **Device Type**.
- Trailing blanks are stripped from Device Family, Domain Name, Software Name, and Software Revision.
- If **Software Name** or **Software Revision** is composed of all blanks, then the underscore that would have proceeded is omitted to prevent names with two adjacent underscores, or from having the underscore character appear directly before the ".ffd".

For example, if the file contains the following header values,

Manufacturer ID = "48574C"

Device Type = "0006"

Domain Name = "FD-DOM"

Software Name = "FD_SW"

Software Revision = "2-41"

Then the file name would be:

"48574C0006_0006_FD-DOM_FD-SW_2-41.ffd".

ATTENTION

In the STT850 FF temperature transmitter, only communication board firmware can be upgraded using the class 3 download. Display and sensor boards' firmware upgrade is not possible through FF link in the current release

6. STT850 FF troubleshooting

6.1 Troubleshooting overview

This section contains information to help you identify the faults in devices and the recommended actions to correct them. Troubleshooting is performed to determine the cause of the fault by analyzing the device indications (such as device not visible on network or not able to write values to parameters.)

Device status and faults

The transmitter constantly runs internal background diagnostics to monitor the functions and status of the device operations. When errors and/or faults are detected, they are reported in the status bits of certain block parameters, (for example, **BLOCK_ERR**). The other parameters can be seen by viewing the status descriptions and/or a value, which may help to identify a fault.

Device status and operational faults are identified by viewing key parameter values or status and then interpreting their meaning using the following tables.

ATTENTION

Additional diagnostics are available through supervisory and control applications that monitor and control fieldbus networks. These diagnostics and messages are dependent upon the capabilities of the application and the control system that is used.

6.2 Troubleshooting the transmitter

Device not visible on the network

If a device cannot be seen on the fieldbus network, the device may not be powered up or possibly the supervisory or control program is not able to find (or polling) the node address of that device. See the following table for possible causes and recommended actions.

After address is configured for the device, revert back to original configuration of number of unpolled nodes

Symptoms		
Device not visible on the network		
Possible cause	**Things to check**	**Recommended action**
Device may have a node address that is within the “unpolled range” of addresses.	Verify the following settings: • First Unpolled Node • Number of Unpolled Nodes	Set Number of Unpolled Nodes to “0”.
No power to the device.	Measure the DC voltage at the device’s SIGNAL terminals. Voltage must be within the limits.	If no voltage or voltage is out of operating limits, determine the cause and correct it.
Insufficient current to the device.	Measure the DC current to the device. The DC current must be within the limits.	If the current is insufficient, determine the cause and correct it.
More than two or less than two terminators are wired to fieldbus link.	Check to see that only two terminators are present on a link.	Correct, if necessary.
Insufficient signal to the device.	Measure the peak-to-peak signal amplitude. The output must be 0.75 to 1.0 Vp-p. Measure the signal on the + and - SIGNAL terminals and at a frequency of 31.25k Hz.	If the signal amplitude is insufficient, determine the cause and correct it.
Names of parameters are not visible.	Missing or incorrect version of Device Description file on host computer.	Check the path to the Device Description. Load correct version of DD.

Incorrect or non-compatible tools

If non-compatible versions of fieldbus software tools are used, such as Standard Dictionary or Device Description (DD) files, or if you are using the incorrect revision level of device firmware, then device objects or some block objects may not be visible or identified by name. See the following table for the possible causes and recommended actions.

<table>
<tr><th colspan="3">Symptoms</th></tr>
<tr><td colspan="3">Device and/or block objects not identified (Unknown).
Or
Parameters are not visible or identified by name.
Or
Honeywell-defined parameters are not visible.</td></tr>
<tr><th>Possible cause</th><th>Things to check</th><th>Recommended action</th></tr>
<tr><td>Incorrect Standard Dictionary, Device Description (DD) or Symbols on host computer.</td><td>Verify that the Standard Dictionary, the DD or symbols files are correct for the device.</td><td>Install the compatible version of Standard Dictionary and DD for the device on the host computer.</td></tr>
<tr><td>Incorrect pathnames to descriptions on host computer.</td><td>Check that the pathnames to locations of the Standard Dictionary, and DD files on the host computer are correct.</td><td>Make sure that the pathnames of the Standard Dictionary and DD are in the correct location for the fieldbus software application.</td></tr>
<tr><td>Incorrect version of device firmware</td><td>Read the following Resource block parameters:
• DEV_REV (contains the revision level of the resource block).
• DD_REV (contains the revision level of the resource block).</td><td>Perform a code download of the correct device firmware. See section 5.2.</td></tr>
</table>

6.3 Troubleshooting blocks

Non-functioning blocks

Device block objects may not be running (executing their function block schedules) or the blocks may be in Out of Service (OOS) mode due to block configuration error. For example, if the AI function block is in OOS mode, the block does not provide updated output values, although the AI block may be running. While troubleshooting a non-functioning block objects, it is recommended to start with the resource block. For example, if the resource block is in OOS mode, all other blocks in the device are also in the OOS mode.

Troubleshooting block configuration errors

The block configuration errors prevent a device block from leaving the OOS mode. The **BLOCK_ERR** parameter (bit 1) shows whether a block configuration error is present. The following section explains the troubleshooting for all the function blocks:

Troubleshooting the Resource block

Table 20: Resource block

Problem cause	Things to check	Recommended action
Resource block mode is OOS mode and is not going to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE:** If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Resource block is not running.	Check **BLOCK_ERR** for errors.	See Table 3 for details on **BLOCK_ERR**.
Incorrect revision level of the device firmware.	Read **SOFTWARE_REV**	See section 6.2
Block alarms are not reported.	Read **FEATURE_SEL**	**Reports** are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.
Field diagnostics alarms are not reporting.	Check Field Diagnostics MASK.	If the alarms are MASKED, then the alarms do not report. Unmask the alarms.
	Check Field Diagnostics Priority.	If the priority is zero alarms do not report. For information on how set the priority, see Table 4.
	Check Field Diagnostics MAP.	If alarms are not mapped, then Map alarms to any of the Field Diagnostics alarm parameters.

Troubleshooting the Temperature Transducer block

Table 21: Temperature Transducer block

Problem cause	Things to check	Recommended action
Transducer block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE:** If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Transducer block does not produce valid Primary value.	Check the primary Value Range.	Ensure that primary Value Range has valid ranges and units assigned.
	Read **BLOCK_ERR**.	See Table 3 for details on **BLOCK_ERR**.
	Verify parameter: **PRIMARY_VALUE** is not valid **STATUS** = Good or Uncertain **VALUE** = active	Isolate transmitter from process. Perform Correct Reset calibration. Recalibrate the transmitter.
	Check Sensor Configuration	Verify that the **SENSOR_TYPE** is assigned correctly according to the input that is connected. Check sensor connections as per type, connection diagram, and also see that SENSOR_**CONNECTION** is set as per the wire count in case of RTDs. In case of TCs, check whether **CJC_TYPE_PARAM** and **FIXED_CJ_VALUE** are correctly set for appropriate cold junction compensation.
Transducer block does not produce valid Secondary value.	Check the Secondary value Range.	Ensure that Secondary Value Range has valid ranges and units assigned.
	Check Cold junction parameters	Check whether **CJC_TYPE_PARAM** and **FIXED_CJ_VALUE** are correctly set.

Problem cause	Things to check	Recommended action
Transducer block does not produce valid Secondary value.	Check the Secondary value Range.	Ensure that Secondary Value Range has valid ranges and units assigned.
	Check Cold junction parameters	Check whether **CJC_TYPE_PARAM** and **FIXED_CJ_VALUE** are correctly set.
Transducer block does not produce valid Differential/ Average/Redundant/ Split Range Temperature value	Check Transmitter model	Calculated PVs are derived from both Sensor 1 and 2 and hence not available in single input models
	Check **ENABLE_OPT**	The **ENABLE_OPT** corresponding to desired PV must be set for it to function
	Check the Range for the desired PV.	The ranges will default as per chosen **SENSOR_TYPE_1** and **2**. Ensure that desired limits and units are assigned.
Transducer block shows incorrect Electronic Housing temperature value.	Check the Electronic Housing temperature units.	Ensure that proper unit is assigned to Electronic Housing temperature.
LRV calibration failed.	Check the Temperature applied.	Ensure that Temperature (or corresponding voltage/resistance input from calibrator) applied as per the **CAL_POINT_LO** value entered.
URV calibration failed.	Check the Temperature applied.	Ensure that Temperature (or corresponding voltage/resistance input from calibrator) is applied as per the **CAL_POINT_HI** value entered.
Block alarms are not reported.	Read **FEATURE_SEL**.	**Reports** are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**.	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Problem cause	Things to check	Recommended action
Status for one of the PVs is shown as BAD, Sensor Failure	Check Input Connections for the corresponding PV	If Break detect is Enabled, wire breaks could cause this failure.
	Check for Latched States	If PV status does not return to Good after rectifying connection issues and validating with connection diagram, check if latching is Enabled. If yes, there may be a latched Bad state which needs to be cleared manually by running the Clear latch method
Limit Switch is not functioning correctly	Check limit switch configuration	Check that the Limit switch source, setpoint, direction and units are set properly. Most commonly it may be easier to set the units to be same as the PV source, else the setpoint and hysteresis values need to be set as per the new units

Troubleshooting the Diagnostics Transducer block

Table 22: Diagnostics Transducer block

Problem cause	Things to check	Recommended action
Diagnostic Transducer block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE:** If the mode is set to OOS for maintenance then do not change the mode to AUTO.
Sensor Diagnostics, Sensor voltage diagnostics, Electronic temperature diagnostics values are not updating.	Read **UPLOAD_TRACK_DATA**	Select value other than NONE, and then wait for 10 seconds. If no values are updated (for example, if Max and Min still shows 999) in Sensor Diagnostics and Sensor voltage diagnostics, Contact Honeywell TAC.
Block alarms are not reported.	Read **FEATURE_SEL**	**Reports** are not selected in **FEATURE_SEL**. If features do not include reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the LCD Transducer block

Table 23: LCD Transducer block

Problem Cause	Things to check	Recommended Action
LCD Transducer block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE** If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Writing to display parameters fails.	Check for local display.	Either Basic or Advanced Display is required for **LCD_TB** to work.
		If display is available, remove and reconnect the local display, and check if display powers up.
		If display is not powering up contact Honeywell TAC.
Writing to some of display parameter in **SCREEN_1**, **SCREEN_2**, **SCREEN_3** **SCREEN_4** **SCREEN_5, SCREEN_6, SCREEN_7,** or **SCREEN_8** fails.	Check **DISPLAY_TYPE**.	If it shows Basic display, then parameters for which write fails are not supported by Basic display. These parameters are supported only by Advanced display.
Local display shows Attention as title with some text.	Check the **DISPLAY_MESSAGE** parameters.	Transmitter messaging is activated; to clear the message executed the Clear Message method. For more information see section 3.7
Block alarms are not reported.	Read **FEATURE_SEL**.	**Reports** are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Analog Input (AI) block

Table 24: Analog Input block

Problem cause	Things to check	Recommended action
Analog Input block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE:** If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
	Read **WRITE_LOCK** parameter in resource block. Check if device is in Write Protect mode. If **WRITE_LOCK** = Locked (2)	Change Write Protect jumper to "W" position. (See section 6.6) Reset the device. (Cycle power to transmitter or write "Processor" to **RESTART** parameter in Resource block.)
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
Analog Input block mode is in OOS mode with Block Configuration Error.	Read **CHANNEL** parameter and range.	**CHANNEL** must be set to a valid value and cannot be left at the initial value of zero.
		XD_SCALE.UNITS_INDX must be compatible with the units in the transducer block for the channel.
	Read **L_TYPE** parameter.	**L_TYPE** must be set to Direct, Indirect, or Indirect Square Root and cannot be left at the initial value of zero.
	Check if **L_TYPE**= Direct	When **L_TYPE** = Direct, **XD_SCALE** and **OUT_SCALE** must contain the same range values (**EU_0** and **EU_100**).
Value of output seems wrong.	Read Linearization Type.	Check the **L_TYPE** setting.
	Read Scaling.	Check **XD_SCALE** and **OUT_SCALE**
Process and block alarms do not work.	Read **FEATURE_SEL**.	**Reports** are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.
	Read Alarm Summary Disable.	Check that process and block alarms are not disabled.
Cannot set alarm limits.	Read Scaling.	Limit values are outside the **OUT_SCALE.EU_0** and **OUT_SCALE.EU_100** values. Set values within range.

Troubleshooting the Proportional Integral Derivative (PID) block

Table 25: PID block

Problem Cause	Things to check	Recommended action
PID block mode is in OOS mode, and does not change to AUTO, CAS, RCAS and ROUT mode.	Read **MODE_BLOCK.PERMITTED**.	Add AUTO, CAS, RCAS and ROUT modes to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE:** If the mode is set to OOS for maintenance then do not change the mode to AUTO.
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
PID block mode is in OOS mode with Block configuration Error.	Read parameters: **BYPASS** **SHED_OP**	The default values of these parameters are configuration errors and they must be set to a valid range. See Table 31.
	Read **SP_HI_LIM**, **SP_LO_LIM** **OUT_HI_LIM**, **OUT_LO_LIM**	Check that **SP_HI_LIM < SP_LO_LIM**, **OUT_HI_LIM < OUT_LO_LIM**.
Mode does not change from IM, target mode is MAN, AUTO, or Cas.	No path to process.	Assure that the downstream blocks to at least one AO are all in Cas mode and that the path ends in an AO block. All BKCAL connections must be linked.
Mode does not change from MAN; target mode is MAN, AUTO, or Cas.	Check Input blocks.	The status of IN is Bad, not connected.
Mode does not go to Cas, target mode is Cas.	Check Upstream block.	The upstream block cannot not able to complete cascade initialization for some reason. Assure that **BKCAL_OUT** is connected to **BKCAL_IN** of the upstream block.
Value of output does not make sense	Check Cascade Initialization	Assure that the output can move an actuator.
Block alarms are not reported	Read **FEATURE_SEL**	**Reports** are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Input Selector block

Table 26: Input Selector block

Problem Cause	Things to check	Recommended Action
Input Selector block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED** .	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE**: If the mode is set to OOS for maintenance then do not change the mode to AUTO.
	Schedule	Block is not scheduled and therefore cannot execute to go to Target Mode. Schedule the block to execute.
Input Selector block mode is in OOS mode with Block configuration Error.	Check **SELECT_TYPE**	**SELECT_TYPE** must be set to a valid value and cannot be left at 0.
Status of output is Bad.	Check Inputs	Make sure at least one input has status as good.
	Check **OP_SELECT**	**OP_SELECT** is not set to 0 (or it is linked to an input that is not used), and it points to an input that is Bad.
	Check **MIN_GOOD**	Make sure that value entered in **MIN_GOOD** is greater or equal to actual number of Good inputs.
Block alarms are not reported.	Read **FEATURE_SEL.**	Reports are not selected in **FEATURE_SEL**. If features do not include Reports then the host must poll for alarms.
	Read **LIM_NOTIFY**.	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Arithmetic block

Table 27: Arithmetic block

Problem Cause	Things to check	Recommended Action
Arithmetic block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, set **MODE_BLOCK.TARGET** to AUTO. **NOTE**: If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Mode does not change from OOS.	Configuration error.	**BLOCK_ERR** shows the Block Configuration Error condition, since **ARITH_TYPE** is not set.
Value of output is incorrect	Error in configuration.	Ensure that engineering units are correct for the computation. If that fails, see section 3.
Block alarms are not reported.	Read **FEATURE_SEL**.	Reports are not selected in **FEATURE_SEL**. If features do not include reports then the host must poll for alarms.
	Read **LIM_NOTIFY**.	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Output Splitter block

Table 28: Output Splitter block

Problem Cause	Things to check	Recommended Action
Arithmetic block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, set **MODE_BLOCK.TARGET** to AUTO. **NOTE**: If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Mode does not change from OOS.	Configuration error.	**BLOCK_ERR** shows the Block Configuration Error condition. This could be because The block **IN_ARRAY** is not configured correctly or **LOCKVAL** is not set to a valid value
Value of output is incorrect	Error in configuration.	Ensure that engineering units are correct for the computation. Also check if IN_ARRAY and OUT_ARRAY are configured correctly.
Block alarms are not reported.	Read **FEATURE_SEL**.	Reports are not selected in **FEATURE_SEL**. If features do not include reports then the host must poll for alarms.
	Read **LIM_NOTIFY**.	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Discrete Input block

Table 29: Discrete Input block

Problem Cause	Things to check	Recommended Action
Discrete Input block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, set **MODE_BLOCK.TARGET** to AUTO. **NOTE**: If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Mode does not change from OOS.	Configuration error.	**BLOCK_ERR** shows the Block Configuration Error condition because no valid CHANNEL selection is made
Value of output is incorrect	Error in configuration.	Ensure that the right Channel is selected for the desired Limit Switch, Also check if the Limit switch selected is correctly configured in the temperature transducer block.
Block alarms are not reported.	Read **FEATURE_SEL**.	Reports are not selected in **FEATURE_SEL**. If features do not include reports then the host must poll for alarms.
	Read **LIM_NOTIFY**.	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Troubleshooting the Signal Characterizer block

Table 30: Signal Characterizer block

Problem cause	Things to check	Recommended action
Signal characterizer block mode is in OOS and does not change to AUTO mode.	Read **MODE_BLOCK.PERMITTED**.	Add AUTO mode to **MODE_BLOCK.PERMITTED**.
	Read **MODE_BLOCK. ACTUAL** of Resource block.	If necessary, Set **MODE_BLOCK.TARGET** to AUTO. **NOTE**: If the mode is set to OOS for maintenance, then do not change the mode to AUTO.
Mode does not change from OOS	Configuration error.	**BLOCK_ERR** shows the Block Configuration Error condition, due to array configuration errors.
Value of output is incorrect	Error in X or Y array.	See section 3.
Block alarms are not reported.	Read **FEATURE_SEL**	**Reports** are not selected in **FEATURE_SEL**. If features do not include reports then the host must poll for alarms.
	Read **LIM_NOTIFY**	Set **LIM_NOTIFY** to a value higher than zero, but not higher than **MAX_NOTIFY**.

Resolving the block configuration errors

Table 31 lists the parameters of all the blocks that can cause the status bit of Block Configuration Error to be set in their respective **BLOCK_ERR** parameters. The following table provides the initial values and the valid range for the parameters.

Table 31: Resolving block configuration errors

Parameter	Initial Value	Valid Range	Corrective Action
ALERT_KEY	0	non-zero	Initial Value is a configuration error. Set value to non-zero number.
SIMULATE	1 (disabled)	1-2 (disabled - enabled)	Set value in valid range.
XD_SCALE	0 to 100 inches of water	EU_100 > EU_0, UNITS_INDEX matches output of transducer block	Set values to valid range(s).
OUT_SCALE	0 to 100 inches of water	EU_100 > EU_0	Set values to valid range.
CHANNEL	0	1-2	Initial Value is a configuration error. Set value to valid range.
L_TYPE	0 (Uninitialize)	1,2,3 (direct, indirect, sq. root)	Initial Value is a configuration error. Set value to valid range.
PV_FTIME	0	0-200	Set value to valid range.
ALARM_HYS	0.5 (%)	0-50 (%)	Set value to valid range.
HI_HI_PRI, HI_PRI, LO_LO_PRI, LO_PRI	0	0-15	Set value to valid range.
HI_HI_LIM, HI_LIM	+INF	+INF or within OUT_SCALE range	Set value to valid range.
LO_LIM, LO_LO_LIM	-INF	-INF or within OUT_SCALE range	Set value to valid range.
BYPASS	0	1:OFF, 2:ON	Initial value is a configuration error. Set value in valid range.
SHED_OPT	0	1-8 see Shed Options in the FF specs.)	Initial value is a configuration error. Set value in valid range.
HI_HI_LIM HI_LIM	+INF +INF	PV_SCALE, +INF	Values must be set in rank order. For example, LO_LIM > LO_LO_LIM but < HI_LIM etc.
LO_LIM LO_LO_LIM	-INF -INF	PV_SCALE, -INF	Values must be set in rank order.

Parameter	Initial Value	Valid Range	Corrective Action
OUT_HI_LIM OUT_LO_LIM	100 0	OUT_SCALE +/- 10%	Verify that OUT_HI_LIM > OUT_LO_LIM.
SP_HI_LIM SP_LO_LIM	100 0	PV_SCALE +/- 10%	Verify that SP_HI_LIM > SP_LO_LIM.

6.4 Device Diagnostics

STT850 FF temperature transmitter memory

The transmitter contains a number of areas of memory. An EEPROM provides a non-volatile memory area for static and non-volatile parameter values. The transmitter also contains areas of RAM and ROM.

Performing diagnostics in the background

Block objects (Resource, Transducer and Function blocks), the communications stack and other device objects, each of them have an allotted area of memory for their corresponding database. Diagnostic routines are performed in the background during device operations that checks the integrity of these individual databases. When a failure is detected, a status bit is set in the **BLOCK_ERR** parameter in the appropriate block object. Diagnostic checks are performed continuously on the device functional databases of the transmitter application shown in Table 32.

Table 32: Diagnostics

Device Functional Area	Location
Block object database (DB)	RAM and EEPROM
Communication stack database (DB)	EEPROM
Boot ROM	ROM
Program ROM	ROM
Trend and link object databases (DB)	ROM

BLOCK_ERR parameter

BLOCK_ERR parameter shows diagnostic faults of hardware and software components within the transmitter. Each block object in the transmitter device application contains a **BLOCK_ERR** parameter. **BLOCK_ERR** is actually a bit string, which provides a means to show multiple status or error conditions. A status message identifying the fault can be viewed by accessing the parameter. Table 3 shows the bit mapping of the **BLOCK_ERR** parameter.

Transmitter Diagnostics

Transmitter faults are grouped into one of these three diagnostic categories and could cause the following results:

1. **Non-Critical Fault** — Transmitter continues to calculate PV output.
2. **Critical Fault** — Transmitter drives PV output to failsafe state.
3. **Block Configuration Errors** — Incorrect parameter values causes the transmitter to generate a fault, for example, **BLOCK_ERR** or **MODE_BLK** = OOS.

A description of each condition in each category is provided in Table 33, Table 34, and Table 35. The condition is described, a probable cause is stated and a recommended corrective action is given for each fault.

Function Block Faults

Checking the status and values of key block parameters helps in identifying the type of function block fault whether it is critical or non-critical. Table 33 helps in identifying the type of function block fault and provides corrective action to restore normal operation.

Table 33: Identifying Critical and Non-critical Function block faults

Block. Parameter	Value	Fault Type	Action
AI.OUT =	Bad/sensor failure	Critical	See **AI.BLOCK_ERR** for message. See Table 3 for details on **BLOCK_ERR**. See **BLOCK_ERR** of all blocks in device for message. See Table 35.
STATUS =	Bad/device failure	Critical	See **AI.BLOCK_ERR** for message. See Table 3. See **BLOCK_ERR** of all blocks in device for message. See Table 35
	Good/constant Uncertain	Non-critical	See Table 34
AI.ALARM_SUM. CURRENT =	Block alarm	Critical/ Non-critical	See **BLOCK_ERR** of all blocks in the device in Table 3.
	Process alarm	Non-critical	See Table 34.
All Blocks **BLOCK_ERR**=	Block Configuration Error (1)	Non-critical	Check the value of all configurable parameters in the block and correct if necessary. See Resolving the block configuration errors.
See Table 3 for description of **BLOCK_ERR** (messages)	Simulation Active (3)	Non-critical	Set "simulate jumper" to "N" on the electronics board, and set the **ENABLE_DISABLE** field to “1” of the **SIMULATE** parameter. See section 6.5.
	Input Failure/Process Variable has Bad Status (7)	Critical	Write Processor or (4) to **RESTART** parameter of resource block. If failure continues, replace the sensor board.
	Memory Failure (9)	Critical	Set Resource block to OOS.
	Lost Static Data (10)	Critical	Write Processor or (4) to RESTART parameter.
	Lost NV Data (11)	Critical	Wait for 10 seconds.
	Readback Check Failed (12)	Critical	See Critical Fault NOTE.

Block. Parameter	Value	Fault Type	Action
	Out-of-Service (15)	Non-critical	Write proper mode to **MODE_BLK** parameter.
Unable to write values to valid device parameters.		Configuration Error	See “Resolving the block configuration errors”.

ATTENTION

Depending on the fieldbus interface application, device operating status and parameter values may appear as text messages. The text in the table is typical of values or messages seen when using the NI-FBUS configurator.

Critical Fault

In the case of a critical fault due to Memory Failure, NV/Static data loss or the readback check failure, writes to the **RESTART** parameter twice, for the transmitter to fully recover from the fault condition. Therefore:

1. Write “4” or “restart processor” to **RESTART** parameter of resource block.
2. Wait until communication is established.
3. If the fault occurs again, repeat the write to the **RESTART** parameter.
4. If the fault occurs again, replace the transmitter communication module.

Note that if a ROM error (Memory Failure) occurs in the resource block, it may take up to 10 seconds for the fault to reappear.

Table 34 summarizes the conditions that could cause a non-critical fault in the transmitter along with recommended actions to correct the fault.

Table 34: Summary of Function blocks Non-critical Faults

Problem/Fault	Probable Cause	Recommended Action
AI block is executing, but status of OUT parameter is: Good::[alarm status]:Constant	AI block is in Manual mode.	Write AUTO to **MODE_BLK** parameter of AI block.
AI block is executing, but status of OUT parameter is: Uncertain::[alarm status]: inaccurate	Transducer block parameter **CAL_SOURCE** = NONE, or a value of “1” (using default characterization values).	Recalibrate transmitter See Calibration
	Excess span correction - correction at **CAL_POINT_HI** is greater than 5% of **CAL_POINT_HI**.	Recalibrate transmitter. See Calibration
	Excess zero correction - correction at **CAL_POINT_LO** is greater than 5% of the "URL" (characterized range of the sensor).	Recalibrate transmitter. See Calibration for Zero Correct
	PV value of transducer block is outside range of **XD_SCALE**. When AI block **CHANNEL** = 1(OR) **OUT** value of AI block is outside of **OUT_SCALE** range.	Sensor board may have been damaged. Check the transmitter for accuracy and linearity. Replace the sensor board and recalibrate, if needed.
AI block is executing, but status of OUT parameter is: One of the following AI alarms is active in **ALARM_SUM.CURRENT**	HI_HI, HI, LO, LO_LO - **OUT** has crossed the corresponding limit HI_HI_LIM, HI_LIM, LO_LIM, LO_LO_LIM, and is either still past the limit or is in the hysteresis range. **ALARM_HYS** is the percentage of **OUT_SCALE** that is used for alarm hysteresis.	Reduce the value or increase limits.
	Block alarm.	Check **BLOCK_ERR** for status bit. See Table 3

Table 35 summarizes the conditions that could cause a critical fault in the transmitter along with recommended actions to correct the fault.

Table 35: Summary of Function blocks Critical Faults

Problem/Fault	Probable Cause	Recommended Action
AI block is executing, but status of output is: Bad:[alarm status]: sensor failure	One of the FAIL conditions in Field Diagnostics has got Set. .	If the diagnostics is related to input being open, check the connections as per the connections diagram. If the failure still exists, write "4" or “restart processor” to **RESTART** parameter of resource block. If the failure persists and sensor related, replace the sensor board if the. If the failure persists and communication board related, replace the communication board.
AI block is executing, but status of output is: Bad::[alarm status]: device failure	Sensor board has stopped communicating with the communication board.	Write "4" “or “restart processor” to **RESTART** parameter of resource block. If failure is still present, replace communication board.

6.5 Understanding simulation mode

About simulation mode jumper

If the process is not running, a simulation mode is available in the transmitter which aids in system debug. When simulation mode is enabled, the **SIMULATE** parameter in the AI and DI blocks provide a user-selected value as the input to the AI or DI block.

Setting simulation jumper

A hardware jumper on the Communication board is set to enable or disable the **SIMULATE** parameter. See Figure 25 for jumper location.

Table 36 shows how to set the simulation jumper on the Communication board.

Figure 25: Simulation Jumper Location on Communication Board

Table 36: Setting the Simulation Jumper

To	Set the Jumper to:
Disable the **SIMULATE** parameter. (Set transmitter for normal operation.)	"OFF" position on the Communication board. 
Enable the **SIMULATE** parameter. (For testing or debugging purposes.)	"ON" position on the Communication board. 

Enabling simulation mode

The **SIMULATE** parameter in AI and DI blocks are enabled by setting the hardware simulation jumper to the "ON" position.

In addition, the AI block **SIMULATE** parameter must be set to the following values:

- **SIMULATE.STATUS** = Good::[alarm status]:constant (suggested setting)
- **SIMULATE.SIMULATE_VALUE** = (supplied by user) Used as the input to the AI block.
- **SIMULATE.ENABLE_DISABLE** = Active Enabled.

Simulation mode truth table

Table 37 shows the states of the simulation jumper and **SIMULATE** parameter shows how to activate the simulation mode.

Table 37: Simulation Mode Truth Table

When the Simulation Jumper on Transducer board is set to:	and the SIMULATE Enable_Disable is set to:	
	(Disabled)	**(Active)**
"OFF" Position	Simulation Disabled	Simulation Disabled
"ON" Position	Simulation Disabled	Simulation Active

Setting AI or DI block mode

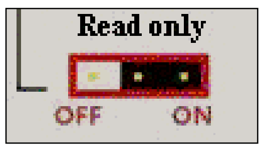
To connect the AI or DI block input to the output, the AI block must be in AUTO mode.

6.6 Understanding write protection

The hardware and software write lock features are controlled using the FEATURE_SEL parameter in the resource block. The software write lock feature can be enabled, only if the hardware write lock feature is disabled. If the software write lock feature is enabled without disabling the hardware write lock feature, then the software write lock feature gets disabled automatically. The hardware write lock feature must be enabled before placing the hardware write lock jumper in the On position. If the hardware write lock feature is selected with the hardware jumper being enabled, the selection is rejected. See Figure 25 for jumper location.

For more information on write protection, see Table 38.

Table 38: Write Lock

To	Set the Jumper to:
Disable the Read and Write lock. (In this mode, perform Read and Write operation.)	“OFF” position on the Communication board. 
Enable the Write lock. (In this mode, read operation can be performed, but the write operation is disabled.)	“ON” position on the Communication board. 

Sales and Service

For application assistance, current specifications, pricing, or name of the nearest Authorized Distributor, contact one of the offices below.

ASIA PACIFIC

Honeywell Process Solutions,
(TAC) hfs-tac-support@honeywell.com

Australia
Honeywell Limited
Phone: +(61) 7-3846 1255
FAX: +(61) 7-3840 6481
Toll Free 1300-36-39-36
Toll Free Fax:
1300-36-04-70

China – PRC - Shanghai
Honeywell China Inc.
Phone: (86-21) 5257-4568
Fax: (86-21) 6237-2826

Singapore
Honeywell Pte Ltd.
Phone: +(65) 6580 3278
Fax: +(65) 6445-3033

South Korea
Honeywell Korea Co Ltd
Phone: +(822) 799 6114
Fax: +(822) 792 9015

EMEA

Honeywell Process Solutions,
Phone: + 80012026455 or
+44 (0)1344 656000

Email: (Sales)
FP-Sales-Apps@Honeywell.com
or
(TAC)
hfs-tac-support@honeywell.com

AMERICAS

Honeywell Process Solutions,
Phone: (TAC) 1-800-423-9883 or
215/641-3610
(Sales) 1-800-343-0228

Email: (Sales)
FP-Sales-Apps@Honeywell.com
or
(TAC)
hfs-tac-support@honeywell.com

Specifications are subject to change without notice.

For more information
To learn more about SmartLine Tempertaure Transmitters, visit
www.honeywellprocess.com/smartline-temperature
Or contact your Honeywell Account Manager

Process Solutions
Honeywell
1250 W Sam Houston Pkwy S
Houston, TX 77042

Honeywell Control Systems Ltd
Honeywell House, Skimped Hill Lane
Bracknell, England, RG12 1EB

Shanghai City Centre, 100 Jungi Road
Shanghai, China 20061

Honeywell

34-TT-25-07 Rev.2
March 2016

www.honeywellprocess.com