ABB MEASUREMENT & ANALYTICS | OPERATING INSTRUCTION ADDENDUM

# 266 with FOUNDATION Fieldbus Communication

## Pressure transmitters

Engineered solutions for all applications

**Measurement made easy**

266 models

## Introduction

The 2600T family provides comprehensive range of top quality pressure measurement products, specifically designed to meet the widest range of applications ranging from arduous conditions in offshore oil and gas to the laboratory environment of the pharmaceutical industry.

## For more information

Further publications for 2600T series pressure products are available for free download from www.abb.com/pressure

# 266 Models - FOUNDATION Fieldbus

## Table of contents

# Appendix A – Device Data Blocks

The device parameters are listed in the following tables. You can access the parameters by means of the index number.
The individual blocks each contain standard parameters, block parameters and manufacturer-specific parameters. If you use the DD based configuration tools as an operating program, input screens are available as a user interface.

### General explanatory remarks

**Object Type**

Object type for the parameter value.

**S** – Simple variable.
**R** – Record.
**A** – Array of simple variables

**Data Type** – Data type for the parameter value.
**Name** – Simple variable or array.
**DS-n** – Data structure (Record) of index n.
**Storage Class –** Class of memory required

**S** – Static. Writing to the parameter changes the static revision counter ST_REV.
**N** – Non-volatile parameter which must be remembered through a power cycle, but which is not under the static update code.
**D** – Dynamic. The value is calculated by the block, or read from another block.

**Size** – Number of octets.

### Analogue variable format

The output of each AI block as well as many variables calculated and available from the different blocks of the transmitter is composed of 5 bytes. The Variable is of 32 bit size in Floating Point format (4 bytes) plus a Status Byte (1 Byte).

### Variable format - Floating Point Format IEEE-754

| Byte n | | | | | | | | Byte n+1 | | | | | | | | Byte n+2 | | | | | | | | Byte n+3 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | | | | | | | Bit 7 | Bit 6 | | | | | | | Bit 7 | | | | | | | | Bit 7 | | | | | | | |
| S | $2^{7}$ | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-8}$ | $2^{-9}$ | $2^{-10}$ | $2^{-11}$ | $2^{-12}$ | $2^{-13}$ | $2^{-14}$ | $2^{-15}$ | $2^{-16}$ | $2^{-17}$ | $2^{-18}$ | $2^{-19}$ | $2^{-20}$ | $2^{-21}$ | $2^{-22}$ | $2^{-23}$ |
| | EXPONENT | | | | | | | MANTISSA | | | | | | | | MANTISSA | | | | | | | | MANTISSA | | | | | | | |

Example: 40 F0 00 00 (hex) = 0100 000 111 000 000 000 000 000 (binary)
Calculation: Value = $(-1)^{S} * 2^{(Exponent - 127)} * (1 + Mantissa)$
Value = $(-1)^{0} * 2^{(129 - 127)} * (1 + 2^{-1} + 2^{-2} + 2^{-3})$
Value = 1 * 4 * (1 + 0.5 + 0.25 + 0.125) = 7.5

### Status

The Status byte is the fifth byte of any out value and represents the Quality of the variable. Each Transducer and Function Block produces a specific set of Status Bytes.

**Important**
Refer to the specific Block in order to see which Status bytes it produces

# Device Application Process (DAP) blocks

# Resource Block (RB)

### Overview

This block contains data that is specific to the hardware that is associated with the resource. All data is modelled as Contained, so there are no links to this block. The data is not processed in the way that a function block processes data, so there is no function schematic. This parameter set is intended to be the minimum required for the Function Block Application associated with the resource in which it resides. Some parameters that could be in the set, like calibration data and ambient temperature, are more appropriately part of their respective transducer blocks. The ITK_VER parameter identifies the version of the Interoperability Tester used by the Fieldbus Foundation in certifying the device as interoperable.

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Notes | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | | | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / OOS | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / OOS | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | 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. | |
| | | | | | | | | Bit 3 = Simulate Active | The Switch 4 of the electronics has been set in ON position in order to enable the Simulation. |
| | | | | | | | | Bit 6 = Device Needs Maintenance Soon | SEE its correspondence to Device root errors in the section “Diagnostic, Block_err and Status Byte” from page 29 |
| | | | | | | | | Bit 11 = Lost NV Data | |
| | | | | | | | | Bit 15 = Out of Service | The **MODE_BLK_ACTUAL** = Out of Service. The **MODE_BLK_ACTUAL** of all the other blocks is forced to Out of Service too |
| 7 | RS_STATE | | R | S | U8 | 8 | D | State machine of the function block application. | |
| 8 | TEST_RW | | RW | R | DS-85 | 112 | D | Read/Write test parameter – used only for conformance testing. | |
| 9 | DD_RESOURCE | | R | S | V_STR | 32 | S | String identifying the tag of the resource, which contains the Device Description for this resource. | |
| 10 | MANUFAC_ID | | R | S | U32 | 4 | S | Manufacturer Identification number – used by an interface device to locate the DD file for the resource | ABB = 0x000320 |
| 11 | DEV_TYPE | | R | S | U16 | 2 | S | Manufacturer’s model number associated with the resource – used by interface devices to locate the DD file for the resource. | 2600T 266 PdP = 0x0007 |
| 12 | DEV_REV | | R | S | U8 | 1 | S | Manufacturer’s revision number associated with the resource – used by interface devices to locate the DD file for the resource. | 0x01 |
| 13 | DD_REV | | R | S | U8 | 1 | S | Revision of the DD associated with the resource – used by interface devices to locate the DD file for the resource. | 0x01 |
| 14 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | |
| 15 | HARD_TYPES | | R | S | B_STR | 2 | S | Bit 0 | Scalar Input | The type of Hardware available as channel numbers |
| 16 | RESTART | | R/W | S | U8 | 1 | D | 0 | Uninitialized | |
| | | | | | | | | 1 | Run | |
| | | | | | | | | 2 | Restart resource | |
| | | | | | | | | 3 | Restart with default | |
| | | | | | | | | 4 | Restart process | |
| | | | | | | | | 5 | Special Restart | See also **SPECIAL_RESTART** in the block mapping |
| | | | | | | | | 6 | Special Operations | See also **SPECIAL_OPERATION** in the block mapping |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Notes | | |
|---|---|---|---|---|---|---|---|---|---|
| 17 | FEATURES | R | S | B_STR | 2 | S | Used to show supported resource block options | | |
| 18 | FEATURES_SEL | RW | S | B_STR | 2 | S | Used to select resource block options. For the 266 PdP they are: | | |
| | | | | | | | Bit 1 | Reports Supported | |
| | | | | | | | Bit 2 | Fault State Supported | |
| | | | | | | | Bit 3 | SW Write Lock Supported | |
| | | | | | | | Bit 4 | HW Write Lock Supported | |
| | | | | | | | Bit 10 | Multi-bit Alarm (Bit-Alarm) Support | |
| 19 | CYCLE_TYPE | R | S | B_STR | 2 | S | Identifies the block execution methods for this resource | | |
| 20 | CYCLE_SEL | RW | S | B_STR | 2 | S | Bit 1 | Scheduled | Used to select the block execution methods for this resource. |
| | | | | | | | Bit 2 | Completion of block execution | |
| 21 | MIN_CYCLE_T | R | S | U32 | 4 | S | Time duration of the shorted cycle interval of which the resource is capable. | | |
| 22 | MEMORY_SIZE | R | S | U16 | 2 | S | Available configuration memory in the empty resource. To be checked before attempting a download | | |
| 23 | NV_CYCLE_TIME | R | S | U32 | 4 | S | Minimum time interval for writing copies of NV parameters to non-volatile memory. Zero means it will be never automatically copied. | | |
| 24 | FREE_SPACE | R | S | FLT | 4 | D | Percent of memory available for further configuration. Zero in a preconfigured device | | |
| 25 | FREE_TIME | R | S | FLT | 4 | D | Percent of the block processing time that is free to process additional blocks. | | |
| 26 | SHED_RCAS | RW | S | U32 | 4 | S | Time duration at which to give up on computer writes to function block Rcas locations. Shed from Rcas shall never happen when Shed_Rcas = 0 | | |
| 27 | SHED_ROUT | RW | S | U32 | 4 | S | Time duration at which to give up on computer writes to function block Rout locations. Shed from Rout shall never happen when Shed_Rout = 0 | | |
| 28 | FAULT_STATE | R | S | U8 | 1 | N | Fault State | | |
| 29 | SET_FSTATE | RW | S | U8 | 1 | D | Set Fault State | | |
| 30 | CLR_FSTATE | RW | S | U8 | 1 | D | Clear Fault State | | |
| 31 | MAX_NOTIFY | R | S | U8 | 1 | S | Maximum number of unconfirmed alert notify messages possible | | |
| 32 | LIM_NOTIFY | RW | S | U8 | 1 | S | Maximum number of unconfirmed alert notify messages allowed | | |
| 33 | CONFIRM_TIME | RW | S | U32 | 4 | S | The min time between retries of alert report. Retries shall not happen when Confirm_Time = 0 | | |
| 34 | WRITE_LOCK | RW | S | U8 | 1 | S | 1 | Unlocked (default), | |
| | | | | | | | 2 | Locked | If set, no writes are allowed except to clear Write_Lock. Block inputs will continue to be updated |
| 35 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | |
| 36 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed | | |
| 37 | ALARM_SUM | RW | R | DS-74 | 8 | mix | The alert status associated to the function block | | |
| 38 | ACK_OPTION | RW | S | B_STR | 2 | S | 0 | Auto Ack Disabled (default) | Selection of whether alarms associated the function block will be automatically acknowledged |
| | | | | | | | 1 | Auto Ack Enabled | |
| 39 | WRITE_PRI | RW | S | U8 | 1 | S | Priority of the alarm generated by clearing the write_lock | | |
| 40 | WRITE_ALM | RW | R | DS-72 | 13 | D | This alert is generated if the write_lock parameter is cleared | | |
| 41 | ITK_VER | R | S | U16 | 2 | S | Major revision number of the Interoperability test case used in certifying this device as interoperable. See the ITK version used for the 266 PdP registration from www.fieldbus.org | | |

| Enhanced parameters | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Idx** | **Parameter** | | | **Data** | **Type** | **Size** | **Storage** | **Description / Range / Selections / Notes** | |
| 42 | FAIL_ACTIVE | | R | S | B_STR | 4 | D | Active error conditions of Failure category | According to the NAMUR NE107 Categories. Bit = 0 – No Error active Bit = 1 – Error active – It is bit string, and multiple conditions may be shown. SEE its bit MAPPING DETAIL in the dedicated section |
| 43 | OFFSPEC_ACTIVE | | R | S | B_STR | 4 | D | Active error conditions of Out of Specification category | |
| 44 | MAINTENANCE_ACTIVE | | R | S | B_STR | 4 | D | Active error conditions of Maintenance category | |
| 45 | CHECK_ACTIVE | | R | S | B_STR | 4 | D | Active error conditions of Check Function category. | |
| 46 | RECOMMEND_ACTION | | R | S | U16 | 2 | D | Numeric Code defining the corrective action to be taken for the problem solution. When the Device's DD has been imported in the Hosts it converts the numeric code into a Textual info comprehensible for the user | |
| 47 | EXTENDED_ACTIVE | | R | S | B_STR | 4 | D | Some of the Active Errors of Fail, Maint, Out of Spec, Function Check classification could be produced by more root causes that can be seen by reading this parameter. SEE its bit MAPPING DETAIL in the dedicated section | |
| 48 | DIAGNOSIS_HISTORY | | R | S | B_STR | 4 | D | The bit associated at each error condition is permanently set after the condition became set at least one time Its bit mapping is the same as for x_ACTIVE. SEE its bit MAPPING DETAIL in the dedicated section | |
| 49 | DIAGNOSIS_CONDITION_IDX | | RW | S | U8 | 1 | N | The writing of an Error code in this parameter updates the DIAGNOSIS_DETAILS with details of that error. | |
| 50 | DIAGNOSIS_ DETAILS | COUNTER | RW | S | U16 | 2 | D | After the idx 49 has been written with an error code, all its details are returned by reading this parameter | N° of times the specified error has been detected during the device's life |
| | | TIME_COUNTER | R | S | TIME_DIF | 6 | D | | Sum of all the periods of time the specified error has been active as ddd/hh/mm/ss |
| | | LAST_TIME | R | S | DATE_S | 8 | D | | Time of when the error became active the last occurrence as ddd/hh/mm/ss |
| 51 | DIAGNOSIS_SIMULATION_ STATUS | | RW | S | U8 | 1 | N | 0: | Simulation disabled (default) |
| | | | | | | | | 1: | Simulation enabled — Enable the writing of conditions to be simulated to DIAGNOSIS_SIMULATION. When error condition is active, this parameter clarify if it is real or simulated |
| 52 | DIAGNOSIS_SIMULATION | | RW | S | B_STR | 4 | N | Allow the Simulation of any individual error condition. Only one error per time can be simulated. Refer to the dedicated section | |
| 53 | DIAGNOSIS_MASK | | RW | S | B_STR | 4 | N | Allow to Mask of one or more error conditions. the Critical errors cannot be masked | |
| 54 | DEVICE_SER_NUM | | R | S | V_STR | 16 | N | Serial Number of the Transmitter as printed on the main Type Plate (on the housing). | |
| 55 | CB_FW_REVISION | | R | S | V_STR | 8 | N | Electronics Software Revision | xx.yy.zz |
| 56 | CB_HW_REVISION | | R | S | V_STR | 8 | N | Electronics Hardware Revision | xx.yy.zz |
| 57 | FE_FW_REV | | R | S | V_STR | 8 | N | Sensor Software Revision | xx.yy.zz |
| 58 | FE_HW_REV | | R | S | V_STR | 8 | N | Sensor Hardware Revision | xx.yy.zz |
| 59 | MESSAGE | | RW | S | O_STR | 32 | S | Message | Reserved to the User. |
| 60 | DESCRIPTOR | | RW | S | O_STR | 32 | S | Descriptor | |
| 61 | INSTALLATION_DATE | | RW | S | O_STR | 16 | S | Installation date | |
| 62 | SPECIAL_RESTART | | RW | S | B_STR | 4 | S | Bit 11 | AR pre-setting — After the selection of one or more blocks from this list and its writing to SPECIAL_RESTART, then the operation is really executed writing the command "Special Restart" in the RB_RESTART. All the selected Blocks are set with a pre-defined configuration allowing their switching to AUTO Mode. PS: The Function Blocks must have been previously instantiated into a Function Block Application otherwise cannot move out from OOS. |
| | | | | | | | | Bit 12 | IS pre-setting |
| | | | | | | | | Bit 14 | IT pre-setting |
| | | | | | | | | Bit 17 | SC pre-setting |
| | | | | | | | | Bit 23 | PID pre-setting |
| | | | | | | | | Bit 25 | CS pre-setting |
| | | | | | | | | Bit 29 | AI pre-setting |
| | | | | | | | | Bit 30 | RB pre-setting |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Notes | | |
|---|---|---|---|---|---|---|---|---|---|
| 63 | SPECIAL_OPERATION | RW | S | U8 | 1 | N | 0 | Do nothing | See the dedicated section |
| | | | | | | | 8 | Reset Device Configuration to Default Configuration | |
| | | | | | | | 9 | Reset PdP Sensor Trimming to Factory Trim | |
| | | | | | | | 12 | Reset PdP Sensor Trimming to User Trim | |
| | | | | | | | 10 | Reset Static Press Sensor Trimming to Factory Trim | |
| | | | | | | | 13 | Reset Static Press Sensor Trimming to User Trim | |
| | | | | | | | 11 | Reset Sensor Temp Trimming to Factory Trim | |
| | | | | | | | 14 | Reset Sensor Temp Trimming to User Trim | |
| | | | | | | | 1 | Save current Device Configuration as Default Configuration | |
| | | | | | | | 2 | Save actual PdP Sensor Trimming as PdP Factory Trim | |
| | | | | | | | 3 | Save actual Static Press Sensor Trimming as Static Press Factory Trim | |
| | | | | | | | 4 | Save actual Sensor temp Trimming as Sensor Temp Factory Trim | |
| | | | | | | | 5 | Save actual PdP Sensor Trimming as PdP User Trim | |
| | | | | | | | 6 | Save actual Static Press Sensor Trimming as Static Press User Trim | |
| | | | | | | | 7 | Save actual Sensor temp Trimming as Sensor Temp User Trim | |
| 64 | LOCAL_OPERATIONS | RW | S | U8 | 1 | N | 0: | disabled | Local operation via PUSH BUTTONS are not allowed |
| | | | | | | | 1: | enabled (default) | Local operation via PUSH BUTTONS are allowed |

## Operations

### Savings

In order to keep a valid device setting to be used as reference when a valid condition has to be recovered in case of wrong operations, it is possible save all the above calibrations as Factory or User calibrations and the complete device configuration.
The possible savings are the following and are executed in two steps:
– Selecting and writing the proper save operation in the **RB_SPECIAL_OPERATION**
– Selecting and writing in the **RB_RESTART = Special Operations**

| | |
|---|---|
| **Save Configuration as Default** | When this operation is executed, the complete device configuration is saved as default configuration at which the device returns when the Reset to Default configuration is executed. After the device has been properly configured, the user can decide to save it as a default configuration in order to recover it if necessary |
| **Save P-dP Trimming as Factory** | The P-dP Sensor calibration/trimming is saved as Factory Calibration. This operation is typically executed in the Factory after the Sensor has been calibrated to the customer's specified measuring range or, in case the customer didn't requested any measuring range, at the maximum sensor range |
| **Save Static P Trimming as Factory** | The Static P Sensor calibration/trimming is saved as Factory Calibration. |
| **Save Sensor Temp Trimming as Factory** | The Sensor Temp. calibration/trimming is saved as Factory Calibration |
| **Save P-dP Trimming as User** | The P-dP Sensor calibration/trimming is saved as User Calibration. This operation is typically executed by the user after the Sensor has been calibrated at the desired measuring range. |
| **Save Static P Trimming as User** | The Static P Sensor calibration/trimming is saved as User Calibration |
| **Save Sensor Temp Trimming as User** | The Sensor Temp. calibration/trimming is saved as User Calibration |

### Resets

The transmitter offers some reset operations executed in two steps:
– Selecting and writing the proper reset code in the **RB_SPECIAL_OPERATION**
– Selecting and writing in the **RB_RESTART = Special Operations**

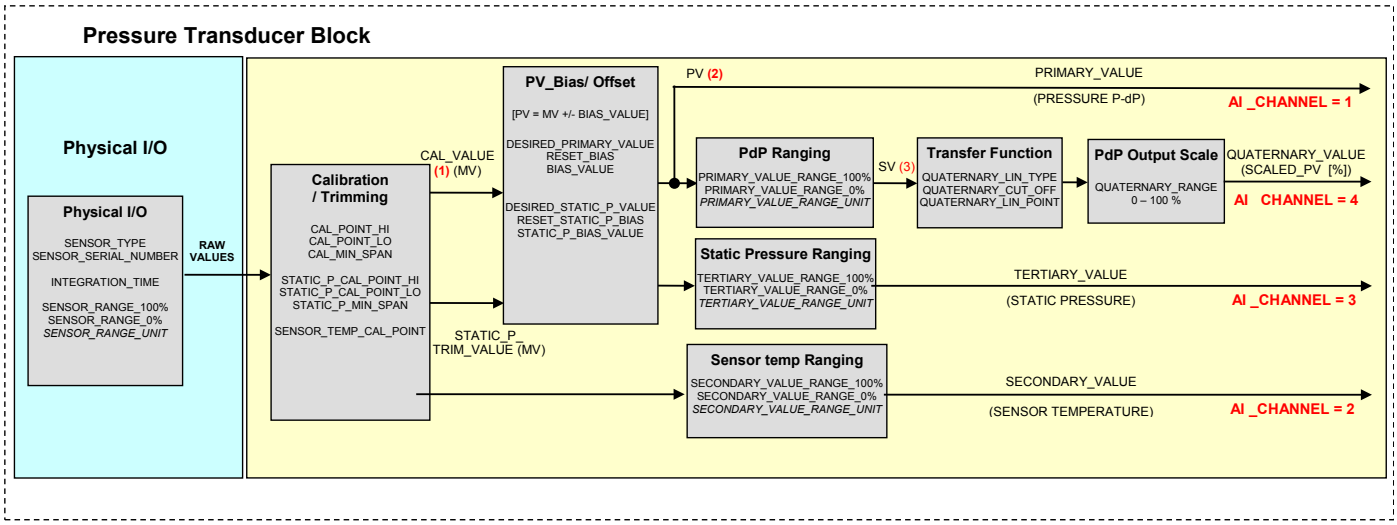
| | |
|---|---|
| **Reset Configuration to Default Values** | When this operation is executed, the complete device configuration returns to the configuration previously saved as default configuration. |
| **Reset P-dP Trimming to Factory** | Return the P-dP Sensor calibration/trimming at the calibration previously saved as Factory Calibration |
| **Reset Static P Trimming to Factory** | Return the Static Pressure Sensor calibration/trimming at the calibration previously saved as Factory Calibration |
| **Reset Sensor Temp Trimming to Factory** | Return the Sensor temperature calibration/trimming at the calibration previously saved as Factory Calibration |
| **Reset P-dP Trimming to User** | Return the P-dP Sensor calibration/trimming at the calibration previously saved as User Calibration. |
| **Reset Static P Trimming to User** | Return the Static Pressure Sensor calibration/trimming at the calibration previously saved as User Calibration |
| **Reset Sensor Temp Trimming to User** | Return the Sensor temperature calibration/trimming at the calibration previously saved as User Calibration |

# Pressure transducer block (PRTB)

## Overview

This pressure transducer block is implemented within devices whose primary process sensor has the purpose to measure pressure, or differential pressure (P-dP). In addition, at the pressure value as primary measurement, there are other variables that can be selected trough the Channel as input for the Analog Input blocks, these are the Sensor Temperature, the Static Pressure, for Differential pressure sensors only, and the Scaled PV identified respectively as Secondary, Tertiary and Quaternary variables.

## Block diagram




## Description

The Physical I/O represents the physical interface with the process and is part of the device's Pressure Transducer. The physical I/O takes care to execute the basic manufacturer device specific algorithm with the purpose to convert the raw signal representing the measured process value into a digital format. The physical I/O operations are:

– Sampling of the primary raw signal changing according the process changes.

– Validation and Elaboration of the sampled primary raw signal

– Linearization and Compensation

Result of the above operations is the RAW_VALUES produced in output of the physical I/O, see the Block Diagram, and used as input for the Pressure Transducer Block.

The first Pressure Transducer Block operation is the Calibration/trimming of the RAW_VALUES in order to adjust its digital value to match the real pressure measured by the Sensor block.

The RAW_VALUES after the calibration became the calibrated Measured Values (MV) represented by the **PRTB_CAL_VALUE** and, for differential sensors only, **PRTB_STATIC_P_TRIM_VALUE.**

These Measured Values matches and represents the real inputs sampled by the sensor and any further calculation has the scope to transform them to a Process Variables (PV).

In this perspective the first calculation applied is the elevation/suppression within the PV-Bias/Offset step executed for different reasons like the correction of the mounting position or for example in any condition where part of the measure must not be considered as part of the process.

## Equations

Once the MV and PV are calculated and available in the PRTB then it can produces different type of measurements depending by the selected **PRTB_QUATERNARY_LIN_TYPE** and applying the following formula:

| Measurement Type | QUATERNARY_LIN_TYPE | | Formula |
|---|---|---|---|
| Pressure / Level | | Linear | PRIMARY_VALUE = CAL_VALUE [MV] +/- BIAS_VALUE |
| Flow | QLT | Linear | PV = CAL_VALUE [MV] +/- BIAS_VALUE |
| | | Square root | SV = (PV – PV_RANGE_0%) / (PV_RANGE_100% - PV_RANGE_0%) |
| | | SQRT 3° pow | |
| | | SQRT 5° pow | QUATERNARY_VALUE [%] = (QLT (SV) * (QUATERNARY_100% - QUATERNARY_0%) + |
| | | Bidirectional Flow | QUATERNARY_0% |
| Volume | QLT | cylindrical lying container | PV = CAL_VALUE [MV] +/- BIAS_VALUE<br>SV = (PV – PV_RANGE_0%) / (PV_RANGE_100% - PV_RANGE_0%) |
| | | spherical container | QUATERNARY_VALUE [%] = (QLT (SV) * (QUATERNARY_100% - QUATERNARY_0%)) + QUATERNARY_0% |

## Transfer function

The transfer output functions available in the 266 Pressure Transducer Block are described in details
– Linear for differential, gauge and absolute pressure or level measurements
– Sq. Root (x) for flow measurements using restriction type primary element, like orifice plate, integral orifice, Venturi or Dall tube and similar.
– Sq. Root (x3) for open channel flow measurements using rectangular or trapezoidal weir
– Sq. Root (x5) for open channel flow measurements using V-notch (triangular) weir.
– Bidirectional Flow
– Custom linearization table
– Cylindrical lying tank
– Spherical tank
These output functions can be selected writing in **PRTB_QUATERNARY_LIN_TYPE** activated using a DD based Configuration Tool. The transfer function can be applied to the Process Variable only or also to the indication (in engineering units).

## Linear

Using this function, the relationship between the input (measured value), expressed in % of the calibrated span and the output is linear (i.e.: at 0% input, corresponds 0% output - at 50% input corresponds 50% output - and at 100% input corresponds 100% output). No further settings are possible here

## Square root

Using the Square Root function, the output (in % of the span) is proportional to the square root of the input signal in percentage of the calibrated span (i.e.: the instrument gives an analog output proportional to the rate of flow). The possibility to have the full Square Root function is given. To avoid the extremely high gain error with the input approaching zero, the transmitter output is linear with the input up with a slope of 1 up to 0.5% and then still linear with the appropriated slope to a programmable percentage value between 10 % and 20%. This option is offer in order to ensure a more stable output when the signal is close to zero avoiding errors due to the high gain of the square root. To neglect the values with the input approaching zero, the transmitter output is zero with the input up to a programmable percentage value between 0 % and 20%. This option is offer in order to ensure a more stable flow measure. This option is possible for all the listed output functions.

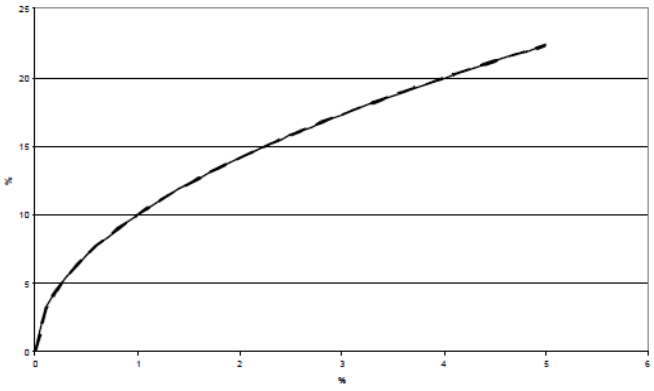



**Figure 45: Linear output**

### Square root to the 3rd power

The $x^3$ Square root Transfer function can be used for open channel (see figures on the right) flow measurement using ISO 1438 rectangular weirs (Hamilton Smith, Kindsvater-Carter, Rehbock formulas) or trapezoidal weirs (Cippoletti formulas) and ISO 1438 Venturi flumes. In these types of devices the relationship between the flow and the developed head h (the differential pressure measured by the transmitter) is proportional to h3/2 or square root of h3. Other types of Venturi or Parshall flume do not follow this relationship.

Using this function, the output (in % of the span) is proportional to the square root of the third power of the input signal in % of the calibrated span: the instrument gives an output proportional to the rate of flow calculated using the above mentioned formulas.

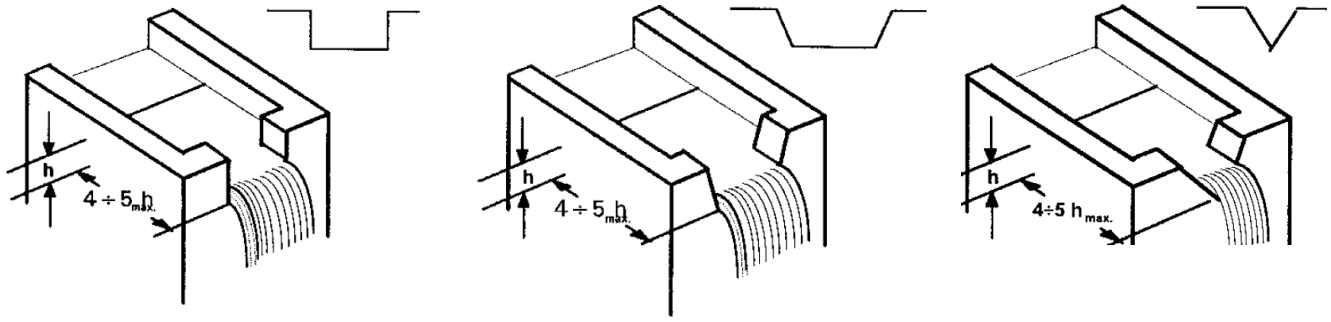



**Figure 47: Tanks (respectively rectangula weir, trapezoidal weir and V-notch weir)**

### Square root to the 5th power

The $x^5$ Square root Transfer function can be used for open channel flow measurement using ISO 1438 Vnotch (triangular) weirs (see figure on the right) where the relationship between the flow and the developed head h (the differential pressure measured by the transmitter) is proportional to h5/2 or square root of h5.

Using this function, the output (in % of the span) is proportional to the square root of the fifth power of the input signal in % of the calibrated span: the instrument (it gives an output proportional to the rate of flow calculated using the Kingsvater-Shen formula).

### Bidirectional Flow

The bidirectional function, applied to the transmitter input (x) expressed in percentage of the calibrated span, has the following form:

Output = ½ + ½ sign (x) · x ½

where “x” and “Output” should be normalized in the range 0 to 1 for calculation purpose, with the following Output meaning:

– Output = 0 means Analog out 4 mA;

– Output = 1 means Analog out 20 mA.

This function can be used for flow measurement purpose when the flow is in both the directions and the primary elements are designed to perform this type of measure.

As an example, if we have a bidirectional flow measurement application with the following data: Max reverse flow rate: -100 l/h

Max flow rate: +100 l/h

The differential pressure generated by the flow primary is for the maximum flow rate 2500 mmH2O, for the max reverse flow rate 2500 mmH2O. The transmitter will have to be configured as follows: Calibrated span: 4mA = LRV = -2500mmH2O

20mA = URV = +2500mmH2O

Transfer function: Bidirectional flow. Once configured as above the transmitter will deliver: flowrate 100 l/h reverse: output= 4mA

no flowrate: output= 12mA

Flow rate 100 l/h: output= 20mA

### Cylindric lying tank

This function is used to measure the volumetric level into a cylindrical horizontal tank with flat ends. The transmitter calculates the volume from the measured filling level.

### Spherical Tank

This function is used to measure the volumetric level into a spherical tank. The transmitter calculates the volume from the measured filling level.

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Notes | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / OOS | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / OOS | The allowed modes the operator can select as Target |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | 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. | |
| | | | | | | | | Bit 0 = Other | |
| | | | | | | | | Bit 4 =Local Override | |
| | | | | | | | | Bit 7 = Sensor Failure detected by this block/process variable has a status of BAD, Sensor Failure | |
| | | | | | | | | Bit 15 = Out of Service | |
| 7 | UPDATE_EVT | | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | |
| 8 | BLOCK_ALM | | RW | R | DS-72 | 13 | D | 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 sub-code field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the sub-code has changed | |
| 9 | TRANSDUCER_DIRECTORY | | R | A | U16 | 2 | S | Directory that specifies the number and starting indices of the transducers in the transducer block | |
| 10 | TRANSDUCER_TYPE | | R | S | U16 | 2 | S | Identifies the transducer type. | TN-016 – 100 = Standard Pressure with calibration |
| 11 | XD_ERROR | | R | S | U8 | 1 | D | Transducer block error sub-code | |
| 12 | COLLECTION_ DIRECTORY | | R | A | U32 | 4 | S | Directory that specifies the number, starting indices, and the DD items IDs of the data collections in each transducer within a transducer block | |
| 13 | PRIMARY_VALUE_TYPE | | RW | S | U16 | 2 | S | Selects the type of measurement represented in **PRIMARY_VALUE** | TN-016 - Changing has no calculation effect |
| 14 | PRIMARY_VALUE | | R | R | DS-65 | 5 | D | Pressure Process (PdP) in output from PRTB and input to the AI with Channel = 1 | |
| 15 | PRIMARY_VALUE_ RANGE | | RW | R | DS-68 | 11 | S | The High and Low range limit values, the engineering units code and the number of digits to the right of the decimal point Used as input scaling for the production of the Scaled PV | |
| 16 | CAL_POINT_HI | | RW | S | FLT | 4 | S | PdP Sensor Calibration point High value expressed in **CAL_UNIT** | |
| 17 | CAL_POINT_LO | | RW | S | FLT | 4 | S | PdP Sensor Calibration point Low value expressed in CAL_UNIT | |
| 18 | CAL_MIN_SPAN | | R | S | FLT | 4 | D | PdP Sensor Calibration minimum Span value expressed in CAL_UNIT. When calibration is done, the two calibrated points (high and low) must not be too close together. | |
| 19 | CAL_UNIT | | RW | S | U16 | 2 | S | Calibration Unit. | Only Pressure Units are allowed |
| 20 | SENSOR_TYPE | | R | S | U16 | 2 | S | Type of Sensor | TN-016 – 121 = Pressure sensor unknown |
| 21 | SENSOR_RANGE | | R | R | DS-68 | 11 | S | The High and Low PdP physical sensor limits with the engineering units code (Press Only) | |
| 22 | SENSOR_SERIAL_NUMBER | | R | S | V_STR | 32 | S | Serial Number of the sensor | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Notes | |
|---|---|---|---|---|---|---|---|---|
| 23 | SENSOR_CAL_METHOD | RW | S | U8 | 1 | S | The method of last sensor calibration. ISO defines several standard methods of calibration. This parameter is intended to record that method, or if some other method was used. | |
| 24 | SENSOR_CAL_LOC | RW | S | V_STR | 32 | S | The location of last sensor calibration. | |
| 25 | SENSOR_CAL_DATE | RW | S | DATE | 7 | S | The date of the last sensor calibration. | |
| 26 | SENSOR_CAL_WHO | RW | S | V_STR | 32 | S | The name of the person responsible for the last sensor calibration. | |
| 27 | SENSOR_DIAPHRAGM_MATERIAL | RW | S | U16 | 2 | S | Defines the construction material of the isolating diaphragms. | |
| 28 | SENSOR_FILL_FLUID | RW | S | U16 | 2 | S | Defines the type of fill fluid used in the sensor | |
| 29 | SECONDARY_VALUE | R | R | DS-65 | 5 | D | Sensor Temperature in output from PRTB and input to the AI with Channel = 2 | |
| 30 | SECONDARY_VALUE_RANGE | RW | R | DS-68 | 11 | S | The High and Low Sensor Temp Limits with the engineering units code (Temp Only) | Only the Unit is changeable. The limits/ranges are automatically converted. |
| | | | | | | | **Enhanced parameters** | |
| 31 | TERTIARY_VALUE | R | R | DS-65 | 5 | D | Static Pressure in output from PRTB and input to the AI with Channel = 3 | |
| 32 | TERTIARY_VALUE_RANGE | RW | R | DS-68 | 11 | S | The High and Low Static Pressure Sensor Limits with the engineering units code (Press Only) | Only the Unit is changeable. The limits/ranges are automatically converted. |
| 33 | QUATERNARY_VALUE | R | R | DS-65 | 5 | D | Scaled_PV in in output from PRTB and input to the AI with Channel = 4 | |
| 34 | QUATERNARY_VALUE_RANGE | R | R | DS-68 | 11 | S | The High and Low Scaled_PV Limits with the engineering unit code.Used as Output Scaling for the production of the Scaled_PV | Fixed set to 0 / 100 % with 2 decimals |
| 35 | QUATERNARY_LIN_TYPE | RW | S | U8 | 1 | S | 0: Linear (default)<br>1 Square root<br>2 SQRT 3° pow<br>3 SQRT 5° pow<br>4 cylindrical lying container<br>5 spherical container<br>6: Bidirectional Flow | Transfer Function for Scaled_PV |
| 36 | QUATERNARY_VALUE_CUT_OFF | RW | S | FLT | 4 | S | 0% to 20% [default = 6%] | Valid Only for **QUATERNARY_LIN_TYPE** = Square Root or Bidirectional Flow |
| 37 | QUATERNARY_VALUE_LINEAR_POINT | RW | S | FLT | 4 | S | 0% or 5% to 20% [default = 5%] | |
| 38 | CAL_VALUE | R | R | DS-65 | 5 | D | PdP Process Value after the Calibration. Reference value to be adjusted with the calibration operations | |
| 39 | DESIDERED_PV | RW | S | FLT | 4 | D | Expressed in **PRIMARY_VALUE_UNIT** | Force the Measured Pressure to a selected Value setting an offset between Measured and Process values > PV = MV +/- BIAS > PRIMARY_VALUE = CAL_VALUE +/- BIAS_VALUE |
| 40 | RESET_BIAS | W | S | U8 | 1 | N | Reset BIAS_VALUE to 0.0 so that > PRIMARY_VALUE = CAL_VALUE | |
| 41 | BIAS_VALUE | R | S | FLT | 4 | N | Expressed in **PRIMARY_VALUE_UNIT** | Read the offset between the Measured and Process values > BIAS_VALUE = CAL_VALUE – PRIMARY_VALUE |
| 42 | STATIC_P_CAL_POINT_HI | RW | S | FLT | 4 | S | Static Pressure Calibration point High expressed in **STATIC_P_CAL_UNIT** | |
| 43 | STATIC_P_CAL_POINT_LO | RW | S | FLT | 4 | S | Static Pressure Calibration point Low expressed in **STATIC_P_CAL_UNIT** | |
| 44 | STATIC_P_CAL_MIN_SPAN | R | S | FLT | 4 | D | Expressed in **STATIC_P_CAL_UNIT.** | Static Pressure Sensor Calibration minimum Span value. When calibration is done, the two calibrated points (high and low) must not be too close together. |
| 45 | STATIC_P_CAL_UNIT | RW | S | U16 | 2 | S | Static Pressure Calibration Only Pressure Units are allowed | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Notes | |
|---|---|---|---|---|---|---|---|---|
| 46 | STATIC_P_TRIM_VALUE | R | R | DS-65 | 5 | D | Static Pressure Value after the Calibration. Reference value to be adjusted with the calibration operations | |
| 47 | MAX_WORKING_PRESSURE | RW | S | FLT | 4 | S | Expressed in **TERTIARY_VALUE_RANGE** Unit | Max Sensor Working Pressure |
| 48 | DESIDERED_STATIC_P_VALUE | RW | S | FLT | 4 | D | Expressed in **TERTIARY_VALUE_RANGE** Unit | Force the Measured Static Pressure to a selected Value setting an offset between Measured and Process value. > SP = MV +/- BIAS > TERTIARY_VALUE = STATIC_P_TRIM_VALUE +/- STATIC_P_BIAS_ VALUE |
| 49 | RESET_STATIC_P_BIAS | W | S | U8 | 1 | N | Reset BIAS_VALUE to 0.0 so that > TERTIARY_VALUE = STATIC_P_TRIM_VALUE | |
| 50 | STATIC_P_BIAS_VALUE | R | S | FLT | 4 | N | Expressed in **TERTIARY_VALUE_RANGE** Unit | Read the offset between Measured and Process values > STATIC_P_ BIAS_VALUE = STATIC_P TRIM_VALUE – TERTIARY_VALUE |
| 51 | SENSOR_TEMP_CAL_POINT | R/W | S | FLT | 4 | S | Expressed in **SECONDARY_VALUE_RANGE** Unit | Sensor Temperature Calibration Point. An adjustment of the sensor temperature is effected by writing the correct temperature value. This setting has no influence to the accuracy of the pressure measurement |
| 52 | SET_UPPER_RANGE_POINT_PV | R/W | S | U8 | 1 | N | SPAN Button emulation for Process Value | The instant measured value is written to **PRIMARY_VALUE_RANGE_100%** |
| 53 | SET_LOWER_RANGE_POINT_PV | R/W | S | U8 | 1 | N | ZERO Button emulation for Process Value. | The instant measured value is written to **PRIMARY_VALUE_RANGE_0%** |
| 54 | PARALLEL_SHIFT_PV | R/W | S | FLT | 4 | N | Shift the **PRIMARY_VALUE_RANGE** values in order to produce the desired percentage in output. | The **PRIMARY_VALUE_RANGE** span remains unchanged |
| 55 | MODULE_TYPE | R | S | U8 | 1 | | | Type of Transducer technology (piezo, inductive, capacitive) |
| 56 | SENSOR_O_RING_MATERIAL_HSP | RW | S | U16 | 2 | S | Sensor O-Ring Material | High Side |
| 57 | PROCESS_CONNECTION_TYPE_HSP | RW | S | U16 | 2 | S | Process connection type | |
| 58 | PROCESS_CONNECTION _ MATERIAL_HSP | RW | S | U16 | 2 | S | Process connection material | |
| 59 | DRAIN_VENT_MATERIAL_HSP | R/W | S | U16 | 2 | S | Drain vent Material | |
| 60 | SENSOR_O_RING_MATERIAL_LSP | R/W | S | U16 | 2 | S | Sensor O-Ring Material | Low Side |
| 61 | PROCESS_CONNECTION_TYPE_LSP | R/W | S | U16 | 2 | S | Process connection type | |
| 62 | PROCESS_CONNECTION _ MATERIAL_LSP | R/W | S | U16 | 2 | S | Process connection material | |
| 63 | DRAIN_VENT_MATERIAL_LSP | R/W | S | U16 | 2 | S | Drain vent Material | |
| 64 | GAUGE_ABS_PROC_CONNECT_MTL | R | S | U16 | 2 | N | Process connection material for Gauge or Absolute sensor types | |
| 65 | REMOTE_SEALS_TYPE_HSP | R | S | U16 | 2 | N | Remote Seal type | High Side |
| 66 | REMOTE_SEALS_FILL_FLUID_HSP | R | S | U16 | 2 | N | Remote Seal Fill Fluid | |
| 67 | REMOTE_SEALS_ISOLATOR_HSP | R | S | U16 | 2 | N | Remote Seal Isolator | |
| 68 | REMOTE_SEALS_TYPE_LSP | R | S | U16 | 2 | N | Remote Seal type | Low Side |
| 69 | REMOTE_SEALS_FILL_FLUID_LSP | R | S | U16 | 2 | N | Remote Seal Fill Fluid | |
| 70 | REMOTE_SEALS_ISOLATOR_LSP | R | S | U16 | 2 | N | Remote Seal Isolator | |

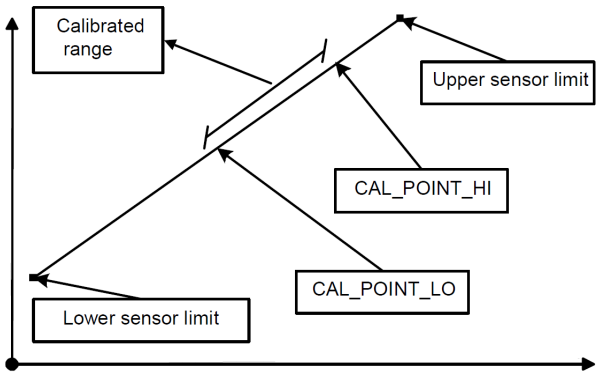
| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Notes | | |
|---|---|---|---|---|---|---|---|---|---|
| 71 | REMOTE_SEALS_NUMBER | R | S | U8 | 1 | N | 1 | One | |
| | | | | | | | 2 | Two | |
| | | | | | | | 3 | One on low side | |
| | | | | | | | 4 | One on high side | |
| | | | | | | | 251 | None | |
| 72 | PRESSURE_SIMULATION_ENABLE | R/W | S | U8 | 1 | S | 0: | Disabled/OFF | |
| | | | | | | | 1: | Enabled/ON | |
| 73 | PRESSURE_SIMULATION_VALUE | R/W | S | FLT | 4 | S | Expressed in **CAL_UNIT** | | Pressure/dP Simulation Value to be simulated in input |
| 74 | STATIC_PR_SIMULATION_ENABLE | R/W | S | U8 | 1 | S | 0: | Disabled/OFF | |
| | | | | | | | 1: | Enabled/ON | |
| 75 | STATIC_PR_SIMULATION_VALUE | R/W | S | FLT | 4 | S | Expressed in **STATIC_P_CAL_UNIT** | | Static Pressure Simulation Value to be simulated in input |
| 76 | SENSOR_TEMP_SIMULATION_ENABLE | R/W | S | U8 | 1 | S | 0: | Disabled/OFF | |
| | | | | | | | 1: | Enabled/ON | |
| 77 | SENSOR_TEMP_SIMULATION_VALUE | R/W | S | FLT | 4 | S | Expressed in SECONDARY_VALUE_RANGE.Unit | | Sensor temperature Simulation Value to be simulated in input |
| 78 | INTEGRATION_TIME | R/W | S | U8 | 1 | S | From 0.01 to 1.28 seconds<br>This parameter is available only for piezo-resistive and capacitive sensor types | | The Integration Time of the A/D converter can be changed between 0.1s and 1.28s in steps of 0.01s. The accuracy of the transmitter will be higher with a high Integration Time. The transmitter will be faster with a short Integration Time but the output will be more noisy depending on the process conditions. The output will rise linear with 10ms cycle after a step of the input value. The end value will be reached with expiration of the Integration Time. The default value of the Integration Time is 0.3s for transmitters with ≥0.04% accuracy and 1.28s for transmitter with 0.025% accuracy. The setting of the integration time is independent from the Damping.<br>The Integration Time shall be set to 1.28s for the calibration of the pressure measurement. |

### Sensor calibration

The transmitter makes available to the user some operations that can be useful during the device life cycle. These operations are supported and can be executed with the EDD based configuration tools, or also by following the instructions/descriptions below.

### Sensor trimming/calibration

The scope of the sensor trimming/calibration is to adjust and make accurate as much as possible the sensor conversion to a pressure value in digital format.
The sensors of the 266 are calibrated/trimmed in the factory to the customer's specified measuring range therefore it could be necessary change or correct the sensor calibration later on as maintenance operation.
Two points are necessary to perform a sensor calibration. Low sensor calibration point (Zero) writing in **PRTB_CAL_POINT_LO** and High sensor calibration point (Span) writing in **PRTB_CAL_POINT_HI**. The minimum distance from the two points must be greater than minimum span **PRTB_CAL_MIN_SPAN**.




### P-dP sensor low trimming

With this operation the **PRTB_CAL_VALUE** is automatically adjusted, in order to match the real value of the pressure applied in input, in the low part of the working range. The following sequence of operations is required:
– Apply a reference pressure in input using a reference pressure generator.
– Select the engineering unit of the measure in the **PRTB_CAL_UNIT** (Pressure Unit Only)
– Read the measure produced by the transmitter from the **PRTB_CAL_VALUE**.
– If this value doesn't match the pressure applied in input, enter the correct known applied pressure value in the **PRTB_CAL_POINT_LO** and write to the transmitter. This writing executes an internal algorithm that produces the new correction coefficients.
– Read again the **PRTB_CAL_VALUE** and check if its value now matches the applied pressure.

### P-dP sensor high trimming

With this operation the **PRTB_CAL_VALUE** is automatically adjusted, in order to match the real value of the pressure applied in input, in the high part of the working range. The following sequence of operations is required:
– Apply a reference pressure in input using a reference pressure generator.
– Select the engineering unit of the measure in the **PRTB_CAL_UNIT** (Pressure Unit Only)
– Read the measure produced by the transmitter from the **PRTB_CAL_VALUE**.
– If this value doesn't match the pressure applied in input, enter the correct known applied pressure value in the **PRTB_CAL_POINT_HI** and write to the transmitter. This writing executes an internal algorithm that produces the new correction coefficients.
– Read again the **PRTB_CAL_VALUE** and check if its value now matches the applied pressure.

### Static pressure low trimming

With this operation the **PRTB_STATIC_P_TRIMMED_VALUE** is automatically adjusted, in order to match the real value of Static Pressure applied at the transducer in the lower part of the range. The following sequence of operations is required:
– Select the engineering unit of the measure in the **PRTB_STATIC_P_CAL_UNIT** (Pressure Unit Only)
– Read the Static Pressure value from the **PRTB_STATIC_P_TRIMMED_VALUE**.

– If this value doesn't match the known Static Pressure applied in input at the transducer, enter the correct value in the **PRTB_STATIC_P_CAL_POINT_LO** and write to the transmitter. This writing executes an internal algorithm that produces the new correction coefficients.
– Read again the **PRTB_STATIC_P_TRIMMED_VALUE** and check if its value now matches the real Static Pressure value coefficients.

### Static pressure high trimming (for piezo dP sensor only)

With this operation the **PRTB_STATIC_P_TRIMMED_VALUE** is automatically adjusted, in order to match the real value of Static Pressure applied at the transducer in the upper part of the range. The following sequence of operations is required:
– Select the engineering unit of the measure in the **PRTB_STATIC_P_CAL_UNIT** (Pressure Unit Only)
– Read the Static Pressure value from the **PRTB_STATIC_P_TRIMMED_VALUE**.
– If this value doesn't match the known Static Pressure applied in input at the transducer, enter the correct value in the **PRTB_STATIC_P_CAL_POINT_HI** and write to the transmitter. This writing executes an internal algorithm that produces the new correction coefficients.
– Read again the **PRTB_STATIC_P_TRIMMED_VALUE** and check if its value now matches the real Static Pressure value.

### Sensor temperature trimming

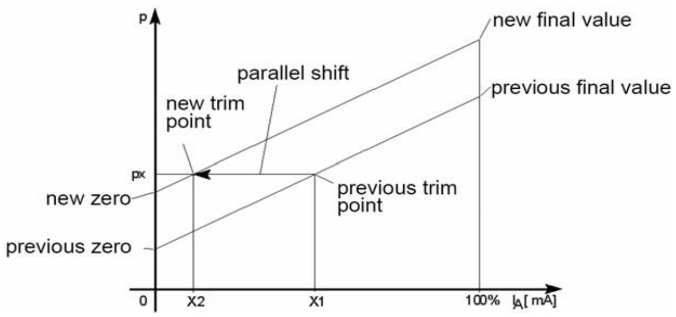
With this operation the **PRTB_SECONDARY_VALUE** (Sensor Temperature) is automatically adjusted, in order to match the real value of the sensor temperature. The following sequence of operations is required:
– Select the engineering unit of the temperature in the **PRTB_SECONDARY_VALUE_RANGE_UNIT** (Temperature Unit Only)
– Read the Sensor Temperature value from the **PRTB_SECONDARY_VALUE.**
– If this value doesn't match the known Sensor Temperature of the transducer, enter the correct value in the **PRTB_SENSOR_TEMP_CAL_POINT** and write to the transmitter. This writing executes an internal algorithm that produces the new correction coefficients.
– Read again the **PRTB_SECONDARY_VALUE** and check if its value now matches the real Sensor temperature value.

### Parallel shift (P-dP)

In case the process (dp or p) cannot be led to 0 it is possible correct the measure performing the Parallel Shift operation. Typically this operation is applicable for Level measurements.
Having the possibility to see/read the actual measure in percent, if it is not what expected, enter the percent of what the process should measure. The correction consists in the shift of the calibration range values **PRTB_PRIMARY_VALUE_RANGE 0%** and **PRTB_PRIMARY_VALUE_RANGE 100%** in order to produce in output the measure, **PRTB_QUATERNARY_VALUE** at the desired percentage. The parallel shift is executed by writing the desired percent value in the **PTRB_PARALLEL_SHIFT_PV**.

> **Important**
> After the parallel shift execution, the percent value of the **PRTB_QUATERNARY_VALUE** matches the desired percentage only if the **PRTB_LIN_TYPE** is set to Linear. If an AI block is set to **CHANNEL = 4** it receives in input the **PRTB_QUATERNARY_VALUE** and in this case the **AI_OUT** matches the desired percentage as well only if the **AI_L_TYPE** is set to Linear

This makes it possible to set the output signal of several measuring devices that measure the same process variable to the same value without having to perform a calibration with applied pressure. E.G. the transmitter output can be adjusted to gauge-glass for level measurement. This function can - under the following circumstances - be carried out at any point on the characteristic:
– Process variable within the adjusted measuring range - transmitter with linear transfer function.
– Write protection on the transmitter must not be activated.



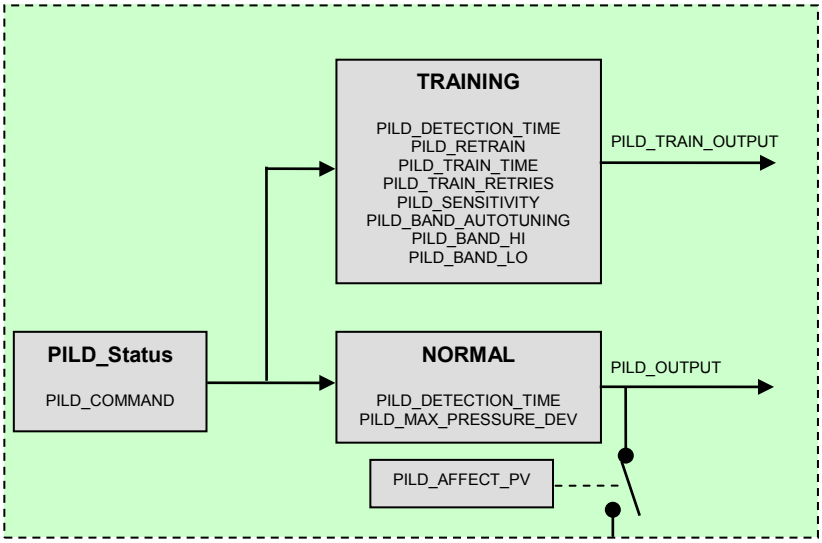
When a pressure px is applied, the transmitter displays the standardized output value x1 in percent. Due to the present application the value x2 should be displayed. Enter this new value x2 in the line **PRTB_ PARALLEL_SHIFT_PV**, the transmitter calculates the new zero and the new final value and adopts these new settings in the **PRTB_PRIMARY_VALUE_RANGE 0%** and **PRTB_PRIMARY_VALUE_RANGE 100%**

## Advanced diagnostic transducer block (ADTB)

### Overview

The advanced diagnostic transducer block contains some historic/statistical information and all the parameters related with the PILD algorithm. The goal of this block is to supervise the device and set diagnostic alarms under transducer abnormal condition to the control system modifying the pressure transducer block primary value status and raising the proper alarm bit in the **ADTB_BLOCK_ERR** and **RB_MAINTENANCE_ACTIVE**.

### Block diagram




### Description

The Plugged Impulse Line Detection (PILD) is a function aimed at detecting the blockage of the process connections of the instrument and any type of problem occurring at the sensor internal hydraulic circuit.
The PILD algorithm is executed in two distinct phases:

### Training phase:

Selecting **ADTB_PILD_COMMAND = TRAIN** the training phase starts analyses and learns the process dynamics in term of noises of the primary signal detected when the process is working at its normal conditions. The Training Phase can take long time depending by the PILD settings of **ADTB_PILD_TRAIN_TIME, ADTB_PILD_RETRIES……**, then if the training phase is successfully completed with good result, **ADTB_PILD_TRAINING_OUTPUT = PILD_TRAIN_OK** the PILD pass to the second phase of process monitoring otherwise it is possible read from the **ADTB_PILD_TRAINING_OUTPUT** the possible cause like:
– Process Instable during training
– Process not available during training
– Not good process condition for training
– Training not done

### Monitoring phase:

The algorithms perform a continuous sampling and comparison of the current process noises with what memorized during the training phase. Differences have been experienced being consequences of something bad in process connections to the sensor like dirty, ice and so on which tap/plug the pipe/s partially or totally.
Whenever a pipe plugging/tapping is detected, the **ADTB_PILD_OUTPUT** that was set to NORMAL during the monitoring phase changes to one of the following conditions:

**– Line on side H plugged**
**– Line on side L plugged**
**– Both lines H and L plugged**
**– An undefined line plugged**

In this case also the device diagnosis is affected setting the bit **Pild_Output** in the **RB_MAINTENANCE_ACTIVE** and the specific bit of the above detailed 4 info in the **RB_EXTENDED_ACTIVE**.

When one of the above conditions has been detected, there is the possibility that the process variables in output from the PRTB continue to be produced with GOOD status.

In this way the AI blocks receiving in input the variables from the **PRTB** works normally and the operator could have not evidence of the wrong conditions. For this reason it is possible make a choice in order to decide to affect or not the **PRTB** variables when the plugging conditions have been detected. This selection is possible with the **ADTB_PILD_AFFECT_PV** variable. When it is selected to true, and the Plugging conditions are detected, the GOOD status that would be produced in output for the **PRTB_PRIMARY_VALUE**, **PRTB_SECONDARY_VALUE**, **PRTB_TERTIARY_VALUE**, **PRTB_QUATERNARY_VALUE** are all forced to BAD status. The PILD algorithm loses the train every time it is switched off. The algorithm is switched off automatically for every error condition, except when the pressure violates the maximum pressure deviation and the retrain is selected.

### Historical/statistical info

From this block can be also read the Minimum and Maximum values measured by the transducer of Pressure, Static Pressure and Sensor Temperature plus the total working time and the number of device power cycles. These information can allow the user to do detailed/specific diagnostic analysis and evaluations.

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / OOS | | The selectable modes by the operator. | |
| | | ACTUAL | R | | | | D | | | The mode the block is currently in. | |
| | | PERMITTED | RW | | | | S | AUTO / OOS | | The allowed modes the operator can select as Target | |
| | | NORMAL | RW | | | | S | AUTO | | The common mode for the Actual. | |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | 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. | | | |
| | | | | | | | | Bit 6 = Device Needs Maintenance Soon | | Refer to the section “Diagnostic, Block_err and Status Byte” from page 29 | |
| | | | | | | | | Bit 15 = Out of Service | | The MODE_BLK_ACTUAL = Out of Service. | |
| 7 | UPDATE_EVT | | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | | |
| 8 | BLOCK_ALM | | RW | R | DS-72 | 13 | D | 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 sub-code field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the sub-code has changed | | | |
| 9 | TRANSDUCER_DIRECTORY | | R | A | U16 | 2 | S | Directory that specifies the number and starting indices of the transducers in the transducer block | | | |
| 10 | TRANSDUCER_TYPE | | R | S | U16 | 2 | S | Identifies the transducer type. | | | |
| 11 | XD_ERROR | | R | S | U8 | 1 | D | Transducer block error sub-code | | | |
| 12 | COLLECTION_DIRECTORY | | R | A | U32 | 4 | S | Directory that specifies the number, starting indices, and the DD items IDs of the data collections in each transducer within a tb | | | |
| 13 | PILD_COMMAND | | RW | S | U8 | 1 | N | Activation / deactivation of the PILD algorithms | 0: | IDLE | default value |
| | | | | | | | | | 1: | GO_OFF | Switch OFF the PILD algorithm |
| | | | | | | | | | 2: | TRAIN | Start the training phase |
| | | | | | | | | | 3: | STOP TRAINING | Stop the training phase of the algorithm before its natural ending |
| 14 | PILD_STATUS | | R | S | U8 | 1 | D | Status of the PILD algorithm | 0: | OFF | The algorithm is Inactive (Default value) |
| | | | | | | | | | 1: | NORMAL | The algorithm is Active |
| | | | | | | | | | 2: | TRAINING | The algorithm is in training phase |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 15 | PILD_OUTPUT | R | S | U8 | 1 | D | Status of the Impulse Lines | Bit 0 | Normal | The process connections are OK and the device is working normally. Lines Not Plugged |
| | | | | | | | | Bit 1 | Not Valid | When the PILD algorithm is not working like, for example, during the Training phase or if the training phase didn't produce a valid result |
| | | | | | | | | Bit 2 | Max Pressure Deviation | The pressure value currently detected is too different from what used for the Training. A new Training is necessary for this new process condition |
| | | | | | | | | Bit 3 | One Line Plugged | One undetected process connection is plugged. It was not possible identify which one |
| | | | | | | | | Bit 4 | Two Lines Plugged | Both the Process connections, high side (+) and low side (-) are plugged |
| | | | | | | | | Bit 5 | Line H Plugged | The Process connection on the high side (+) is plugged |
| | | | | | | | | Bit 6 | Line L Plugged | The Process connection on the low side (-) is plugged |
| | | | | | | | | Bit 7 | not used | |
| 16 | PILD_AFFECT_PV | RW | S | U8 | 1 | S | This parameter indicates if the PILD algorithm must affect the PRTB_PRIMARY_VALUE. if YES, when the PILD reveals an abnormal situation, it sets the PRTB_PRIMARY_VALUE Status to BAD. | 0 | No | Doesn't affect primary value status (default value) |
| | | | | | | | | 1 | Yes | Affect primary value status |
| 17 | PILD_DETECTION_TIME | RW | S | U8 | 1 | S | This parameter represents the length of the algorithm slot. This is the time interval (minutes) over which the algorithm bases the decision on the plugging state of the impulse lines | | | |
| 18 | PILD_MAX_PRESSURE_DEV | RW | S | U8 | 1 | S | This parameter is used in the normal operation checks. It is the maximum allowed deviation of the differential pressure from the mean differential pressure Red in the training phase. If the deviation is greater, than the PILD output is set to OUTPUT NOT VALID, because the conditions are too different from the training phase | | | |
| 19 | PILD_RETRAIN | RW | S | U8 | 1 | S | The PILD algorithm can be forced to train again when the process conditions pass the maximum allowed deviation | 0 | No | Re-trainings are Disabled / not executed |
| | | | | | | | | 1 | Yes | Affect primary value status |
| 20 | PILD_TRAIN_TIME | RW | S | U8 | 1 | S | This parameter represents the duration of the training period | | | |
| 21 | PILD_TRAIN_RETRIES | RW | S | U8 | 1 | S | At the end of the training procedure, there are the training checks on the Red data. If they fail, the algorithm is allowed to retry the procedure adding a further slot of data. This parameter is the max number of allowed retries | | | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 22 | PILD_TRAIN_OUTPUT | R | S | U8 | 1 | D | This parameter gives information on the status of the training phase | Bit 0: | PILD Train Not Done | Training not yet executed |
| | | | | | | | | Bit 1: | PILD Train OK | Training correct |
| | | | | | | | | Bit 2: | PILD Train Frequency Tuning | The training phase is checking the signal frequency for its execution |
| | | | | | | | | Bit 3: | PILD Train Power Instable | Signal power has passed the max. allowed deviation This process condition is considered instable for a good training |
| | | | | | | | | Bit 4: | PILD Train Pressure Instable | Pressure has passed the max.allowed deviation. This process condition is considered instable for a good training |
| | | | | | | | | Bit 5: | PILD Train Low Noise | The Noise of the process is too low for allowing a good training |
| | | | | | | | | Bit 6: | not used | |
| | | | | | | | | Bit 7: | not used | |
| 23 | PILD_SENSITIVITY | RW | S | U8 | 1 | S | Algorithm sensibility | 1: | LOWEST | |
| | | | | | | | | 2: | VERY LOW | |
| | | | | | | | | 3: | LOW | |
| | | | | | | | | 4: | MEDIUM (Default) | |
| | | | | | | | | 5: | HIGH | |
| | | | | | | | | 6: | VERY HIGH | |
| | | | | | | | | 7: | HIGHEST | |
| 24 | PILD_BAND_ AUTOTUNING | RW | S | U8 | 1 | S | PILD auto tuning enable/disable | 0 | No | Doesn't perform auto tune (default value) |
| | | | | | | | | 1 | Yes | Performs Auto Tune |
| 25 | PILD_BAND_LO | RW | S | U8 | 1 | S | | | Writable only if PILD_BAND_AUTOTUNING is set to 0 | |
| 26 | PILD_BAND_HI | RW | S | U8 | 1 | S | | | | |
| 27 | PWR_ON_CNT | R | S | U16 | 2 | D | Power On Counter. Number of the device Power on | | | |
| 28 | TOT_WORK_TIME | R | S | DS-13 | 6 | D | Total Working hours. Total amount of time the transmitter has been kept switched on | | | |
| 29 | PAR_WORK_TIME | RW | S | DS-13 | 6 | D | Partial Working hours. Partial amount of time the transmitter has been switched on. | | | The user can clear this counter |
| 30 | MAX_PdP_VALUE | R | S | FLT | 4 | N | Max Historical Sensor value | | Resettable from RESET_MIN_MAX_VALUE | |
| 31 | MIN_PdP_VALUE | R | S | FLT | 4 | N | Min Historical Sensor value | | | |
| 32 | MAX_SENS_TEMP_VAL | R | S | FLT | 4 | N | Max Historical temp. value | | | |
| 33 | MIN_SENS_TEMP_VAL | R | S | FLT | 4 | N | Min Historical temp. value | | | |
| 34 | MAX_STATIC_PRESS_VAL | R | S | FLT | 4 | S | Max Historical Static Press value | | | |
| 35 | MIN_STATIC_PRESS_VAL | R | S | FLT | 4 | S | Min Historical Static Press value | | | |
| 36 | RESET_MIN_MAX_VALUE | W | S | U8 | 1 | N | Reset the selected User Min Max values | 0 | None | |
| | | | | | | | | 1 | Reset PdP Values | |
| | | | | | | | | 2 | Reset Sensor Temperature Values | |
| | | | | | | | | 3 | Reset Static Pressure Values | |

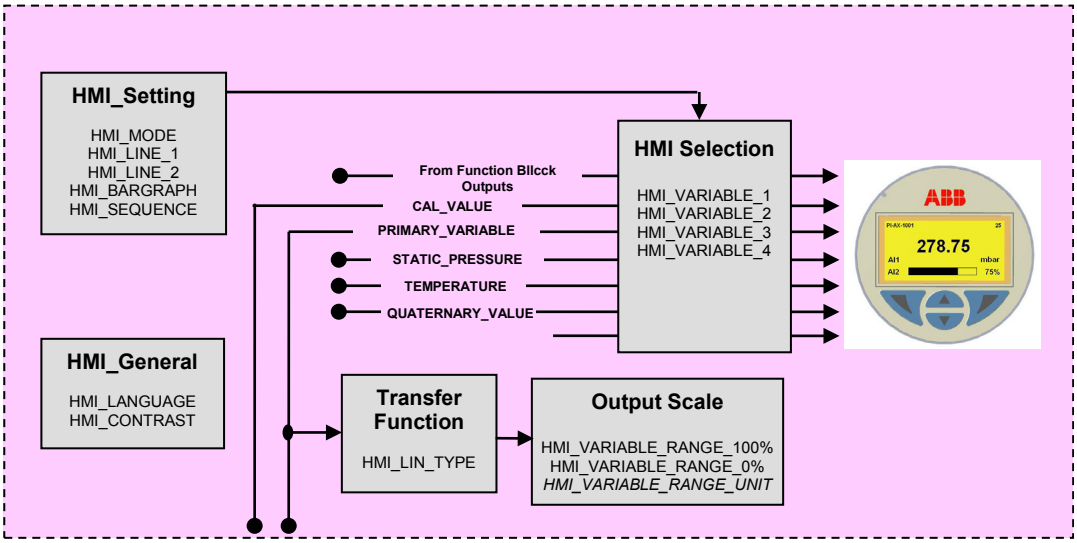
| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|
| 37 | SERVICE_MAX_PdP_VALUE | R | S | FLT | 4 | N | For service Use. Max Historical Sensor value | Resettable from RESET_SERV_MIN_MAX_VALUE |
| 38 | SERVICE_MIN_PdP_VALUE | R | S | FLT | 4 | N | For service Use. Min Historical Sensor value | |
| 39 | SERV_MAX_SENS_TEMP_VAL | R | S | FLT | 4 | N | For service Use. Max Historical temp. value | |
| 40 | SERV_MIN_SENS_TEMP_VAL | R | S | FLT | 4 | N | For service Use. Min Historical temp. value | |
| 41 | SERV_MAX_STATIC_PRESS_VAL | R | S | FLT | 4 | S | For service Use. Max Historical Static Press value | |
| 42 | SERV_MIN_STATIC_PRESS_VAL | R | S | FLT | 4 | S | For service Use. Min Historical Static Press value | |

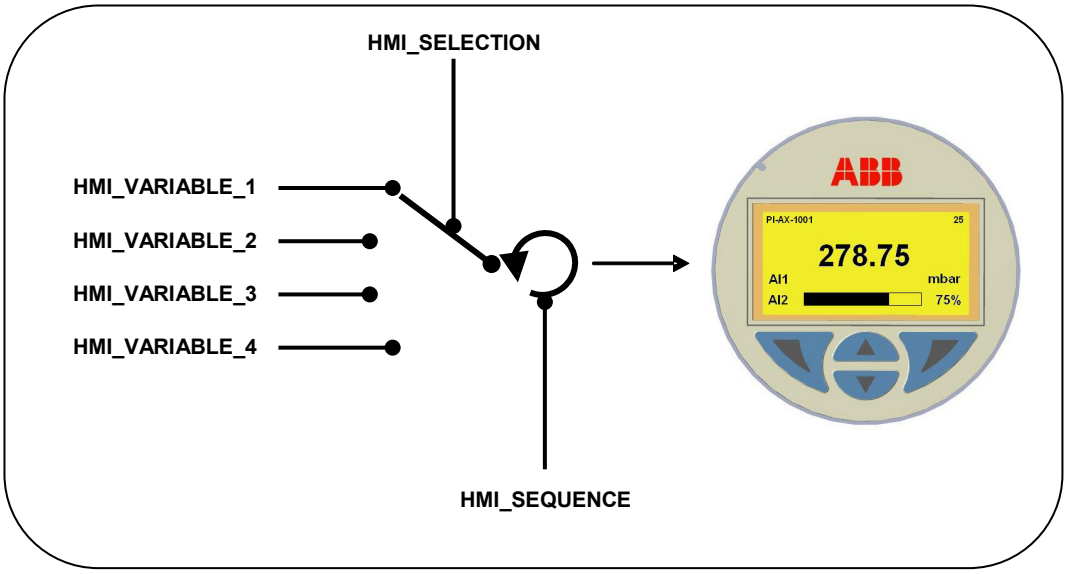
## HMI transducer block (HMITB)

### Overview

The display transducer block is an independent block dedicated to the management of the local display and the local operations via push buttons.

### Block diagram




### Description

The HMI Transducer Block contains all the parameters allowing the display configuration.
Refer also to section 10 of OI/266/FF manual.

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | | | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / OOS | The selectable modes by the operator. | |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. | |
| | | PERMITTED | RW | | | | S | AUTO / OOS | The allowed modes the operator can select as Target | |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. | |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | 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. | | |
| | | | | | | | | Bit 15 = Out of Service | The **MODE_BLK_ACTUAL** = Out of Service. | |
| 7 | UPDATE_EVT | | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | |
| 8 | BLOCK_ALM | | RW | R | DS-72 | 13 | D | 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 sub-code field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the sub-code has changed. | | |
| 9 | TRANSDUCER_DIRECTORY | | R | A | U16 | 2 | S | Directory that specifies the number and starting indices of the transducers in the transducer block | | |
| 10 | TRANSDUCER_TYPE | | R | S | U16 | 2 | S | Identifies the transducer type. | | |
| 11 | XD_ERROR | | R | S | U8 | 1 | D | Transducer block error sub-code | | |
| 12 | COLLECTION_DIRECTORY | | R | A | U32 | 4 | S | Directory that specifies the number, starting indices, and the DD items IDs of the data collections in each transducer within a tb | | |
| 13 | HMI_CONTRAST | | RW | S | U8 | 1 | S | Display Contrast 0 ….100 [50] | | |
| 14 | HMI_LANGUAGE | | RW | S | U8 | 1 | S | 0: | English (default) | |
| | | | | | | | | 1: | German | |
| | | | | | | | | 2 | French | |
| | | | | | | | | 3 | Spanish | |
| | | | | | | | | 4: | Italian | |
| 15 | HMI_MODE | | RW | S | U8 | 1 | S | 5: | One Line | |
| | | | | | | | | 6: | One Line with Bargraph (default) | |
| | | | | | | | | 9: | Two Lines | |
| | | | | | | | | 10: | Two Lines with Bargraph | |
| 16 | HMI_SW_REV | | R | S | U8 | 1 | N | 0 | Not Installed | |
| | | | | | | | | xxx | Display SW Revision | |
| 17 | HMI_LIN_TYPE | | RW | S | U8 | 1 | S | 0: | Linear (default) | Active only if one of the 4 HMI_Variable_x is set to HMI_Variable. In this case the PRTB_PRIMARY_VALUE is calculated with the selected HMI_LIN_TYPE and the result ready to be scaled with the HMI_VARIABLE_RANGE. |
| | | | | | | | | 1: | Square root | |
| | | | | | | | | 2: | SQRT 3° pow | |
| | | | | | | | | 3: | SQRT 5° pow | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 17 | HMI_LIN_TYPE | RW | S | U8 | 1 | S | 4: | cylindrical lying container | Active only if one of the 4 HMI_Variable_x is set to HMI_Variable. In this case the PRTB_PRIMARY_VALUE is calculated with the selected HMI_LIN_TYPE and the result ready to be scaled with the HMI_VARIABLE_RANGE. |
| | | | | | | | 5: | spherical container | |
| | | | | | | | 6: | Bidirectional Flow | |
| 18 | HMI_VARIABLE_RANGE | RW | R | DS-68 | 11 | S | The High and Low HMI Scale limits with the engineering units code | | Scaling applied at the PRTB_PRIMARY_VALUE for displaying purpose only and has not effect on the PRTB_PRIMARY_VALUE in input at the AI blocks. |
| 19 | HMI_VARIABLE_CUSTOM_UNIT | RW | S | V_STR | 8 | S | Textual custom unit | | |
| 20 | HMI_LINE_1 | RW | S | U8 | 1 | S | 1: | HMI_Variable 1 (default) | |
| | | | | | | | 2: | HMI_Variable 2 | |
| | | | | | | | 3: | HMI_Variable 3 | |
| | | | | | | | 4: | HMI_Variable 4 | |
| 21 | HMI_LINE_2 | RW | S | U8 | 1 | S | Same as HMI_LINE_1 | | |
| 22 | HMI_BARGRAPH | RW | S | U8 | 1 | S | Same as HMI_LINE_1 | | |
| 23 | HMI_SEQUENCE | RW | S | U8 | 1 | S | 0 | Sequence/Autoscrolling OFF | |
| | | | | | | | 1 | Sequence/Autoscrolling ON | |
| 24 | HMI_VARIABLE_1 | RW | S | U8 | 1 | S | In order to recognize the displayed variable among all those in this list, it appears a three character string in the left side of the value when two lines mode is selected and below the value when One Line Mode is selected. The strings for any variables are: | | |
| | | | | | | | 0: | P-dP (default) | ‘PV’ |
| | | | | | | | 6: | OUT AI_1 | ‘AI1’ |
| | | | | | | | 7: | OUT % AI_1 | ‘%1’ |
| | | | | | | | 8: | OUT AI_2 | ‘AI2’ |
| | | | | | | | 9: | OUT % AI_2 | ‘%2’ |
| | | | | | | | 10: | OUT AI_3 | ‘AI3’ |
| | | | | | | | 11: | OUT % AI_3 | ‘%3’ |
| | | | | | | | 1: | Sensor Temp | ‘ST’ |
| | | | | | | | 2: | Static Pressure | ‘SP’ |
| | | | | | | | 19: | HMI Variable | ‘HMI’ |
| | | | | | | | 4: | PV Trim Value | ‘TPV’ |
| | | | | | | | 5 | Static Pressure Trim Value | ‘TSP’ |
| | | | | | | | 3 | Scaled PV | ‘LIN’ |
| | | | | | | | 16 | OUT1 SC | ‘SC1’ |
| | | | | | | | 17 | OUT2 SC | ‘SC2’ |
| | | | | | | | 12 | OUT PID | ‘PID’ |
| | | | | | | | 18 | OUT AR | ‘AR’ |
| | | | | | | | 14 | OUT IS | ‘IS’ |
| | | | | | | | 15 | OUT IT | ‘IT’ |
| | | | | | | | 13 | OUT CS | ‘CS’ |
| 25 | HMI_VARIABLE_2 | RW | S | U8 | 1 | S | Same as HMI_VARIABLE_1 | | |
| 26 | HMI_VARIABLE_3 | RW | S | U8 | 1 | S | Same as HMI_VARIABLE_1 | | |
| 27 | HMI_VARIABLE_4 | RW | S | U8 | 1 | S | Same as HMI_VARIABLE_1 | | |

# Device diagnostic

The 266 PdP FF produces different type of diagnostic information:
– Device Diagnostic
These are the diagnostic information produced by the Resource and Transducer Blocks and refer to the device status/health
– Process Diagnostic
These are the diagnostic information which are seen through process variable status and process alarms such as the HI, HI HI, LO, LO LO, DV HI and DV LO alarms implemented in various Function blocks
Scope of this section is to describe the Device Diagnostic to be used by the Asset Management Software.

### Standard errors

The FF standard parameter defining the device diagnostic conditions is the BLOCK_ERR contained by each block and mapped as follow:

| BLOCK_ERR mapping | |
|---|---|
| Bit 0 | Other |
| Bit 1 | Block Configuration Error |
| Bit 2 | Link Configuration Error |
| Bit 3 | Simulate Active |
| Bit 4 | Local Override |
| Bit 5 | Device Fault State Set |
| Bit 6 | Device Needs Maintenance Soon |
| Bit 7 | Sensor Failure detected by this block/process variable has a status of BAD, Sensor Failure |
| Bit 8 | Output Failure detected by this block/backcalculation input has a status of BAD, Device Failure |
| Bit 9 | Memory Failure |
| Bit 10 | Lost Static Data |
| Bit 11 | Lost NV Data |
| Bit 12 | Readback Check Failed |
| Bit 13 | Device Needs Maintenance Now |
| Bit 14 | Power Up |
| Bit 15 | Out of Service |

### Field Diagnostic Profile (NAMUR NE107)

The device diagnostic info are split into four different alert type parameters relating the four NAMUR NE107 categories, they are:
– RB_FAIL_ACTIVE
– RB_OFFSPEC_ACTIVE
– RB_MAINTENANCE_ACTIVE
– RB_CHECK_ACTIVE
The 4 bit-string parameters are mapped in the same way but each error condition became set only within one of the four parameters. For some error conditions, the bit mapped within the 4 parameters above is not clear enough for the maintenance person to get the root of the problem, for this reason it has been defined an additional parameter RB_EXTENDED_ACTIVE with additional detailed information about the specific error condition.
Only the errors without details mapped in the RB_x_ACTIVE and all the RB_EXTENDED_ACTIVE errors can be simulated from the RB_DIAGNOSIS_SIMULATION.

**Important**
Only one condition/bit per time can be simulated.

**Important**
Critical error conditions cannot be masked.

The RB_SIMULATION_STATUS must be enabled before to simulate any individual error condition. The same parameter can be read with the scope to clarify if an error condition is active because simulated or calculated.
Some of the error bits mapped in the RB_x_ACTIVE can be masked from the RB_DIAGNOSIS_MASK.

# 266 Models - FOUNDATION Fieldbus

## Device Diagnostic Mapping

This table shows the bit mapping of **RB_x_ACTIVE** and **RB_EXTENDED_ACTIVE** parameters with the relating error name and NAMUR NE107 category.

– Bit x° = Error conditions that can be Masked from **RB_DIAGNOSIS_MASK**

– Bit x* = Error conditions that can be simulated from **RB_DIAGNOSIS_SIMULATION**

| Cat. | Error | RB_fail / maintenance / offspec / check_ACTIVE (indexes 42, 43, 44, 45) | | RB_EXTENDED_ACTIVE (index 47) | |
|---|---|---|---|---|---|
| | | Bit 0* | | | |
| | Sensor Invalid | Bit 1 | The transducer is not able to generate a valid signal due to one of the following conditions (see RB_EXTENDED_ACTIVE) | Bit 0* | The primary signal of the sensor is no longer available |
| | Sensor Memory Fail | Bit 2* | The data in the sensor memory are corrupted precluding the correct functionality of the device | Bit 1* | The sensor and the connected electronics are incompatible |
| | Memory Failure | Bit 3* | The device data loaded at the start up are corrupted precluding the correct functionality of the device | | |
| | P-dP Sensor Fail | Bit 4* | The sensor signal value is incorrect due to a mechanical failure i.e. Loss of fill fluid from the cell; ruptured diaphragm, broken sensor… | | |
| | Static Pressure Sensor Fail | Bit 5* | The sensor signal value is incorrect due to a mechanical failure i.e. The circuitry for the sampling of the static pressure has failed... Valid for Differential pressure models | | |
| | Sensor Temperature Fail | Bit 6* | The circuitry for the sampling of the temperature has failed. The measurement accuracy is decreased more than the acceptable error | | |
| | Pressure Overrange | Bit 7*° | The Pressure is outside the overpressure limit and risk to damage the sensor | | |
| | P-dP Out Of Limit | Bit 8*° | The measured Process Pressure is outside the sensor limits and no longer representing the true applied process value | | |
| | Static Pressure Out Of Limit | Bit 9*° | The measured Static pressure is above its operational limit | | |
| | Input Simulation Active | Bit 10° | The Input Simulation function is Active At least one of these variables is simulated | Bit 2* | The Pressure Value produced in output is calculated starting from a simulated input |
| | | | | Bit 3* | The Static Pressure Value produced in output is calculated starting from a simulated input |
| | | | | Bit 4* | The Sensor Temp Value produced in output is calculated starting from a simulated input |
| | Sensor Temperature Out Of Limit | Bit 11*° | The measured sensor temperature is outside of its operational limits | | |
| | Max. Working Pressure Exceeded | Bit 12*° | The measured Static Pressure is higher than the acceptable mechanical limit of the process connection elements. | | |
| | Primary Variable Out of Range | Bit 13*° | Process value is outside its High or Low working range | | |

| Cat. | Error | RB_fail / maintenance / offspec / check_ACTIVE (indexes 42, 43, 44, 45) | | RB_EXTENDED_ACTIVE (index 47) | |
|---|---|---|---|---|---|
| | Electronic Interface error. | Bit 14*° | Exchange of non-critical data between sensor and electronics is precluded due to problem in the transmitter circuit of the electronics or in receiver circuit of the sensor | | |
| | Non-Volatile Sensor memory burn error | Bit 15*° | Writings to the Sensor non-Volatile Memory were not successful. The device works without problems but any replacement operation is compromised because the back-up configuration is not updated | | |
| | Non-Volatile Electronics memory burn error | Bit 16*° | Writings to the electronic Non-Volatile Memory were not successful. The device continue to work without problems but after the next power cycle the last configuration will be lost | | |
| | Replace Info | Bit 17° | An element of the transmitter has been changed (sensor or electronics) and the correct replacement operation must be executed | Bit 5* | The Replace operation is required after the changing of the electronics or of the sensor. Both the directions are allowed, from Electronic (CB) to Sensor (FE) or from Sensor (FE) to Electronic (CB) |
| | | | | Bit 6* | The Replace operation is required after the changing of the electronics or of the sensor but it is allowed only from Electronic (CB) to Sensor (FE) |
| | | | | Bit 7* | The Replace operation has been attempted but with wrong direction |
| | PILD Output | Bit 18° | The PILD algorithm has detected impulse lines plugged. The Plugged Line can be one among: | Bit 10* | PILD algorithm has detected both impulse lines plugged. |
| | | | | Bit 11* | PILD algorithm has detected a plugged impulse line on the HIGH side. |
| | | | | Bit 12* | PILD algorithm has detected a plugged impulse line on the LOW side. |
| | | | | Bit 13* | PILD algorithm has detected one plugged impulse line. |
| | PILD Changed Operating Conditions | Bit 19*° | The pressure value currently detected is too different from what used for the PILD Training | | |

## Diagnostic, Block_err and Status Byte

The table below shows all the error conditions grouped according the element producer of the error itself among Electronics, Sensor, Installation/configuration and Process. For each error condition is also shown:

– The Block_Err bit associated with the error condition as BLOCK.Block_Err bit

– The Error code available from the HMI Diagnostic... Refer to the Display section

– Status Byte of the Pressure Transducer Block Variables

| Cat. | Error | Block err | HMI code | PRTB_x_VALUE Status | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Electronics errors | | | PRIMARY | SECONDARY | TERTIARY | QUATERNARY | PDP_TRIM | SP_TRIM |
| | Memory Failure | RB.Lost NV Data | F116.023 | BAD Device Failure No Limit | | | | | |
| | Electronic Interface error. | RB.Device Needs Maintenance Soon | M030.020 | GOOD_NC Non Specific No Limit | | | | | |
| | Non-Volatile memory burn error | RB.Device Needs Maintenance Soon | M026.024 | GOOD_NC Non Specific No Limit | | | | | |
| | Pressure sensor errors | | | PRIMARY | SECONDARY | TERTIARY | QUATERNARY | PDP_TRIM | SP_TRIM |
| | Sensor Invalid | PRTB.Sensor Failure | F120.016 | BAD Sensor Failure No Limit | | | | | |
| | Sensor Memory Fail | PRTB.Sensor Failure | F118.017 | BAD Sensor Failure No Limit | | | | | |
| | P-dP Sensor Fail | PRTB.Sensor Failure | F114.000 | BAD Sensor Failure No Limit | GOOD_NC Non Specific No Limit | GOOD_NC Non Specific No Limit | BAD Sensor Failure No Limit | | GOOD_NC Non Specific No Limit |
| | Static Pressure Sensor Fail | PRTB.Sensor Failure | F112.001 | UNCERTAIN Sensor Conversion Not Accurate No Limit | GOOD_NC Non Specific No Limit | BAD Sensor Failure No Limit | UNCERTAIN Sensor Conversion Not Accurate No Limit | | BAD Sensor Failure No Limit |
| | Sensor Temperature Fail | PRTB.Sensor Failure | F110.002 | UNCERTAIN Sensor Conversion Not Accurate No Limit | BAD Sensor Failure No Limit | UNCERTAIN Sensor Conversion Not Accurate No Limit | | | |
| | NonVolatile memory burn error | RB.Device Needs Maintenance Soon | M028.018 | GOOD_NC Non Specific No Limit | | | | | |
| | Installation / start-up errors | | | PRIMARY | SECONDARY | TERTIARY | QUATERNARY | PDP_TRIM | SP_TRIM |
| | Out of Service | RB.Out of Service | | BAD Out of Service | | | | | |
| | Input Simulation Active | PRTB.Local Override | C088.030 | GOOD_NC Non Specific No Limit | | | | | |
| | Replace Info | RB.Device Needs Maintenance Soon | M020.042 | GOOD_NC Non Specific No Limit | | | | | |
| | Process errors | | | PRIMARY | SECONDARY | TERTIARY | QUATERNARY | PDP_TRIM | SP_TRIM |
| | Pressure Overrange | PRTB.Other | F104.032 | BAD Non Specific No Limit | GOOD_NC Non Specific No Limit | | BAD Non Specific No Limit | | GOOD_NC Non Specific No Limit |
| | P-dP Out Of Limit | PRTB.Other | F102.004 | UNCERTAIN Non Specific No Limit | GOOD_NC Non Specific No Limit | | UNCERTAIN Non Specific No Limit | | GOOD_NC Non Specific No Limit |
| | Static Pressure Out Of Limit | PRTB.Other | F100.005 | UNCERTAIN Sensor Conversion Not Accurate No Limit | GOOD_NC Non Specific No Limit | UNCERTAIN Non Specific No Limit | UNCERTAIN Sensor Conversion Not Accurate No Limit | | UNCERTAIN Non Specific No Limit |
| | Sensor Temperature Out Of Limit | PRTB.Other | S054.006 | UNCERTAIN Sensor Conversion Not Accurate No Limit | UNCERTAIN Non Specific No Limit | UNCERTAIN Sensor Conversion Not Accurate No Limit | | | |
| | Max. Working Pressure Exceeded | PRTB.Other | M052.031 | GOOD_NC Non Specific No Limit | | | | | |

| Cat. | Error | | Block err | HMI code | PRTB_x_VALUE Status | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Process errors | | | | | PRIMARY | SECONDARY | TERTIARY | QUATERNARY | PDP_TRIM | SP_TRIM |
| | Primary Variable Out of Range | High Range | PRTB.Other | S050.010 | GOOD_NC Non Specific No Limit | | | UNCERTAIN engineering unit range violation High Limit | GOOD_NC Non Specific No Limit | |
| | | Low Range | | | | | | UNCERTAIN engineering unit range violation Low Limit | | |
| | PILD Output | PILD Affect PV = 0 | ADTB.Device Needs Maintenance Soon | M018.038 | GOOD_NC Non Specific No Limit | | | | | |
| | | PILD Affect PV = 1 | | | BAD Sensor Failure No Limit | GOOD_NC Non Specific No Limit | BAD Sensor Failure No Limit | | | |
| | PILD Changed Operating Conditions | PILD Affect PV = 0 | ADTB.Device Needs Maintenance Soon | M018.038 | GOOD_NC Non Specific No Limit | | | | | |
| | | PILD Affect PV = 1 | | | BAD Sensor Failure No Limit | GOOD_NC Non Specific No Limit | BAD Sensor Failure No Limit | | | |
| | Out Of Service | | PRTB.Out of Service | | BAD Out of Service | | | | | |

## Device troubleshooting

| Cat | Error | | HMI code | Description | Possible Cause | Suggested Actions |
|---|---|---|---|---|---|---|
| **Electronics errors** | | | | | | |
| | Memory Failure | | F116.023 | The device data loaded at the start up are corrupted precluding the correct functionality of the device | Electronic memory corrupted | The electronics must be replaced |
| | Electronic Interface error. | | M030.020 | Exchange of data between Electronics and Sensor have problems | Exchange of non-critical data between sensor and electronics is precluded due to problem in the transmitter circuit of the electronics or in receiver circuit of the sensor | Power cycle the device and retry the operation, if the error persist the electronics should be replaced |
| | Non-Volatile memory burn error | | M026.024 | The device continue to work without problems but at the next power cycle the new configuration will be lost | Writings to the electronic non-Volatile Memory was not successful | The electronics should be replaced as soon as possible. |
| **Pressure sensor errors** | | | | | | |
| | Sensor Invalid | Missing Primary Signal | F120.016 | The primary signal of the sensor is no longer available. The transducer is not in a condition to generate a valid signal. | The sensor signal is not being updated correctly as a result of an electronics failure, sensor error or a poorly connected sensor cable. | Check cable connection, check sensor and if problem persists, the sensor must be replaced. |
| | | Invalid Sensor | | Sensor and/or the connected electronics are incompatible | The sensor model/version is not longer compatible with the connected electronic version | The sensor must be replaced |
| | Sensor Memory Fail | | F118.017 | The data in the sensor memory are corrupted precluding the correct functionality of the device | Sensor memory corrupted | The Sensor must be replaced |
| | P-dP Sensor Fail | | F114.000 | The sensor signal value is incorrect due to a mechanical failure | Mechanical damage to the sensor. Loss of fill fluid from the cell; ruptured diaphragm, broken sensor…. | The Sensor must be replaced |
| | Static Pressure Sensor Fail | | F112.001 | The sensor signal value is incorrect due to a mechanical failure. Valid only for Differential pressure models | The circuitry for the sampling of the static pressure has failed. | The Sensor must be replaced |

| Cat | Error | | HMI code | Description | Possible Cause | Suggested Actions |
|---|---|---|---|---|---|---|
| | | | | **Pressure sensor errors** | | |
| | Sensor Temperature Fail | | F110.002 | The measurement accuracy is decreased more than the acceptable error | The circuitry for the sampling of the temperature has failed. | The Sensor must be replaced |
| | Non-Volatile memory burn error | | M028.018 | The device continue to work without problems but any replacement operation is compromised because the back-up configuration is not updated | Writings to the Sensor non-Volatile Memory was not successful | The Sensor should be replaced as soon as possible. |
| | | | | **Installation / start-up errors** | | |
| | RB Out of Service | | | Device configured to be Out of Service or initializing | Resource Block is configured to be Out of Service | The TARGET MODE of the Resource Block must be switched in AUTO |
| | Input Simulation Active | PdP simulation | C088.030 | The Process Value is simulated to became the P-dP value measured in input | The P-dP Value in output is calculated from a value simulated in input | Use DD based configurator (AVB Professional - Hand held) to place device back into normal operating mode (Remove the input simulation) |
| | | Static Pressure simulation | | The Process Value is simulated to became the Static Pressure value measured in input | The Static Pressure Value in output is calculated from a value simulated in input | |
| | | Sensor Temp simulation | | The Sensor Temperature Value is simulated to became the measured Sensor Temperature value | The Sensor Temperature Value in output is calculated from a value simulated in input | |
| | Replace Info | Replace required – Both data direction valid | M020.042 | The Replace operation is required after the changing of the electronics or of the sensor | The Electronics or the Sensor have been changed but the replacement operation has not been executed | The replacement operation must be executed:<br>– Move the SW 1 of the electronics in position 1 (= Enable replace mode).<br>– Select the SW 2 the element that has been changed between new Sensor or new electronics<br>– Power Cycle the device<br>– Move the SW 1 of the electronics in position 0 |
| | | Replace required – FE to CB not applicable | | The Replace operation is required after the changing of the electronics or of the sensor | The Electronics or the Sensor has been changed and a replacement operation for a new sensor has to be executed. | The replacement operation must be executed:<br>Only electronics data can be copied into the sensor<br>– Move the SW 1 to Enable replace mode (1)<br>– Select with the SW 2 to New Sensor (1)<br>– Power Cycle the device<br>– Move the SW 1 to Disable replace mode (0) |
| | | Replace enabled – FE to CB not applicable | | The Replace operation has been attempted but with wrong direction | The Electronics or the Sensor have been changed, The replacement has been enabled but with a wrong direction (SW 2 = 0) | Change the replacement direction (if possible)<br>– The SW 1 is already set to Enable replace mode<br>– Select with the SW 2 to New Sensor (1)<br>– Power Cycle the device<br>– Move the SW 1 to Disable replace mode (0) |
| | | | | **Process errors** | | |
| | Pressure Overrange | | F104.032 | An overpressure has been detected | This effect could be produced by other equipment on the process, (valves…). Exceeding the pressure range can cause reduced accuracy or mechanical damage to the diaphragm material and may require calibration/replacement. | The compatibility of pressure transmitter model and process conditions has to be checked.<br>A different transmitter type could be required |

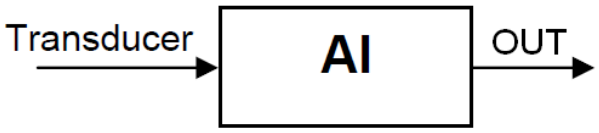
| Cat | Error | HMI code | Description | Possible Cause | Suggested Actions |
|---|---|---|---|---|---|
| | | | **Process errors** | | |
| | P-dP Out Of Limit | F102.004 | The measured Process Pressure value is outside the sensor limits and no longer representing the true applied process value. | The measurement range has not been correctly calculated OR an incorrect transducer model has been selected. | The compatibility of pressure transmitter model and process conditions has to be checked. Probably a different transmitter type is required. |
| | Static Pressure Out Of Limit | F100.005 | The measured Static pressure is above its operational limit | The static pressure of the process exceeds the limit of the sensor. Exceeding the Static Pressure can reduce accuracy, mechanically damage the diaphragm and may require calibration/replacement. An incorrect transducer model could have been selected. | The compatibility of pressure transmitter model and process conditions has to be checked. Probably a different transmitter type is required. |
| | Sensor Temperature Out Of Limit | S054.006 | The measured sensor temperature is outside of its operational limits | The temperature of the process environment affects the pressure transmitter; Excess temperature can reduce accuracy, degrade device components and may require calibration/replacement. | The compatibility of pressure transmitter model and process conditions has to be checked. A different installation type could be required e.g. use of remote seals. |
| | Max. Working Pressure Exceeded | M052.031 | The measured Static Pressure is higher than the acceptable mechanical limit for the process connection elements. | The static pressure of the process exceeds the limit of the max working Pressure supported by the transmitter. Exceeding the Max Working Pressure can mechanically damage the process connections (flanges, pipes….) and/or be dangerous | The compatibility of the process connection type and material with process conditions has to be checked. A different installation type could be required e.g. use of remote seals. |
| | Primary Variable Out of Range | S050.010 | Process value is outside its working range | The measured pressure value is beyond its Low or High scaling limits | Adjust the working range if possible. |
| | PILD Output – Both Impulse Lines Plugged | M018.038 | PILD algorithm has detected both impulse lines plugged. | Both connections between the pressure sensor and the process are blocked either by plugging or closed valves. | Check valves and impulse line. Clean impulse line if necessary and initiate PILD training |
| | PILD Output – High Side Plugged | | PILD algorithm has detected a plugged impulse line on the HIGH side. | The connection between the pressure sensor and the process on the HIGH side is blocked either by plugging or closed valves. | |
| | PILD Output – Low Side Plugged | | PILD algorithm has detected a plugged impulse line on the LOW side. | The connection between the pressure sensor and the process on the LOW side is blocked either by plugging or closed valves | |
| | PILD Output – Undefined line plugged | | PILD algorithm has detected one plugged impulse line. | One of the connections between the pressure sensor and the process is blocked either by plugging or closed valves. | |
| | PILD Changed Operating Conditions | M016.039 | The pressure value currently detected is too different from what used for the PILD Training | Process conditions have changed to an extent that new settings for the PILD algorithm are needed. | A new Training is necessary for this new process condition |
| | PRTB Out Of Service | | Transducer Block is Out of Service. | Power has been reapplied resulting in the re-initialization of the device. | Check transducer block configuration and make sure that the Target Mode has been set to Automatic Mode. |

## Control Application Process (CAP) Blocks

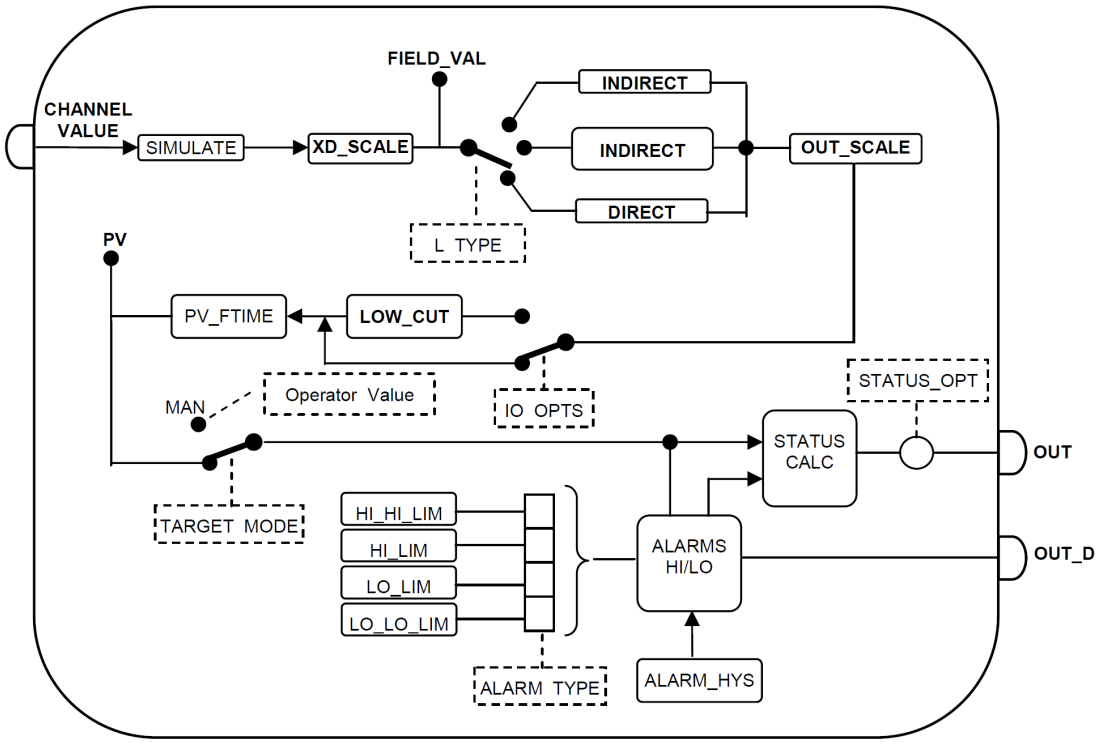
## Enhanced-Analog Input Function Block (E-AI)

### Overview

The Analog Input receives in input variables produced in the Transducer Block via the selected CHANNEL. It is possible that some transmitters have more Transducer Blocks or more variables produced within one Transducer Block. The CHANNEL setting allows the user to select the desired variable to be used in input for the AI.




### Block diagram




### Description

Transducer scaling (XD_SCALE) is applied to the value from the channel to produce the FIELD_VAL in percent. The XD_SCALE units code must match the channel units code (if one exists), or the block will remain in O/S mode after being configured. A block alarm for units mismatch will be generated. The OUT_SCALE is normally the same as the transducer, but if L_TYPE is set to Indirect or Ind.Sqr.Root, OUT_SCALE determines the conversion from FIELD_VAL to the output. PV and OUT always have identical scaling.

OUT_SCALE provides scaling for PV. The PV is always the value that the block will place in OUT if the mode is Auto. If Man is allowed, someone may write a value to the output. The status will prevent 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 true, any calculated output below the low cut-off value will be changed to zero. This is only useful for zero based measurement devices, such as flow.

The PV filter, whose time constant is PV_FTIME, is applied to the PV, and not the FIELD_VAL.

## Equations

The Analog Input receive in input the Transduce  Block variable Value selected with the CHANNEL. The Input Value is represented as CHANNEL Value in the following formula.

$$FIELD\_VAL = 100 * \frac{CHANNEL\_VALUE - XD\_SCALE\_0\%}{XD\_SCALE\_100\% - XD\_SCALE\_0\%}$$

| Depending by the L_TYPE selection the following signal conversions are applied: | |
|---|---|
| **L_TYPE** | **Formula** |
| **Direct** | PV = CHANNEL_VALUE |
| **Indirect** | This conversion is applied when the XD_SCALE values are different from the OUT_SCALE values<br>PV = FIELD_VAL% * (OUT_SCALE_100% - OUT_SCALE_0%) + OUT_SCALE_0% |
| **Indirect Square Root** | IF FIELD_VAL < 0.0<br>PV = OUT_SCALE 0%<br>ELSE IF FIELD_VAL < LOW_CUT<br>PV = OUT_SCALE 0%<br>ELSE<br>PV = $\sqrt{\text{Field\_Val\%}}$ * (OUT_SCALE_100% - OUT_SCALE_0%) + OUT_SCALE_0% |

## Configuration hints

The minimum configuration for having the AI working and/or moving out from the OOS needs at least the following settings:
– CHANNEL different by 0 (uninitialized)
– XD_SCALE = OUT_SCALE
– L_TYPE = Direct
** The minimum configuration can be set also via the RB_SPECIAL_RESTART

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | | |
| 7 | PV | | R | R | DS-65 | 5 | D | The process variable used in block execution, expressed in **XD_SCALE Unit** Code | | |
| 8 | OUT | | RW | R | DS-65 | 5 | N | The block output value calculated as a result of the block execution, expressed in **XD_SCALE unit** code | | Writeable only if **MODE_BLK.ACTUAL = MAN** |
| 9 | SIMULATE | | RW | R | DS-82 | 11 | D | Allow to simulate Value and Status that should be received from the PRTB. | | |
| 10 | XD_SCALE | | RW | R | DS-68 | 11 | S | Input Scale | | All the values are associated with the channel input value |
| 11 | OUT_SCALE | | RW | R | DS-68 | 11 | S | Output scale | | All the values are associated with the OUT |
| 12 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | | |
| 13 | IO_OPTS | | RW | S | B_STR | 2 | S | Option which the user can select to alter Input and Output block processing | | |
| | | | | | | | | Bit 10 | Low Cut Off | Enable/Disable the LOW_CUT Off effect in the AI calculation |
| | | | | | | | | Bit 12 | Unit conversion | Enable/Disable the automatic Unit conversion of the variables in input at the AI from the PRTB when their unit is different by the **XD_SCALE.Unit** |
| 14 | STATUS_OPTS | | RW | S | B_STR | 2 | S | Options which the user can select for the block processing of status. The available selections are: | | |
| | | | | | | | | Bit 3 | Propagate Fault Forward | Enable/Disable the propagation of the Status byte from the PRTB in input at the AI to its Output |
| | | | | | | | | Bit 6 | Uncertain if Limited | |
| | | | | | | | | Bit 7 | BAD if Limited | |
| | | | | | | | | Bit 8 | Uncertain if MAN Mode | |
| 15 | CHANNEL | | RW | S | U16 | 2 | S | 0 | Uninitialized | ** Doesn't allow at the AI to move out from OOS |
| | | | | | | | | 1 | Pressure Process Value | P-dP |
| | | | | | | | | 2 | Sensor temperature | ST |
| | | | | | | | | 3 | Static Pressure | SP |
| | | | | | | | | 4 | Scaled Process Value | Lin PV |

| Idx | Parameter | | Data | Type | Size | Stor-age | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 16 | L_TYPE | RW | S | U8 | 1 | S | 0 | Uninitialized | ** Doesn't allow at the AI to move out from OOS |
| | | | | | | | 1 | Direct | PV = CHANNEL_VALUE – To be used when XD_SCALE = OUT_SCALE |
| | | | | | | | 2 | Indirect | PV = FIELD_VAL% * (OUT_SCALE_100% - OUT_SCALE_0%) + OUT_SCALE_0%<br>To be used when XD_SCALE != OUT_SCALE |
| | | | | | | | 3 | Indirect Square Root | IF FIELD_VAL < 0.0 when PV = OUT_SCALE 0%<br>ELSE IF FIELD_VAL < LOW_CUT when PV = OUT_SCALE 0%<br>ELSE when PV = $\sqrt{\text{Field\_Val\%}}$ * (OUT_SCALE_100% - OUT_SCALE_0%) + OUT_SCALE_0% |
| 17 | LOW_CUT | RW | S | FLT | 4 | S | 0 or >0 | | Limit used in square root processing. A value of zero percent of scale is used in block processing if the transducer falls below this limit, in % of scale. The features may be used to eliminate noise near zero for a flow sensor. |
| 18 | PV_FTIME | RW | S | FLT | 4 | S | 0….60 seconds | | Time constant of a single exponential filter for the PV, expressed in seconds. This is the time necessary for reach the 63% of the variation in input. |
| 19 | FIELD_VAL | R | R | DS-65 | 5 | D | The percent of the value from the Transducer block or from the simulation value, when enabled, before the characterisation (**L_TYPE**) and Filtering (**PV_FTIME**). | | $\text{FIELD\_VAL} = 100 * \frac{\text{CHANNEL\_VALUE} - \text{XD\_SCALE\_0\%}}{\text{XD\_SCALE\_100\%} - \text{XD\_SCALE\_0\%}}$ |
| 20 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | |
| 21 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed. | | |
| 22 | ALARM_SUM | RW | R | DS-74 | 8 | mix | The summary alarm is used for all process alarm in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Actve Status, if the subcode has changed. | | |
| 23 | ACK_OPTION | RW | S | B_STR | 2 | S | Used to set auto acknowledgment of the alarms | | |
| 24 | ALARM_HYS | RW | S | FLT | 4 | S | 0 or > 0 expressed as percent of the **OUT_SCALE** span (default =[0.5%]) | | Alarm Hysteresis is the amount the **PV** must return within the alarm limit before the alarm condition clears. |
| 25 | HI_HI_PRI | RW | S | U8 | 1 | S | 0 - 15 | | Expressed in OUT_SCALE unit. |
| 26 | HI_HI_LIM | RW | S | FLT | 4 | S | Critical Limit High | | |
| 27 | HI_PRI | RW | S | U8 | 1 | S | 0 - 15 | | |
| 28 | HI_LIM | RW | S | FLT | 4 | S | Advisory Limit High | | |
| 29 | LO_PRI | RW | S | U8 | 1 | S | 0 - 15 | | |
| 30 | LO_LIM | RW | S | FLT | 4 | S | Advisory Limit Low | | |
| 31 | LO_LO_PRI | RW | S | U8 | 1 | S | 0 - 15 | | |
| 32 | LO_LO_LIM | RW | S | FLT | 4 | S | Critical Limit Low | | |
| 33 | HI_HI_ALM | RW | R | DS-71 | 16 | D | Critical High Alarm | | |
| 34 | HI_ALM | RW | R | DS-71 | 16 | D | Advisory High Alarm | | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 35 | LO_ALM | RW | R | DS-71 | 16 | D | Advisory Low Alarm | | |
| 36 | LO_LO_ALM | RW | R | DS-71 | 16 | D | Critical Low Alarm | | |
| ENHANCED PARAMETERS | | | | | | | | | |
| 37 | OUT_D | RW | R | DS-66 | 2 | N | Digital Output Value set when the AI_OUT Value, over-cross the thresholds selected with the **ALARM_SEL_TYPE**. The **ALARM_HYS** enter in the calculation for setting and clearing the Digital state in order to avoid continuous changing whenever the Out Float value is around to the threshold. | | Writeable only if **MODE_BLK.ACTUAL = MAN** |
| 38 | ALARM_SEL_TYPE | RW | S | B_STR | 1 | S | Bit 0 | HI_HI_LIM | 0 – Alarm Diabled<br>1 – Alarm Enabled<br>Selection of the AI Limits used as threshold for producing the Digital output when the OUT Value over-cross the selected limits<br>(more of one limit can be simultaneously enabled) |
| | | | | | | | Bit 1 | HI_LIM | |
| | | | | | | | Bit 2 | LO_LIM | |
| | | | | | | | Bit 3 | LO_LO_LIM | |

## Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | – CHANNEL = 0 (uninitialized)<br>– L_TYPE = 0 (uninitialized)<br>– XD_SCALE != OUT_SCALE and LIN_TYPE = DIRECT<br>– CHANNEL = 4 (Scaled_PV), and L_TYPE = Indirect Square Root<br>– CHANNEL = 4 (Scaled_PV), and XD_SCALE different by 0/100 % | When the Block Configuration Error is set, the AI cannot exit from OOS and the OOS condition is also set in the AI Block Error.<br>In this case the OUT_STATUS = BAD-OOS overrides the BAD-Configuration Error |
| Simulate active | The Simulation has been set with the HW switch and the SIMULATE_EN/DIS = Active | The OUT_STATUS is produced as result of the normal calculation within the AI block but starting from a simulated Status instead of the real TB Status |
| Input Failure/process variable has BAD status | The value in input coming from the TB has BAD Status. | IF STATUS_OPTS-Propagate Fault Forward = Set:<br>– BAD Sensor Fail<br>– BAD Device Fail<br>Received in input from the TB are produced also as AI OUT Status.<br>All the other BAD status are propagated as:<br>– BAD-not specific.<br>IF STATUS_OPTS-Propagate Fault Forward = Clear:<br>All the BAD status are propagated as<br>– BAD-not specific |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | The OUT_STATUS is BAD-OOS. In case of concomitance with other conditions this is the status produced in output because this is the High priority Status condition |

## OUT Status

| Binary Code | Decimal Code | Quality | Sub-Status | Status_Opts | Descrition |
|---|---|---|---|---|---|
| 0000 0000 | 0 | BAD | non specific | Propagate Fault Forward = Clear | The value in input at the AI has BAD status |
| 0000 00xx | 1-3 | BAD | non specific | BAD if Limited = Set | The value in input at the AI has status limit set “low limited” or “high limited” or “constant” |
| 0000 1100 | 12 | BAD | Device Failure | Propagate Fault Forward = Set | The value in input at the AI has BAD-Device Failure status |
| 0001 0000 | 16 | BAD | Sensor Failure | Propagate Fault Forward = Set | The value in input at the AI has BAD-Sensor Failure status |
| 0001 1111 | 31 | BAD | Out of Service | | The AI_MODE_BLK.ACTUAL = OOS |
| 0100 0000 | 64 | UNCERTAIN | non specific | Propagate Fault Forward = Clear | The value in input at the AI has UNCERTAIN status |
| 0100 00xx | 65--67 | UNCERTAIN | non specific | UNCERTAIN if Limited = set | The value in input at the AI has status limit set “low limited” or “high limited” or “constant”. See NOTE A |
| 0100 1000 | 72 | UNCERTAIN | Substitute set | UNCERTAIN if Man Mode = set | The MODE_BLK.ACTUAL of the AI = MAN |
| 0101 0100 | 84 | UNCERTAIN | engineering unit range violation | | IF (OUT > (OUT_SCALE_100% + 10%)) OR (OUT < (OUT_SCALE_0% - 10%))<br>In case of reverse range, See NOTE B<br>IF (OUT < (OUT_SCALE_100% - 10%) OR. (OUT > (OUT_SCALE_0% + 10%) |
| 1000 0000 | 128 | GOOD_NC | ok | | The value in input at the AI has GOOD_NC status |
| 1000 0100 | 132 | GOOD_NC | Active block alarm | ACK_OPTION = set | When an AI BLOCK_ERR condition is set, if the AI goes in OOS, the OUT status cannot be set to GOOD_NC.Active block alarm. See NOTE C |
| 1000 1010 | 138 | GOOD_NC | Active Advisory Alarm | ACK_OPTION = set | The OUT_VALUE is outside the limits (HI_HI, HI, LO, LO_LO) and the priority of the limits is between 3 and 7 |
| 1000 1110 | 142 | GOOD_NC | Active Critical Alarm | ACK_OPTION = set | The OUT_VALUE is outside the limits (HI_HI, HI, LO, LO_LO) and the priority of the limits is between 8 and 15 |
| 1000 1001 | 137 | GOOD_NC | Unack block alarm | ACK_OPTION = clear | When an AI BLOCK_ERR condition is set, if the AI goes in OOS, the OUT status cannot be set to GOOD_NC. Unack block alarm. See NOTE C below |

| Binary Code | Decimal Code | Quality | Sub-Status | Status_Opts | Descrition |
|---|---|---|---|---|---|
| 1000 1101 | 141 | GOOD_NC | Unack advisory alarm | ACK_OPTION = clear | The OUT_VALUE is outside the limits (HI_HI, HI, LO, LO_LO) and the priority of the limits is between 3 and 7 |
| 1010 0100 | 164 | GOOD_NC | Unack critical alarm | ACK_OPTION = clear | The OUT_VALUE is outside the limits (HI_HI, HI, LO, LO_LO) and the priority of the limits is between 8 and 15 |

**NOTE A:** When the Transducer Block value goes outside the Range Limits, it should be Limited at the Range Limit high or low, and the Status should be set to GOOD_NC- High Limited or Low Limited.

**NOTE B:** The XD_SCALE and/or OUT_SCALE can be set with EU0% value greater than EU100% value. In this case the test of the OUT value with the range values is inverted.

**NOTE C:** This status can be set only if the specific AI Block_Err condition doesn't force the AI to OOS

## Troubleshooting

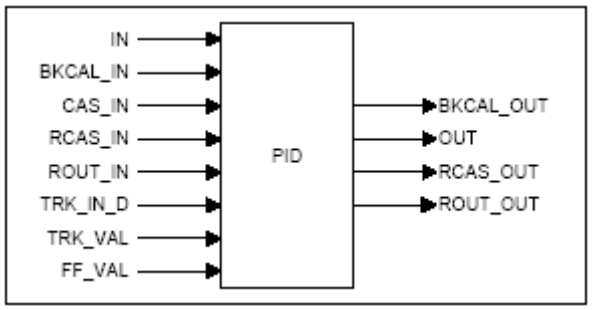
| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is set to OOS | Set the Target Mode to something different by OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | – Set the CHANNEL to a valid value different by 0<br>– Set L_TYPE = DIRECT – If XD_SCALE = OUT_SCALE<br>– Set LIN_TYPE = INDIRECT or IND.SQ ROOT – if XD_SCALE different by OUT_SCALE<br>– IF the AI_CHANNEL = 4 set the AI_LIN_TYPE = indirect<br>– IF the AI_CHANNEL = 4 set the XD_SCALE = 0 / 100 % |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Target Mode is not set to AUTO | Design the FB Application correctly and download it to the devices |
| The Block cannot be switched in AUTO mode | The Simulation has been set with the HW switch and the SIMULATE_EN/DIS = Active | The OUT_STATUS is produced as result of the normal calculation within the AI block but starting from a simulated Status instead of the real TB Status |
| The OUT Status has the Limit bits (0, 1) set to Constant | The Target Mode is not set to AUTO | Set the Target Mode to AUTO |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |
| | LIM_NOTIFY value is less of the MAX_ NOTIFY value | Set the value of LIM_NOTIFY equal, at least, to the MAX_NOTIFY value |
| | STATUS_OPTS has the Propagate Fault Forward bit Set | This bit should be cleared for producing the alarm |

# Enhanced-PID function block (E-PID)

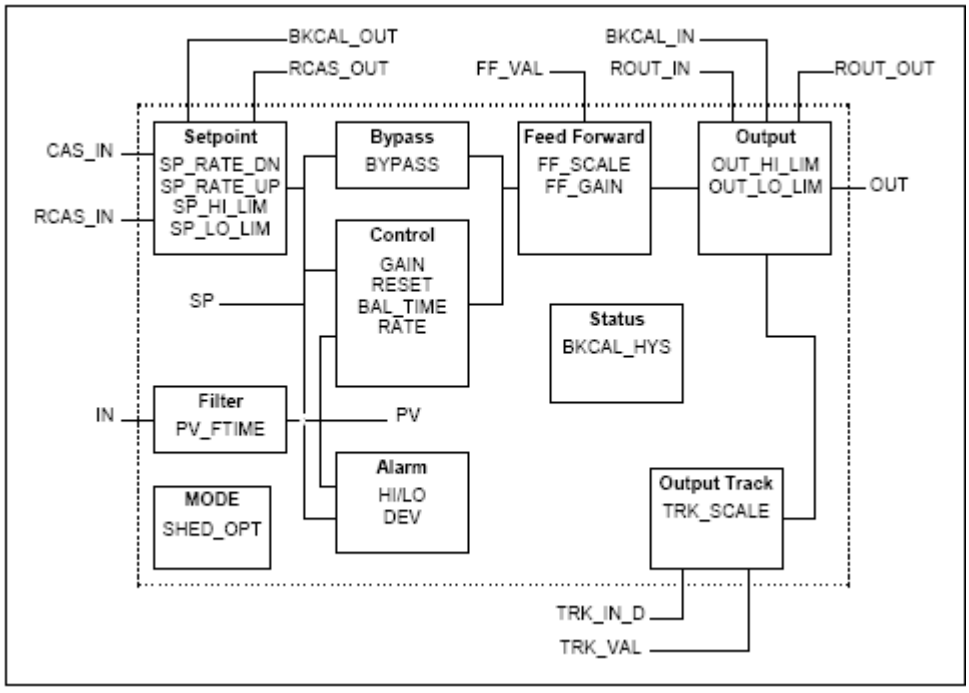
## Overview

The PID block is key to many control schemes and is used almost universally, with the exception of PD, which is used when the process itself does the integration. As long as an error exists, the PID function will integrate the error, which moves the output in a direction to correct the error. PID blocks may be cascaded when the difference in process time constants of a primary and secondary process measurement makes it necessary or desirable.

The PID receives in input the value produced in output from an upstream function block like Analog Input, and provides to apply the algorithm with the Proportional, Integral, Derivative contribute as previously configured.




## Block diagram




## Description

The Process Value to be controlled is connected to the IN input. This value is passed through a filter whose time constant is PV_FTIME. The value is then shown as the PV, which is used in conjunction with the SP in the PID algorithm. A PID will not integrate if the limit status of IN is constant. 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. There are additional control options which will cause the SP value to track the PV value when the block is in an actual mode of IMan, LO, Man or ROut. Limits do not cause SP-PV tracking.

There is a switch for BYPASS, which is available to the operator if the Bypass Enable control option is true. Bypass is used in secondary cascade controllers that have a bad PV.

The Bypass Enable option is necessary because not all cascade control schemes will be stable if BYPASS is true. BYPASS can only be changed when the block mode is Man or O/S. While it is set, the value of SP, in percent of range, is passed directly to the target output, and the value of OUT is used for BKCAL_OUT. When the mode is changed to Cas, the upstream block is requested to initialize to the value of OUT. When a block is in Cas mode, then on the transition out of bypass, the upstream block is requested to initialize to the PV value, regardless of the "Use PV for BKCAL_OUT" option.
GAIN, RESET, and RATE are the tuning constants for the P, I, and D terms, respectively. Gain is a dimensionless number. RESET and RATE are time constants expressed in seconds. There are existing controllers that are tuned by the inverse value of some or all of them, such as proportional band and repeats per minute. The human interface to these parameters should be able to display the user's preference.
The Direct Acting control option, if true, causes the output to increase when the PV exceeds the SP. If false, the output will decrease when the PV exceeds the SP. It will make the difference between positive and negative feedback, so it must be set properly, and never changed while in an automatic mode. The setting of the option must also be used in calculating the limit state for BKCAL_OUT.
The output supports the feed forward algorithm. The FF_VAL input brings in an external value which is proportional to some disturbance in the 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 will be used, because this prevents bumping the output. When the status returns to good, the block will adjust its integral term to maintain the previous output. The output supports the track algorithm.
There is an option to use either the SP value after limiting or the PV value for the BKCAL_OUT value.

## Equations

The algorithm applied is as in the following formula:

$$OUT = GAIN \left[ (BETA \cdot SP - PV) + \frac{1}{RESET \cdot s} (SP - PV) + \frac{RATE \cdot s}{T1\_RATE \cdot s + 1} (GAMMA \cdot SP - PV) \right] + FF\_VAL$$

Where the standard variables are:

| | |
|---|---|
| GAIN: | Proportional Gain Value |
| RESET: | Integral action Time constant in seconds |
| s: | Laplace operator |
| RATE: | Derivative action time constant in seconds |
| FF_VAL: | Feed-forward contribution from the feed-forward input |
| SP: | Setpoint |
| PV: | Process Variable |

Where the the enhanced variables are:

| | |
|---|---|
| T1_RATE: | Derivative 1st order filter |
| BETA: | Setpoint weight proportional part [0…1] |
| GAMMA: | Setpoint weight derivative part [0…1] |

## Configuration hints

The minimum configuration for having the PID working and/or moving out from the OOS needs at least the following settings:
– OUT_HI_LIM > OUT_LO_LIM
– SP_HI_LIM > SP_LO_LIM
– BYPASS = OFF
– SHED_OPT = Normal Shed Normal Return
– GAIN > 0

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / CAS / RCAS / ROUT / OOS | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO/MAN/OOS/IMAN/CAS/RCAS/ROUT/LO | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO / CAS | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | |
| 7 | PV | | R | R | DS-65 | 5 | D | The process variable used in block execution, expressed in **PV_SCALE unit** Code | |
| 8 | SP | | R/W | R | DS-65 | 5 | N | The analog Set Point value of this block, expressed in **PV_SCALE Unit** Code | Acceptable value: **PV_SCALE** +/- 10% |
| 9 | OUT | | RW | R | DS-65 | 5 | N | The block output value calculated as a result of the block execution, expressed in **OUT_SCALE** unit code | Writeable only if **MODE_BLK.ACTUAL** = MAN |
| 10 | PV_SCALE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters which have the same scaling as PV. | |
| 11 | OUT_SCALE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters which have the same scaling as OUT. | |
| 12 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | |
| 13 | CONTROL_OPTS | | RW | S | B_STR | 2 | S | Options the user may select to alter the calculation done in a control loop | |
| | | | | | | | | Bit 0 | Bypass Enable |
| | | | | | | | | Bit 1 | SP-PV Track in Man |
| | | | | | | | | Bit 2 | SP-PV Track in ROut |
| | | | | | | | | Bit 3 | SP-PV Track in LO or IMan |
| | | | | | | | | Bit 4 | SP Track retained target |
| | | | | | | | | Bit 5 | Direct Acting |
| | | | | | | | | Bit 6 | Track if Bad TRK_IN_D |
| | | | | | | | | Bit 7 | Track Enable |
| | | | | | | | | Bit 8 | Track in Manual |
| | | | | | | | | Bit 9 | Use PV for BKCAL_OUT |
| | | | | | | | | Bit 12 | Obey limits if CAS or RCAS |
| | | | | | | | | Bit 13 | No out limits in Manual |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 14 | STATUS_OPTS | RW | S | B_STR | 2 | S | Options which the user can select for the block processing of status. The available selections are: | | |
| | | | | | | | Bit 0 | Initiate Fault Sate if BAD IN | |
| | | | | | | | Bit 1 | Initiate Fault Sate if BAD CAS_IN | |
| | | | | | | | Bit 2 | Use Uncertain as Good | |
| | | | | | | | Bit 5 | Target to Manual if BAD IN | |
| | | | | | | | Bit 9 | Target AUTO if BAD CAS_IN | |
| | | | | | | | Bit 10 | Target to Man if BAD TRK_IN_D | |
| | | | | | | | Bit 11 | IFS if BAD TRK_IN_D | |
| 15 | IN | RW | R | DS-65 | 5 | N | The Primary Input Value for the block coming from another block. Expressed in **PV_SCALE Unit** | | |
| 16 | PV_FTIME | RW | S | FLT | 4 | S | 0….60 seconds | | Time constant of a single exponential filter for the PV, expressed in seconds. This is the time necessary for reach the 63% of the variation in input. |
| 17 | BYPASS | RW | S | U8 | 1 | S | The normal control algorithm may be bypassed trough this parameter. When bypass is set, the set point value (in percent) will be directly transferred to the output. | | |
| | | | | | | | 1 | | OFF |
| | | | | | | | 2 | | ON |
| 18 | CAS_IN | RW | R | DS-65 | 5 | N | Remote set point value from another block. Expressed in **PV_SCALE** Unit Code | | |
| 19 | SP_RATE_DN | RW | S | FLT | 4 | S | 0 or > 0<br>Expressed in PV_SCALE Unit per seconds | | Ramp rate at which downward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero, then the setpoint will be used immediately. For control blocks, rate limiting will apply only in Auto. |
| 20 | SP_RATE_UP | RW | S | FLT | 4 | S | | | Ramp rate at which upward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero, then the setpoint will be used immediately. For control blocks, rate limiting will apply only in Auto. |
| 21 | SP_HI_LIM | RW | S | FLT | 4 | S | Acceptable value: PV_SCALE +/- 10%<br>Expressed in PV_SCALE Unit | | The setpoint high limit is the highest setpoint operator entry that can be used for the block. |
| 22 | SP_LO_LIM | RW | S | FLT | 4 | S | | | The setpoint low limit is the lowest setpoint operator entry that can be used for the block. |
| 23 | GAIN | RW | S | FLT | 4 | S | 0 or > 0 | | The proportional gain value. |
| 24 | RESET | RW | S | FLT | 4 | S | 0 or > 0 | | The integral time constant, expressed in seconds per repeat |
| 25 | BAL_TIME | RW | S | FLT | 4 | S | 0 or > 0 | | The specified time for the internal working value of bias to return to operator set bias. Also used to specify the time constant at which the integral term will move to obtain balance when the output is limited and the mode is AUTO, CAS, or RCAS. Expressed in seconds |
| 26 | RATE | RW | S | FLT | 4 | S | 0 or > 0 | | The derivative action time constant expressed in seconds |
| 27 | BKCAL_IN | RW | R | DS-65 | 5 | N | The analog input value from another block's **BKCAL_OUT** output that is used to prevent reset windup and to initialize the control loop. Expressed in **OUT_SCALE** Unit Code | | |
| 28 | OUT_HI_LIM | RW | S | FLT | 4 | S | Acceptable value: OUT_SCALE +/- 10%<br>Expressed in OUT_SCALE Unit | | Limits the maximum output value. |
| 29 | OUT_LO_LIM | RW | S | FLT | 4 | S | | | Limits the minimum output value. |
| 30 | BCAL_HYS | R | S | FLT | 4 | S | 0 to 50% [Default = 0.5%] – Expressed as percent of the OUT_SCALE span | | The amount that the output must change away from its output limit before the limit status is turned off. |
| 31 | BKCAL_OUT | RW | R | DS-65 | 5 | N | Expressed in PV_SCALE Unit | | Value and status required by an upper block's **BKCAL_IN** so that it may prevent reset windup and provide bumpless transfer to closed loop control. |
| 32 | RCAS_IN | RW | R | DS-65 | 5 | N | Expressed in **PV_SCALE Unit**<br>Used when mode is RCAS | | Target setpoint value provided by a supervisory host. |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|
| 33 | ROUT_IN | RW | R | DS-65 | 5 | N | Expressed in **OUT_SCALE Unit**<br>Used when the mode is ROUT. | Target output value provided by a supervisory host |
| 34 | SHED_OPT | RW | S | U8 | 1 | S | Define actions to be taken on remote control device timeout | |
| 35 | RCAS_OUT | R | R | DS-65 | 5 | D | Expressed in **PV_SCALE Unit.**<br>Used when mode is RCAS. | Block setpoint Value after ramping – provided by a supervisory host for back calculations and to allow action to be taken under limiting conditions or mode change |
| 36 | ROUT_OUT | R | R | DS-65 | 5 | D | Expressed in **OUT_SCALE Unit.**<br>Used when mode is ROUT. | Block output Value provided to a supervisory host for a back calculation to allow action to be taken under limiting conditions or mode change |
| 37 | TRK_SCALE | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point, associated with **TRK_VAL**. | |
| 38 | TRK_IN_D | RW | R | DS-66 | 2 | N | This discrete input is used to initiate external tracking of the block output to the value specified by TRK_VAL. | |
| 39 | TRK_VAL | RW | R | DS-65 | 5 | N | Expressed in **TRK_SCALE Unit**. | This input is used as the track value when external tracking is enabled by TRK_IN_D. |
| 40 | FF_VAL | RW | R | DS-65 | 5 | N | Expressed in FF_SCALE Unit. | The feed forward value and status |
| 41 | FF_SCALE | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point associated with FF_VAL | |
| 42 | FF_GAIN | RW | S | FLT | 4 | S | The gain that the feed forward inpt is multiplied by before it is added to the calculated control output. | |
| 43 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | |
| 44 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed | |
| 45 | ALARM_SUM | RW | R | DS-74 | 8 | mix | The summary alarm is used for all process alarm in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Actve Status, if the subcode has changed. | |
| 46 | ACK_OPTION | RW | S | B_STR | 2 | S | Used to set auto acknowledgment of the alarms | |
| 47 | ALARM_HYS | RW | S | FLT | 4 | S | Alarm Hysteresis is the amount the PV must return within the alarm limit before the alarm condition clears. | 0 or > 0 expressed as percent of the **OUT_SCALE** span (default =[0.5%]) |
| 48 | HI_HI_PRI | RW | S | U8 | 1 | S | 0 - 15 | Expressed in **OUT_SCALE unit.** |
| 49 | HI_HI_LIM | RW | S | FLT | 4 | S | Critical Limit High producing the High-High Alarm | |
| 50 | HI_PRI | RW | S | U8 | 1 | S | 0 - 15 | |
| 51 | HI_LIM | RW | S | FLT | 4 | S | Advisory Limit High producing the High Alarm | |
| 52 | LO_PRI | RW | S | U8 | 1 | S | 0 - 15 | |
| 53 | LO_LIM | RW | S | FLT | 4 | S | Advisory Limit Low producing the Low Alarm | |
| 54 | LO_LO_PRI | RW | S | U8 | 1 | S | 0 – 15 | |
| 55 | LO_LO_LIM | RW | S | FLT | 4 | S | Critical Limit Low producing the Low-Low Alarm | |
| 56 | DV_HI_PRI | RW | S | U8 | 1 | S | 0 - 15 | |
| 57 | DV_HI_LIM | RW | S | FLT | 4 | S | Deviation High Limit producing the Deviation High Alarm | |
| 58 | DV_LO_PRI | RW | S | U8 | 1 | S | 0 - 15 | |
| 59 | DV_LO_LIM | RW | S | FLT | 4 | S | Deviation Low Limit producing the Deviation Low Alarm | |
| 60 | HI_HI_ALM | RW | R | DS-71 | 16 | D | High-High Alarm | |
| 61 | HI_ALM | RW | R | DS-71 | 16 | D | High Alarm | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|
| 62 | LO_ALM | RW | R | DS-71 | 16 | D | Low Alarm | |
| 63 | LO_LO_ALM | RW | R | DS-71 | 16 | D | Low-Low Alarm | |
| 64 | DV_HI_ALM | RW | R | DS-71 | 16 | D | Deviation High Alarm | |
| 65 | DV_LO_ALM | RW | R | DS-71 | 16 | D | Deviation Low Alarm | |
| | | | | | | | ENHANCED PARAMETER | |
| 66 | T1_RATE | RW | S | FLT | 4 | S | Derivative 1st order filter | |
| 67 | BETA | RW | S | FLT | 4 | S | Set-point weight proportional part | |
| 68 | GAMMA | RW | S | FLT | 4 | S | Set-point weight derivative part | |

## Diagnostic

| Block_Err | Possible reason | OUT status |
|---|---|---|
| Block Configuration error | – SHED_OPT = 0 (uninitialized)<br>– BYPASS = 0 (uninitialized)<br>– OUT_HI_LIM =< OUT_LO_LIM<br>– SP_HI_LIM =< SP_LO_LIM | BAD + Out Of Service<br>See Note A |
| Local Override | MODE_BLK.Actual = Local Override | NO EFFECT |
| Input Failure/process variable has BAD status | BAD quality Status in input at the PID_IN. | Depends by the STATUS_OPTS |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

## OUT status

The OUT Status can be affected by the setting of the STATUS_OPTS

## Troubleshooting

| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is not set different of OOS | Set the Target Mode to something different by OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | – Set the OUT_HI_LIM > OUT_LO_LIM<br>– Set the SP_HI_LIM > SP_LO_LIM<br>– Set BYPASS to ON or OFF but different by 0 (uninitialized)<br>– Set SHED_OPT different by 0 |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Block is not scheduled | Design the FB Application correctly and download it to the devices |
| The Block cannot be removed from IMAN mode | Something wrong in the BKCAL_IN | – The Status received in input of the BKCAL_IN is BAD Not Connected. Configure the link with the downstream block<br>– The downstream block is producing a BAD status or Not Invited. Check the reason on the downstream block |
| The Block cannot be switched in AUTO mode | The Target Mode is not set to AUTO | Set the Target Mode to AUTO |
| | Something wrong in the IN | – The Status received in input of the IN is BAD Not Connected. Configure the link with the upstream block<br>– The upstream block is producing a BAD status or Not Invited. Check the reason on the upstream block |
| The Block cannot be switched in CAS mode | The Target Mode is not set to CASCADE | Set the Target Mode to CASCADE |
| | Something wrong in the CAS_IN | – The Status received in input of the CAS_IN is BAD Not Connected. Configure the link of the CAS_IN with another block<br>– The upstream block is producing a BAD status or Not Invited. Check the reason on the upstream block |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |
| | LIM_NOTIFY value is less of the MAX_NOTIFY value | Set the value of LIM_NOTIFY equal, at least, to the MAX_NOTIFY value |

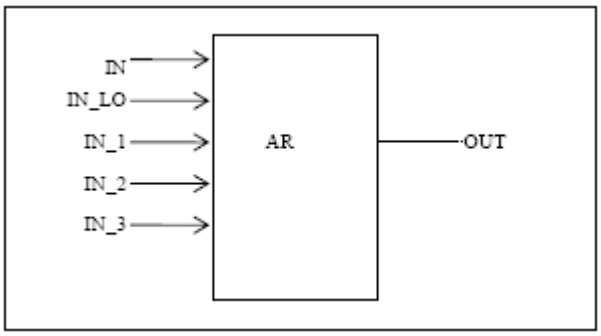
## Arithmetic function block (AR)

### Overview

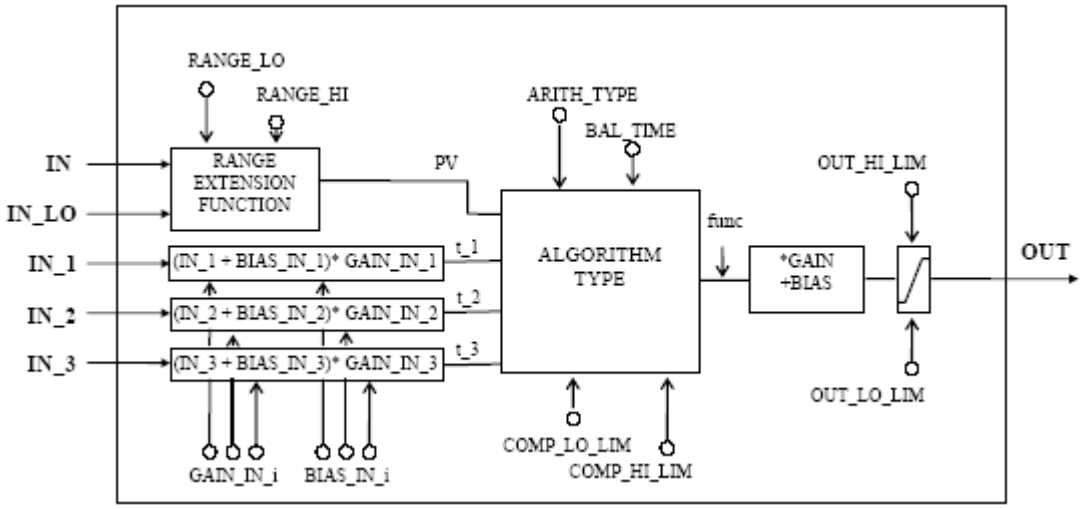
This block is designed to permit simple use of popular measurement math functions. The user does not have to know how to write equations. The math algorithm is selected by name, chosen by the user for the function to be done.

The following algorithms are available selectable from ARTH_TYPE:

– 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.




### Block diagram




### Description

The AR block is intended for use in calculating measurements from combinations of signals from sensors. It is not intended to be used in a control path, so it does not support control status propagation or back calculation. It has no process alarms.

The block has 5 inputs. The first two are dedicated to a range extension function that results in a PV, with status reflecting the input in use. The remaining three inputs are combined with the PV in a selection of four term math functions that have been found useful in a variety of measurements. The inputs used to form the PV should come from devices with the desired engineering units, so that the PV enters the equation with the right units. Each of the additional inputs has a bias and gain constant. The bias can be used to correct for absolute temperature or pressure. The gain can be used to normalize terms within a square root function. The output also has gain and bias constants for any further adjustment required. 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 * IN + (1-g) * IN_LO**

If the status of IN_LO is unusable and IN is usable and greater than RANGE_LO, then g should be set to one. If the status of IN is unusable, and IN_LO is usable and less than RANGE_HI, then g should be set to zero. In each case the PV should have a status of Good until the condition no longer applies. Otherwise, the status of IN_LO is used for the PV if g is less than 0.5, while IN is used for g greater than or equal to 0.5. An optional internal hysteresis may be used to calculate the status switching point. Six constants are used for the three auxiliary inputs. Each has a BIAS_IN_i and a GAIN_IN_i. The output has a BIAS and a GAIN static constant. 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) * GAIN_IN_i**.

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

## Equations

| Algorythm type | Description | Function |
|---|---|---|
| Flow Compensation Linear | Used for density compensation of Volume flow | **OUT= {f · PV · GAIN + BIAS}**<br>Where **f** = $\frac{t\_1}{t\_2}$ is limited |
| Flow Compensation Square Root | Usually:<br>– IN_1 is pressure – (t_1)<br>– IN_2 is temperature – (t_2)<br>– IN_3 is the compressibility factor Z – (t_3) | **OUT= {f · PV · GAIN + BIAS}**<br>Where **f** = $\sqrt{\frac{t\_1}{t\_2 \cdot t\_3}}$ for Volumetric Flow is limited<br>For the calculation of the Volumetric Flow t_3 = Z<br>The compressibility factor Z can be set writing into the IN_3 a constant value Z or can be calculated by a previous block linked in the IN_3.<br>**OUT= {f · PV · GAIN + BIAS}**<br>Where **f** = $\sqrt{\frac{t\_1 \cdot t\_3}{t\_2}}$ for Volumetric Flow is limited<br>In case it would be necessary produce the Mass Flow, the compressibility factor Z must be set as into the IN_3 as $\frac{1}{Z}$ |
| Flow Compensation Approximate | Both IN_1 and IN_2 would be connected to the same temperature<br>NOTE:<br>– The Square Root of the third power can be achieved connecting the input to IN and IN_1.<br>– The Square Root of the fifth power can be achieved connecting the input to IN, IN_1, IN_3. | **OUT= {f · PV · GAIN + BIAS}**<br>Where **f** = $\sqrt{t\_1 \cdot t\_2 \cdot t\_3^2}$ is limited |
| BTU Flow | – IN_1 is the inlet temperature<br>– IN_2 is the outlet temperature | **OUT= {f · PV · GAIN + BIAS}**<br>Where **f = t_1 – t_2** is limited |
| Traditional Multiply Divide | | **OUT= {f · PV · GAIN + BIAS}**<br>Where **f =** $\frac{t\_1}{t\_2}$ **+ t_3** is limited |
| Average | | **OUT=** $\frac{PV + t\_1 + t\_2 + t\_3}{f}$ **· GAIN + BIAS**<br>**f=** number of inputs used in computation |
| Traditional Summer | | **OUT= (PV + t_1 + t_2 + t_3) · GAIN + BIAS** |
| Fourth Order Polynomial | All inputs except IN_LO (not used) are linked together | **OUT= (PV + $t\_1^2$ + $t\_2^3$ + $t\_3^4$) · GAIN + BIAS** |
| Simple HTG Compensated Level | – The PV is the tank base pressure<br>– IN_1 is the top pressure – (t_1)<br>– IN_2 is the density correction pressure – (t_2)<br>– GAIN is the height of the density tap | **OUT=** $\frac{PV - t\_1}{PV - t\_2}$ **· GAIN + BIAS** |

## Configuration hints

The minimum configuration for having the AR working and/or moving out from the OOS needs at least the following settings:
– Set ARITH_TYPE with a valid value. It must be different by 0 and in the range 1 – 9
– If the selected ARITH_TYPE is in the range between 1-5 (limited functions), the output limits COMP_HI_LIM > COMP_LO_LIM
– The BAL_TIME must be greater than the Block Execution Time
– When the ARITH_TYPE = 6 (Average) in case of no inputs available the output will be set to NaN (Not a Number)
– Set the GAIN with value different by 0

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | |
| 7 | PV | | R | R | DS-65 | 5 | D | The process variable used in block execution, expressed in **PV_SCALE unit** Code | |
| 8 | OUT | | RW | R | DS-65 | 5 | N | The block output value calculated as a result of the block execution, expressed in **OUT_SCALE** unit | Writeable only if **MODE_BLK.ACTUAL = MAN** |
| 9 | PRE_OUT | | R | R | DS-65 | 5 | D | expressed in **OUT_SCALE** unit | Displays what would be the OUT value and status if the mode was Auto or lower. |
| 10 | PV_SCALE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters which have the same scaling as PV. | |
| 11 | OUT_RANGE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the scaling for the output. It has no effect on the block | |
| 12 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | |
| 13 | INPUT_OPTS | | RW | S | B_STR | 2 | S | Options the user may select to alter the calculation done in a control loop | |
| | | | | | | | | Bit 0 | IN Use uncertain as good |
| | | | | | | | | Bit 1 | IN_LO Use uncertain as good |
| | | | | | | | | Bit 2 | IN_1 Use uncertain as good |
| | | | | | | | | Bit 3 | IN_1 Use bad as good |
| | | | | | | | | Bit 4 | IN_2 Use uncertain as good |
| | | | | | | | | Bit 5 | IN_2 Use bad as good |
| | | | | | | | | Bit 6 | IN_3 Use uncertain as good |
| | | | | | | | | Bit 7 | IN_3 Use bad as good |
| 14 | IN | | RW | R | DS-65 | 5 | N | The Primary Input Value for the block coming from another block. Expressed in **PV_SCALE Unit** | |
| 15 | IN_LO | | RW | R | DS-65 | 5 | N | Input for the low range transmitter, in a range extension application. Expressed in **PV_SCALE Unit** | |
| 16 | IN_1 | | RW | R | DS-65 | 5 | N | The Primary Input Value for the block coming from another block. Expressed in **PV_SCALE Unit** | |
| 17 | IN_2 | | RW | R | DS-65 | 5 | N | The Primary Input Value for the block coming from another block. Expressed in **PV_SCALE Unit** | |
| 18 | IN_3 | | RW | R | DS-65 | 5 | N | The Primary Input Value for the block coming from another block. Expressed in **PV_SCALE Unit** | |
| 19 | RANGE_HI | | RW | S | FLT | 4 | S | Constant Value above which the range extension has switched to the high range transmitter Expressed in **PV_SCALE Unit** | |
| 20 | RANGE_LO | | RW | S | FLT | 4 | S | Constant Value below which the range extension has switched to the low range transmitter Expressed in **PV_SCALE Unit.** | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note |
|---|---|---|---|---|---|---|---|
| 21 | BIAS_IN_1 | RW | S | FLT | 4 | S | The constant to be added to IN_1 |
| 22 | GAIN_IN_1 | RW | S | FLT | 4 | S | The constant to be multiplied times (IN_1 + Bias) |
| 23 | BIAS_IN_2 | RW | S | FLT | 4 | S | The constant to be added to IN_2 |
| 24 | GAIN_IN_2 | RW | S | FLT | 4 | S | The constant to be multiplied times (IN_2 + Bias) |
| 25 | BIAS_IN_3 | RW | S | FLT | 4 | S | The constant to be added to IN_3 |
| 26 | GAIN_IN_3 | RW | S | FLT | 4 | S | The constant to be multiplied times (IN_3 + Bias) |
| 27 | COMP_HI_LIM | RW | S | FLT | 4 | S | The high limit imposed on the PV compensation term. Expressed in PV_SCALE Unit Code |
| 28 | COMP_LO_LIM | RW | S | FLT | 4 | S | The low limit imposed on the PV compensation term. Expressed in PV_SCALE Unit Code |
| 29 | ARTH_TYPE | RW | S | U8 | 1 | S | The identification number of the arithmetic algorithm<br>1 Flow Compensation, Linear<br>2 Flow Compensation, Square Root<br>3 Flow Compensation, Approximate<br>4 BTU Flow<br>5 Traditional Multiple Divide<br>6 Average<br>7 Traditional Summer<br>8 Fourth Order Polynomial<br>9 Simple HTG compensated Level |
| 30 | BAL_TIME | RW | S | FLT | 4 | S | Acceptable value: **OUT_SCALE** +/- 10%<br>Expressed in **OUT_SCALE Unit**<br>The specified time for the internal working value of bias to return to operator set bias. Also used to specify the time constant at which the integral term will move to obtain balance when the output is limited and the mode is AUTO, CAS, or RCAS. Expressed in seconds |
| 31 | BIAS | RW | S | FLT | 4 | N | Expressed in **OUT_SCALE Unit**<br>The bias value used in computing the function block output |
| 32 | GAIN | RW | S | FLT | 4 | S | 0 or > 0<br>Dimensionless value used by the block algorithm in calculating the block output |
| 33 | OUT_HI_LIM | RW | S | FLT | 4 | S | Acceptable value: **OUT_SCALE** +/- 10%<br>Expressed in **OUT_SCALE Unit**<br>Limits the maximum output value. |
| 34 | OUT_LO_LIM | RW | S | FLT | 4 | S | Limits the minimum output value. |
| 35 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data |
| 36 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed |

## Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | – ARITH_TYPE = 0 (uninitialized)<br>– GAIN = 0<br>– if COMP_HI_LIM =< COMP_LO_LIM and ARITH_TYPE in the range 1-5<br>– if BAL_TIME =< macrocycle and different by 0 | BAD + Out Of Service<br>See Note A |
| Input Failure/process variable has BAD status | At least one of the inputs used in the Output calculation is not usable**:<br>**For the inputs IN and IN_LO usable status are::<br>– GOOD_NC<br>– GOOD_C<br>– UNCERTAIN with INPUT_OPTION = Use uncertain | The worst Status of the used inputs |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

## OUT status

Status of PV depends by the factor g. If it is less than 0,5 it will be used the Status of IN_LO otherwise it will use the Status of IN
The inputs with status byte different by GOOD are controlled by the INPUT_OPTS. The status of unused inputs is ignored. The Status of the OUT will be the same of PV except when the PV is GOOD and the Status of the auxiliary inputs is NOT GOOD and the INPUT_OPTS is not configured to use it. In this case the Status of the OUT is UNCERTAIN. Otherwise the OUT Status id the worst of the inputs used in the calculation after applying the INPUT_OPTS.

## Troubleshooting

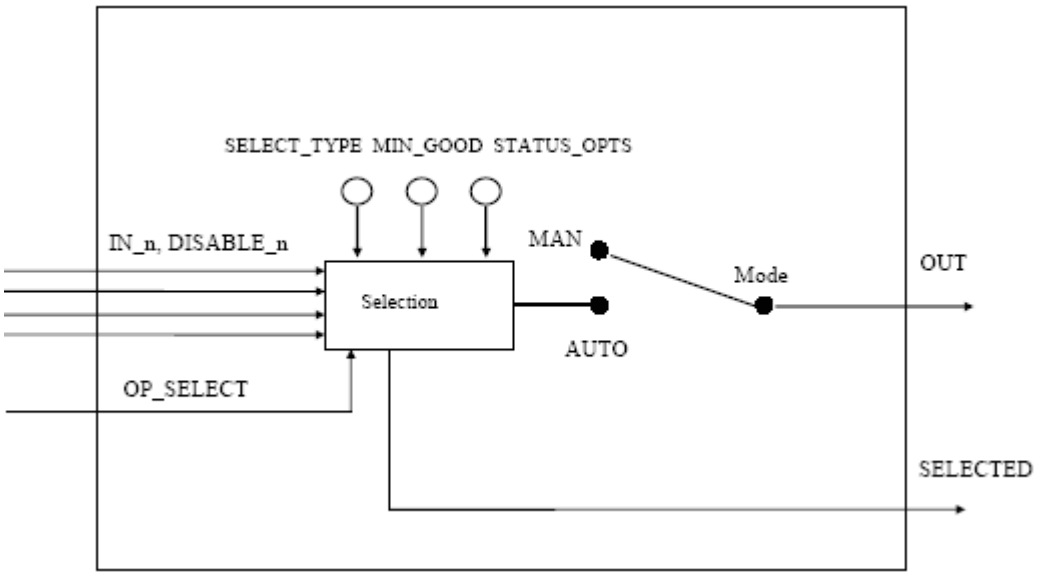
| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is not set to AUTO | Set the Target Mode to AUTO and/or remove the OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | – Set the ARITH_TYPE with a valid value. It must be different by 0 and in the range 1 – 9<br>– Set the GAIN with value different by 0<br>– Set COMP_HI_LIM > COMP_LO_LIM when ARITH_TYPE in the range 1-5<br>– Set BAL_TIME > of the Macrocycle IF different by 0 |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Block is not scheduled | Design the FB Application correctly and download it to the devices |
| The OUT Status is BAD | At least one of used inputs have a BAD status | Check the upstream blocks |
| The OUT Status is UNCERTAIN | At least one of the used inputs have an UNCERTAIN status | Check the upstream blocks |
| The OUT Status has the Limit bits (0, 1) set to Constant | The Actual Mode is set to MAN | Set the Target Mode to AUTO |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |
| | LIM_NOTIFY value is less of the MAX_NOTIFY value | Set the value of LIM_NOTIFY equal, at least, to the MAX_NOTIFY value |

# Input selector function block (AR)

## Overview

The signal selector block provides selection of up to four inputs and generates an output based on the configured action. This block normally receives its inputs from AI blocks. The block performs maximum, minimum, middle, average and 'first good' signal selection. With a combination of parameter configuration options the block can function 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 can receive switching information from either the connected inputs or from an operator input. The block also supports the concept of a middle selection. Although the normal configuration for this feature would be with three signals the block should generate an average of the middle two if four signals are configured or the average of two if three are configured and a bad status is passed to one of the inputs. Logic is provided for handling uncertain and bad signals in conjunction with configured actions. The intended application of this block is to provide control signal selection in the forward path only, therefore, no back calculation support is provided. SELECTED is a second output that indicates which input has been selected by the algorithm.


## Block diagram




## Description

This block is intended to be used in a forward path only and is not intended to receive signals from the output of a controller. There is no back calculation support or propagation of control status values. The processing of the block is as follows.

**Input processing**

If DISABLE_n is true then don't process (ignore) the respective input IN_n.

Process the Use Uncertain as Good status options. Discard (ignore) inputs whose status is BAD.

If there are no inputs left, or fewer than MIN_GOOD inputs, then set the value of SELECTED to zero. Do not do selection processing.

**Selection Processing**

If OP_SELECT is non-zero, the OP_SELECT value shall determine the selected input, regardless of the SELECT_TYPE selection. Set SELECTED to the number of the input used. If SELECT_TYPE is First Good, transfer the value of the first remaining input to the output of the block. Set SELECTED to the number of the input used.

If SELECT_TYPE is Minimum, sort the remaining inputs by value. Transfer the lowest value to the output of the block. Set SELECTED to the number of the input with the lowest value.
If SELECT_TYPE is Maximum, sort the remaining inputs by value. Transfer the highest value to the output of the block. Set SELECTED to the number of the input with the highest value.
If SELECT_TYPE is Middle, sort the remaining inputs by value. If there are 3 or 4 values, discard the highest and lowest value. If two values are left, compute their average. Transfer the value to the output of the block. Set SELECTED to zero if an average was used, else set SELECTED to the number of the input with the middle value.
If SELECT_TYPE is Average compute the average of the remaining inputs and transfer the value to the output of the block.
Set SELECTED to the number of inputs used in the average.

**Limit Processing**

The computations to determine high and low limit conditions for the output can be complex. They should be done to the best of the designer's ability. The limits of OUT should be able to tell a PID to stop integrating if the measurement cannot move.

### Equations

With the SELECT_TYPE it is possible select the following algorithms

| | |
|---|---|
| **First Good** | Select the first available Input with Good Status |
| **Minimum** | Select the minimum value of the Inputs |
| **Maximum** | Select the maximum value of the Inputs |
| **Middle** | Calculate the middle of three inputs or the average of the two middle inputs if four inputs are defined |
| **Average** | Calculate the average value of the inputs |

### Configuration hints

The minimum configuration for having the IS working and/or moving out from the OOS needs at least the following settings:
– Set the SELECT_TYPE with a valid value. It must be different by 0 and in the range 1 – 5.

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | | |
| 7 | OUT | | RW | R | DS-65 | 5 | N | The block output value calculated as a result of the block execution, expressed in **OUT_SCALE unit** | | Writeable only if **MODE_BLK.ACTUAL** = MAN |
| 8 | OUT_RANGE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the scaling for the output. It has no effect on the block | | |
| 9 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | | |
| 10 | STATUS_OPTS | | RW | S | B_STR | 2 | S | Options which the user can select for the block processing of status. The available selections are: | | |
| | | | | | | | | Bit 3 | Propagate Fault Forward | Enable/Disable the propagation of the Status byte from the PRTB in input at the AI to its Output |
| | | | | | | | | Bit 6 | Uncertain if Limited | |
| | | | | | | | | Bit 7 | BAD if Limited | |
| | | | | | | | | Bit 8 | Uncertain if MAN Mode | |
| 11 | IN_1 | | RW | R | DS-65 | 5 | N | Input 1 Value and Status | | |
| 12 | IN_2 | | RW | R | DS-65 | 5 | N | Input 2 Value and Status | | |
| 13 | IN_3 | | RW | R | DS-65 | 5 | N | Input 3 Value and Status | | |
| 14 | IN_4 | | RW | R | DS-65 | 5 | N | Input 4 Value and Status | | |
| 15 | DISABLE_1 | | RW | R | DS-66 | 2 | N | 0 | Use | Parameter to switch off the input 1 from being used |
| | | | | | | | | 1 | Disable | |
| 16 | DISABLE_2 | | RW | R | DS-66 | 2 | N | 0 | Use | Parameter to switch off the input 2 from being used |
| | | | | | | | | 1 | Disable | |
| 17 | DISABLE_3 | | RW | R | DS-66 | 2 | N | 0 | Use | Parameter to switch off the input 3 from being used |
| | | | | | | | | 1 | Disable | |
| 18 | DISABLE_4 | | RW | R | DS-66 | 2 | N | 0 | Use | Parameter to switch off the input 4 from being used |
| | | | | | | | | 1 | Disable | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 19 | SEL_TYPE | RW | S | U8 | 1 | S | This parameter specifies the type of selector action | | |
| | | | | | | | 1 | First Good | |
| | | | | | | | 2 | Minimum | |
| | | | | | | | 3 | Maximum | |
| | | | | | | | 4 | Middle | |
| | | | | | | | 5 | Average | |
| 20 | MIN_GOOD | RW | S | U8 | 1 | S | 0 - 4 | | If the number of inputs which are good is less than the value of MIN_GOOD then set the out status to bad. |
| 21 | SELECTED | RW | R | DS-66 | 2 | D | 0 - 4 | | An integer indicating which input has been selected |
| 22 | OP_SELECTED | RW | R | DS-66 | 2 | N | 0 - 4 | | An operator settable parameter to force a given input to be used |
| 23 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | |
| 24 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed | | |

## Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | SELECT_TYPE = 0 (uninitialized) | BAD + Out Of Service<br>See Note A |
| Input Failure/process variable has BAD status | SELECT_TYPE = AVERAGE and at least one IN is BAD | BAD + non specific |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

## OUT status

When in AUTO mode the OUT reflects the Value and Status of the selected input (IN_x).
If there are no inputs used, or the number of inputs with GOOD status is less than the MIN_GOOD value, the OUT status shall be BAD-Non Specific. The SELECTED output shall have Good(NC) status, unless the block is out of service.
With the STATUS_OPTS it is possible selects the following options:
– **Use Uncertain as Good:** Set the IS_OUT status to Good when the Selected Input Status is Uncertain
– **Uncertain if Manual Mode:** The Status of the IS_OUT is set to Uncertain when the Mode is set to Manual

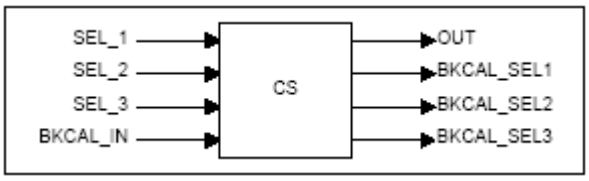
## Troubleshooting

| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is not set to AUTO | Set the Target Mode to AUTO and/or remove the OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | Set the SELECT_TYPE with a valid value. It must be different by 0 and in the range 1 – 5 |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Block is not scheduled | Design the FB Application correctly and download it to the devices |
| The OUT Status is BAD | All the Inputs have a BAD status | Check the upstream blocks |
| | The number of inputs with GOOD status is less than the MIN_GOOD value | |
| | The OP_SELECT is different by 0 and force in output and Input with BAD status | |
| | The SELECT_TYPE = AVERAGE and at least one Input has Status BAD | |
| The OUT Status has the Limit bits (0, 1) set to Constant | The Actual Mode is set to MAN | Set the Target Mode to AUTO |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |

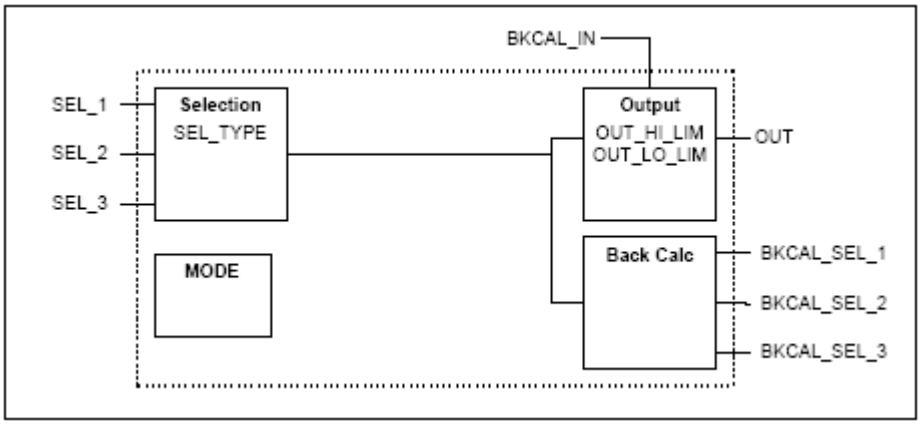
## Control Selector function block (CS)

### Overview

The control selector block is intended to select one of two or three control signals in a manner determined by SEL_TYPE, when the block is in Auto mode. A different block, described in Part 3, is used for selecting a measurement from input or calculation blocks.




### Block diagram




### Description

All inputs to the selector block are assumed to have the same scaling as OUT, since any one of them may be selected to be OUT. Three separate BKCAL_SEL_N outputs are available, one for each SEL_N input. The status will indicate those inputs that are not selected. Control blocks that are not selected are limited in one direction only, determined by the type of selector. The value of each BKCAL_SEL_N output is the same as OUT. The limits of back calculation outputs corresponding to deselected inputs will be high for a low selector and low for a high selector, or one of each for a mid selector.

### Equations

With the SEL_TYPE it is possible select the following algorithms:

1. High
2. Low
3. Middle

### Configuration hints

The minimum configuration for having the CS working and/or moving out from the OOS needs at least the following settings:

– Set the SEL_TYPE with a valid value. It must be different by 0 and in the range 1 – 3

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | The selectable modes by the operator. | | |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. | | |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | Allowed modes that the target may take on | | |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. | | |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | | | |
| 7 | OUT | | RW | R | DS-65 | 5 | N | The block output value calculated as a result of the block execution, expressed in **OUT_SCALE** unit | Writeable only if **MODE_BLK.ACTUAL** = MAN | | |
| 8 | OUT_SCALE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters which have the same scaling as OUT. | | | |
| 9 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | | | |
| 10 | STATUS_OPTS | | RW | S | B_STR | 2 | S | Options which the user can select for the block processing of status. The available selections are: | | | |
| | | | | | | | | Bit 0 | IFS if BAD IN | | |
| | | | | | | | | Bit 2 | Use Uncertain as Good | | |
| 11 | SEL_1 | | RW | R | DS-65 | 5 | N | First input value to the selector | Expressed in **OUT_SCALE Unit** | | |
| 12 | SEL_2 | | RW | R | DS-65 | 5 | N | Second input value to the selector | | | |
| 13 | SEL_3 | | RW | R | DS-65 | 5 | N | Third input value to the selector | | | |
| 14 | SEL_TYPE | | RW | S | U8 | 1 | S | This parameter specifies the type of selector action | 2 | Low | |
| | | | | | | | | 1 | High | 3 | Middle |
| 15 | BKCAL_IN | | RW | R | DS-65 | 5 | N | The analog input value from another block's **BKCAL_OUT** output that is used to prevent reset windup and to initialize the control loop. Expressed in **OUT_SCALE Unit** | | | |
| 16 | OUT_HI_LIM | | RW | S | FLT | 4 | S | Acceptable value: **OUT_SCALE** +/- 10% | Limits the maximum output value. | | |
| 17 | OUT_LO_LIM | | RW | S | FLT | 4 | S | Expressed in **OUT_SCALE Unit** | Limits the minimum output value. | | |
| 18 | BKCAL_SEL_1 | | R | R | DS-65 | 5 | D | Control selector Value and Status associated with **SEL_1** input which is provided to **BKCAL_IN** of the block connected to **SEL_1** in order to prevent reset windup. Expressed in **OUT_SCALE Unit** | | | |
| 19 | BKCAL_SEL_2 | | R | R | DS-65 | 5 | D | Control selector Value and Status associated with **SEL_2** input which is provided to **BKCAL_IN** of the block connected to **SEL_2** in order to prevent reset windup. Expressed in **OUT_SCALE Unit** | | | |
| 20 | BKCAL_SEL_3 | | R | R | DS-65 | 5 | D | Control selector Value and Status associated with **SEL_3** input which is provided to **BKCAL_IN** of the block connected to **SEL_3** in order to prevent reset windup. Expressed in **OUT_SCALE Unit** | | | |
| 21 | UPDATE_EVT | | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | | |
| 22 | BLOCK_ALM | | RW | R | DS-72 | 13 | D | 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 alert will set the Active Status in the status parameter. When 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 | | | |

## Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | SELECT_TYPE = 0 (uninitialized) | BAD + Out Of Service<br>See Note A |
| Input Failure/process variable has BAD status | The value linked in input coming from the upstream blocks has BAD Status. | As Calculated and depending by the STATUS_OPTS |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

## OUT status

The OUT Status of the CS block is the same of the Selected Input exception for:
– If input is Uncertain, the output is Bad unless the STATUS_OPTS is set to **Use Uncertain as Good**.
– If all the inputs are Bad the CS mode goes to MAN as well as it does the PID. This condition produces the OUT Status to be set to IFS if the STATUS_OPTS is set to **IFS if BAD IN.**
– If no inputs have been linked or are valid the OUT Status is set to Bad - Configuration Error
**Supported STATUS_OPTS:**
– IFS if BAD IN
– Use Uncertain as GOOD
**Status supported for other output variables:**
– If the BKCAL_IN status is NI or IR, this status is transferred to the three BKCAL_SEL_x.
– If the BKCAL_IN status is not normal it is transferred to the selected BKCAL_SEL_x output.
– The BKCAL_SEL_x Status of the deselected inputs is set to Not Selected with the appropriate high or low limit set.
– When the CS is in MAN no inputs are selected. All the BKCAL_SEL_x status are set to Not Invited and Constant limits with the same value of OUT.

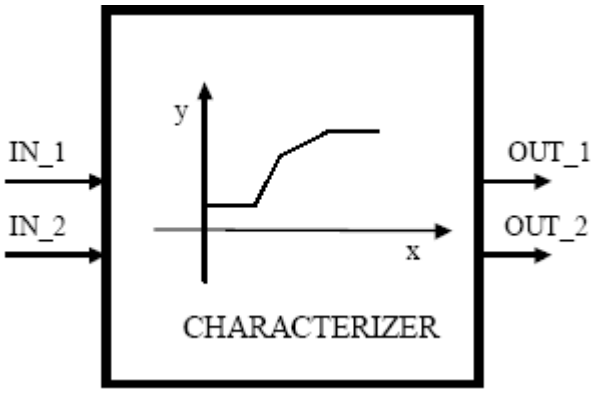
## Troubleshooting

| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is not set to AUTO | Set the Target Mode to AUTO and/or remove the OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | – Set the SEL_TYPE with a valid value. It must be different by 0 and in the range 1 – 3<br>– Set OUT_HI_LIM > OUT_LO_LIM |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Block is not scheduled | Design the FB Application correctly and download it to the devices |
| The Block is in MAN mode | The Target Mode is set to MAN | Set the Target Mode to AUTO |
| | An used input has Bad Status | Check the upstream blocks |
| | The Selected input has UNCERTAIN Status | Set the STATUS_OPTS to Use Uncertain as Good |
| The OUT Status is BAD | There are no inputs linked in<br>(OUT Status = BAD Configuration Error) | Review the FB application design |
| The OUT Status has the Limit bits (0, 1) set to Constant | The Actual Mode is set to MAN | Set the Target Mode to AUTO |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |
| | LIM_NOTIFY value is less of the MAX_NOTIFY value | Set the value of LIM_NOTIFY equal, at least, to the MAX_NOTIFY value |

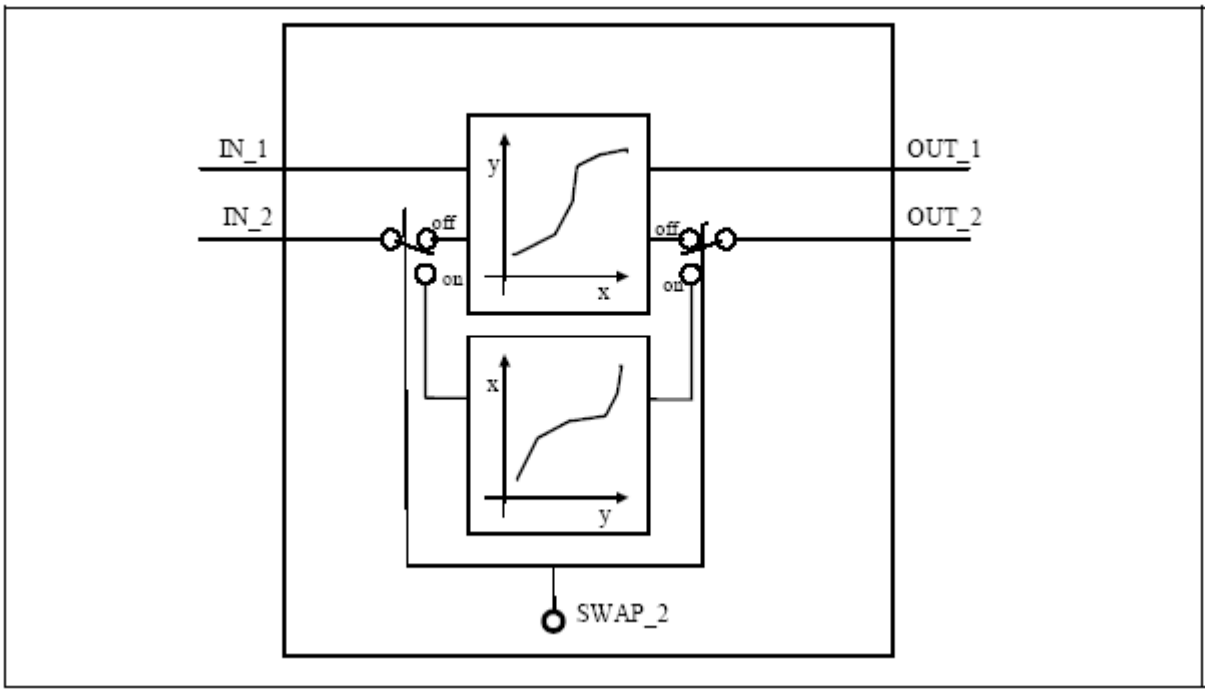
# Signal Characterizer function block (SC)

## Overview

The signal characterizer block has two sections, each with an output that is a non-linear function of the respective input.
The non-linear function is determined by a single look-up table with 21 arbitrary x-y pairs. The status of an input is copied to the corresponding output, so the block may be used in the control or process signal path. An option can swap the axes of the function for section 2, so that it can be used in the backward control path.




## Block diagram



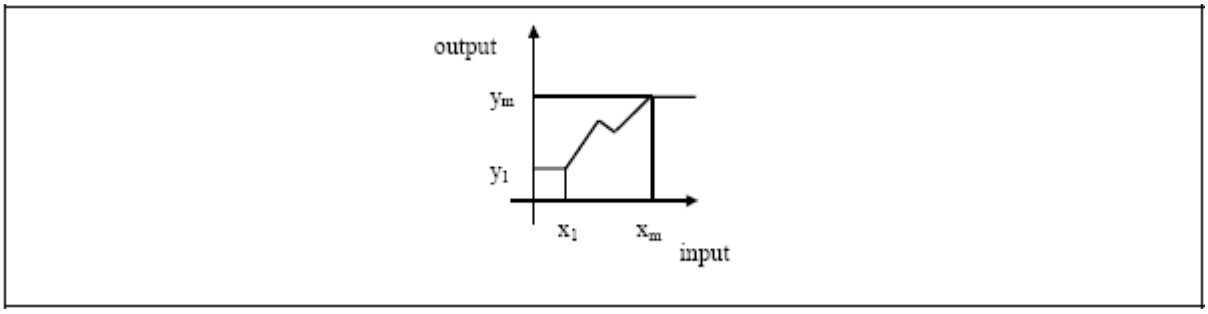

## Description

The block calculates OUT_1 from IN_1 and OUT_2 from IN_2 using a curve given by the points:
[x1 ;y1 ], [x2 ; y2 ] ... [x21 ; y21 ] where **x** corresponds to the Input and **y** to the Output. The **x**-coordinates are given in engineering units of X_RANGE. The **y**-coordinates are given in engineering units of Y_RANGE. The only useful mode is Auto.

**Calculation and the curve:**

OUT_1 is related to IN_1 and OUT_2 to IN_2 by the same curve, but there is no relation between IN_1 and IN_2 or between OUT_1 and OUT_2.
An output value may be calculated by linear interpolation between two points bracketing the input value.
Values of x should increase monotonically, so that interpolation may be possible.

If not, a configuration error shall be set in BLOCK_ERR and the actual mode of the block shall go to Out of Service. Write checks may also be implemented, but they may force the order of entry of the x terms.
If the curve has m points, m<21, the non-configured points, [xm+1; ym+1], [xm+2; ym+2], ... [x21; y21] shall be set to +INFINITY to mark them as unused. Since x1 is the smallest specified value for the input and xm is the largest, the output shall be at y1 when the input is smaller than x1 and at ym when the input is larger than xm. Since the ends of the y curve act as limits, the OUT status shall show when either limit is active.




**Reversing path 2:**
A reverse function swaps the interpretation of IN_2 and OUT_2, which provides a way to do back calculation using the same curve. If the parameter SWAP_2 is set true, the block shall provide:
IN_1 = x and OUT_1 = y while IN_2 = y and OUT_2 = x
If the function is not monotonic in y and SWAP_2 is true, then BLOCK_ERR shall indicate a configuration error and the actual mode go to Out of Service as above for x. A function is called monotonic when **y** values always increase or decrease when x values increase, e.g. the function does not present peaks, valleys, or flat spots.
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.

### Configuration hints

The minimum configuration for having the SC working and/or moving out from the OOS needs at least the following settings:
– Set SWAP_2 different by 0
– Set at least one X and Y pairs
– Set the X values monotonically increasing or the Y values monotonically increasing or decreasing

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | |
| 7 | OUT_1 | | RW | R | DS-65 | 5 | N | The block output 1 value and Status calculated as a result of the block execution, | Writeable only if **MODE_BLK.ACTUAL** = MAN |
| 8 | OUT_2 | | RW | R | DS-65 | 5 | N | The block output 2 value and Status calculated as a result of the block execution, | |
| 9 | X_RANGE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the variables corresponding to the x-axis for display. It has no effect on the block | |
| 10 | Y_RANGE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the variables corresponding to the y-axis for display. It has no effect on the block. | |
| 11 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | |
| 12 | IN_1 | | RW | R | DS-65 | 5 | N | Input 1 Value and Status | |
| 13 | IN_2 | | RW | R | DS-65 | 5 | N | Input 2 Value and Status | |
| 14 | SWAP_2 | | RW | S | U8 | 1 | S | Changes the algorithm in such a way that IN_2 corresponds to “y” and OUT_2 to “x”. | |
| | | | | | | | | 1 | No Swap |
| | | | | | | | | 2 | Swap |
| 15 | CURVE_X | | RW | A | FLT | 84 | S | Curve input points. The xi points of the curve are defined by an array of 21 points | |
| 16 | CURVE_Y | | RW | A | FLT | 84 | S | Curve output points. The yi points of the curve are defined by an array of 21 points | |
| 17 | UPDATE_EVT | | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | |
| 18 | BLOCK_ALM | | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active Status, if the subcode has changed. | |

## Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | – SWAP_2 = 0 (uninitialized)<br>– No X, Y point set<br>– SWAP = FALSE and X values do not increase monotonically<br>– SWAP = TRUE and X values do not increase monotonically OR Y values do not increase or decrease monotonically | BAD + Out Of Service<br>See Note A |
| Input Failure/process variable has BAD status | The value linked in input coming from the upstream blocks has BAD Status. | – The Status of IN_1 is propagated to the OUT_ 1<br>– The Status of IN_2 is propagated to the OUT_2 |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

## OUT status

OUT_1 shall reflect the status of IN_1 and OUT_2 shall reflect the status of IN_2. The sub-status shall also be passed to the outputs. If one of the curve limits is reached or the input is limited, the appropriate limit should 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 O/S mode, the cascade to both the lower and upper blocks shall be broken by Bad status at the outputs. When this block goes to Auto mode, the lower block can begin cascade initialization with status values that pass through this block to the upper block. Answering status signals from the upper block pass through this block to the lower block.
– The block does not use STATUS_OPTS.

## Troubleshooting

| Problem | Possible cause | Solution |
|---|---|---|
| The Block cannot be removed from OOS mode | The Target Mode is not set to AUTO | Set the Target Mode to AUTO and/or remove the OOS |
| | The Configuration Error bit is set in the BLOCK_ERR | – Set SWAP_2 different by 0<br>– Set at least one X, Y pairs<br>– IF SWAP = FALSE set X points with increasing monotonically values<br>– IF SWAP = TRUE set X points with increasing monotonically values and Y points with increasing or decreasing monotonically values |
| | The RESOURCE BLOCK is not in AUTO mode | Set the Target Mode of the RESOURCE BLOCK to AUTO mode |
| | The Block is not scheduled | Design the FB Application correctly and download it to the devices |
| The OUT Status is BAD | The used input has Bad Status | Check the upstream blocks |
| | There are no inputs linked in (OUT Status = BAD Configuration Error) | Review the FB application design |
| The OUT Status has the Limit bits (0, 1) set to Constant | The Actual Mode is set to MAN | Set the Target Mode to AUTO |
| Block Alarm Not Working (Events not notified) | The FEATURE_SEL has not the Reports bit Set | Set the REPORTS bit in the FEATURE_SEL of the RESOURCE BLOCK |
| | LIM_NOTIFY value is less of the MAX_NOTIFY value | Set the value of LIM_NOTIFY equal, at least, to the MAX_NOTIFY value |

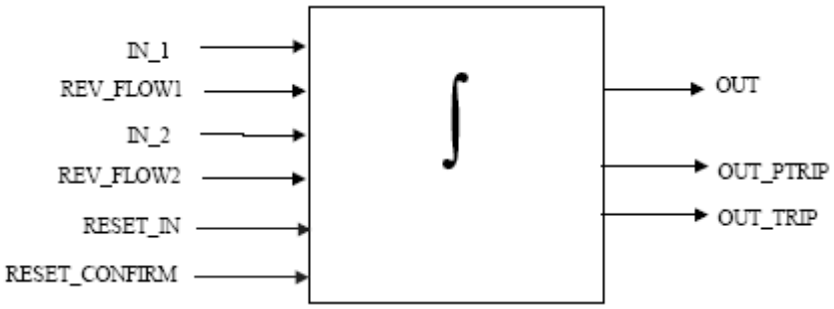
# Integrator function block (IT)

## Overview

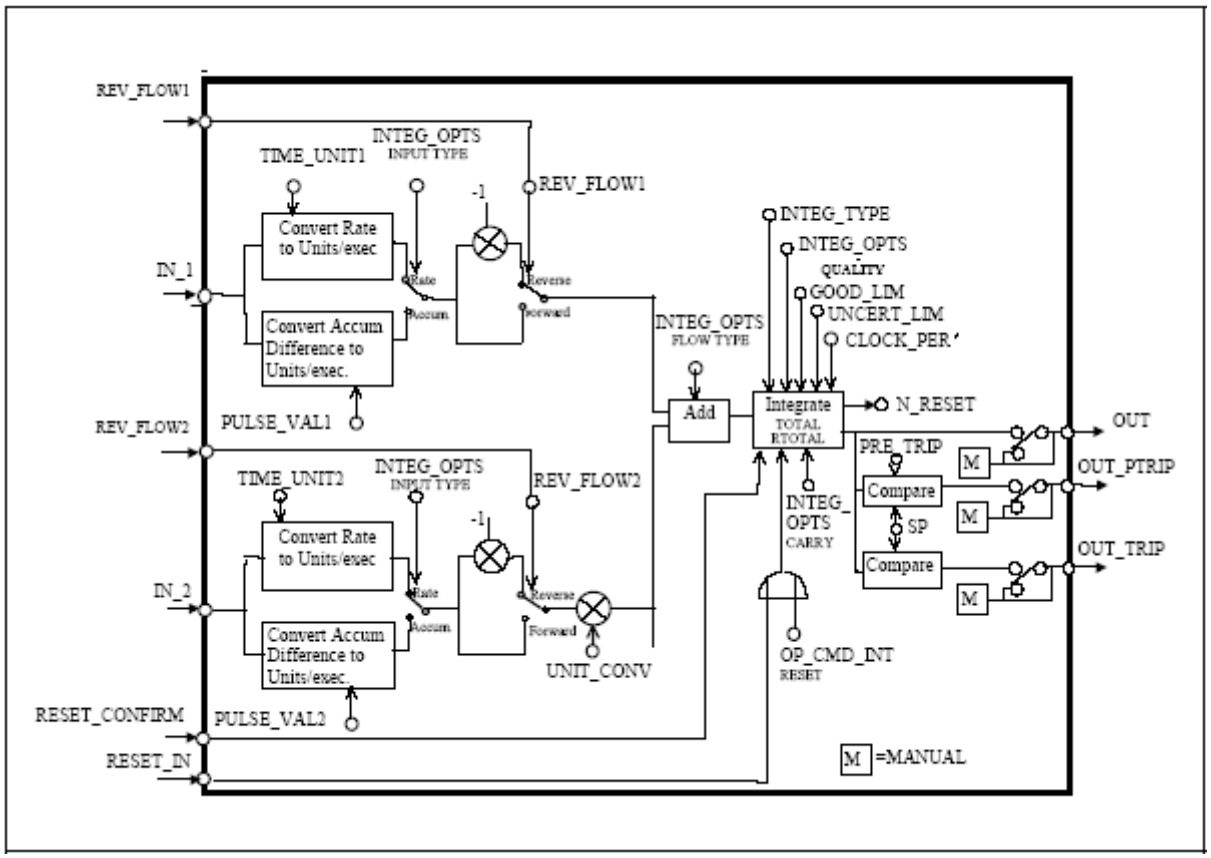
The Integrator Function Block integrates a variable as a function of the time or accumulates the counts from a Pulse Input block (to be described in another specification). The block may be used as a totalizer that counts up until reset or as a batch totalizer that has a setpoint, where the integrated or accumulated value is compared to pre-trip and trip settings, generating discrete signals when these settings are reached. The integrated value may go up, starting from zero, or down, starting from the trip value. The block has two flow inputs so that it can calculate and integrate net flow. This can be used to calculate volume or mass variation in vessels or as an optimizing tool for flow ratio control.

In order to determine the amount of uncertain or bad readings, the block integrates the variables with bad or bad and uncertain status separately. The values used in this second integration are the values with good status just before they went from good to bad or uncertain.

The ratio of good to total counts determines the output status. Absolute values are used to avoid problems with changing signs.




## Block diagram



## Description

The function of this block is in keeping with common industry practice. There is nothing to be tested here concerning the calculation of the integral term. The following description is a guide to the use of the parameters. The basic function of the Integrator block is to integrate an analog value over time. It can also accumulate the pulses coming from Pulse Input blocks or from other Integrator blocks. This block is normally used to totalize flow, giving total mass or volume over a certain time, or totalize power, giving the total energy.

**Inputs**

The block has two dual purpose inputs, IN_1 and IN_2. If IN_2 is not connected (does not have a corresponding link object) then calculations for IN_2 may be omitted. Each input can receive a measurement per unit of time (rate) or an accumulated number of pulses. The usage is as follows:

– Rate: used when the variable connected to the input is a rate, i.e., Kg/s, w, Gal/hour, etc. This input can come from the rate output OUT of a Pulse Input block or from the output of an Analog Input block.

– Accum: used when the input comes from the OUT_ACCUM output of a Pulse Input block, which represents a continuous accumulation of pulse counts from a transducer, or from the output of another Integrator block.

The input type is configured in the bit string parameter INTEG_OPTS. The bits corresponding to IN_1 and IN_2 can be set false for Rate or true for Accum.

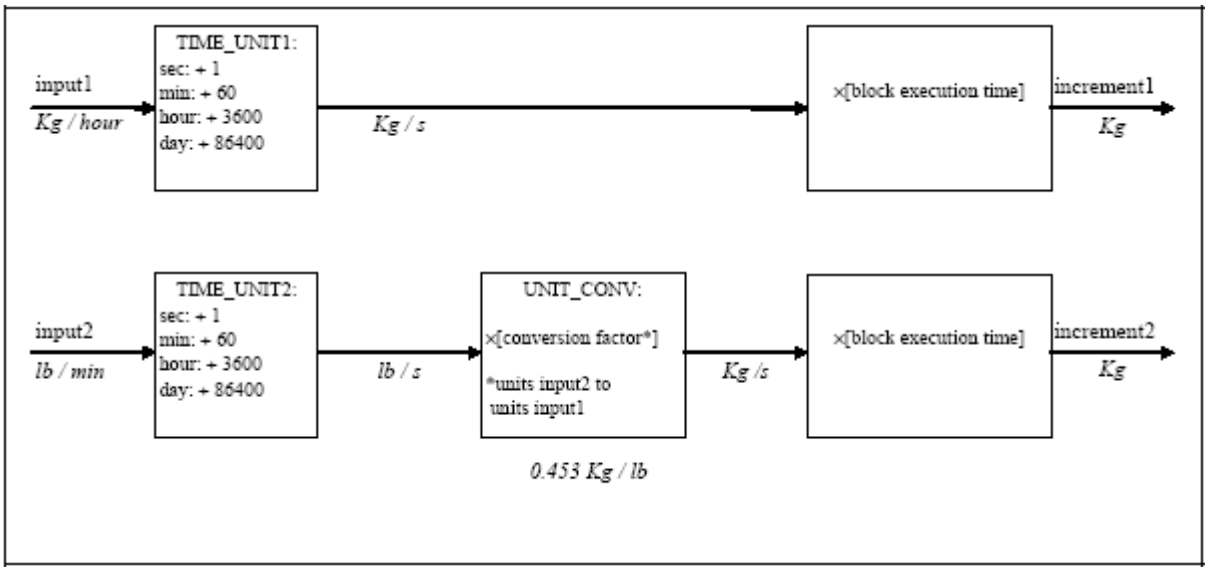
**If the input option is Rate:**

Each input needs a parameter to define the rate time unit: TIME_UNIT1 or TIME_UNIT2. The time units are used to convert the two rates in units of mass, volume or energy per second

The second analog input may have to be converted into the same units of the first input. This is achieved by a unit conversion factor, given by the parameter UNIT_CONV.

Each rate, multiplied by the block execution time, gives the mass, volume or energy increment per block execution. This increment should be added or subtracted in a register, according to some rules defined below.

The following diagram is an example of the use of two Rate inputs:



**If the input option is Accum:**

A counter input normally comes from a Pulse Input block OUT_ACCUM. It can also be connected to the output of another integrator block. The OUT_ACCUM of the Pulse Input block represents a continuous accumulation of pulses from the flow transducer, while the output OUT of an Integrator represents an integration or accumulation of analog inputs.

The Integrator block should determine the number of additional counts from the counter input readings since the last execution. As the output ACCUM_OUT of the Pulse Input block wraps up when the counting reaches 999,999 and does not increment or decrement by more than 499,999 per cycle, the difference in counts is determined as follows:

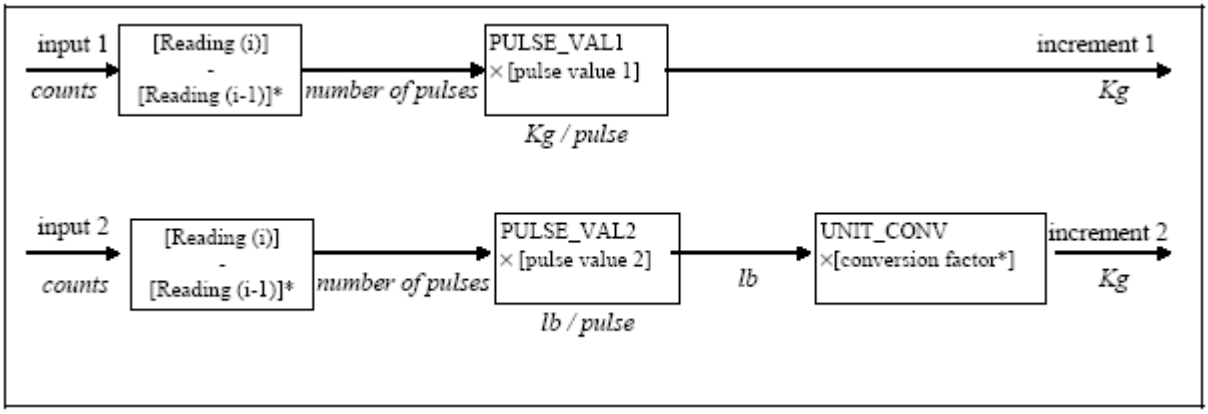
– If the difference between the reading in one cycle and the reading in the preceding cycle is less than 500,000 or greater than (-500,000), the difference should be taken as the variation.

– If the difference between the reading in one cycle and the reading in the preceding cycle is greater than or equal to (+500,000), add (-1,000,000) and use the result as the variation.

– If the difference between the reading in one cycle and the reading in the preceding cycle is more negative than or equal to (-500,000), add (+1,000,000) and use the result as the variation.

If the output OUT of another integrator block is used, that block should be programmed to obey the rules listed above.

The variation of each input should be multiplied by the value, in engineering units, of each pulse given by PULSE_VAL1 or PULSE_VAL2, as appropriate. The result is the increment in engineering units of, for example, mass, volume or energy per block execution (please see diagram below).




**Net Flow**

In order to discern between forward and reverse flows, the Integrator block considers a negative sign as an indication of reverse flow. Some flowmeters already indicate forward and reverse flows by adding a sign to the measurement value. Others use a separate binary signal. This signal can be connected to the inputs REV_FLOW1 and REV_FLOW2, where True should invert the signal of the corresponding input. The net flow is obtained by adding the two increments. The net increment should have a positive or negative signal to indicate the net flow direction. In order to integrate the difference between the inflow and outflow of a tank, for example, the second one can be assigned to be negative.

The net flow direction to be considered in the totalization is defined in INTEG_OPTS. The following options are available:

FORWARD = only positive flows (after application of REV_FLOWi ) are totalized. The negative values should be treated as zero. FORWARD is selected when the bit corresponding to Forward is set to true.

REVERSE = only negative flows are totalized. The positive values should be treated as zero. The option bit Reverse should be set to true

TOTAL = both positive and negative values should be totalized. Both option bits Forward and Reverse should be set to true or to false.

**Integration of Inputs:**

There are three internal registers used for the totalization:

Total = The net increment is added every cycle, regardless of status.

Atotal = The absolute value of the net increment is added every cycle, regardless of status.

Rtotal = The absolute value of the net increments with bad status (rejects) are added to this register.

These internal registers may have greater precision than the standard floating point value. The value of Rtotal requires the same precision as Atotal in order to be able to accumulate floating point fractions so that they are not lost as arithmetic underflow.

The value of a register that corresponds to standard floating point is called the most significant part of the register. The most significant part of Total can be read in the output OUT, and of Rtotal in RTOTAL. OUT_RANGE is used only for display of the totals by a host. The

high and low range values of OUT_RANGE have no effect on the block.

**Types of integration:**
The value of OUT can start from zero and go up or it can start from a Setpoint value (TOTAL_SP) and go down. Reset may be automatic, periodic, or on demand. This is defined by the enumerated parameter INTEG_TYPE:
– UP_AUTO Counts up with automatic reset when TOTAL_SP is reached
– UP_DEM Counts up with demand reset
– DN_AUTO Counts down with automatic reset when zero is reached
– DN_DEM Counts down with demand reset
– PERIODIC Counts up and is reset periodically according to CLOCK_PER
– DEMAND Counts up and is reset on demand
– PER&DEM Counts up and is reset periodically or on demand
The first four types indicate use as a batch totalizer with a setpoint TOTAL_SP. This is not the standard SP because it does not have the structure of SP that is defined in FF-890. The count does not stop at TOTAL_SP going up or zero going down, as it is important to get the true total of flow. Two outputs, OUT_TRIP and OUT_PTRIP, are associated with the four types. See Batch totalizer outputs below.
The next three types indicate that TOTAL_SP and the trip outputs are not used. The Periodic type (5) disables operator reset.
The internal registers always add the net increments. Counting down is done by setting OUT to the value of TOTAL_SP minus the most significant part of Total.

**Resetting the totals:**
The block shall use a discrete input RESET_IN to reset the internal integration registers. The operator can send an operator command to reset the same registers by making OP_CMD_INT = RESET. This is a momentary switch, which shall be turned off when the block is evaluated. Either shall cause reset to occur. Reset should occur after the totals have been adjusted in the same block evaluation. The block should take a snapshot of the most significant part of Total, Rtotal and TOTAL_SP just prior to the reset and move the values to the registers STOTAL, SRTOTAL and SSP, respectively. The information should be kept until the next reset. The integrator should reject reset requests for at least 5 seconds after a reset. This is to allow time for other devices to read the snapshot values before they can be overwritten. The option Confirm Reset in INTEG_OPTS, if set, prevents another reset from occurring until the value 1 has been written to RESET_CONFIRM. This is an Input that behaves like a momentary dynamic parameter if it is not connected. This provides a guarantee that a host has recorded the snapshot values before the next reset can occur. The number of resets is counted in the register N_RESET. This counter can not be written or reset. It provides verification that the total has not been reset since N_RESET was last checked. The counter should roll over from 999999 to 0. Reset always clears the internal registers Total, Atotal and Rtotal, except that when the option UP_AUTO or DN_AUTO is selected, a residual value beyond the trip value may be carried to the next integration if the option Carry is set in INTEG_OPTS. In this case, TOTAL_SP is subtracted from Total, leaving the residual value. The option Generate reset event in INTEG_OPTS shall cause an analog event (DS-75) to be generated at each reset. This messages provides a timestamp and the most significant part of Total just prior to the reset. The Standard Type (4.3) shall be 14, Reset event. The Subcode (4.8) shall be the status byte of OUT. The Value (4.9) shall be the most significant part of Total just prior to the reset. The Unit Index (4.11) shall be the units code of OUT_RANGE.

**Batch totalizer outputs:**
When the integration is counting up (type 1 or 2) and the value of OUT equals or exceeds a value given by TOTAL_SP minus PRE_TRIP then the discrete output OUT_PTRIP is set. When it equals or exceeds a value given by the parameter TOTAL_SP, the discrete output OUT_TRIP is set. OUT_PTRIP remains set. When the integration is counting down (type 3 or 4), it starts from a value given by TOTAL_SP. When the value of OUT is equal to or less than PRE_TRIP, the discrete output OUT_PTRIP is set. When the count reaches zero, the discrete output OUT_TRIP is set. OUT_PTRIP remains set. When a reset occurs, the comparisons that set OUT_PTRIP and OUT_TRIP are no longer true, so they are cleared. OUT_TRIP shall remain set for five seconds after an automatic reset (type 1 or 3) if RESET_CONFIRM is not connected or the option to Confirm Reset in INTEG_OPTS is not set.

### Configuration hints
The minimum configuration for having the IT working and/or moving out from the OOS needs at least the following settings:
– Set TIME_UNIT1 different by 0
– Set TIME_UNIT2 different by 0
– Set INTEG_TYPE different by 0

## Block mapping

| Idx | Parameter | | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BLOCK_OBJ | | mix | R | Mix | 62 | | In the Block Object data structure, there are different items describing the block characteristics. Execution period, Number of parameters in the block, the DD Revision, Profile Revision, View Objects characteristics and so on | | |
| 1 | ST_REV | | R | S | U16 | 2 | N | The revision level of the Static data associated with the Function Block. The revision level is incremented each time a static parameter value (S – under Storage) in the block is changed. | | |
| 2 | TAG_DESC | | RW | S | O_STR | 32 | S | The user description of the intended application of the block | | |
| 3 | STRATEGY | | RW | S | U16 | 2 | S | The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block. | | |
| 4 | ALERT_KEY | | RW | S | U8 | 1 | S | The identification number of the plant unit. This information may be used in the host for sorting alarms, etc. | | |
| 5 | MODE_BLK | TARGET | RW | R | DS-69 | 4 | S | AUTO / MAN / OOS | | The selectable modes by the operator. |
| | | ACTUAL | R | | | | D | | | The mode the block is currently in. |
| | | PERMITTED | RW | | | | S | AUTO / MAN / OOS | | Allowed modes that the target may take on |
| | | NORMAL | RW | | | | S | AUTO | | The common mode for the Actual. |
| 6 | BLOCK_ERR | | R | S | B_STR | 2 | D | This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. | | |
| 7 | TOTAL_SP | | RW | S | FLT | 4 | N | 0 or > 0 | | Set Point for a batch UP totalization. When the **OUT** reaches it, the **OUT** is reset and the **N_RESET** is incremented. |
| 8 | OUT | | RW | R | DS-65 | 5 | N | This variable is the most significant part of the internal Total (Total = net increment added every cycle regardless of status). The OUT value is expressed in **OUT_RANGE Unit**. | | Writeable only if **MODE_BLK.ACTUAL = MAN** |
| 9 | OUT_RANGE | | RW | R | DS-68 | 11 | S | The high and low scale values, engineering units code, and number of digits to the right of the decimal point used only for displaying of the totals (**OUT, STOTAL, RTOTAL SRTOTAL**) by a host. The high and low range values of **OUT_RANGE** have no effect on the block | | |
| 10 | GRANT_DENY | | RW | R | DS-70 | 2 | S | | | |
| 11 | STATUS_OPTS | | RW | S | B_STR | 2 | S | Options which the user can select for the block processing of status. The available selections are: | | |
| | | | | | | | | Bit 8 | Uncertain if MAN Mode | |
| 12 | IN_1 | | RW | R | DS-65 | 5 | N | Input 1. If the input is not set to Accumulate in the **INTEG_OPTS** it is expressed in unit/sec, unit/min, unit/h or unit/day, if the input is set to Accumulate it is expressed in number of pulses. | | |
| 13 | IN_2 | | RW | R | DS-65 | 5 | N | Input 2. If the input is not set to Accumulate in the **INTEG_OPTS** it is expressed in unit/sec, unit/min, unit/h or unit/day, if the input is set to Accumulate it is expressed in number of pulses. | | |
| 14 | OUT_TRIP | | RW | R | DS-66 | 2 | N | 0 | OFF | The first discrete output. Set to ON when the **TOTAL_SP** is reached. This value is set to ON when **OUT** ≥ **TOTAL_SP** in UP totalization or **OUT** ≤ **0** in DOWN totalization. When a reset occurs, **OUT_TRIP** is no longer true, so it is cleared. It shall remain set for 5 seconds after an automatic reset if **RESET_CONFIRM** is not connected or if the I**NTEG_OPTS** is not set to Confirm Reset. |
| | | | | | | | | 1 | ON | |
| 15 | OUT_PTRIP | | RW | R | DS-66 | 2 | N | 0 | OFF | The second discrete output. This value is set to ON when **OUT** ≥ **(TOTAL_SP - PRE_TRIP)** in UP totalization or **OUT** ≤ **PRE_TRIP** in DOWN totalization. When a reset occurs, **OUT_PTRIP** is no longer true, so it is cleared. |
| | | | | | | | | 1 | ON | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 16 | TIME_UNIT_1 | RW | S | U8 | 1 | S | Time unit of the IN_1. It is used to convert the IN_1 in unit per seconds. | | |
| | | | | | | | 0 | Not Initialized | |
| | | | | | | | 1 | Seconds | |
| | | | | | | | 2 | Minutes | |
| | | | | | | | 3 | Hours | |
| | | | | | | | 4 | days | |
| 17 | TIME_UNIT_2 | RW | S | U8 | 1 | S | Time unit of the IN_2. It is used to convert the IN_2 in unit per seconds. | | |
| | | | | | | | 0 | Not Initialized | |
| | | | | | | | 1 | Seconds | |
| | | | | | | | 2 | Minutes | |
| | | | | | | | 3 | Hours | |
| | | | | | | | 4 | days | |
| 18 | UNIT_CONV | RW | S | FLT | 4 | S | Only Positive values accepted, not zero | | Factor to convert the engineering unit of the input 2 into the engineering unit of input 1 |
| 19 | PULSE_VAL 1 | RW | S | FLT | 4 | S | 0 or > 0 | | If the input 1 is set to Accumulation this value converts the number of pulses in the engineer unit. It is expressed in unit per pulse |
| 20 | PULSE_VAL 2 | RW | S | FLT | 4 | S | 0 or > 0 | | If the input 2 is set to Accumulation this value converts the number of pulses in the engineer unit. It is expressed in unit per pulse |
| 21 | REV_FLOW 1 | RW | R | DS-66 | 2 | N | 0 | FORWARD | In order to discern between forward and reverse flows, the Integrator block considers the negative sign as an indication of reverse flow. Some flowmeters already indicate forward and reverse flows by adding a sign to the measurement value. Others use a separate binary signal. This signal can be connected to the inputs **REV_FLOW** 1 for the **IN_1**, when it is True the **IN_1** is inverted. |
| | | | | | | | 1 | REVERSE | |
| 22 | REV_FLOW 2 | RW | R | DS-66 | 2 | N | 0 | FORWARD | In order to discern between forward and reverse flows, the Integrator block considers the negative sign as an indication of reverse flow. Some flowmeters already indicate forward and reverse flows by adding a sign to the measurement value. Others use a separate binary signal. This signal can be connected to the inputs **REV_FLOW 2** for the **IN_2**, when it is True the **IN_2** is inverted. |
| | | | | | | | 1 | REVERSE | |
| 23 | RESET_IN | RW | R | DS-66 | 2 | N | 0 | OFF | External signal used to reset the totalizer |
| | | | | | | | 1 | ON | |
| 24 | STOTAL | R | S | FLT | 4 | N | After Reset the block should take a snapshot/copy of the **OUT** just prior to the reset and move the value to the register **STOTAL.** The information should be kept until the next reset. Expressed in **OUT_RANGE Unit** | | |
| 25 | RTOTAL | RW | S | FLT | 4 | N | This value is the most significant part of the internal **RTOTAL (RTOTAL** = The absolute value of the net increments with bad status (rejects) are added to this register). Expressed in **OUT_RANGE Unit** | | |
| 26 | SRTOTAL | R | S | FLT | 4 | N | After Reset the block should take a snapshot/copy of the **RTOTAL** just prior to the reset and move the value to the register **SRTOTAL**. The information should be kept until the next reset. Expressed in **OUT_RANGE Unit** | | |
| 27 | SSP | R | S | FLT | 4 | N | After Reset the block should take a snapshot/copy of the **TOTAL_SP** just prior to the reset and move the value to the register **SSP**. The information should be kept until the next reset. Expressed in **OUT_RANGE Unit** | | |

| Idx | Parameter | | Data | Type | Size | Storage | Description / Range / Selections / Note | | |
|---|---|---|---|---|---|---|---|---|---|
| 28 | INTEG_TYPE | RW | S | U8 | 1 | S | Defines the type of counting (up or down) and the type of resetting (demand or periodic) | | |
| | | | | | | | 0 | Not Initialized | |
| | | | | | | | 1 | UP AUTO | |
| | | | | | | | 2 | UP DEM | |
| | | | | | | | 3 | DN AUTO | |
| | | | | | | | 4 | DN DEM | |
| | | | | | | | 5 | PERIODIC | |
| | | | | | | | 6 | DEMAND | |
| | | | | | | | 7 | PER & DEM | |
| 29 | INTEG_OPTS | RW | S | B_STR | 2 | S | A bit string to configure the type of input (rate or accum.) used in each input, the flow direction to be considered in the totalization, the status to be considered in TOTAL and if the totalization residue shall be used in the next batch (only when I**NTEG_TYPE = UP_AUTO or DN_AUTO).** | | |
| | | | | | | | Bit 0 | Input 1 Accumulate | |
| | | | | | | | Bit 1 | Input 2 Accumulate | |
| | | | | | | | Bit 2 | Flow Forward | |
| | | | | | | | Bit 3 | Flow Reverse | |
| | | | | | | | Bit 4 | Use Uncertain | |
| | | | | | | | Bit 5 | Use Bad | |
| | | | | | | | Bit 6 | Carry | |
| | | | | | | | Bit 7 | Add zero if Bad | |
| | | | | | | | Bit 8 | Confirm Reset | |
| | | | | | | | Bit 9 | Generate Reset Event | |
| 30 | CLOCK_ERR | RW | S | FLT | 4 | S | 0 or > 0 | | Establishes the period for periodic reset, in seconds |
| 31 | PRE_TRIP | RW | S | FLT | 4 | S | 0 or > 0 | | Value used for the **OUT_PTRIP** setting. It adjusts the amount of mass, volume or energy that should set OUT_PTRIP when the integration reaches **(TOTAL_SP - PRE_TRIP)** when counting UP or **PRE_TRIP** when counting DOWN. Expressed in **OUT_RANGE Unit** |
| 32 | N_RESET | R | S | FLT | 4 | N | The number of resets is counted in the register **N_RESET**. This counter can not be written or reset. It provides verification that the TOTAL has not been reset since **N_RESET** was last checked. The counter should roll over from 999999 to 0. | | |
| 33 | PCT_INCL | R | S | FLT | 4 | D | This value is the % of the absolute net increment with good status respect of the absolute net increment regardless of the status. | | |
| 34 | GOOD_LIM | RW | S | FLT | 4 | S | If **PCT_INCL ≥ GOOD_LIM** and the mode is AUTO the status of the OUT is GOOD_NC otherwise check the other limit. Expressed in % | | |
| 35 | UNCERT_LIM | RW | S | FLT | 4 | S | If **PCT_INCL ≥ UNCERT_LIM** and the mode is AUTO the status of the OUT is UNCERTAIN otherwise is BAD. Expressed in % | | |
| 36 | OP_CMD_INT | RW | S | U8 | 1 | D | 0 | OFF | Operator reset command. Resets the totalizer |
| | | | | | | | 1 | ON | |
| 37 | OUTAGE_LIM | RW | S | FLT | 4 | S | 0 or > 0 | | The max, tolerated duration for power failure. This value is used by the host expressed in seconds |
| 38 | RESET_CONFIRM | RW | R | DS-66 | 2 | N | 0 | OFF | If the Confirm Reset in the **INTEG_OPTS** is set, any further reset will be prevented until a logic 1 is not received in input to the **RESET_CONFIRM**. If not connected it acts like a momentary dynamic parameter. This ensures that a host has recorded the snapshot values before the next reset can occur. |
| | | | | | | | 1 | CONFIRM | |
| 39 | UPDATE_EVT | R | R | DS-73 | 14 | D | This alert is generated by any change to the static data | | |
| 40 | BLOCK_ALM | RW | R | DS-72 | 13 | D | The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active Status in the status parameter. When 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 | | |

### Operating Mode

In manual mode, the outputs are disconnected from the algorithm and the user can set the values of OUT, RTOTAL, OUT_TRIP and OUT_PTRIP for test purposes. No integration takes place. When the block is switched to Auto, the integration starts from the value set manually. Each write to OUT or RTOTAL shall increment the N_RESET counter. In auto mode, the outputs follow the algorithm.

### Diagnostic

| Block_Err | Possible Reasons | OUT Status |
|---|---|---|
| Block Configuration error | – TIME_UNIT1 = 0<br>– TIME_UNIT2 = 0<br>– INTEG_TYPE = 0<br>– IF INTEG_OPTS = IN_1 ACCUMULATE<br>o PULSE_VAL1 = 0<br>– IF INTEG_OPTS = IN_2 ACCUMULATE<br>o PULSE_VAL2 = 0<br>– IF INTEG_TYPE = PERIODIC<br>o CLOCK_PER = 0 | BAD + Out Of Service<br>See Note A |
| Input Failure/process variable has BAD status | The value linked in input coming from the upstream blocks has BAD Status. | Calculated according the algorithm.<br>See the OUT STATUS section below |
| Out-of-Service | The Actual_Mode is OUT OF SERVICE | BAD + Out Of Service |

**NOTE A:** The specific block cannot be switched out from OUT OF SERVICE due to the Configuration Error. The Bad-Configuration Error Status is overridden by the Bad-Out Of Service Status.

### OUT status

If an input has a status of Uncertain or Bad, it shall be treated as explained below. The limit status of the inputs is ignored, as is the substatus. Either Good(C) or Good(NC) are accepted as good.

The increment calculated from an input has an internal status that is either good or bad. If the input status is Good(C) or Good(NC) the increment status is good. If the input status is Uncertain, the increment status is bad, and the last good value is used unless the option Use Uncertain is set in INTEG_OPTS, and then the increment status is good and the new value is used. If the input status is Bad, the increment status is bad, and the last good value is used unless the option Use Bad is set in INTEG_OPTS, and then the increment status is good and the last good value is used.

The two increments are added together, and the resulting status is the worst of the two.

The option Add zero if bad in INTEG_OPTS causes the net increment to be zero if its status is bad.

The percentage of bad or uncertain and bad counts may be determined by calculating the value of PCT_INCL from Rtotal and Atotal. Since Atotal is the sum of increments with good and bad status, and Rtotal is the sum of increments with bad status, Atotal minus Rtotal is exactly equal to the total of increments with good status. If msp is used to mean “most significant part” and Atotal is not zero then the percent of good values may be calculated as:

PCT_INCL = 100 * ( 1 - (msp of Rtotal) / (msp of Atotal) )

If Atotal is zero, then PCT_INCL shall be 100 if Rtotal is also zero, or 0 if Rtotal is not zero.

If the block mode is Auto, if PCT_INCL ≥ GOOD_LIM, the status of OUT shall be Good, or else if PCT_INCL ≥ UNCERT_LIM, the status of OUT shall be Uncertain, or else the status of OUT shall be Bad.

If the block mode is Manual, then the status of OUT, OUT_PTRIP, and OUT_TRIP will be Good (NC) constant when then status option Uncertain if Man is not selected. If this status option is selected and the block mode is manual, then the status of these three outputs will be Uncertain constant. No limits are applied to the output.

# Appendix B – Device installation and commissioning into ABB Control System

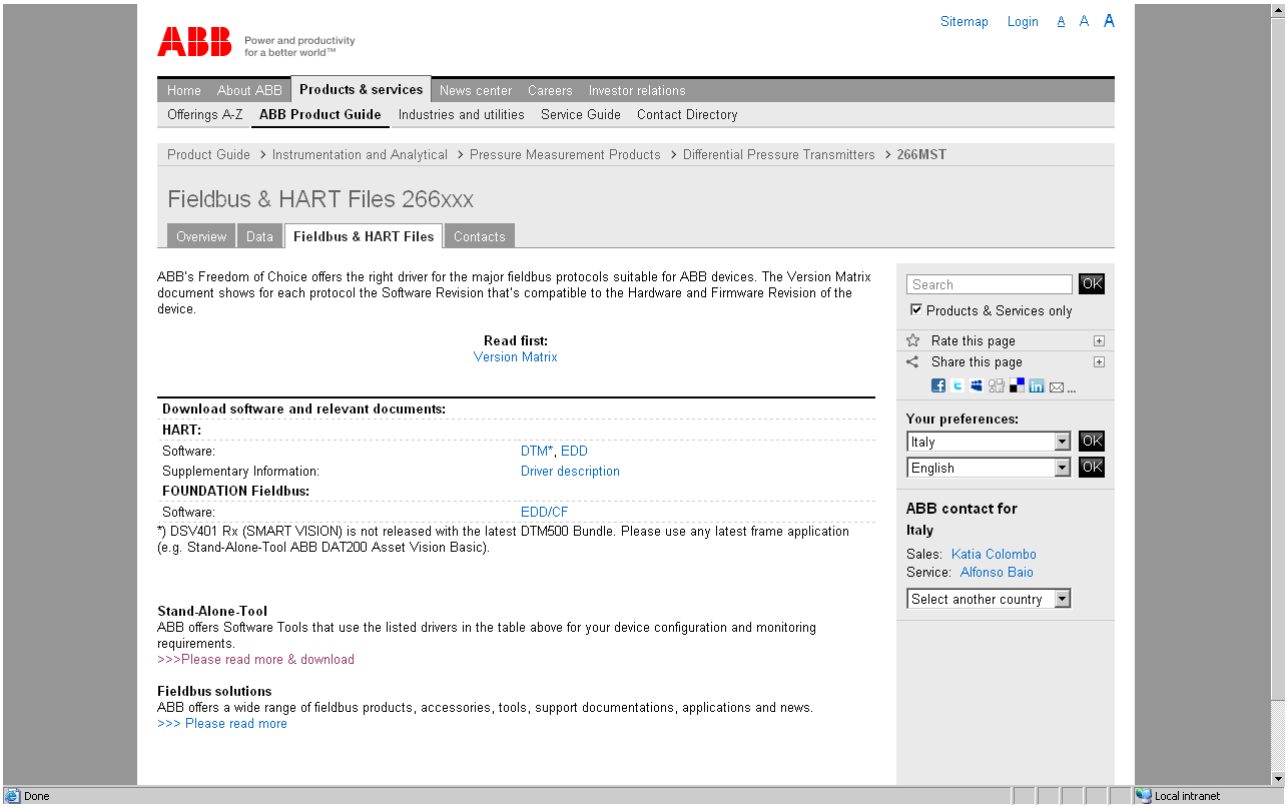
In order to make 266 PdP working with any FF host it is necessary perform some operations as described in the following sections. The description below is based on the 266 PdP connected to an ABB System but a similar approach is in general valid also for other non ABB hosts. A summary of the required operations is:

– Off Line Configuration
  Importing of the FF device drivers DD&CFF in the host
  Design of the FF H1 network
  Design of the FBAP
– ON Line Configuration
  Assignment of the FF device
  Downloading of the FBAP to the H1 network and devices
  Device and/or Blocks Configuration

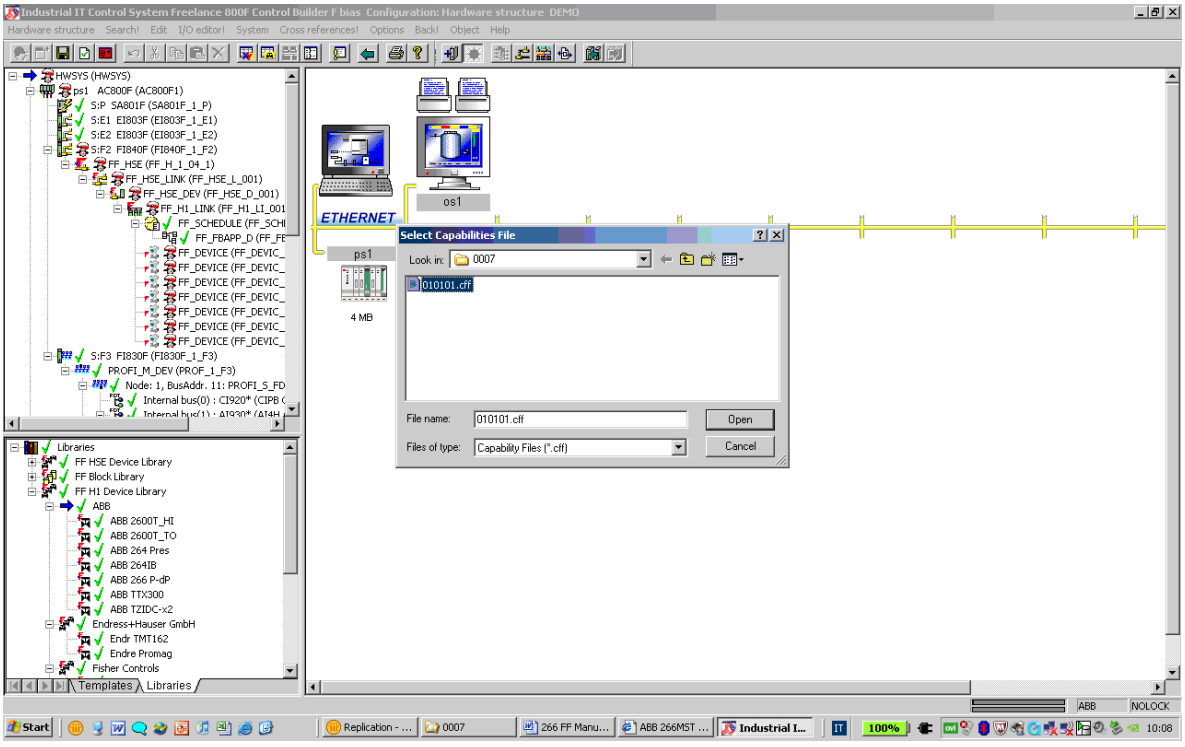
> **Attention**
> The first part of the operations is executed in OFF-Line. OFF line means that is not necessary has the real device connected on the FF H1 network to the host.

## Importing of the FF device drivers DD&CFF in the host

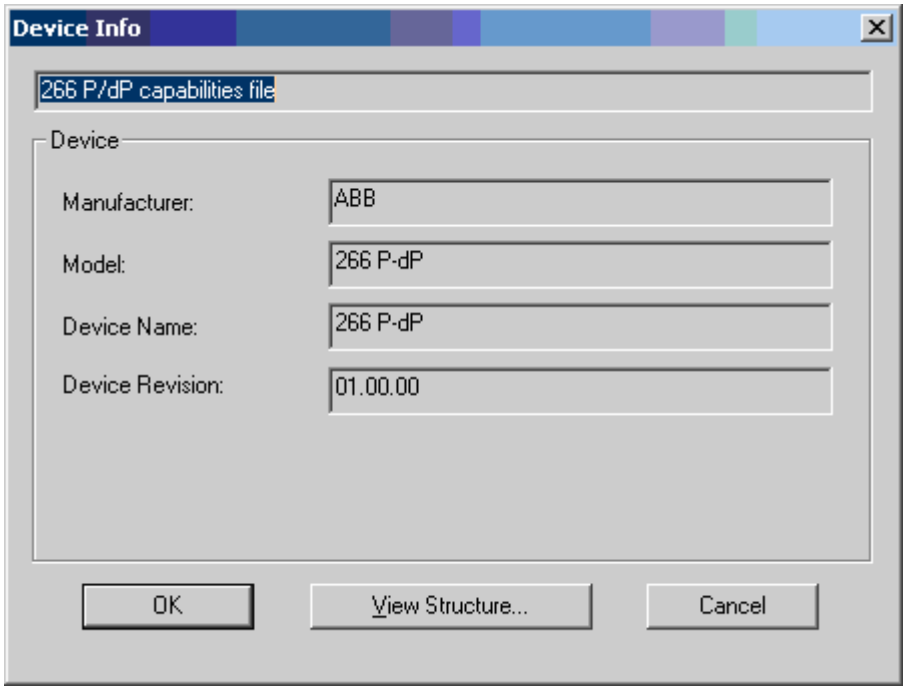
The DD&CFF drivers of the 266 PdP FOUNDATION™ Fieldbus have to be previously downloaded from the From the ABB website, www.abb.com/instrumentation select the 266 PdP and from the Fieldbus&Hart page download the FOUNDATION Fieldbus EDD/CF file into a dedicate directory



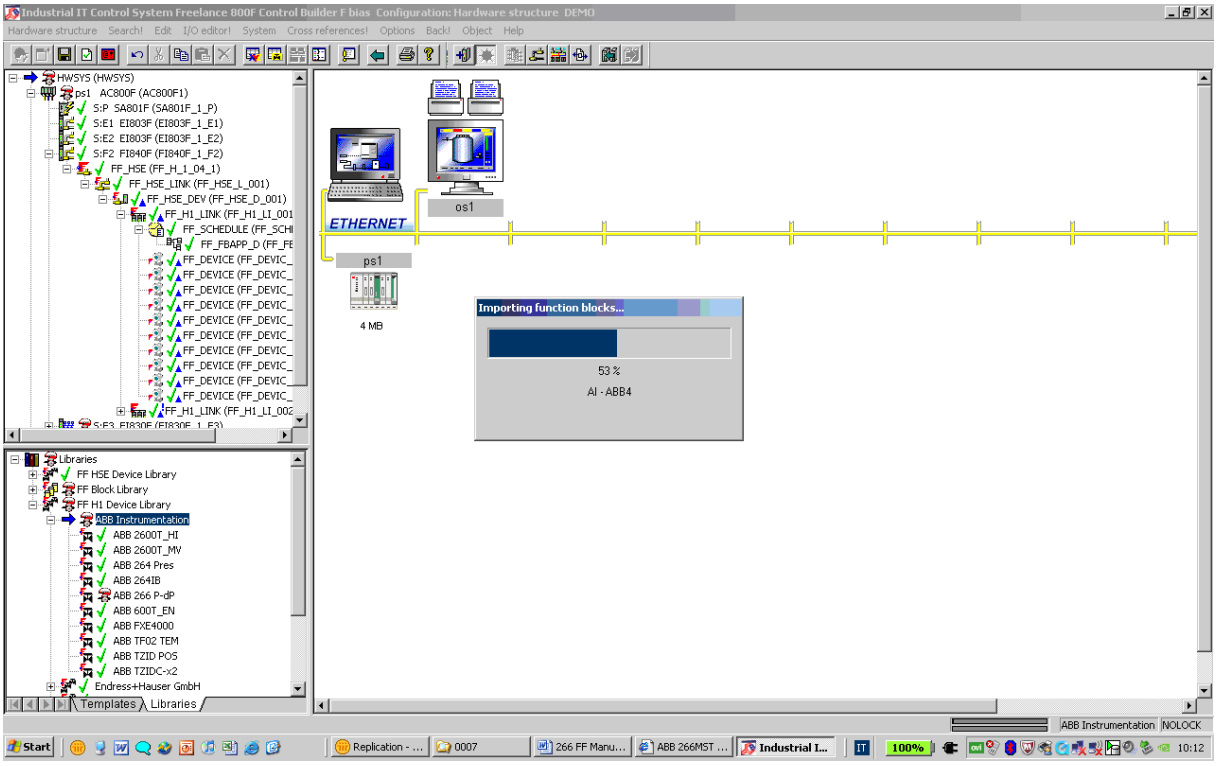
Then from the Engineering Station of the System (Control Builder) open Libraries and select “insert”.
The “Select Capability File” window is opened and from its browser search the downloaded 266 PdP Capability File (CFF) in the hard disk. Once it has been found press “Open”.



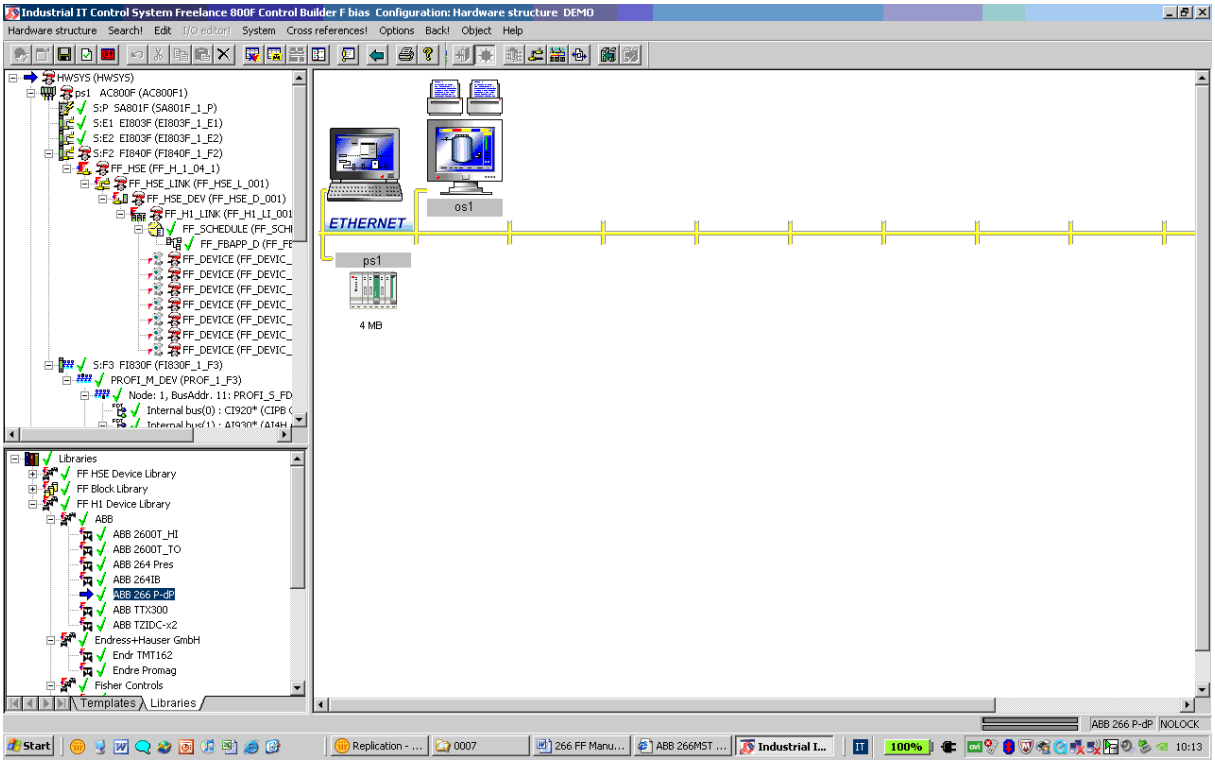

The “Device Info” box appears and then press “OK”



The “Importing Function blocks….” get start. Wait until all the blocks are fully imported….. (100%)



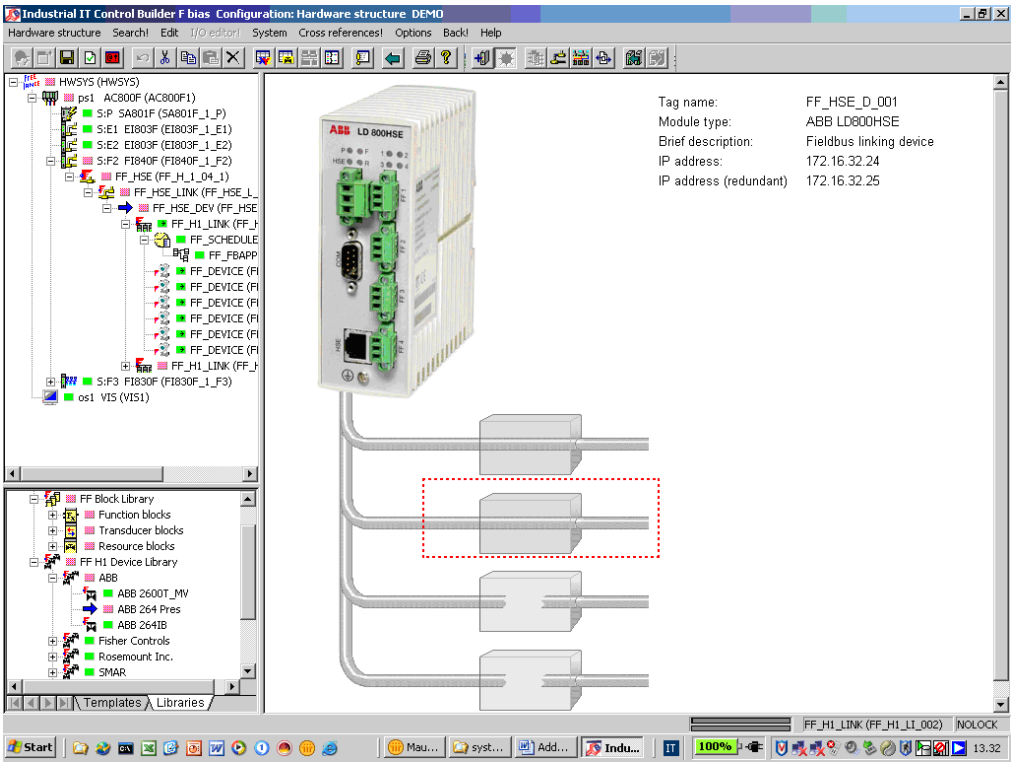

When completed, the “ABB 266 PdP” appears now in the FF library



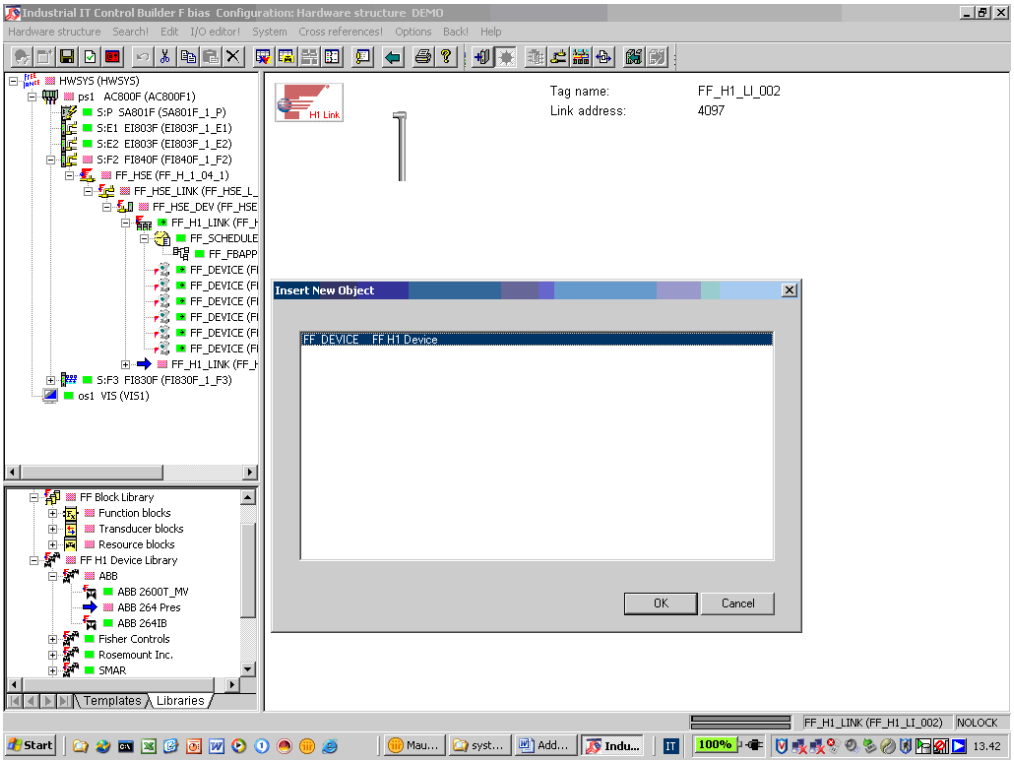
# 266 Models - FOUNDATION Fieldbus

## Design of the FF H1 network

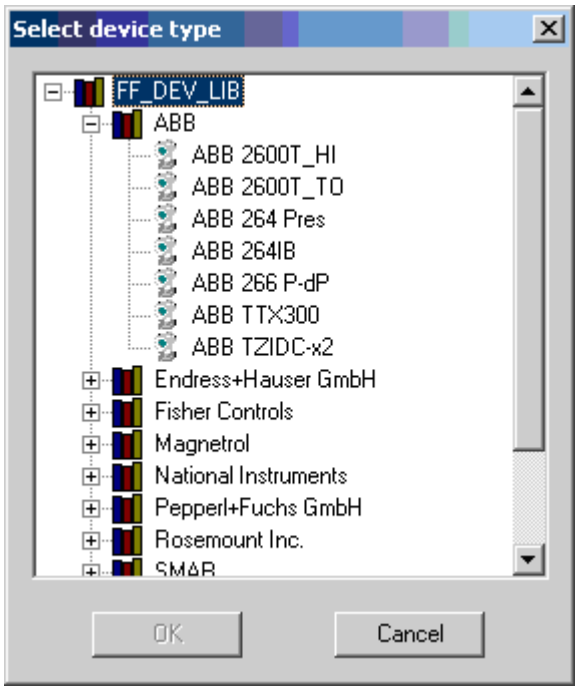
Select one of the four (4) FF H1 segments supported by the ABB Linking Device LD800HSE with a double right mouse click on the desired line.



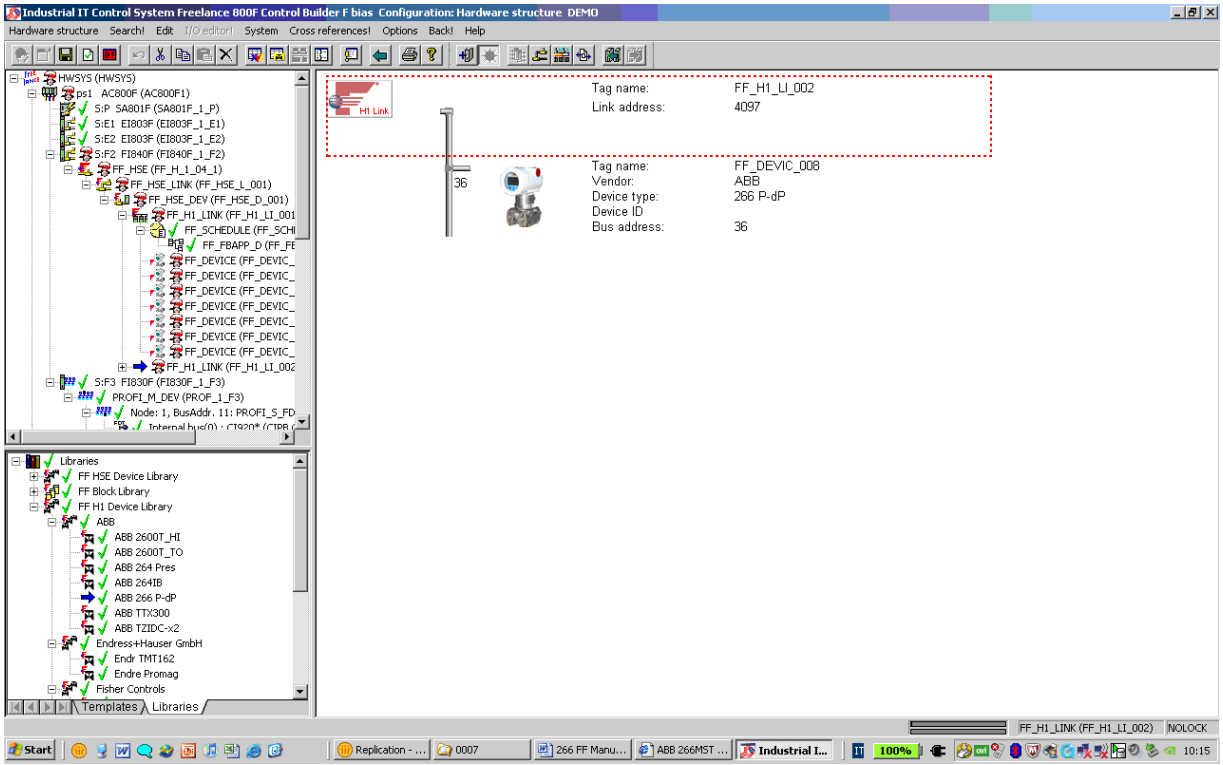

Select Insert with a right mouse click, the “Insert new object” box appears and press “OK”.



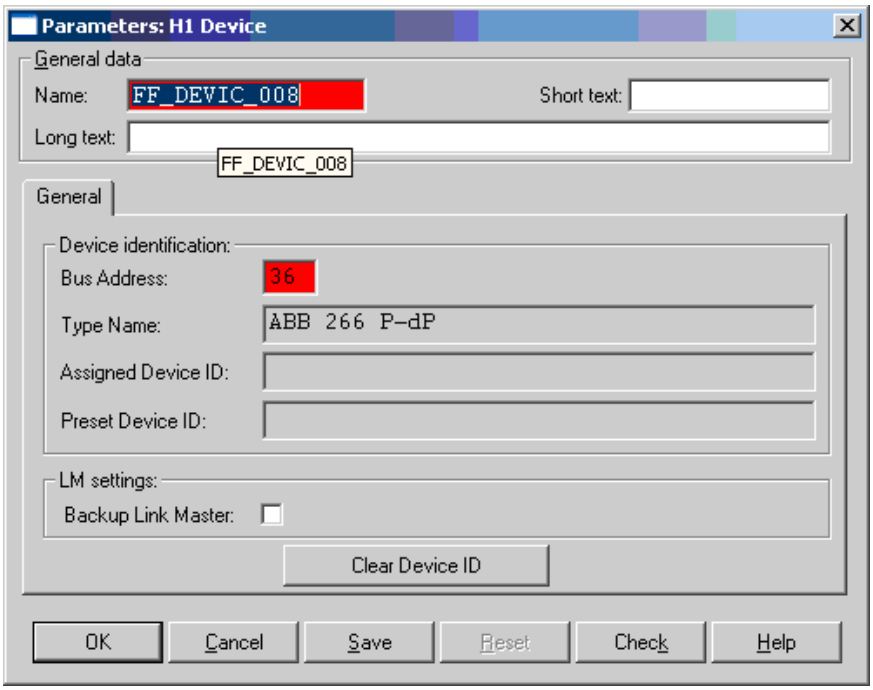
The “Select device type” box appears with a list of manufacturers and device types depending by the DD/CFF drivers imported in the host. Under ABB are available the drivers of the FF ABB devices.




Select “ABB 266 PdP” and press OK. The 266 PdP appears now in the H1 segment with predefined TAG and Address……

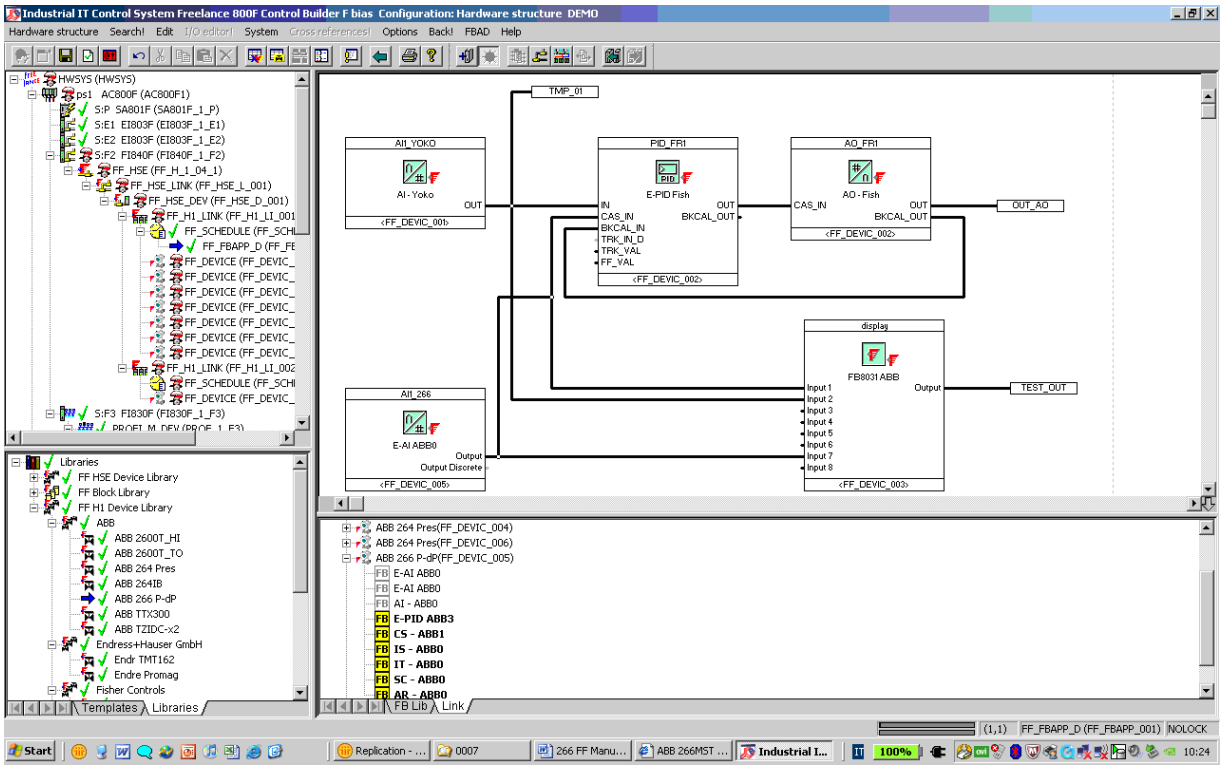


……parameters that can be changed as desired opening the “Parameters” box of the device with a right mouse click. From this box is also possible select the Backup Link Master function (LAS) of the device.




### Design of the Function Block Application (FBAP)

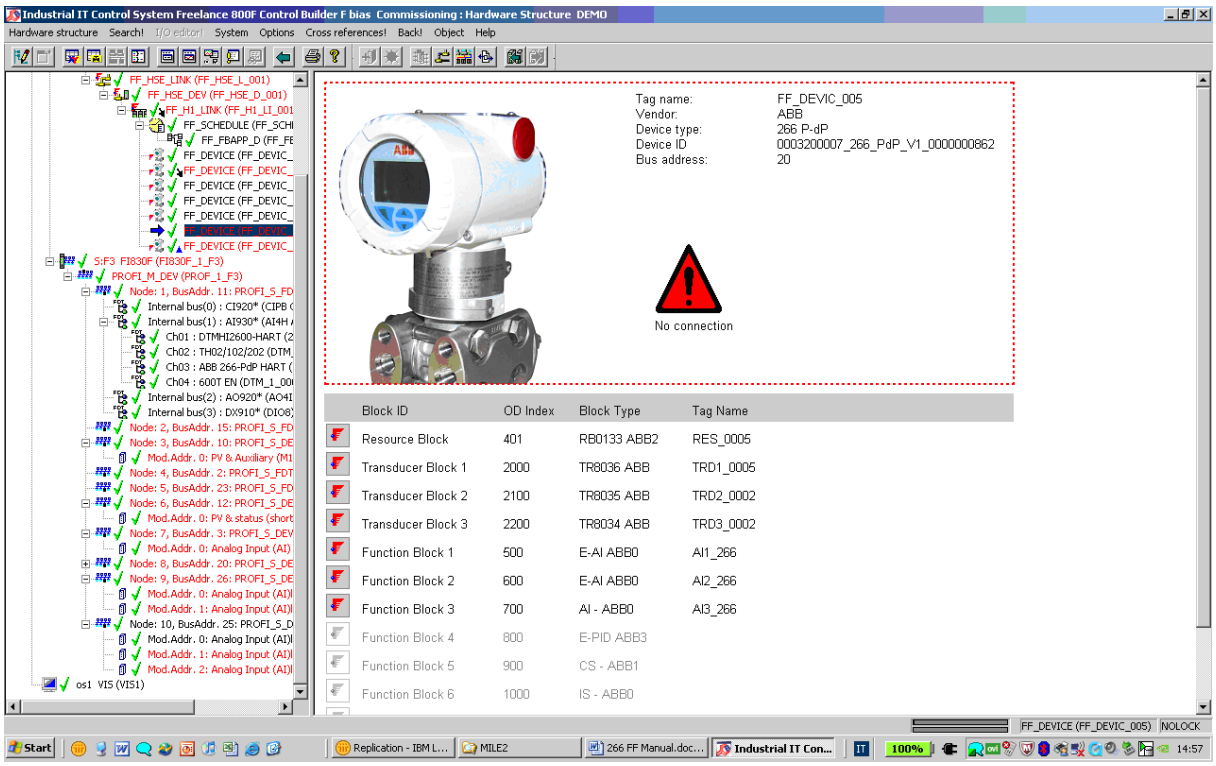
Select the FF Function Block Application section. In the lower part of the screen the list of the selected devices and their function blocks will appear. The yellow blocks mean that they are not in use and thus available. Drag and drop these blocks to move them in the upper box, rename and link them with other blocks in order to achieve the desired control strategy.



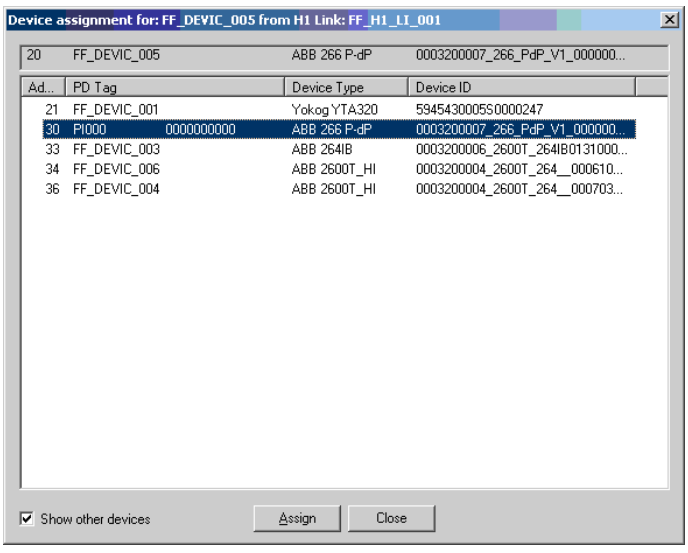
> **Attention**
>
> The second part of the operations is executed in ON-Line. ON line means that the real device has to be connected on the FF H1 network to the host.

## Assignment of the FF devices

Verify that the 266 PdP appears in the “live List” of the Linking Device.
Then with the right mouse click select “Pre-commissioning” and then with the left mouse click select “Assign device”.

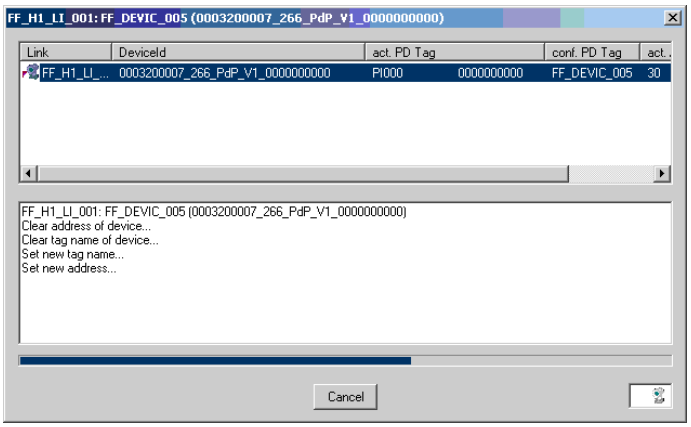



The current configuration of the device appears in the top of the “Device Assignment for: xxxxx” where, in the grey field is shown the configured/desired settings of the device to be commissioned in term of Address, TAG, Device Type and Device_ID, while in the white field are listed all the devices in the live list with their real settings.
Select the new device to be commissioned/assigned with the mouse and press “Assign”

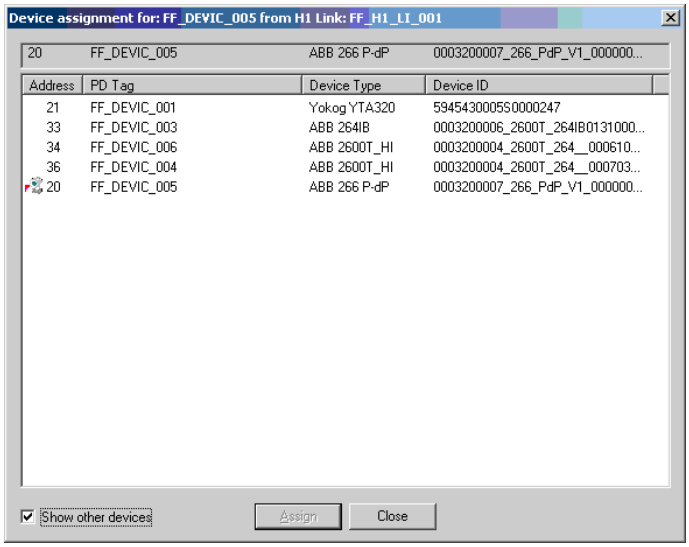


# 266 Models - FOUNDATION Fieldbus

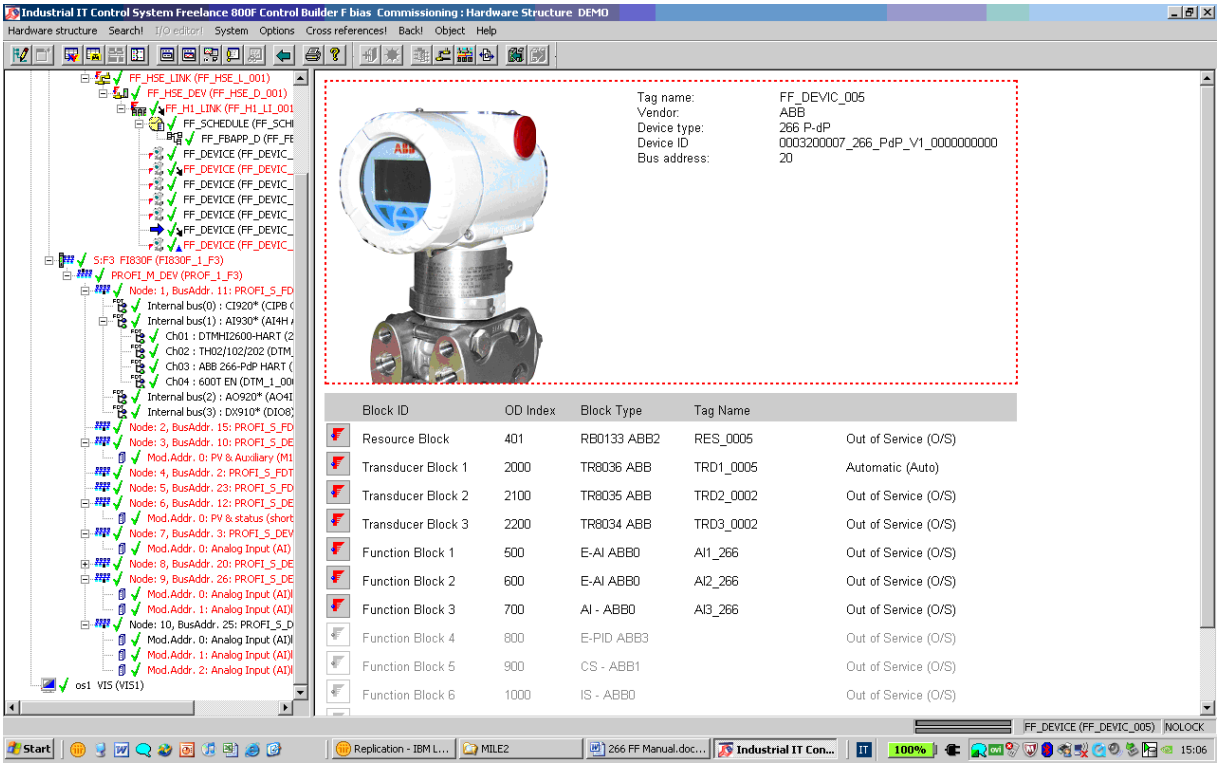
The Assignment get start and step by step it changes the device Address, and TAG as decided in the configuration….(What written in grey field of the“Device Assignment for: xxxxx” window above).




At the end the ABB 266 PdP is displayed with its new setting. In this example the address has been changed from 30 to 20 and the TAG from PI000 to FF_DEVICE_005.

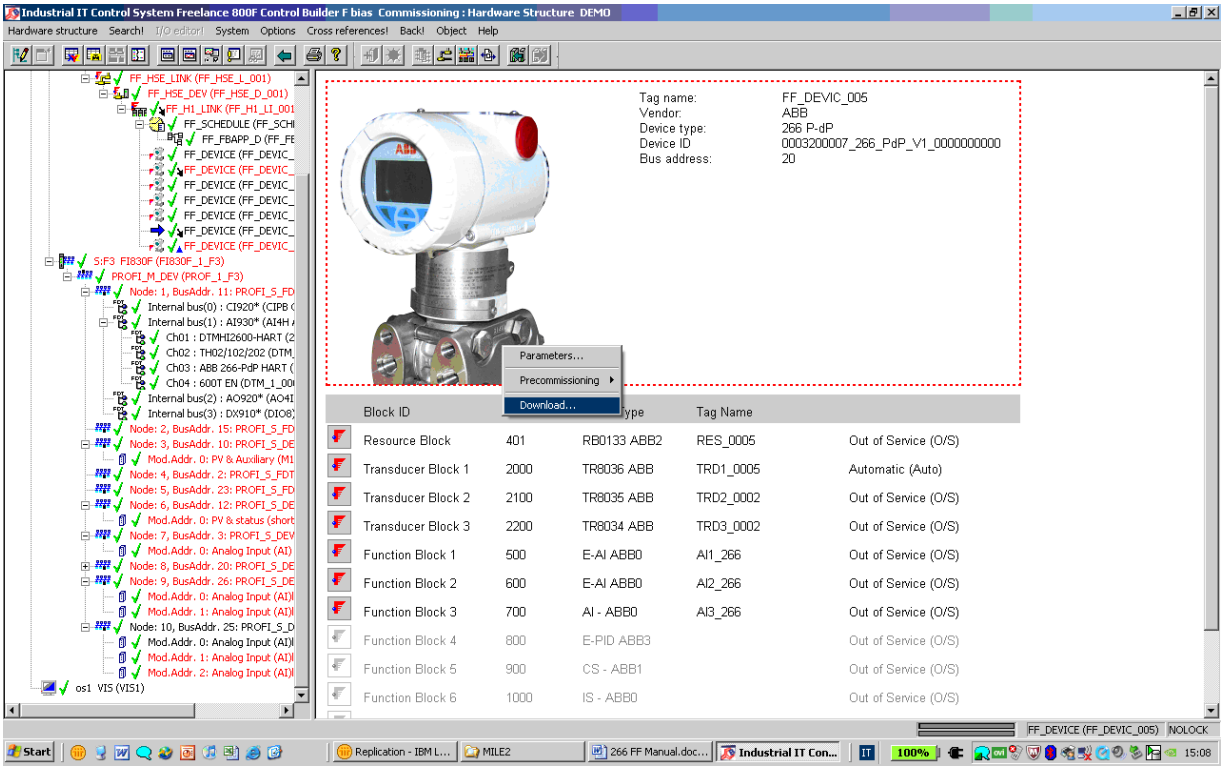



The device appears now alive in the Host, with all the blocks in Out of Service

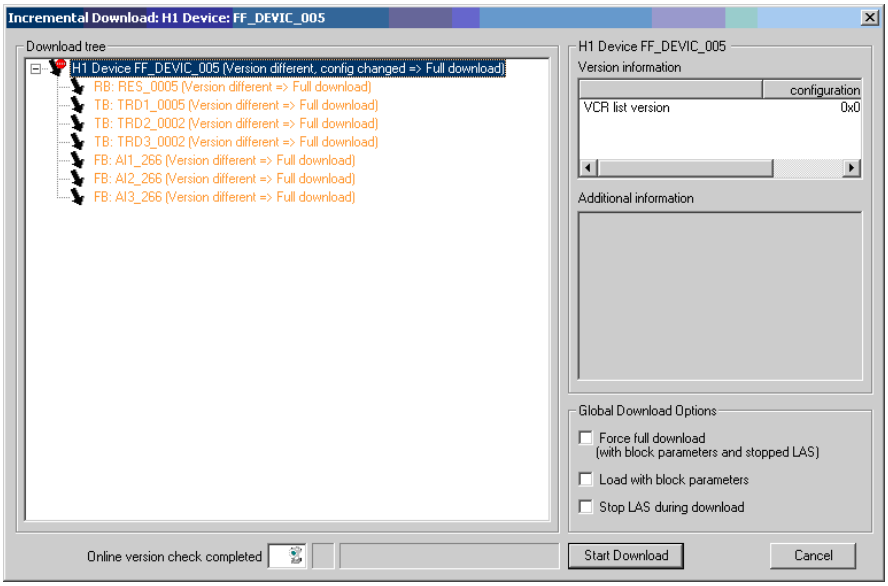


## Downloading of the FBAP into the H1 network and devices

With the right mouse click select “Download”

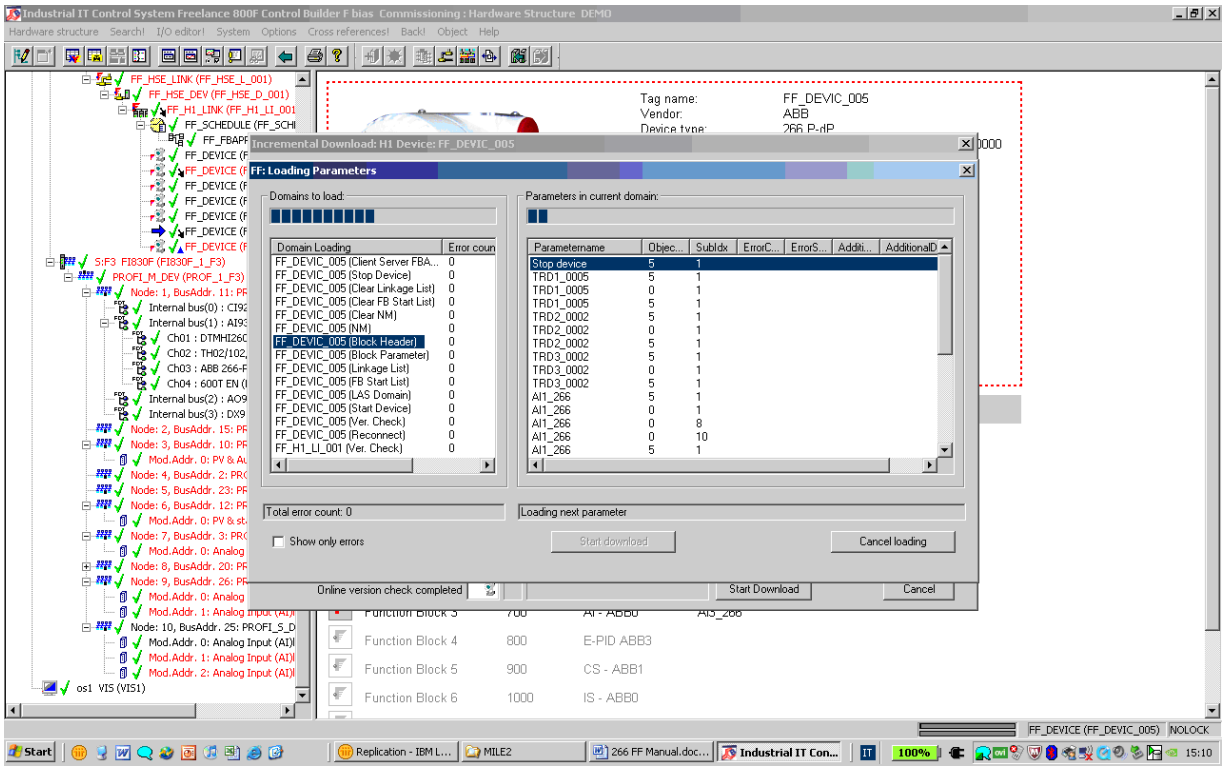



The “Incremental Download” window is open and press “Start Download”

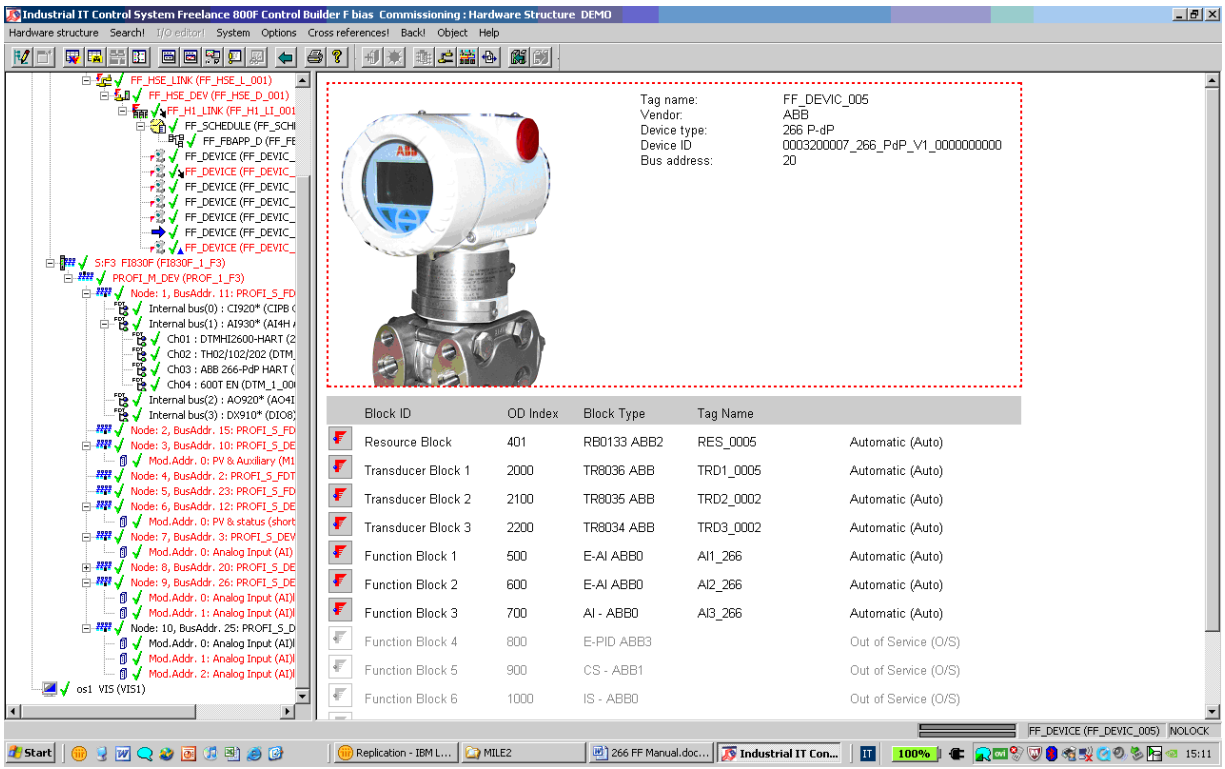


# 266 Models - FOUNDATION Fieldbus

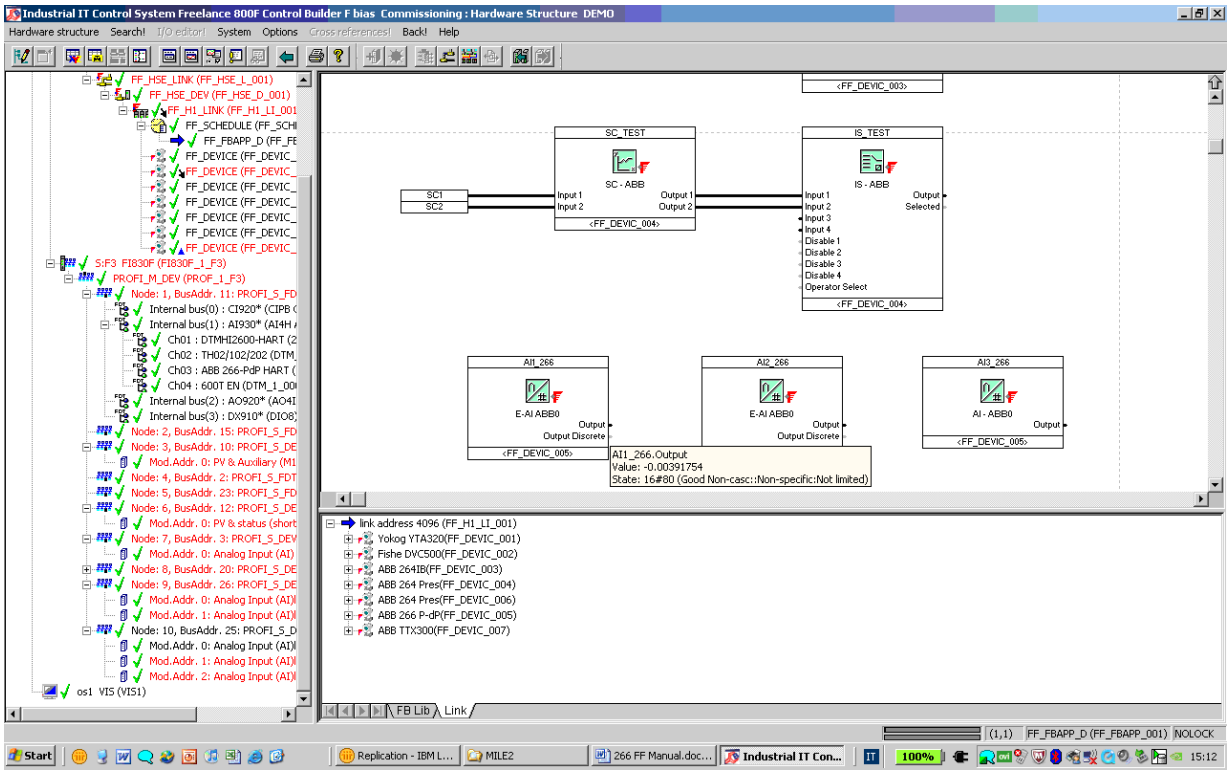
The Parameter downloading gets start and at the end.....




.....the transmitter's blocks used in the FBAP are moved in the AUTO or their Normal Mode.

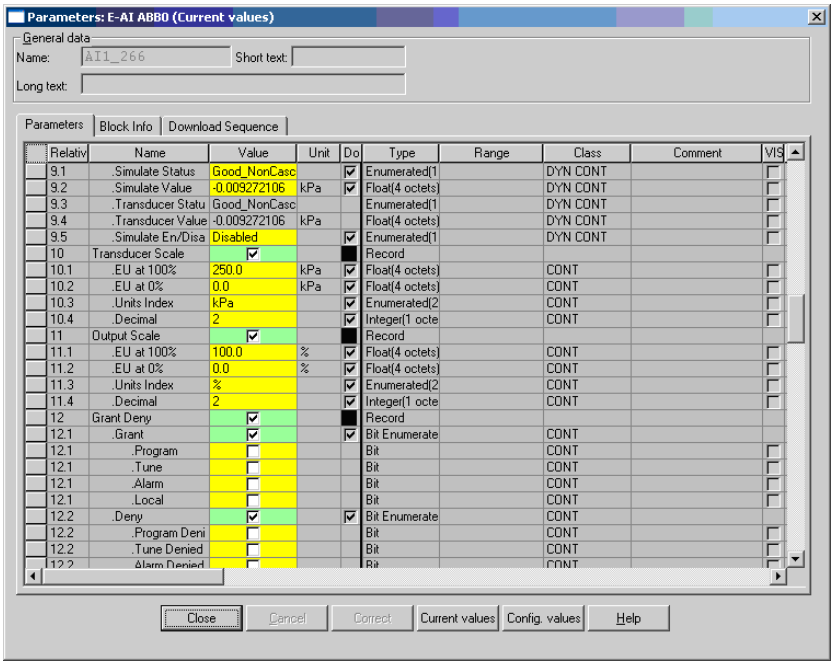


……and the Function Blocks start to work normally. In the example below the 266 PdP Analog Input block produces the measured pressure value in output

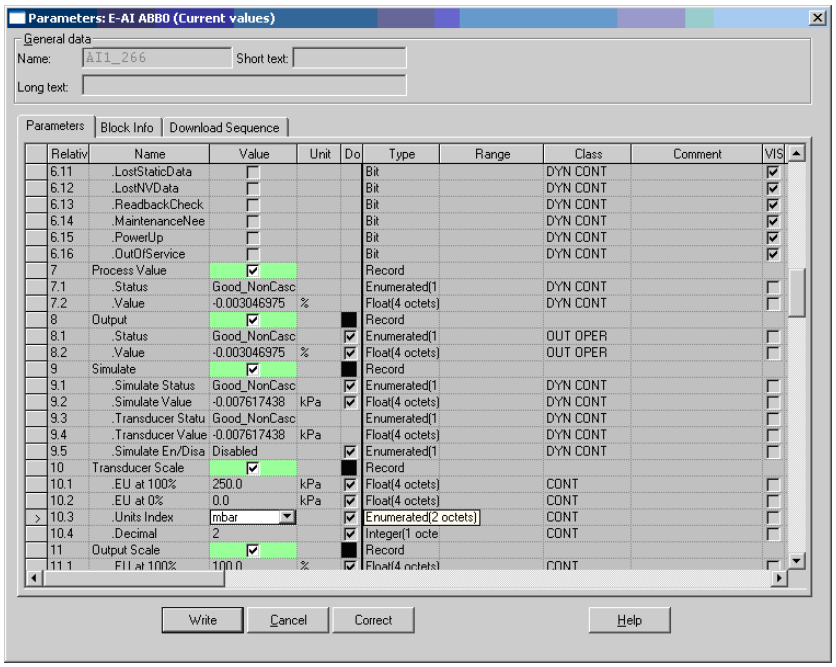



## Device and/or Blocks configuration

Whenever the 266 PdP is in this condition, it is then possible open any of the used blocks for read/write operations.
A double right mouse click, when the cursor is over the desired block, open it and the contained variables are read and shown.



Then variables in white or yellow fields can be changed and written in the device pressing the buttons “Write” or “Correct”. The “Write” button change only the actual values. The “Correct” button change both the actual values than the configuration values




## Appendix C – Device Configuration/Setting through FF communication

When the 266 PdP transmitter has to be used in a FF project and/or connected to any type of Host, the first operation is to import in the Host the DD and CF files of the device. The DD and CF files can be downloaded from the ABB website www.abb.com/intrumentation or from the FF organization website www.fieldbus.org under Registered Products

When the DD and CF files has been imported in the Host then:

– All the device blocks are visible and it is possible access at their parameters for operations like configuration/parameterization, maintenance, monitoring by reading or writing the parameters mapped in the transmitter’s blocks and addressed via index

– The device can be instantiated in a network design and its Function Blocks can be instantiated into a Function Block Application (FBAP) for the plant control strategy.

### Commissioning

Once the transmitter has been installed, it is put into operation by switching on the operating voltage.

Check the following before switching on the operating voltage:

– Process and electrical connections

– The impulse line/s and the measuring chamber of the measuring equipment must be completely filled with the measuring medium. The transmitter can then be put into operation. To do this, the shut-off valves must be actuated in the following sequence (in the default setting, all valves are closed):

### (Differential models) 266Dx or 266Mx

– Open the shut-off valves on the pressure tap connection (if present).

– Open the pressure equalization valve of the manifold.

– Open the positive shut-off valve (on the manifold).

– Open the negative shut-off valve (on the manifold).

– Close the pressure equalization valve.

To put the transmitter out of operation, carry out the steps in reverse order.

### (Gauge & Absolute models) 266Gx, 266Ax, 266Hx, 266Nx, 266Px, 266Vx, 266Rx

– Open the shut-off valve on the pressure tap connection (if present).
– Open the positive shut-off valve.
To put the transmitter out of operation, carry out the steps in reverse order.

> **Important**
> In case of the 266 transmitter for absolute pressure (266Vx, 266Rx, 266Ax and 266Nx) with a measuring range less than or equal 650 mbar abs., please be aware that the measuring equipment will have been overloaded by the atmospheric pressure due to the long periods of transport and storage involved. For this reason, you will need to allow a starting time of approx. 30 minutes for 266Vx, 266Rx and 266Nx models and 3 hours for 266Ax models after commissioning, until the sensor has stabilized to such an extent that the specified accuracy can be maintained.

### Correction of the mounting position

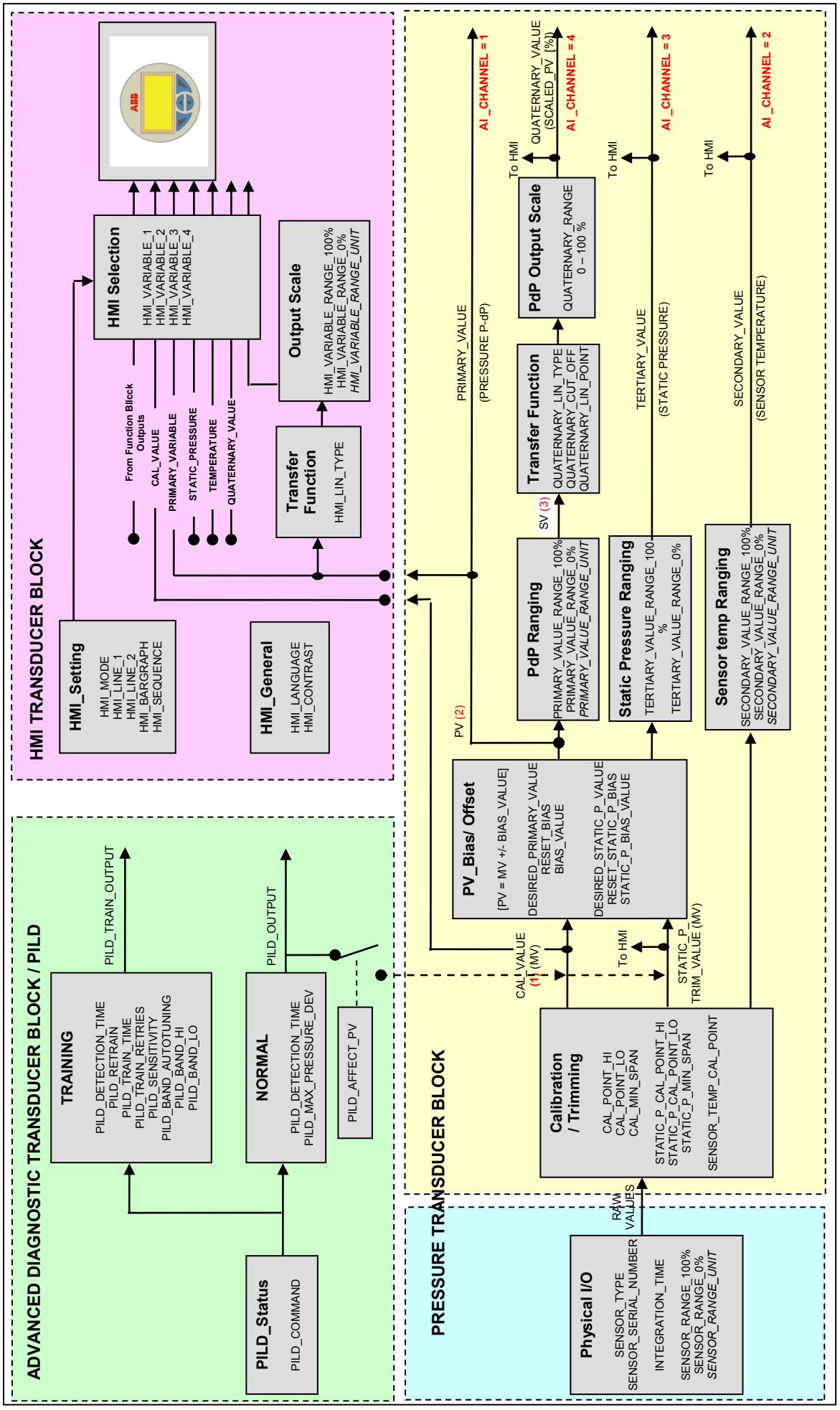
During installation of the transmitter, zero shifts caused by mounting (e.g., a slightly oblique mounting position due to a remote seal, etc.) may occur; these must be corrected.

> **Important**
> The transmitter must have reached its operating temperature (approx. 5 min. after startup, if the transmitter has already reached the ambient temperature).

This correction can be executed only if the Calibration Lower Range value is 0.0 and must be made with process (dp or p) = 0.
The correction consists in the Zero elevation/suppression operation and can be done in two ways:
– Locally by acting on the Z push button when the electronic switch SW 3 is set to 0
– From remote station via FF communication writing 0.0 in the **PTRB_DESIRED_PRIMARY_VALUE**
In case the Calibration Lower Range value is not 0.0 then the correction cannot be made with the local Z push button but it can be done in the following way:
– From remote station via FF communication writing the correct measure value in the **PTRB_DESIRED_PRIMARY_VALUE**

> **Important**
> After the above operations the Calibration Range Values are not changed. The desired process output value is produced through an internal calculation by applying an offset at the measured value

Transducer Blocks Diagram



## Initialization

The 266 PdP implements up to three Analog Input Blocks, 2 of Enhanced category and 1 Standard. Each AI produce in output one variable (**AIx_OUT**) suitable to be linked to other downstream function blocks. The three AIs receive in input one of the variables produced by the Pressure Transducer Block depending by their own **AIx_CHANNEL** setting.
The default AIx_CHANNEL setting is:
– The Analog Input 1 receives in input the **PRTB_PRIMARY_VALUE** (Process Pressure) through the **AI1_CHANNEL = 1**.
– The Analog Input 2 receives in input the **PRTB_TERTIARY_VALUE** (Static Pressure) through the **AI2_CHANNEL = 3.**
– The Analog Input 3 receives in input the **PRTB_SECONDARY_VALUE** (Sensor Temperature) through the **AI3_CHANNEL = 2**.
Hovewer all the 3 **AIx_CHANNEL** can be switched to receive in input up to different 4 PRTB variables:

| AI CHANNEL | Variables | PRTB_Variable |
|---|---|---|
| 0 | Uninitialized | None |
| 1 | Pressure Process Value | **PRIMARY_VALUE** |
| 2 | Sensor Temperature | **SECONDARY_VALUE** |
| 3 | Static Pressure | **TERTIARY_VALUE** |
| 4 | Scaled Process Value | **QUATERNARY_VALUE** |

## Factory settings

Transmitters are calibrated at the factory to the customer's specified measuring range. The calibrated range and tag number are provided on the name plate. If this data has not been specified, the transmitter will be delivered with the following configuration:

| Process Info | Parameter | Factory setting |
|---|---|---|
| Node Address | | 248 |
| TAG | **PD_TAG** | "PI000" |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_0%** | 0.0 |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE_100%** | **PTRB_SENSOR_RANGE_100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE_UNIT** | Kpa |
| Transfer function | **PTRB_QUATERNARY_VALUE_LIN_TYPE** | Linear |
| Display Mode | **HMI_MODE** | One Line |
| Display Variable | **HMI_LINE1** | **HMI_VARIABLE_1 = PRTB_PRIMARY_VALUE** |
| Analog Input 1 setting | | |
| Channel | **AI1_CHANNEL** | 1 = PV |
| Damping | **AI1_PV_FTIME** | 0 second |
| Calibation Lower Range Value 0% | **AI1_XD_SCALE 0%** | 0.0 |
| Calibation Upper Range Value 100% | **AI1_XD_SCALE 100%** | **PTRB_SENSOR_RANGE_100%** |
| Calibration Unit | **AI1_XD_SCALE_UNIT** | Kpa |
| Output scale 0% | **AI1_OUT_SCALE 0%** | 0.0 |
| Output scale 100% | **AI1_OUT_SCALE 100%** | **PTRB_SENSOR_RANGE_100%** |
| Output Scale Unit | **AI1_OUT_SCALE_UNIT** | Kpa |
| Linearization | **AI1_L_TYPE** | Direct |
| Critical Limit Low | **AI1_LO_LO_LIM** | **AI1_OUT_SCALE 0% - 10% of the SPAN** |
| Advisory Limit Low | **AI1_LO_LIM** | |
| Advisory Limit High | **AI1_HI_LIM** | **AI1_OUT_SCALE 100% + 10% of the SPAN** |
| Critical Limit High | **AI1_HI_HI_LIM** | |
| Alarm Hysteresis | **AI1_ALARM_HYS** | 0.5% of the SPAN |
| Analog Input 2 setting (applicable only for differential pressure sensor types) | | |
| Channel | **AI2_CHANNEL** | 3 = Static Pressure |
| Damping | **AI2_PV_FTIME** | 0 second |
| Calibation Lower Range Value 0% | **AI2_XD_SCALE 0%** | 0.0 |
| Calibation Upper Range Value 100% | **AI2_XD_SCALE 100%** | **PTRB_TERTIARY_VALUE_RANGE_100%** |
| Calibration Unit | **AI2_XD_SCALE_UNIT** | MPa |
| Output scale 0% | **AI2_OUT_SCALE 0%** | 0.0 |

| | | |
|---|---|---|
| Output scale 100% | **AI2_OUT_SCALE 100%** | PTRB_TERTIARY_VALUE_RANGE_100% |
| Output Scale Unit | **AI2_OUT_SCALE_UNIT** | MPa |
| Linearization | **AI2_L_TYPE** | Direct |
| Critical Limit Low | **AI2_LO_LO_LIM** | **AI2_OUT_SCALE 0% - 10% of the SPAN** |
| Advisory Limit Low | **AI2_LO_LIM** | |
| Advisory Limit High | **AI2_HI_LIM** | AI2_OUT_SCALE 100% + 10% of the SPAN |
| Critical Limit High | **AI2_HI_HI_LIM** | |
| Alarm Hysteresis | **AI2_ALARM_HYS** | 0.5% of the SPAN |
| | Analog Input 3 setting | |
| Channel | **AI2_CHANNEL** | 3 = Static Pressure |
| Channel | **AI3_CHANNEL** | 2 = Sensor temperature |
| Damping | **AI3_PV_FTIME** | 0 second |
| Calibation Lower Range Value 0% | **AI3_XD_SCALE 0%** | **PTRB_ SECONDARY_VALUE_RANGE_0%** |
| Calibation Upper Range Value 100% | **AI3_XD_SCALE 100%** | **PTRB_ SECONDARY_VALUE_RANGE_100%** |
| Calibration Unit | **AI3_XD_SCALE_UNIT** | °C |
| Output scale 0% | **AI3_OUT_SCALE 0%** | **PTRB_ SECONDARY_VALUE_RANGE_0%** |
| Output scale 100% | **AI3_OUT_SCALE 100%** | **PTRB_ SECONDARY_VALUE_RANGE_100%** |
| Output Scale Unit | **AI3_OUT_SCALE_UNIT** | °C |
| Linearization | **AI3_L_TYPE** | Direct |
| Critical Limit Low | **AI3_LO_LO_LIM** | **AI3_OUT_SCALE 0% - 10% of the SPAN** |
| Advisory Limit Low | **AI3_LO_LIM** | |
| Advisory Limit High | **AI3_HI_LIM** | **AI3_OUT_SCALE 100% + 10% of the SPAN** |
| Critical Limit High | **AI3_HI_HI_LIM** | |
| Alarm Hysteresis | **AI3_ALARM_HYS** | 0.5% of the SPAN |

**Important**

All the above configurable parameters can be afterward modified via DD based software tools

## User setting

Generally the 266 PdP pressure transmitters are delivered pre-configured as per purchase order request in order to measure Pressure, Level, Flow or Volume.

For the device configuration it is necessary to know at least the following process info as minimum:
– TAG
– Calibration Range/Scale and its engineering unit as range of pressure to be measured in input
– Linearization Type defining the type of linearization to be applied at the pressure measured in input in order to convert it to the output measure
– Output Range/Scale and its engineering unit

| Pressure and level measurement setting | |
|---|---|
| **Process Info** | **Device parameter to be configured** |
| TAG | **PD_TAG** |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_ 0% AI1_XD_SCALE 0%** |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE _100% AI1_XD_SCALE 100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE _UNIT AI1_XD_SCALE_UNIT** |
| Linearization Type | **AI1_L_TYPE = Indirect** |
| Output scale 0% | **AI1_OUT_SCALE 0%** |
| Output scale 100% | **AI1_OUT_SCALE 100%** |
| Output Scale Unit | **AI1_OUT_SCALE Unit Code** |
| | **AI1_CHANNEL = 1** |

| Normal flow measurement setting | |
|---|---|
| **Process Info** | **Device parameter to be configured** |
| TAG | **PD_TAG** |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_ 0% AI1_XD_SCALE 0%** |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE _100% AI1_XD_SCALE 100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE _UNIT AI1_XD_SCALE_UNIT** |
| Linearization Type | **AI1_L_TYPE = Indirect Square Root** |
| Output scale 0% | **AI1_OUT_SCALE 0%** |
| Output scale 100% | **AI1_OUT_SCALE 100%** |
| Output Scale Unit | **AI1_OUT_SCALE Unit Code** |
| | **AI1_CHANNEL = 1** |

| Special flow measurement setting | |
|---|---|
| TAG | **PD_TAG** |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_ 0%** |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE _100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE _UNIT** |
| Linearization Type<br>**QUATERNARY_VALUE_ LIN_TYPE** | Square Root<br>SQRT 3° pow<br>SQRT 5° pow<br>Bidirectional Flow |
| Output scale 0% | **AI1_OUT_SCALE 0%** |
| Output scale 100% | **AI1_OUT_SCALE 100%** |
| Output Scale Unit | **AI1_OUT_SCALE Unit Code** |
| | **AI1_CHANNEL = 4** |
| | **AI1_L_TYPE = Indirect** |
| | **AI1_XD_XCALE = 0.0 ... 100.0 / %** |

| Linear volume measurement setting | |
|---|---|
| **Process Info** | **Device parameter to be configured** |
| TAG | **PD_TAG** |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_ 0% AI1_XD_SCALE 0%** |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE _100% AI1_XD_SCALE 100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE _UNIT AI1_XD_SCALE_UNIT** |
| Linearization Type | **AI1_L_TYPE = Indirect** |
| Output scale 0% | **AI1_OUT_SCALE 0%** |
| Output scale 100% | **AI1_OUT_SCALE 100%** |
| Output Scale Unit | **AI1_OUT_SCALE Unit Code** |
| | **AI1_CHANNEL = 1** |

| Special volume measurement setting | |
|---|---|
| **Process Info** | **Device parameter to be configured** |
| TAG | **PD_TAG** |
| Calibration Lower Range Value 0% | **PRTB_PRIMARY_VALUE_RANGE_ 0%** |
| Calibration Upper Range Value 100% | **PRTB_PRIMARY_VALUE_RANGE _100%** |
| Calibration Unit | **PRTB_PRIMARY_VALUE_RANGE _UNIT** |
| Linearization Type | Linear |
| **QUATERNARY_VALUE_ LIN_TYPE** | cylindrical lying container |
| | spherical container |
| Output scale 0% | **AI1_OUT_SCALE 0%** |
| Output scale 100% | **AI1_OUT_SCALE 100%** |
| Output Scale Unit | **AI1_OUT_SCALE Unit Code** |
| | **AI1_CHANNEL = 4** |
| | **AI1_L_TYPE = Indirect** |
| | **AI1_XD_XCALE = 0.0 ... 100.0 / %** |

| Further common setting | |
|---|---|
| **Process Info** | **Device parameter to be configured** |
| Node Address | |
| Damping | **AI1_PV_FTIME** |
| Critical Limit Low | **AI1_LO_LO_LIM** |
| Advisory Limit Low | **AI1_LO_LIM** |
| Advisory Limit High | **AI1_HI_LIM** |
| Critical Limit High | **AI1_HI_HI_LIM** |
| Alarm Hysteresis | **AI1_ALARM_HYS** |

No field calibration is normally requested, the transmitter has been trimmed to the calibration points (URV and LRV) to provide the best performances in the real operating range.

**Important**

In case the calibrated range has to be changed, please refer to the section “Sensor Calibration” in this manual

# Appendix D – 266 PdP Fieldbus FOUNDATION electronics replacement

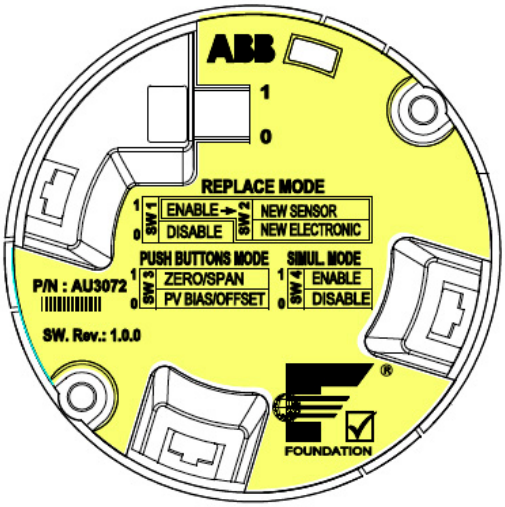
In order to perform the replacement of a Fieldbus FOUNDATION electronic module, please follow the steps listed here below:

– Remove the cover of the electronics/display side.

– Remove the display (if installed) and be careful to the plastic clips of the electronic module.

– Remove the 2 fixing screws from the electronic module.

– Extract the electronics from the housing, and disconnect the flat cable that links the sensor primary electronics to the communication board.




– Take the new electronics and put the switch 1 in UP position (1) since it enables the Replacement operation. It must be used in combination with the SW 2.

– Put dip switch 2 in OFF position (0) selects the Electronics Replacement. The entire transmitter's configuration data are kept valid in the sensor memory and copied into the memory of the new electronics once it is connected.

– Connect the sensor flat cable to the new electronics and insert it into the housing (be careful with the two in-housing jack connectors).

– Power on the transmitter and keep it powered-on for few seconds (at least 30).

– Power-Off the transmitter again, and put the switch 1 and 2 in OFF (0) position.

– Fix the electronics with the two screws.

– Insert the display, (be carefully with the 6 pins connector).
Maybe removing again the electronics from the housing make easier the connection of the display.

– Mount the display cover again.

The operation is completed and the device will run with default configuration:
PD_TAG = PI000
ADDRESS = 248

**ABB Limited**
**Measurement & Analytics**
Howard Road, St. Neots
Cambridgeshire, PE19 8EU
UK
Tel: +44 (0)870 600 6122
Fax: +44 (0)1480 213 339
Email: enquiries.mp.uk@gb.abb.com

**ABB S.p.A.**
**Measurement & Analytics**
Via Luigi Vaccani 4
22016 Tremezzina (CO)
Italy
Tel: +39 0344 58111

**ABB Inc.**
**Measurement & Analytics**
125 E. County Line Road
Warminster, PA 18974
USA
Tel: +1 215 674 6000
Fax: +1 215 674 7183

**abb.com/pressure**

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.
We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB.
© ABB 2020
All rights reserved

3KXP000008R4201

OI/266/FF/ADD-EN Rev. B 05.2020