

smar  
First in Fieldbus

- FIELDBUS



# LIBRARY B FUNCTION BLOCKS INSTRUCTION MANUAL





Specifications and information are subject to change without notice.  
Up-to-date address information is available on our website.

web: [www.smar.com/contactus.asp](http://www.smar.com/contactus.asp)

# INTRODUCTION

Fieldbus is not a replacement for 4-20 mA or Intelligent/Smart Transmitter Protocols, it provides much more. Fieldbus is a complete Control System Architecture enabling distribution of the control function to equipment in the field; therefore it is a replacement for the DCS Architecture of the 1970s.

To achieve the desired control the devices must be configured. That includes calibration but also building of a control strategy. The latter is covered in this manual.

One of the major advantages of Fieldbus is interoperability. Some blocks described in this manual are used not only by Smar devices, but other Foundation Fieldbus devices too. No particular configuration tool is addressed in this manual, because the devices are independent of configuration tool due to the DD technology.

Get the best results of the Fieldbus System by carefully reading these instructions.

This manual presents the necessary background knowledge to understand the programming language of Function Blocks diagram focusing on Foundation Fieldbus technology.

Smar devices, that are members of System302, may not be covered by this manual, because they have specific manuals.

The **Library B** Function Blocks, which are described in this manual, refer to the most recent FOUNDATION Fieldbus function blocks specification. As one may notice, the block library is composed of more Enhanced blocks to guarantee greater functionality to the function block applications. Moreover, this manual version presents the Flexible Mathematical Function Block allowing the user to create customized mathematical expressions.

## **NOTE:**

**This manual is related to the DF62, DF63, DF73, DF75, DF79, DF81, DF89, DF95, DF97, DF99, and DF100 controllers.**



# TABLE OF CONTENTS

|   |            |
|---|------------|
| INTRODUCTION .....  | III        |
| DEFINITIONS .....   | IX         |
| ABBREVIATIONS .....   | XIII       |
| <b>CHAPTER 1 - INTRODUCTION TO FUNCTION BLOCK APPLICATION .....</b> | <b>1.1</b> |
| OVERVIEW .....  | 1.1        |
| FUNCTION BLOCK .....  | 1.1        |
| TRANSDUCER BLOCK .....  | 1.1        |
| RESOURCE BLOCK .....  | 1.1        |
| FUNCTION BLOCK DEFINITIONS .....                                    | 1.1        |
| FUNCTION BLOCK LINKAGES .....                                       | 1.1        |
| LINK WITH FUNCTION BLOCK EXECUTING IN DFI302 .....                  | 1.2        |
| INFORMATION ACCESS .....  | 1.2        |
| FUNCTION BLOCK APPLICATION STRUCTURE .....                          | 1.2        |
| BLOCK OBJECT .....  | 1.2        |
| BLOCK PARAMETERS .....  | 1.2        |
| PARAMETER IDENTIFIERS .....   | 1.3        |
| PARAMETER STORAGE .....   | 1.3        |
| PARAMETER USAGE .....   | 1.3        |
| PARAMETER RELATIONSHIPS .....                                       | 1.4        |
| PARAMETER STATUS .....  | 1.4        |
| COMPOSITION OF STATUS .....   | 1.5        |
| PROCESS VARIABLE CALCULATION .....                                  | 1.7        |
| SETPOINT CALCULATION .....  | 1.7        |
| OUTPUT CALCULATION .....  | 1.8        |
| CASCADE CONTROL .....   | 1.8        |
| MODE PARAMETER .....  | 1.10       |
| SCALING PARAMETERS .....  | 1.17       |
| MODBUS SCALE CONVERSION .....                                       | 1.18       |
| FAULT STATE HANDLING .....  | 1.20       |
| FAULT STATE ACTIVE .....  | 1.21       |
| ALARMS AND EVENTS – ALERT PROCESSING .....                          | 1.23       |
| SIMULATION .....  | 1.29       |
| I/O MAPPING FORMS .....   | 1.30       |
| NUMERIC MAPPING .....   | 1.30       |
| ROM MAPPING .....   | 1.31       |
| BLOCK INSTANTIATION .....   | 1.33       |
| ORDER OF PARAMETERS DURING DOWNLOAD .....                           | 1.33       |
| DATA TYPE AND DATA STRUCTURE DEFINITION .....                       | 1.34       |
| BLOCK STRUCTURE – DS-64 .....                                       | 1.34       |
| VALUE & STATUS - FLOATING POINT STRUCTURE – DS-65 .....             | 1.35       |
| VALUE & STATUS - DISCRETE STRUCTURE – DS-66 .....                   | 1.35       |
| SCALING STRUCTURE – DS-68 .....                                     | 1.35       |
| MODE STRUCTURE – DS-69 .....  | 1.35       |
| ACCESS PERMISSIONS – DS-70 .....                                    | 1.35       |
| ALARM FLOAT STRUCTURE – DS-71 .....                                 | 1.36       |
| ALARM DISCRETE STRUCTURE – DS-72 .....                              | 1.36       |
| EVENT UPDATE STRUCTURE – DS-73 .....                                | 1.36       |
| ALARM SUMMARY STRUCTURE – DS-74 .....                               | 1.36       |
| SIMULATE - FLOATING POINT STRUCTURE – DS-82 .....                   | 1.36       |
| SIMULATE - DISCRETE STRUCTURE – DS-83 .....                         | 1.36       |
| TEST STRUCTURE – DS-85 .....  | 1.37       |
| BITSTRING32 ALARM - DS-87 .....                                     | 1.37       |
| SIMULATION – FIELD DIAGNOSTICS - DS-89 .....                        | 1.37       |
| BITSTRING STRUCTURE – DS-158 .....                                  | 1.37       |
| DISCRETE STRUCTURE – DS-159 .....                                   | 1.38       |
| DISCRETE STRUCTURE – DS-160 .....                                   | 1.38       |
| FLOAT STRUCTURE – DS-174 .....                                      | 1.38       |
| HART EQUIPMENT INFORMATION - DS-175 .....                           | 1.39       |
| EXPECTED ELEMENT - DS-176 .....                                     | 1.39       |

|   |            |
|---|------------|
| ACTUAL ELEMENT - DS-177 .....                               | 1.39       |
| ELEMENT STATUS - DS-178 .....                               | 1.39       |
| THRESHOLD ANALOG I/O - DS-179 .....                         | 1.40       |
| INFORMATION OF HART BURST CONTROL - DS-183 .....            | 1.40       |
| INDICATION ALARM OF FOREIGN PROTOCOL - DS-184 .....         | 1.40       |
| MANUFACTURER SPECIFIC DATA STRUCTURE .....                  | 1.41       |
| SCALING CONVERSION STRUCTURE - DS-256 .....                 | 1.41       |
| SCALING CONVERSION STRUCTURE WITH STATUS - DS-257 .....     | 1.41       |
| SCALING LOCATOR STRUCTURE - DS-258 .....                    | 1.41       |
| SCALING LOCATOR STRUCTURE WITH STATUS- DS-259 .....         | 1.42       |
| MODBUS VARIABLE LOCATOR STRUCTURE - DS-260 .....            | 1.42       |
| MODBUS VARIABLE LOCATOR STRUCTURE WITH STATUS- DS-261 ..... | 1.42       |
| FF PARAMETER ID STRUCTURE - DS-262 .....                    | 1.42       |
| SLAVE ADDRESS STRUCTURE - DS-263 .....                      | 1.43       |
| <b>CHAPTER 2 - BLOCK LIBRARY .....</b>                      | <b>2.1</b> |
| DESCRIPTION OF BLOCK TYPES .....                            | 2.2        |
| BLOCK TYPE AVAILABILITY AND INITIAL BLOCK SET .....         | 2.5        |
| RESOURCE .....  | 2.7        |
| RS – RESOURCE BLOCK .....                                   | 2.7        |
| RS2 - RESOURCE BLOCK .....                                  | 2.11       |
| TRANSDUCER BLOCKS .....                                     | 2.14       |
| DIAG – DIAGNOSTICS TRANSDUCER BLOCK .....                   | 2.14       |
| TEMP – DF45 TEMPERATURE TRANSDUCER .....                    | 2.16       |
| HC – HARDWARE CONFIGURATION TRANSDUCER .....                | 2.19       |
| TRDRED - REDUNDANCY TRANSDUCER .....                        | 2.22       |
| ABC – ASSOCIATION BLOCK FOR COMPONENTS .....                | 2.24       |
| TB8BI – 8 BINARY INPUTS TRANSDUCER BLOCK .....              | 2.44       |
| TB8BO – 8 BINARY OUTPUTS TRANSDUCER BLOCK .....             | 2.46       |
| TB8AI – 8 ANALOG INPUTS TRANSDUCER BLOCK .....              | 2.48       |
| TB4AO - 4 ANALOG OUTPUTS TRANSDUCER BLOCK .....             | 2.54       |
| TBHG - TRANSDUCER BLOCK FOR HART GATEWAY .....              | 2.58       |
| TBWH - TRANSDUCER BLOCK FOR <i>WIRELESSHART</i> .....       | 2.61       |
| INPUT FUNCTION BLOCKS .....                                 | 2.69       |
| AI - ANALOG INPUT .....                                     | 2.69       |
| EAI – ENHANCED ANALOG INPUT .....                           | 2.72       |
| DI - DISCRETE INPUT .....                                   | 2.74       |
| MAI - MULTIPLE ANALOG INPUT .....                           | 2.76       |
| MDI - MULTIPLE DISCRETE INPUT .....                         | 2.78       |
| PUL – PULSE INPUT .....                                     | 2.80       |
| ROMAI – ROM ANALOG INPUT .....                              | 2.84       |
| MAI16 –16 MULTIPLE ANALOG INPUT .....                       | 2.86       |
| MBI64 – MULTIPLE BINARY INPUTS 64 .....                     | 2.88       |
| CONTROL AND CALCULATION FUNCTION BLOCKS .....               | 2.91       |
| PID - PID CONTROL .....                                     | 2.91       |
| EPID – ENHANCED PID CONTROL .....                           | 2.99       |
| APID – ADVANCED PID CONTROL .....                           | 2.102      |
| ARTH - ARITHMETIC .....                                     | 2.108      |
| SPLT-SPLITTER .....   | 2.115      |
| CHAR - SIGNAL CHARACTERIZER .....                           | 2.119      |
| ECHAR – ENHANCED SIGNAL CHARACTERIZER .....                 | 2.122      |
| INTG - INTEGRATOR .....                                     | 2.124      |
| AALM - ANALOG ALARM .....                                   | 2.132      |
| EAALM – ENHANCED ANALOG ALARM .....                         | 2.138      |
| ISEL - INPUT SELECTOR .....                                 | 2.139      |
| SPG - SETPOINT RAMP GENERATOR .....                         | 2.142      |
| ESPG – ENHANCED SETPOINT RAMP GENERATOR .....               | 2.148      |
| TIME – TIMER AND LOGIC .....                                | 2.149      |
| LLAG - LEAD LAG .....                                       | 2.157      |
| OSDL - OUTPUT SIGNAL SELECTOR AND DYNAMIC LIMITER .....     | 2.161      |
| DENS - DENSITY .....  | 2.165      |
| CTRW – CONSTANT AND CONTAINED RW .....                      | 2.169      |
| FFET - FLIP-FLOP AND EDGE TRIGGER .....                     | 2.175      |
| AEQU – ADVANCED EQUATIONS .....                             | 2.177      |
| PRED – SMITH PREDICTOR .....                                | 2.179      |

|   |            |
|---|------------|
| TF – TRANSFER FUNCTION .....                | 2.181      |
| LCF – LIQUID CORRECTION FACTORS .....       | 2.183      |
| CCF CALCULATION.....                        | 2.183      |
| BSW CALCULATION– DUAL RANGE .....           | 2.184      |
| BSW CALCULATION – LAB ANALYSIS.....         | 2.186      |
| FMTH – FLEXIBLE MATHEMATICAL BLOCK .....    | 2.190      |
| ADT – ANALOG DATA TRANSFER .....            | 2.201      |
| DDT – DISCRETE DATA TRANSFER.....           | 2.204      |
| FLEXIBLE FUNCTION BLOCK 1131.....           | 2.207      |
| FFB_1131 – FLEXIBLE FUNCTION BLOCK .....    | 2.207      |
| FFB2_1131 – FLEXIBLE FUNCTION BLOCK 2 ..... | 2.209      |
| MODBUS FUNCTION BLOCKS.....                 | 2.211      |
| MBCF – MODBUS CONFIGURATION .....           | 2.211      |
| MBCS – MODBUS CONTROL SLAVE .....           | 2.213      |
| MBSS – MODBUS SUPERVISION SLAVE .....       | 2.220      |
| MBCM – MODBUS CONTROL MASTER .....          | 2.224      |
| MBSM – MODBUS SUPERVISION MASTER .....      | 2.229      |
| OUTPUT FUNCTION BLOCKS.....                 | 2.233      |
| AO - ANALOG OUTPUT .....                    | 2.233      |
| DO - DISCRETE OUTPUT .....                  | 2.237      |
| MAO - MULTIPLE ANALOG OUTPUT .....          | 2.240      |
| MDO - MULTIPLE DISCRETE OUTPUT .....        | 2.242      |
| STEP – STEP OUTPUT PID .....                | 2.244      |
| ROMAO – ROM ANALOG OUTPUT .....             | 2.250      |
| MAO16 – 16 MULTIPLE ANALOG OUTPUT .....     | 2.251      |
| MBO64 – 64 MULTIPLE BINARY OUTPUT .....     | 2.254      |
| BLOCK OPTIONS .....                         | 2.257      |
| RESOURCE BLOCK OPTIONS.....                 | 2.257      |
| IO_OPTS.....                                | 2.259      |
| CONTROL_OPTS .....                          | 2.260      |
| STATUS_OPTS .....                           | 2.261      |
| ALARM_SUM AND ACK_OPTION .....              | 2.262      |
| PID_OPTS .....                              | 2.263      |
| INTEG_OPTS.....                             | 2.263      |
| INVERT_OPTS .....                           | 2.264      |
| INPUT_OPTS.....                             | 2.264      |
| OSDL_OPTS.....                              | 2.265      |
| MO_STATUS_OPTS.....                         | 2.265      |
| MO_OPTS.....                                | 2.266      |
| CT_OPTS.....                                | 2.266      |
| HARDWARE CONFIGURATION BLOCK OPTIONS .....  | 2.267      |
| ELEM_MISMAT_BS_1[4].....                    | 2.269      |
| ELEM_MISMAT_BS_2[4].....                    | 2.269      |
| MO_FSTATE_OPTS_16[2] .....                  | 2.269      |
| MO_FSTATE_OPTS_64[8] .....                  | 2.269      |
| MO_POWERUP_OPTS_16[2] .....                 | 2.270      |
| MO_POWERUP_OPTS_64[8] .....                 | 2.270      |
| FSTATE_ST_16[2].....                        | 2.270      |
| FSTATE_ST_64[8].....                        | 2.270      |
| MAP_MM_16.....                              | 2.271      |
| MAP_MM_64.....                              | 2.271      |
| <b>CHAPTER 3 - EXAMPLES .....</b>           | <b>3.1</b> |
| SIMPLE CONTROL APPLICATION .....            | 3.1        |
| CASCADE CONTROL.....                        | 3.1        |
| CORRESPONDING CONFIGURATION .....           | 3.2        |
| PARAMETERIZATION.....                       | 3.3        |
| RATIO CONTROL .....                         | 3.3        |
| CORRESPONDING CONFIGURATION .....           | 3.4        |
| PARAMETERIZATION.....                       | 3.4        |
| FEEDFORWARD CONTROL.....                    | 3.5        |
| CORRESPONDING CONFIGURATION .....           | 3.5        |
| PARAMETERIZATION.....                       | 3.6        |
| SPLIT RANGE CONTROL .....                   | 3.6        |
| CORRESPONDING CONFIGURATION .....           | 3.7        |

|   |      |
|---|------|
| PARAMETERIZATION .....  | 3.7  |
| LEVEL CONTROL .....   | 3.8  |
| CORRESPONDING CONFIGURATION .....                             | 3.8  |
| PARAMETERIZATION .....  | 3.9  |
| RATE CONTROL LOOP WITH LEAD - LAG .....                       | 3.9  |
| CORRESPONDING CONFIGURATION .....                             | 3.10 |
| PARAMETERIZATION .....  | 3.10 |
| FLOW COMPENSATION CONFIGURATION WITH TOTALIZATION .....       | 3.11 |
| CORRESPONDING CONFIGURATION .....                             | 3.11 |
| PARAMETERIZATION .....  | 3.11 |
| HYDROSTATIC TANK GAUGING .....                                | 3.12 |
| CORRESPONDING CONFIGURATION .....                             | 3.13 |
| PARAMETERIZATION .....  | 3.13 |
| COMBUSTION CONTROL WITH DOUBLE CROSS LIMITS .....             | 3.15 |
| CORRESPONDING CONFIGURATION .....                             | 3.16 |
| PARAMETERIZATION .....  | 3.16 |
| 3 ELEMENT BOILER LEVEL/FEEDWATER CONTROL .....                | 3.18 |
| CORRESPONDING CONFIGURATION .....                             | 3.19 |
| PARAMETERIZATION .....  | 3.19 |
| APPLICATION OF THE FLEXIBLE MATHEMATICAL FUNCTION BLOCK ..... | 3.20 |
| SMITH PREDICTOR AND TRANSFER FUNCTION BLOCK APPLICATION ..... | 3.22 |
| CORRESPONDING CONFIGURATION .....                             | 3.23 |
| PARAMETERIZATION .....  | 3.24 |



# DEFINITIONS

| Term                                  | Description   |
|---------------------------------------|---|
| Action Object (action)                | Object that supports object instantiation and exclusion (blocks or links) in the resource.  |
| Alarm                                 | Detection when the block leaves a particular status and when it returns to this status.   |
| Alert                                 | Alarms and Events.  |
| Algorithm                             | Rule set for problem solution following some steps  |
| Application                           | Software functional unit which consists of a set of interconnected function blocks, events and objects, that can be distributed, have interfaces with their applications and posses other applications.                     |
| Array                                 | Array with all elements that are of the same type and each one can be uniquely referenced by a subscript reference.   |
| Attribute                             | Property or characteristic of an entity for instantiation. Value and status are attributes for an output parameter.   |
| Bitstring                             | Data type which consists of a bit array. Each bit corresponds to a specific characteristic that is allowed to the user make a composition of some bits (some characteristics at the same time).                             |
| Block or block instantiation          | Software logic processing unit which consists of a single copy named of the block and of the specific parameters associated by the block type, it can have several PID block instantiations, for example.                   |
| Block Type                            | Common qualities for all type instantiations. The PID block is a block type, for example.   |
| BLOCK.PARAMETER                       | Definition used to indicate the block and the parameter associated, for example: DIAG.BEHAVIOR.   |
| Bypass                                | This parameter indicates a deviation, normally from the input to the block output, without passing through the algorithm. In the PID block, for example, it means the SP goes to the output.                                |
| Configuration (of a system or device) | Steps for a system: selecting function units, indicating their positions and identifiers, and defining their interconnections.  |
| Constant status                       | Status attribute which has both the indications for high and low limits, used to indicate the lower block can not respond to the output of the higher block.  |
| Container Parameter                   | Block parameter that is accessible in the communication network but can not be connected to the input or output parameter. The value of the parameter can be used in the block algorithm or written by the block algorithm. |
| Critical Alert                        | Alert object with priority from 8 to 15.  |
| Data Structure                        | A set, whose elements need to be of the same data type and each one can be referenced uniquely by an identification   |

| Term                           | Description   |
|--------------------------------|---|
| Data Type                      | Value set for a set of allowed operations.  |
| Dd Item Id                     | A unique number which identifies the description of the object class associated, that is developed as part of the Equipment Description.  |
| Dd member Id                   | A unique number which identifies the function of the object class associated, that is developed as part of the Equipment Description.   |
| Device                         | Physical entity able to perform one or more specific functions into a particular context and delimited for its interfaces.  |
| Download                       | Function which consists of loading data from the client to the server.  |
| Engineering Unit               | Terms of data measuring, such as Celsius degrees, Pound, grams and others.  |
| Enhanced                       | The enhanced block has additional characteristics related to the standard block. For example: EPID block has all characteristics of the PID block plus additional characteristics.          |
| Event                          | Instantaneous occurrence that is important to schedule the algorithm execution.   |
| Execution                      | Process of executing an operation sequence for an algorithm.  |
| Feed forward                   | Process anticipation.   |
| Function Block                 | Block which consists of one or more input or output parameters.   |
| Input Parameter                | Block parameter that has data structure of value and status elements that can be updated through link for the output parameter.   |
| Interface                      | Limit shared between two function units, defined by the function characteristics, common physic interconnection characteristics, signal characteristics, and other adequate characteristics |
| Interoperability               | Capacity to operate multiple equipment, regardless of the manufacturer, in the same system, without losing the functionality.   |
| Invocation                     | Process of starting the sequence execution of specific operations for an algorithm.   |
| Link                           | Connection between two devices.   |
| Multi-Variable Container (MVC) | Object in the Function Block Application Process referred to a group of Function Block parameters defined by user, that can be one or more function blocks in the equipment.                |
| Output Parameter               | Block parameter which has data structure of value and status elements that can be updated through the block algorithm.  |
| Primary Input                  | Input parameter which is critical for the operation of the control or calculation function block. The value of the primary input can be reflected in the parameter of the process variable. |

| Term                               | Description   |
|------------------------------------|---|
| Primary Output                     | Output parameter which depends on the mode parameter, reflects the results of the block algorithm execution or the input value given by user.   |
| Resource                           | Internal function unit of the equipment that has the control, independent of the operation, and provides characteristics needed for scheduling and execution of application algorithms (resource).  |
| Resource block                     | Block which consists of internal parameters that provide the interface for common resource characteristics or information.  |
| ROM (Remote Operations Management) | New technology of Fieldbus Foundation aiming the following application types: <ul style="list-style-type: none"><li>• Remote devices with great number of I/O points</li><li>• Optimized links of multiple points (64 discretes, 16 analogs)</li><li>• WiredHART and WirelessHART connectivity.</li></ul> |
| Schedule                           | Order of the function block execution.  |
| Shedding                           | The shedding occurs when one or other non fieldbus interface provides data periodically for the algorithm.  |
| Target Mode                        | Mode reached by user for the block acts.  |
| Time stamp                         | Date and time register (hour:minute:millisecond) for the alert occurrence.  |
| Transducer Block                   | Block which consists of internal parameters that provide an interface for one or more function abilities for a resource.  |
| Upload                             | This function allows transmit data from server to client.   |
| View                               | Data structure with the mapping associated to the sub set of block parameters. It is used for optimization of reading parameters.   |



# ABBREVIATIONS

- AI (Analog Input)
- AO (Analog Output)
- Cas (Cascade Mode)
- DCS (Distributed Control System)
- DD (Device Description)
- DI (Discrete Input)
- DO (Discrete Output)
- DS (Data Structure)
- E (Enumerated Parameter)
- EU (Engineering Unit)
- FB (Function Block)
- FF: Fieldbus Foundation
- IA (Initialization Acknowledge): status when the block is in Cascade mode
- IMan (Initialization Manual): status when the block is in Cascade mode
- IR (Initialization Request): status when the block is in Cascade mode
- LO (Local Override)
- NI (Not invited)
- RCas (Remote Cascade Mode): status when the block is in Cascade mode
- Rout (Remote Output Mode): status when the block is in Cascade mode
- ROM (Remote Operations Management)
- TB (Transducer Block)



## INTRODUCTION TO FUNCTION BLOCK APPLICATION

### **Overview**

Function block applications are defined as plant or factory applications that perform one or more automatic monitoring and control functions.

### **Function Block**

Function blocks represent the basic automation functions performed by the function block application. Each function block processes input parameters according to a specified algorithm and an internal set of control parameters. They produce output parameters that are available for use within the same function block application or by other function block applications.

### **Transducer Block**

Transducer blocks insulate function blocks from the specifics of I/O devices, such as sensors, actuators, and switches. Transducer blocks control access to I/O devices through a device independent interface defined for use by function blocks. Transducer blocks also perform functions, such as calibration and linearization, on I/O data to convert it to a device independent representation. Their interface to function blocks is defined as one or more implementation independent I/O channels.

### **Resource Block**

Resource blocks are used to define hardware specific characteristics of function block applications. Similar to transducer blocks, they insulate function blocks from the physical hardware by containing a set of implementation independent hardware parameters.

### **Function Block Definitions**

Function blocks are defined by their inputs, outputs, control parameters, and by the algorithm that operates on these parameters. Function blocks are identified using a name (Tag) and a numeric index.

Tags provide a symbolic reference to function blocks. They are unambiguous within the scope of a Fieldbus system. Numeric indices are numbers assigned to optimize access to function blocks. As opposed to function block tags, which are global, numeric indices have meaning only within the application that contains the function block.

Function block parameters define the inputs, outputs, and the data used to control function block operation. They are visible and accessible over the network. Additional parameters, called “contained within” parameters are used to define the private data of a function block. Although visible over the network, they may not participate in function block linkages.

### **Function Block Linkages**

Function block outputs may be linked to inputs of other function blocks. Each linkage indicates that an input parameter of one function block obtains its value from specific output parameters of another function block. While function block “pull” their values from upstream blocks, which block controls the “pulling” depends on the characteristics of the underlying communications.

Two function blocks linked together may exist in the same function block application, or in separate applications, either in the same device or in different devices.

To transfer the data for a function block link, the communication channel must be known, that provide the transfer of parameter data (and other types of data) between applications.

## ***Link with Function Block executing in DFI302***

Any function block executing in DFI302 may be linked to any other function block being executed in other device connected to any of four H1 channels available.

## ***Information Access***

Function block information may be grouped for access depending on how it is to be used. The following four groups are defined for access purposes:

- 1) Dynamic operation data.
- 2) Static operation data.
- 3) All dynamic data, and
- 4) Other static data.

To support access of operator interface information during function block execution, two levels of network access are defined, one for operational traffic and one for background traffic. Operator interface traffic is transferred as background traffic to prevent it from interfering with the operation of time-critical function blocks.

## ***Function Block Application Structure***

Function block applications are modeled as a set of function blocks coordinated to execute a related set of operations. This set of operations collectively provides a single, higher level control function.

Function block model is real-time algorithm that transforms input parameters into output parameters. Their operation is controlled through the setting of control parameters.

Interoperation between function blocks is modeled by linking an input parameter of one function block to an output parameter of another. Function blocks can be bound together within and across devices. Interfaces between function blocks located in the same function block application are locally defined. Those interfaces between function blocks in different devices use the communication services.

To support function block operation, the function block architecture also provides transducer and resource blocks, and display objects.

Function Block Application Process represents the function block application as an integrated set of these components accessed to its network interface.

## ***Block Object***

A block object represents a logical processing unit composed of a set of input, processing, and control parameters and an associated algorithm.

Each block is identified by its Tag which is defined to be unique throughout the control system at one plant site. Block tags are defined as strings with a maximum length of 32 characters.

During system operation, a short hand reference, known as a numeric index is used for block access purposes. A block's numeric index is unique only within the function block application where it exists.

The algorithm of a block is identified by its type and the revision level of its type. This information indicates how the execution of the algorithm is affected by control parameters.

## ***Block Parameters***

Parameters define the inputs, outputs, and control data for a block. Their relationship to each other and to the block algorithm is shown below.



## Parameter Identifiers

Parameter names are unique within a block. Within a system, a parameter can be unambiguously identified by qualifying its name with the tag of its block. This construction is referred to as "Tag.Parameter".

The Tag.Parameter construct is used to obtain the index of a parameter. This is the second way of identifying a parameter.

## Parameter Storage

Parameter may be classified as dynamic, static, or non-volatile. The value of parameter may need to be restored after a power failure based on its classification:

**Dynamic** - a parameter whose value is calculated by the block algorithm and therefore does not need to be restored after a power failure.

**Static** - a parameter which has a specific configured value that must be restored by a device after power failure. An Interface or temporary devices may write to static parameter on an infrequent basis. Static parameter values are normally tracked by a configuration device. To support tracking changes in static parameter, the associated block's static revision parameter will be incremented and an update event will be generated each time a static parameter value is modified.

**Non-volatile** - a parameter whose value is written on a frequent basis and the last saved value must be restored by the device after a power failure.

The classification of a parameter will determine the manner in which the parameter is stored within a device.

### NOTE

The concepts of Non-Volatile (N) and Dynamic (D) parameters are only applied for transmitters. For controllers, the Dynamic and Non-Volatile parameters will be recovered after the Power Up.

## Parameter Usage

Parameters are defined for a block for a specific purpose. Each is defined for use as an input, an output, or a control parameter. Control parameters are also referred to as "contained" parameters because they may not be linked with parameters in other blocks. Each type of usage is defined as follows:

### Contained

A contained parameter is a parameter whose value is configured, set by an operator, higher level device, or calculated. It may not be linked to another function block input or output. The mode parameter is an example of a contained parameter common to all blocks.

### Output

An output parameter is a parameter that may be linked to an input parameter of another function block. Output parameters contain status. The output status indicates the quality of the parameter value and the mode of the block when it was generated.

The value of an output parameter may not be obtained from a source external to the block. It may be generated by the block algorithm, but does not have to be.

The values of certain output parameters are dependent on the value of the mode parameter of the block. These output parameters may be referred to as mode-controlled output parameters.

Blocks whose purpose is to generate a single output contain one parameter designed as the primary output parameter. Primary outputs are used by other blocks for control or calculation purposes. These blocks also contain secondary output parameters such as alarm and event parameters that play a supporting role to the primary output parameter.

### Input

An input parameter obtains its value from a source external to the block. An input parameter may be linked to an output parameter of another function block. Its value may be used by the algorithm of the block.

Input parameter values are accompanied by status. When an input parameter is linked to an output parameter, the status will be provided as the status of the output parameter. When it is not linked to

an output parameter, the status will indicate that the value was not provided by an output parameter. When an expected input parameter value is not received, the function block supported services responsible for delivering the data will set the status of the input parameter to indicate the failure.

If an input parameter is not linked to an output parameter, then it will be treated as a constant value by the function block application. The difference between unlinked input parameters and contained parameters is that input parameters have the capability to support a linkage and contained parameters do not.

Blocks whose purpose is to transform or operate on a single input will contain one parameter designed as the primary input parameter. One input parameter of some types of blocks is designated as the primary input parameter. Primary inputs are used for control or calculation purposes. These blocks may also contain secondary input parameters that support processing done on the primary input parameter.

## **Parameter Relationships**

The execution of a block involves the inputs, outputs, contained parameters, and the algorithm of the block. The execution time for a block's algorithm is defined as a parameter of the block. Its value is dependent on how the block was implemented.

The input parameters are used by the algorithm in conjunction with the state of the function block application containing the block to determine if the algorithm can achieve the target mode established for it. The target mode is the part of the mode parameter that indicates what mode of operation is desired for the block. It is normally set by a control device or the operator.

Under certain operating condition a block may not be able to function in the requested mode. In such cases, the actual mode reflects the mode it is able to achieve. Comparison of the actual against the target indicates whether the target was achieved.

The values for the mode parameter for a block are defined by the Permitted Mode parameter. Thus, the modes available for controlling a block may vary with each block.

The values assigned to the Permitted Mode are selected from those defined by the block designer. They are assigned during block configuration for the specific use of the function block application.

Once the actual mode is determined, the block execution progresses and the outputs are generated.

## **Parameter Status**

All input and output parameters are structures composed of status and value, but some contained parameter (internal parameter, not accessible by other blocks) have also that data type, for example, RCAS\_IN, ROUT\_IN, SP and PV.

The Status field is composed of three parts: Quality, Sub-Status and Limits.

**Quality** – It indicates the quality of the parameter value.

Good Cascade – The quality of the value is good, and it may be part of a cascade structure.

Good Non-Cascade – The quality of the value is good, and the block doesn't support a cascade path.

Uncertain – The quality of the value is less than normal, but the value may still be useful.

Bad – The value is not useful.

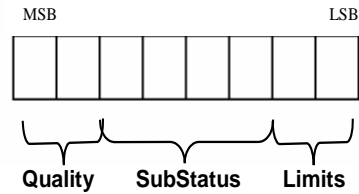
**Sub-Status** – The sub-status is a complement of the quality status and takes information to initialize or break a cascade control, alarms and others. There are different sets of sub-status for each quality.

**Limits** – It provides information whether the associated value is limited or not, as well the direction. The limits are classified as: Not Limited, High Limited, Low Limited, and Constant.

When an input parameter is linked to an output parameter through the link object, the whole structure (status and value) is copied (local link) or received from the bus (external link). If the input is not linked, then the status may be set manually by the user, as well the value.

## Composition of Status

The Status has the following composition:



The quality, sub-status, and limit components of status are defined as follows:

**Quality** - The quality used will be determined by the highest priority condition:

- 0 = Bad
- 1 = Uncertain
- 2 = Good (Non-cascade)
- 3 = Good (Cascade)

**Sub status** - Sub-status values in the status attribute are defined as shown in the following table.

**Limit** - The following limit conditions will be always available in the status attribute.

- 0 = Not limited
- 1 = Low limited
- 2 = High limited
- 3 = Constant

Examples:

0xC1 (in hexadecimal) is "Good-Cascade Non Specific and Low Limited" status

0xCF (in hexadecimal) is "Good-Cascade Not invited and Constant" status

0x4E (in hexadecimal) is "Uncertain Initial Value and High Limited" status

| Quality | Sub-status                                   | Hex value | Not in cascade | Forward path of cascade | Backward path of cascade |
|---------|--|-----------|----------------|-------------------------|--------------------------|
| Bad     | 0 = Non-specific                             | 0x00      | X              | X                       | X                        |
| Bad     | 1 = Configuration Error                      | 0x04      | X              | X                       | X                        |
| Bad     | 2 = Not Connected                            | 0x08      |                |                         |                          |
| Bad     | 3 = Device Failure                           | 0x0c      | X              | X                       | X                        |
| Bad     | 4 = Sensor Failure                           | 0x10      | X              | X                       | X                        |
| Bad     | 5 = No Communication, with last usable value | 0x14      |                |                         |                          |
| Bad     | 6 = No Communication, with no usable value   | 0x18      |                |                         |                          |
| Bad     | 7 = Out of Service (highest priority)        | 0x1c      |                |                         |                          |

| Quality   | Sub-status                           | Hex value | Not in cascade | Forward path of cascade | Backward path of cascade |
|-----------|--------------------------------------|-----------|----------------|-------------------------|--------------------------|
| Uncertain | 0 = Non-specific                     | 0x40      | X              |                         |                          |
| Uncertain | 1 = Last Usable Value                | 0x44      | X              |                         |                          |
| Uncertain | 2 = Substitute                       | 0x48      | X              |                         |                          |
| Uncertain | 3 = Initial Value                    | 0x4c      | X              |                         |                          |
| Uncertain | 4 = Sensor Conversion not Accurate   | 0x50      | X              |                         |                          |
| Uncertain | 5 = Engineering Unit Range Violation | 0x54      | X              |                         |                          |
| Uncertain | 6 = Sub-normal                       | 0x58      | X              |                         |                          |

| Quality | Sub-status                         | Hex value | Not in cascade | Forward path of cascade | Backward path of cascade |
|---------|------------------------------------|-----------|----------------|-------------------------|--------------------------|
| GoodNC  | 0 = Non-specific (lowest priority) | 0x80      | X              |                         |                          |
| GoodNC  | 1 = Active Block Alarm             | 0x84      | X              |                         |                          |
| GoodNC  | 2 = Active Advisory Alarm          | 0x88      | X              |                         |                          |
| GoodNC  | 3 = Active Critical Alarm          | 0x8c      | X              |                         |                          |
| GoodNC  | 4 = Unacknowledged Block Alarm     | 0x90      | X              |                         |                          |
| GoodNC  | 5 = Unacknowledged Advisory Alarm  | 0x94      | X              |                         |                          |
| GoodNC  | 6 = Unacknowledged Critical Alarm  | 0x98      | X              |                         |                          |

| Quality | Sub-status                     | Hex value | Not in cascade | Forward path of cascade | Backward path of cascade |
|---------|--------------------------------|-----------|----------------|-------------------------|--------------------------|
| GoodC   | 0 = Non-specific               | 0xc0      |                | X                       | X                        |
| GoodC   | 1 = Initiate Acknowledge (IA)  | 0xc4      |                | X                       |                          |
| GoodC   | 2 = Initiate Request (IR)      | 0xc8      |                |                         | X                        |
| GoodC   | 3 = Not Invited (NI)           | 0xcc      |                |                         | X                        |
| GoodC   | 4 = Not Selected (NS)          | 0xd0      |                |                         | X                        |
| GoodC   | 6 = Local Override (LO)        | 0xd8      |                |                         | X                        |
| GoodC   | 7 = Fault State Active (FSA)   | 0xdc      |                |                         | X                        |
| GoodC   | 8 = Initiate Fault State (IFS) | 0xe0      |                | X                       |                          |

#### Example: Conversion from the Enumerations to Number

The following formula is used to obtain the enumeration number of a determinate status attribute:

$$\text{Decimal Value Status} = 64 * \text{Quality} + 4 * \text{Sub-Status} + \text{Limit}$$

For example, considering the following status:

“Uncertain - Initial Value - High Limited”

Where:

Quality = “uncertain” = 1

Sub-Status = “Initial Value” = 3

Limit = “High Limited” = 2

Applying the formula:

Decimal Value Status =  $64 * 1 + 4 * 3 + 2 = 78$  (in decimal) or 0x4E (in Hexadecimal)

#### Example: Conversion from Number to Enumerations

There are many forms to convert the enumerate number to the status string. Below is shown two forms to do this.

The number is expressed in binary as:

Hex Value Status = 78 = 0x4E = 01001110 (in binary)

Dividing this binary number in quality, sub-status and limit fields:

Quality = 01 = 1 = “Uncertain”

Sub-Status = 0011 = 3 = “Initial Value”

Limit = 10 = 2 = “High Limited”

The corresponding status is “Uncertain - Initial Value - High Limited”.

The user must use the value of status in decimal format.

Decimal Value Status = 78

Divide the number by 64. The quotient will be the Quality and save the remainder:

Quality =  $78 / 64 = 1$

Remainder = 14

The remainder should be divided by 4. The quotient will be the Sub-Status and the remainder will be the limit:

SubStatus =  $14 / 4 = 3$

Limit = 2

## Process Variable Calculation

The process variable (PV) parameter reflects the value and status of the primary input value or calculated value based on multiple inputs. The PV parameter is the IN parameter after filtering (PID and AALM), or it reflects the value from the transducer after filtering (AI and AO-readback), or the combination of two input parameters for range extension (ARTH).

The PV parameter has a status, although it is a contained parameter. This status is a copy of the primary input status or the worst status when the PV is based on multiple inputs. The PV value reflects the value of the calculated input regardless the mode of the block, unless this input is not usable, then the PV holds the last usable value.

Optionally, a filter may be applied in the process value signal, whose time constant is PV\_FTIME. Considering a step change to the input, this is the time in seconds to the PV reaches 63,2% of the final value. If the PV\_FTIME value is zero, the filter is disabled.

## Setpoint Calculation

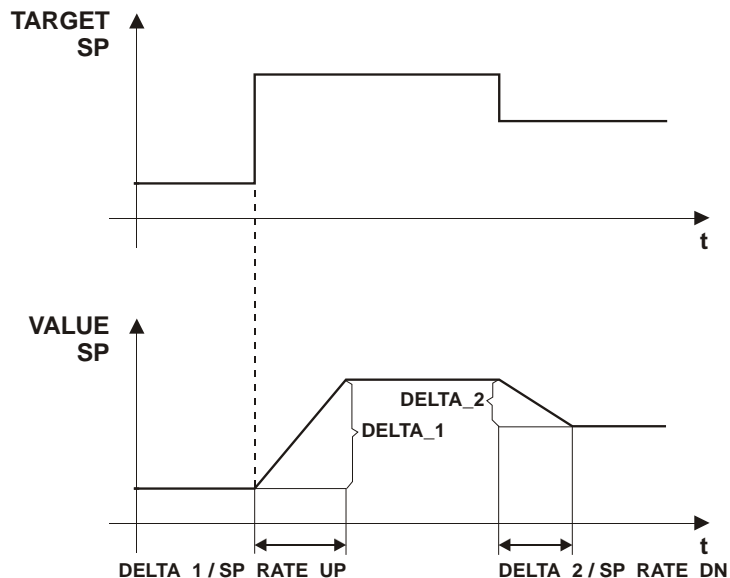
### a) SP limits: SP\_HI\_LIM and SP\_LO\_LIM

At first, the SP will be limited to a range specified by the SP\_HI\_LIM and SP\_LO\_LIM parameters **only in Auto mode**. However in the PID block, if the bit "Obey SP limits if Cas or Rcas" in CONTROL\_OPTS parameter is true, then the setpoint value will be also restricted to setpoint limits in Cas and RCas mode.

### b) SP rate limits: SP\_RATE\_UP and SP\_RATE\_DN

These parameters avoid bump in SP change, and they depend on the block type as well the mode to be effective. The SP rate limits are applied for the PID block in Auto mode, and AO block in Auto, Cas or RCas modes.

When the block is in Auto mode and the user change the SP to a value greater than the current value, then the SP value ramps upward based on the SP\_RATE\_UP parameter. If the new setpoint is less than the current value, the SP value ramps downward based on the SP\_RATE\_DN. When the SP\_RATE\_DN and/or SP\_RATE\_UP is zero the rate limiting is disabled.



The below table summarizes the conditions for SP limits and SP rate limits.

| Block type | Mode     | Required configuration for SP limits<br>(SP_HI_LIM/SP_LO_LIM) | Required configuration for SP rate limits<br>(SP_RATE_UP/SP_RATE_DN) |
|------------|----------|---|--|
| PID        | Auto     | None  | SP_RATE_UP / SP_RATE_DN different of zero                            |
|            | Cas/Rcas | CONTROL_OPTS.“Obey SP limits if Cas or Rcas” is true          | Not apply  |
| AO         | Auto     | None  | SP_RATE_UP / SP_RATE_DN different of zero                            |
|            | Cas/Rcas | Not apply   | SP_RATE_UP / SP_RATE_DN different of zero.                           |

### c) SP tracking PV

Some control strategies require that the transition from a “manual” mode (Rout, Man, LO and Iman) to an “auto” mode (Auto, Cas, Rcas) must be done with error equals to zero, therefore SP must be equal to PV.

The CONTROL\_OPTS of the PID block and the IO\_OPTS of the AO block may be configured for SP tracking PV when the block is in a “manual” mode.

This option is summarized in the following table:

| Bitstring                 | CONTROL_OPTS<br>(PID) | IO_OPTS<br>(AO) | Meaning  |
|---------------------------|-----------------------|-----------------|--|
| SP-PV Track in Man        | X                     | X               | The SP tracks the PV when the target mode is Man.        |
| SP-PV Track in Rout       | X                     |                 | The SP tracks the PV when the actual mode is Rout.       |
| SP-PV Track in LO or Iman | X                     | X               | The SP tracks the PV when the actual mode is LO or Iman. |

## Output Calculation

When the actual mode is AUTO, CAS or RCAS, the normal algorithm is executed. This calculation is specific for each function block type. If the mode is a “manual” mode, the output is just following a value provided by another block (LO, Iman), the user (Man) or a Control Application running on an interface device (Rout).

The output value is limited high and low by the OUT\_HI\_LIM and OUT\_LO\_LIM parameters in PID and ARTH blocks for all modes.

It is possible to disable the output limits in Manual mode in the PID block by setting “No OUT limits in Manual” bit in CONTROL\_OPTS.

## Cascade Control

There is a linked output and input pair involved in each of the different forms of cascade, as shown in the following table.

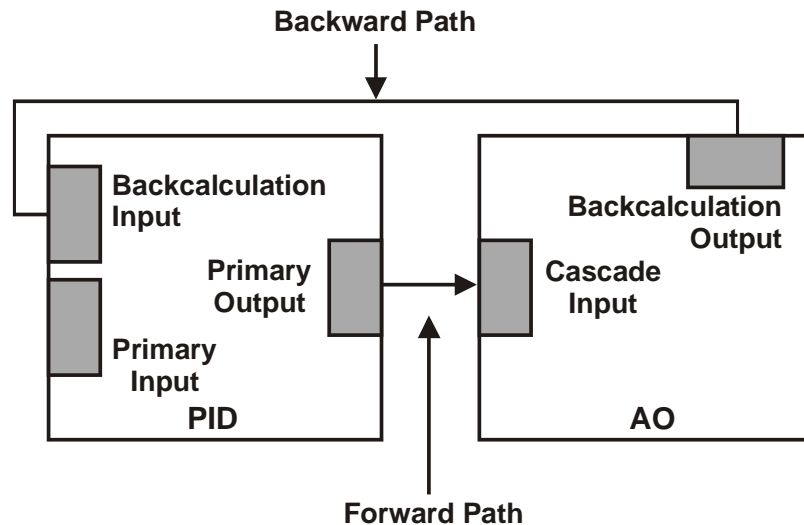
| Mode | Forward | Backward  |
|------|---------|-----------|
| Cas  | CAS_IN  | BKCAL_OUT |
| RCas | RCAS_IN | RCAS_OUT  |
| ROut | ROUT_IN | ROUT_OUT  |

**Table 1 – Parameter Pairs**

In a cascade, the upper control block provides an output value and status, which becomes the cascade input to the lower block.

The lower block in the cascade provides an output value, which is communicated to the upper block as back-calculation input.

Based on the following example, which is the most common form of cascade, it will be shown the process of cascade initialization.



There are four steps to complete a cascade initialization:

**1. Not cascade mode** – As the AO block is in Auto mode, the PID block is not calculating the output (OUT), it is just following the backward value (AO.BKCAL\_OUT -> PID.BKCAL\_IN).

PID  
 MODE\_BLK.Target = Auto  
 MODE\_BLK.Actual = IMan  
 OUT.Status = GoodC-Non-specific

AO  
 MODE\_BLK.Target = Auto  
 MODE\_BLK.Actual = Auto  
 BKCAL\_OUT.Status = GoodC-Not Invited

**2. Initialize** – The user changes the target mode of AO block to Cas, then the AO block sets GoodC-IR in BKCAL\_OUT. The value of BKCAL\_OUT is the initial value for the PID starts to calculate.

The AO block waits for the PID to set GoodC-IA in OUT, which is linked to AO.CAS\_IN (PID.OUT -> AO.CAS\_IN).

PID  
 MODE\_BLK.Target = Auto  
 MODE\_BLK.Actual = IMan  
 OUT.Status = GoodC-Non-specific

AO  
 MODE\_BLK.Target = Cas  
 MODE\_BLK.Actual = Auto  
 BKCAL\_OUT.Status = GoodC-Initialization Request (IR)

**3. Initialization complete** – The AO block goes to Cas, because the PID block sent GoodC-IA.

PID  
 MODE\_BLK.Target = Auto  
 MODE\_BLK.Actual = IMan  
 OUT.Status = GoodC- Initialization Acknowledge (IA)

AO  
 MODE\_BLK.Target = Cas  
 MODE\_BLK.Actual = Cas  
 BKCAL\_OUT.Status = GoodC- Non-specific

**4. Cascade complete** – The PID block changes the status of OUT from GoodC-IA to GoodC-NS.

PID  
 MODE\_BLK.Target = Auto  
 MODE\_BLK.Actual = Auto  
 OUT.Status = GoodC- Non-specific

AO  
MODE\_BLK.Target = Cas  
MODE\_BLK.Actual = Cas  
BKCAL\_OUT.Status = GoodC- Non-specific

**Notes:**

- The remote cascade modes (Rcas and Rout) have a similar mechanism to the process of cascade initialization.
- The function blocks prepared to work as the upper block in the cascade have the BKCAL\_IN parameter, as PID, SPLT, SPG and OSDL
- The function blocks prepared to work as the lower block in the cascade have the BKCAL\_OUT parameter, as PID, AO, SPLT and OSDL.
- The upper blocks will be in Iman mode when the lower block is not in cascade, whose main reasons are:
  - Link failure in backward path (lower.BKCAL\_OUT -> upper.BKCAL\_IN)
  - Lower block can not execute in Cas: the target mode of lower block is not Cas, or there is any condition forcing the lower block to a higher priority mode as fault state (AO in LO), link failure in the forward path (OUT -> CAS\_IN).
  - Tracking (PID in LO), link failure in the primary input (PID in Man), and others. (See details in the section Mode Parameter).
  - Control Application running on as interface device works similarly an upper block in the remote cascade modes (Rcas and Rout).

## Mode Parameter

### a) Mode types

The operation of the block is summarized for each mode type as follows:

**Out of Service (O/S):**

The block is not being evaluated. The output is maintained at last value or, in the case of output class function blocks, the output may be maintained at an assigned Fault State value – last value or configured Fault State value. Setpoint is maintained at last value.

**Initialization Manual (IMan):**

This mode means that the downstream block is not in cascade (Cas mode), therefore the normal algorithm must not be executed and the block output will just follow an external tracking signal (BKCAL\_IN) coming from the downstream block. This mode cannot be requested through the target mode.

**Local Override (LO):**

It applies to control block that supports a track input parameter, when the control block is LO, its output is tracking the TRK\_VAL input parameter.

The LO mode also applies to output block when it is in fault state. This mode cannot be requested through the target mode.

**Manual (Man):**

The block output is not being calculated, although it may be limited. The operator may set directly the outputs of the block.

**Automatic (Auto):**

The normal algorithm calculates the block output. If the block has a setpoint, it will be used a local value that may be written by an operator through an interface device.

**Cascade (Cas):**

The setpoint comes from another block over a link (CAS\_IN), therefore the operator cannot change the setpoint. The normal algorithm calculates the block output based on that setpoint. In order to achieve this mode, the algorithm uses the CAS\_IN input and BKCAL\_OUT output to establish the cascade with the upstream block in a bumpless way.

**Remote Cascade (RCas):**

The block setpoint is being set by a Control Application running on an interface device to the RCAS\_IN parameter. The normal algorithm calculates the block output based on that setpoint, so the block running in Rcas works similarly a “lower block” in cascade. In order to achieve this mode, the block algorithm uses the RCAS\_IN and RCAS\_OUT to **establish a relation like a cascade** with the interface device in a bumpless way. Therefore the Control Application works **similarly as an “upper block”**, but its algorithm is not synchronized to the schedule and neither does it use a link to transfer the setpoint to the block.



**Remote Output (ROut):**

The block output is being set by a Control Application running on an interface Device to the ROut\_IN parameter. In order to achieve this mode, the block algorithm uses the ROut\_IN and ROut\_OUT to **establish a relation like a cascade** with the interface device in a bumpless way. Therefore the Control Application works **similarly as an “upper block”**, but its algorithm is not synchronized to the schedule and neither does it use a link to transfer the output to the block. The block running in ROut works similarly a “lower block” in cascade.

Auto, Cas, and RCas are the “automatic” modes, which calculate the primary output using the normal algorithm. The “manual” modes are IMan, LO, Man, and ROut.

| Mode type | Source of SP   | Source of OUT  |
|-----------|--|--|
| O/S       | User   | User   |
| Iman      | User   | Other function block – following BKCAL_IN parameter  |
| LO        | PID / EPID / APID : User<br>AO / DO : Fault state (last value or FSTATE_VAL) | PID / EPID / APID : Other function block – following TRK_VAL parameter<br>AO / DO : Fault state (last value or FSTATE_VAL) |
| Man       | User   | User   |
| Auto      | User   | Block algorithm  |
| Cas       | Other function block – following CAS_IN parameter                            | Block algorithm  |
| Rcas      | Control Application running on an interface device                           | Block algorithm  |
| Rout      | Block keeps last value   | Control Application running on an interface device   |

**b) Elements of MODE\_BLK**

The mode parameter (MODE\_BLK) is defined in every function block. It is defined as having four elements:

- 1) Target** - This is the mode requested by the operator. Only one mode from those allowed by the permitted mode parameter may be requested, that check will be done by the device.
- 2) Actual** - This is the current mode of the block, which may differ from the target based on operating conditions and block configuration, as input parameter status and bypass configuration, for example. Its value is *always* calculated as part of block execution, therefore the user can not write in this attribute.
- 3) Permitted** – It defines the modes that are allowed for an instance of the block. The permitted mode is configured based on the application requirement. For example, if a PID block does not have link for CAS\_IN, the Cas mode should not be permitted for that block. It is like a list of mode types selected from the supported modes.
- 4) Normal** - This is the mode which the block should be set to during normal operating conditions. The normal attribute is used as a *reminder*. It does not affect the algorithm calculation.

Execution of a function or transducer block will be controlled through the mode parameter. The user sets the target mode, which indicates what mode of operation is desired for the block. Then, the algorithm evaluates if the block can be executed in the *requested mode* (target mode) *or the nearest higher priority mode possible*. The actual mode reflects the mode of block operation.

Other concepts of mode:

**Retained target:** When the target mode is O/S, MAN, RCAS or ROUT the target mode attribute may retain information about the previous target mode. This information may be used by the block in mode shedding and setpoint tracking. This feature is optional and the interface device is responsible to implement it.

**Supported mode:** Each *block type* has a set of mode types supported, it means that *the block definition specifies* in which modes the block may operate.

**c) Priority of mode**

The concept of priority is used when the block calculates the actual mode, and when determining if write access is allowed for a particular mode or other of higher priority.

| Mode | Description           | Priority    |
|------|-----------------------|-------------|
| O/S  | Out of Service        | 7 – highest |
| IMan | Initialization Manual | 6           |
| LO   | Local Override        | 5           |
| Man  | Manual                | 4           |
| Auto | Automatic             | 3           |
| Cas  | Cascade               | 2           |
| Rcas | Remote Cascade        | 1           |
| Rout | Remote Output         | 0 - lowest  |

**Priority of the Mode**

**d) Mode shedding**

Interface devices such as a host computer, distributed control system (DCS) controller, or programmable logic controller (PLC) may exist which not support the function blocks application architecture but have proprietary control applications running on them.

Such applications may adjust the values of the block setpoint (RCas mode) and/or primary output (Rout mode) parameters in a function block. When doing so, they provide the value of each parameter along with its status.

If a new value is not received by the function block within a specified “update time” (defined by SHED\_RCAS and SHED\_ROUT parameters of the Resource block), or a bad status is received, then the function block mode will be changed to a non-remote mode of higher priority.

The SHED\_OPT parameter configures the desired behavior when shedding from a remote mode (Rcas and Rout); therefore **it does not include the Cas mode**. Also, it determines if the shed mode is maintained once the RCAS\_IN or ROUT\_IN parameter updating is recovered (no return – target mode receives the shed mode) or original mode will be restored when the cause of shedding has cleared (normal return – no change in target mode).

The shed option has the following enumeration:

0 = Undefined – Invalid

1 = Normal shed, normal return – Actual mode changes to the next lowest priority non-remote mode permitted but returns to the target remote mode when the remote computer completes the initialization handshake.

2 = Normal shed, no return – Target mode changes to the next lowest priority non-remote mode permitted. The target remote mode is lost, so there is no return to it.

3 = Shed to Auto, normal return.

4 = Shed to Auto, no return – Target mode changes to Auto on detection of a shed condition.

5 = Shed to Manual, normal return.

6 = Shed to Manual, no return – Target mode changes to Man on detection of a shed condition.

7 = Shed to Retained target, normal return.

8 = Shed to Retained target, no return (change target to retained target).

**e) Mode calculation**

The actual mode will be calculated based on the following:

Each mode type has some conditions that force the actual mode to be of higher priority than the target mode.

Starting from the highest priority mode (O/S), it is analyzed its corresponding conditions. If they are present, then the actual mode will be this one, otherwise it is necessary to check the conditions for the next lower priority mode (IMan, LO, Man, Auto, Cas, Rcas and Rout) till the target mode, exclusive. For instance, if the target mode is Cas, it is necessary to check the conditions for O/S, IMan, LO, Man and Auto, in this order. If all those conditions are false, the actual mode will be the target mode.

| Mode        | Conditions  |
|-------------|---|
| <b>O/S</b>  | Resource block is in O/S (resource state is Standby)<br>- Enumerated parameter has an invalid value   |
| <b>IMan</b> | BKCAL_IN.status is Bad<br>BKCAL_IN.status is Good – Fault State Active, Not Invited or Initialization Request.  |
| <b>LO</b>   | Fault state is active (in an output function block)<br>CONTROL_OPTS.Track Enable active and TRK_IN_D is active.<br>If target is Man, then the CONTROL_OPTS.Track in Manual must be active.  |
| <b>Man</b>  | Target mode has just changed from O/S<br>(Status attribute of primary input parameter (IN parameter) is Bad or Uncertain with option to treat Uncertain as Bad) and<br>(Bypass not set).<br>Target mode is RCas or ROut, and SHED_OPT=shed to Manual or shed to next  |
| <b>Auto</b> | Target mode is Cas and (CAS_IN.status=Bad or cascade initialization not completed)<br>Target mode is RCas and RCAS_IN.status=Bad and SHED_OPT=shed to Auto or shed to next<br>Target mode is ROut and ROUT_IN.status=Bad and SHED_OPT = shed to Auto or shed to next.   |
| <b>Cas</b>  | Actual mode last execution was Cas.<br>Target mode is Cas and cascade initialization has just completed<br>Target mode is RCas and RCAS_IN.status=Bad and SHED_OPT=shed to next and cascade initialization has just completed<br>Target mode is ROut and ROUT_IN.status=Bad and SHED_OPT=shed to next and cascade initialization has just completed |
| <b>RCas</b> | RCas cascade initialization has just completed or actual mode last execution was RCas.  |
| <b>ROut</b> | ROut cascade initialization has just completed or actual mode last execution was ROut.  |

When the actual mode is different from the target, a good suggestion to find the cause is an analysis of the corresponding conditions for the actual mode. The most frequent causes are shown below.

| Mode | Frequent cause   |
|------|--|
| O/S  | Check the Resource block mode and all enumerated parameters  |
| Iman | Check the BKCAL_IN.status:<br>Bad – No Comm : link failure in backward path (BKCAL_OUT -> BKCAL_IN)<br>NI : lower block can not execute in Cas (check the target mode of lower block as well fault state condition) or link failure in forward path (OUT -> CAS_IN)  |
| LO   | Check :<br>Value and status of TRK_IN_D, TRK_VAL and CONTROL_OPTS (Track Enable, Track in Manual)<br>Output block : status of CAS_IN and the delay time for fault state established by FSTATE_TIME   |
| Man  | Status of IN<br>If the target is Cas, check the links with the upper block in backward path (BKCAL_OUT -> BKCAL_IN) and forward path (OUT -> CAS_IN)<br>If the block is not reaching the target mode Rcas or Rout, compare the updating rate of RCAS_IN and ROUT_IN by a Control Application to SHED_RCAS and SHED_ROUT. |
| Auto | If the target is Cas, check the links with the upper block in backward path (BKCAL_OUT -> BKCAL_IN) and forward path (OUT -> CAS_IN)<br>If the block is not reaching the target mode Rcas or Rout, compare the updating rate of RCAS_IN and ROUT_IN by a Control Application to SHED_RCAS and SHED_ROUT.                 |
| Cas  | If the block is not reaching the target mode Rcas or Rout, compare the updating rate of RCAS_IN and ROUT_IN by a Control Application to SHED_RCAS and SHED_ROUT.   |

#### f) Cascade initialization

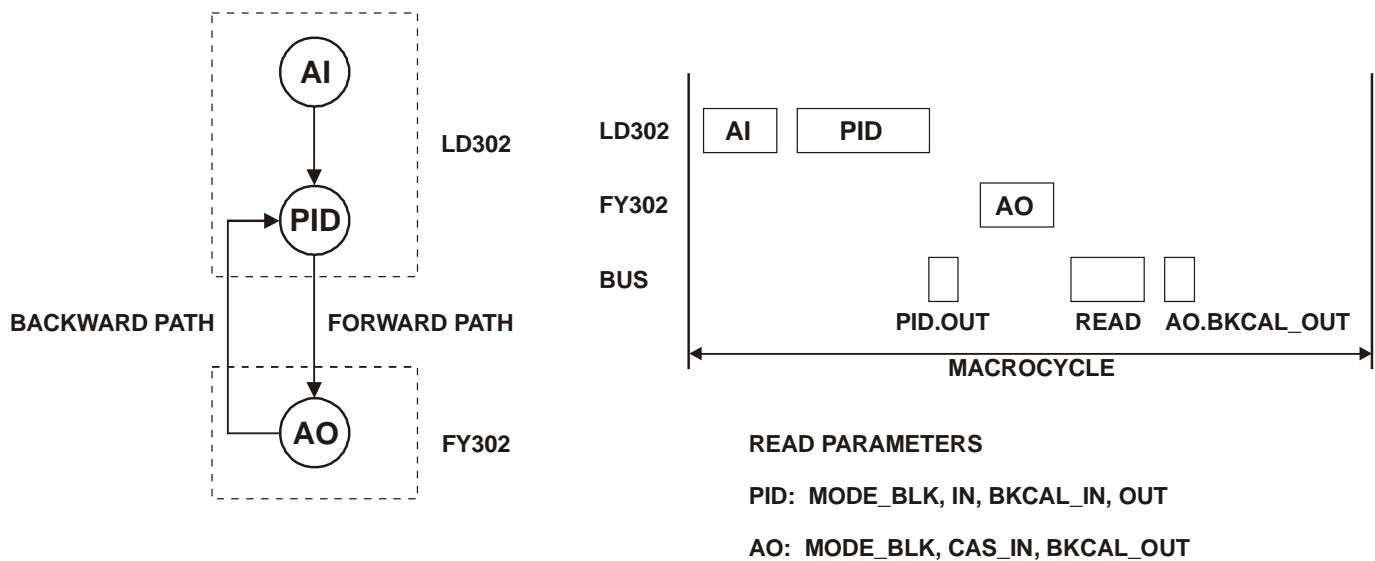


Fig.1 – Example to analyze the cascade initialization

The following table shows the sequence of status exchange between the PID and AO blocks for cascade initialization, and a link failure between PID.OUT and AO.CAS\_IN (forward path) after execution 8.

**PID parameters**

| Execution | 1       | 2       | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   |
|-----------|---------|---------|------|------|------|------|------|------|------|------|------|
| Target    | O/S     | Auto    |      |      |      |      |      |      |      |      |      |
| IN        | Bad     | GNC     | GNC  | GNC  | GNC  | GNC  | GNC  | GNC  | GNC  | GNC  | GNC  |
| BKCAL_IN  | Bad     | Bad-O/S | NI   | IR   | GC   | GC   | GC   | GC   | NI   | NI   | NI   |
| Actual    | O/S     | Iman    | IMan | IMan | Auto | Auto | Auto | Auto | IMan | IMan | IMan |
| OUT       | Bad-O/S | GC      | GC   | IA   | GC   | GC   | GC   | GC   | GC   | GC   | GC   |

**AO parameters**

| Execution | 1       | 2   | 3    | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  |
|-----------|---------|-----|------|-----|-----|-----|-----|-----|-----|-----|-----|
| Target    | O/S     | Cas |      |     |     |     |     |     |     |     |     |
| CAS_IN    | Bad     | GC  | GC   | IA  | GC  | GC  | GC  | Bad | Bad | Bad | Bad |
| Actual    | O/S     | Man | Auto | Cas | Cas | Cas | Cas | LO  | LO  | LO  | LO  |
| BKCAL_OUT | Bad-O/S | NI  | IR   | GC  | GC  | GC  | GC  | NI  | NI  | NI  | NI  |

Meaning of status:

GNC – Good Non-Cascade – Non Specific

GC - Good Cascade – Non Specific

IA - Good Cascade – Initialization Acknowledge

IR - Good Cascade – Initialization Request

NI - Good Cascade – Not Invited

Bad – Bad – any sub-status

The above sequence for cascade initialization applies not only for the Cas mode, but also for Rcas and Rout modes.

**g) Example**

The mode configuration for the control loop of the figure 1.

|     | Supported                          | Target | Actual | Permitted        | Normal |
|-----|------------------------------------|--------|--------|------------------|--------|
| AI  | O/S,Man,Auto                       | Auto   | -      | O/S,Auto         | Auto   |
| PID | O/S,Iman,LO,Man,Auto,Cas,Rcas,Rout | Auto   | -      | O/S,Man,Auto     | Auto   |
| AO  | O/S,Iman,LO,Man,Auto,Cas,Rcas      | Cas    | -      | O/S,Man,Auto,Cas | Cas    |

#### h) Specific information for driver developers

Internally, the each mode attribute is assigned within the bitstring in the following manner:

|           | Hex value | Bit 7                  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------|-----------|------------------------|-------|-------|-------|-------|-------|-------|-------|
| O/S       | 0x80      | 1                      | 0     | 0     | X     | X     | X     | X     | X     |
| Iman & LO | -         | Not Valid Target Modes |       |       |       |       |       |       |       |
| Man       | 0x10      | 0                      | 0     | 0     | 1     | 0     | X     | X     | X     |
| Auto      | 0x08      | 0                      | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| Cas       | 0x0c      | 0                      | 0     | 0     | 0     | 1     | 1     | 0     | 0     |
| Rcas      | 0x0a      | 0                      | 0     | 0     | 0     | 1     | X     | 1     | 0     |
| Rout      | 0x09      | 0                      | 0     | 0     | 0     | 1     | X     | 0     | 1     |

Where: X = indicates the bit setting is retained from the previous target mode.

#### Target and Normal Mode Bitstring

|      | Hex value | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| O/S  | 0x80      | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Iman | 0x40      | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| LO   | 0x20      | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     |
| Man  | 0x10      | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     |
| Auto | 0x08      | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| Cas  | 0x04      | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     |
| Rcas | 0x02      | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     |
| Rout | 0x01      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     |

#### Actual Mode Bitstring

| Block | Permitted modes  | Hex value | O/S | - | - | Man | Auto | Cas | Rcas | Rout |
|-------|------------------|-----------|-----|---|---|-----|------|-----|------|------|
| AI    | O/S,Auto         | 0x88      | 1   | 0 | 0 | 0   | 1    | 0   | 0    | 0    |
| PID   | O/S,Man,Auto     | 0x98      | 1   | 0 | 0 | 1   | 1    | 0   | 0    | 0    |
| AO    | O/S,Man,Auto,Cas | 0x9c      | 1   | 0 | 0 | 1   | 1    | 1   | 0    | 0    |

#### Permitted Mode Bitstring

(Example for the loop control in the previous figure)

The retained target mechanism is suitable to work with toggle switch in the interface device following the rules:

A/M toggle switch:

“Automatic mode” → Man mode: Bit #4 ← 1 and Bit #3 ← 0

Man mode → “automatic mode”: Bit #4 ← 0 and Bit #3 ← 1

Cascade/Local toggle switch:

Cascade → Local: Bit #2 ← 0

Local → Cascade: Bit #2 ← 1

Remote/Non-remote setpoint toggle switch:

Remote → Non-remote: Bit #1 ← 0 and Bit #0 ← 0

Non-remote → Remote: Bit #1 ← 1 and Bit #0 ← 0

Remote/Non-remote output toggle switch:

Remote → Non-remote: Bit #1 ← 0 and Bit #0 ← 0

Non-remote → Remote: Bit #1 ← 0 and Bit #0 ← 1

| Target mode    | Rule for toggle           | Value retained | Possible retained modes  | Target + Retained modes  |
|----------------|---------------------------|----------------|--|--|
| O/S : 100xxxxx | Bit #7 ← 0                | 000xxxxx       | 00010000 (0x10) : Man<br>00001000 (0x08) : Auto<br>00001100 (0x0c) : Cas<br>00001010 (0x0a) : Rcas<br>00001001 (0x09) : Rout | 10010000 (0x90) : Man<br>10001000 (0x88) : Auto<br>10001100 (0x8c) : Cas<br>10001010 (0x8a) : Rcas<br>10001001 (0x89) : Rout |
| Man: 00010xxx  | Bit #4 ← 0 and Bit #3 ← 1 | 00001xxx       | 00001000 (0x08) : Auto<br>00001100 (0x0c) : Cas<br>00001010 (0x0a) : Rcas<br>00001001 (0x09) : Rout                          | 00010000 (0x10) : Auto<br>00010100 (0x14) : Cas<br>00010010 (0x12) : Rcas<br>00010001 (0x11) : Rout                          |
| Rcas: 00001x10 | Bit #1 ← 0 and Bit #0 ← 0 | 00001x00       | 00001000 (0x08) : Auto<br>00001100 (0x0c) : Cas  | 00001010 (0x0a) : Auto<br>00001110 (0x0e) : Cas  |
| Rout: 00001x01 | Bit #1 ← 0 and Bit #0 ← 0 | 00001x00       | 00001000 (0x08) : Auto<br>00001100 (0x0c) : Cas  | 00001001 (0x09) : Auto<br>00001101 (0x0d) : Cas  |

## Scaling Parameters

The scaling parameter defines the operating range and the engineering units associated with a parameter. It also defines the number of digits to the right of the decimal point, which should be used by an interface device in displaying that parameter.

Scaling information is used for two purposes. Display devices need to know the range for bar graphs and trending, as well as the units code. Control blocks need to know the range to use internally as percent of span, so that the tuning constants may remain dimensionless.

The PID blocks take the error and convert it to percent of span using the PV\_SCALE. The algorithm operates on percent of span to produce an output in that form. This is converted back to a number with engineering units by using the range of OUT\_SCALE.

The AI block has the XD\_SCALE parameter to define the engineering units expected from the transducer.

The AO block uses the XD\_SCALE to convert the SP value to the engineering unit expected by the output transducer block, which is also the engineering unit of the readback value.

The following fields form the scale:

- Engineering Units at 100% of scale - The value that represents the upper end of range in engineering unit;
- Engineering Units at 0% of scale - The value that represents the lower end of range in engineering unit;
- Units Index - Device Description units code index for the engineering unit.
- Decimal Point - The number of digits to the right of the decimal point which should be used by an interface device in displaying the specified parameter.

### Example Using Scale Parameter

The PID algorithm works internally with values in percent of span. Therefore the PID block converts the error to percentage (PV\_SCALE), it calculates the output in percentage, and then it converts to engineering unit of output (OUT\_SCALE).

1. The PID takes the input IN and SP and converts to percentage of the PV\_SCALE:

$$\text{VALUE\%} = (\text{VALUE} - \text{EU}_0) * 100 / (\text{EU}_{100} - \text{EU}_0) \quad [\text{PV\_SCALE}]$$

PV\_SCALE:

EU at 100% = 20

EU at 0% = 4

Units Index = mA

Decimal point = 2

SP = 15 mA

PV = 10 mA

The values of SP and PV in percentage are:

$$\text{SP\%} = (15 - 4) * 100 / (20 - 4) = 68.75\%$$

$$\text{PV\%} = (10 - 4) * 100 / (20 - 4) = 37.50\%$$

2. The PID algorithm calculates the error in percentage. If it is configured the reverse action, the error is the difference between SP% and PV%.

$$\text{Error\%} = \text{SP\%} - \text{PV\%} = 31.25\%$$

3. The PID algorithm applies the Error% to the calculation of the P, I and D terms. If only the proportional term is enabled, the value of the output is:

GAIN = 1.0

RESET = +INF

RATE = 0.0

$$\text{OUT\%} = 31.25\%$$

4. The output value is converted from percentage to engineering units of the OUT\_SCALE:  

$$\text{OUT} = \text{OUT\%} / 100 * (\text{EU}_{100} - \text{EU}_0) + \text{EU}_0 \quad [\text{OUT\_SCALE}]$$

OUT\_SCALE:  
 EU at 100% = 15  
 EU at 0% = 3  
 Units Index = psi  
 Decimal point = 2

The output value of this example is:  

$$\text{OUT} = 31.25 / 100 * (15 - 3) + 3 = 6.75 \text{ psi}$$

## Modbus Scale Conversion

The Modbus function blocks can read and write digital and analogical data of other Modbus slave or master devices. For each configured analogical point, which for Modbus reading or writing, it is associated a scale conversion parameter. The parameters and blocks which have Modbus conversion scale are: MBCM.IN\_x, MBCS.IN\_x, MBCS.OUT.x, MBSM.PVALUE\_x, which x=1, 2, until the point limit of that block.

The scale conversion for the Modbus protocol has two purposes:

- Conversion from the Fieldbus analogical value to Modbus value expressed in Engineering Units.
- Conversion from the Modbus analogical value to the Fieldbus value in Engineering Units.

The scale parameters are defined in the data structures DS\_256, DS\_257, DS\_258, and DS\_259 (see the item "Data Structure") and they are composed by the following fields:

- FROM\_EU\_100% - defines the higher input unit value (actual data unit).
- FROM\_EU\_0% - defines the lower input unit value (actual data unit).
- TO\_EU\_100% - defines the higher output unit value (desired data unit).
- TO\_EU\_0% - defines the lower output unit value (desired data unit).
- Data Type – data type which it desires to convert from or to Modbus (in the Fieldbus the data always will be float). The table with the supported data type is below:

| Number of Data Type | Data Type          |
|---------------------|--------------------|
| 1                   | Float              |
| 2                   | Unsigned8          |
| 3                   | Unsigned16         |
| 4                   | Unsigned32         |
| 5                   | Integer8           |
| 6                   | Integer16          |
| 7                   | Integer32          |
| 8                   | Swapped.Float      |
| 9                   | Swapped.Unsigned8  |
| 10                  | Swapped.Unsigned16 |
| 11                  | Swapped.Unsigned32 |
| 12                  | Swapped.Integer8   |
| 13                  | Swapped.Integer16  |
| 14                  | Swapped.Integer32  |

### Note

In despite of there are different data structures which have scale conversion, they follow the same conversion procedure.

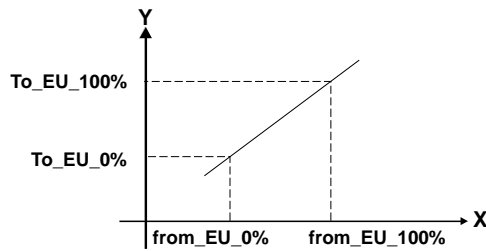
### Procedure to handle the conversion from FF parameter to Modbus variable

- Load INx\_VALUE.
- Calculate  $Y = (A * \text{INx\_VALUE} + B)$ .
- Convert Y to DATA\_TYPE\_IN, generating MOD\_VAR\_IN.
- Write MOD\_VAR\_IN.



**Procedure to handle the conversion from Modbus variable to FF parameter:**

- Read MOD\_VAR\_OUT.
- Convert MOD\_VAR\_OUT to float, generating Y
- Calculate  $OUTx\_VALUE = (A * Y + B)$ .
- Store  $OUTx\_VALUE$ .



Where:

$$A = (To\_EU\_100\% - To\_EU\_0\%) / (From\_EU\_100\% - From\_EU\_0\%)$$

$$B = To\_EU\_0\% - A * From\_EU\_0\%;$$

IN\_VALUE, OUT\_VALUE: FF parameters

MOD\_VAR\_IN, MOD\_VAR\_OUT: Modbus variables

Y: auxiliary float variable

Below, there are examples using the Modbus scale conversion.

**1) Conversion from Modbus to Fieldbus**

It considers a Modbus Slave device which has an analogical value of temperature in percentage (0-10000) with the data type Integer 2 bytes. It desired to use the value in Fieldbus in Celsius degree (0-500). It uses the Master Control Block Modbus (MBCM).

A) In the MBCM block, the scale is configured (OUT\_1.SCALE\_LOC\_OUT\_1) of the following form:

- FROM\_EU\_100% = 10000
- FROM\_0% = 0
- TO\_EU\_100% = 500
- TO\_EU\_0% = 0
- DATATYPE = Integer16

B) The MBCM block reads the Modbus data from the slave and stores in MOD\_VAR\_OUT. Thus, it converts the Y value. For last, it calculates the OUT output using the equation  $OUT = A * Y + B$ . Using the values from the example above and considering the actual value of temperature is 6000, it has:

$$MOD\_VAR\_OUT = 6000 \text{ (value read from Modbus)}$$

And following the equations showed above:

$$A = (TO\_EU\_100\% - TO\_EU\_0\%) / (From\_EU\_100\% - From\_EU\_0\%)$$

$$A = (500 - 0) / (10000 - 0) = 0.05$$

$$B = TO\_EU\_0\% - A * From\_EU\_0\%$$

$$B = 0 - (0.05 * 0) = 0$$

$$OUT = A * Y + B$$

$$OUT = 0.05 * Y$$

$$OUT = 0.05 * 6000 = 300$$

The OUT output value for this example will be:

$$OUT = 300 [^{\circ}C]$$

**2) Conversion from Fieldbus to Modbus**

It considers the DFI is the Modbus slave equipment and it desires to become available one temperature analogical value from one TT302, range 0-500 [°C] for a Modbus master in the range 4-20 [mA] with the data type integer 2 bytes. Using the Modbus Block Control Slave (MBCS).

- A) In the MBCS block, the scale is configured (IN\_1.SCALE\_CONV\_IN\_1) of the following form:
- FROM\_EU\_100% = 500
  - FROM\_0% = 0
  - TO\_EU\_100% = 20
  - TO\_EU\_0% = 4
  - DATATYPE = Integer16

B) The MBCS block reads the data from the Fieldbus in the IN input and stores in IN\_VALUE. It calculates the Y value following the equation  $Y = A * IN\_VALUE + B$ . Thus, it converts the value to the DATATYPE specified and saves in MOD\_VAR\_IN which it will be the value to be sent to the Modbus.

Using the values from the example above and considering the actual value of temperature is 300, it has:

$$IN\_VALUE = 300 \text{ (value read from TT302)}$$

And following the equations showed above:

$$A = (TO\_EU\_100\% - TO\_EU\_0\%) / (From\_EU\_100\% - From\_EU\_0\%)$$
$$A = (20 - 4) / (500 - 0) = 0,032$$

$$B = TO\_EU\_0\% - A * From\_EU\_0\%$$
$$B = 4 - (0,032 * 0) = 4$$

$$MOD\_VAR\_IN = A * IN\_VALUE + B$$
$$MOD\_VAR\_IN = 0.032 * 300 + 4$$
$$MOD\_VAR\_IN = 13.6$$

The value of the Modbus variable read (after the conversion to integer) for this example will be:

$$OUT = 14 \text{ [mA]}$$

## Fault State Handling

### A) Definition

The Fault State is a special state that allows the output block to do safe action when it has been detected an abnormal situation or the user set to Fault State in the Resource block.

The abnormal situation occurs to there is an unusable input (bad sensor, for example) or the loss of the communication between function blocks longer than a specified time (FSTATE\_TIME).

The blocks that support cascade control (as PID, OSDL and SPLT) propagate the fault state status forward to the output block.

When the condition that activated the Fault State is normalized, the Fault State is cleared and the block returns to the normal operation.

### B) Generate the Initiate Fault State status (The fault was detected by the own block)

Blocks like PID, OSDL and SPLT may be configured to send an Initiate Fault State (IFS) status when they detect an unusable input. The bit "IFS if bad IN" and/or the bit "IFS if bad CAS\_IN" in the STATUS\_OPTS or OSDL\_OPTS parameters must be true to generate an IFS status when the corresponding input is unusable.

### C) Propagate the Initiate Fault State status (The fault occurred in the upstream block)

The blocks that support the cascade control have special handling to propagate the fault to the downstream blocks till the output block.

When the block, that is in cascade mode (Cas, RCas), receives an Initiate Fault State (IFS) status, then this status will be reported to forward path. For example, it considers a PID block that is receiving a "Good Cascade IFS" status in CAS\_IN input. If the target mode of the PID is Cas, then the IFS status will be the status of OUT, replacing the normal status. Otherwise, the IFS status will not be propagated forward.

**D) Using the Resource Block to activate the Fault State**

The Fault State of the Resource block forces all output function blocks into the device to go immediately to the fault state condition. The Resource block has some parameters that define the fault state behavior:

- FEATURES\_SEL – The “Fault State supported” bit is used to enable the Fault State characteristic into the Resource block. The default value is disabled.
- FAULT\_STATE – It only indicates the Fault State in the Resource block, not in individual output block. For example, if an AO block is in Fault State because its CAS\_IN input is bad, the FAULT\_STATE parameter will not be active.
- SET\_FSTATE – The user may force FAULT\_STATE to active by setting this parameter to On.
- CLEAR\_FSTATE – The user may force FAULT\_STATE to clear by setting this parameter to On.

**Fault State Active**

When the Output Function Blocks detects an abnormal condition, the block goes to a fault state. The abnormal conditions are:

- Loss of communication to CAS\_IN for a time that exceeds FSTATE\_TIME;
- IFS status in the CAS\_IN input when the target mode is Cas, for a time that exceeds FSTATE\_TIME;
- IFS status in the RCAS\_IN when the target mode is Rcas, for a time that exceeds FSTATE\_TIME;
- The FAULT\_STATE parameter of the Resource block is Active because the user set to on the SET\_FSTATE parameter and the bit “Fault State supported” in the FEATURES\_SEL is true.

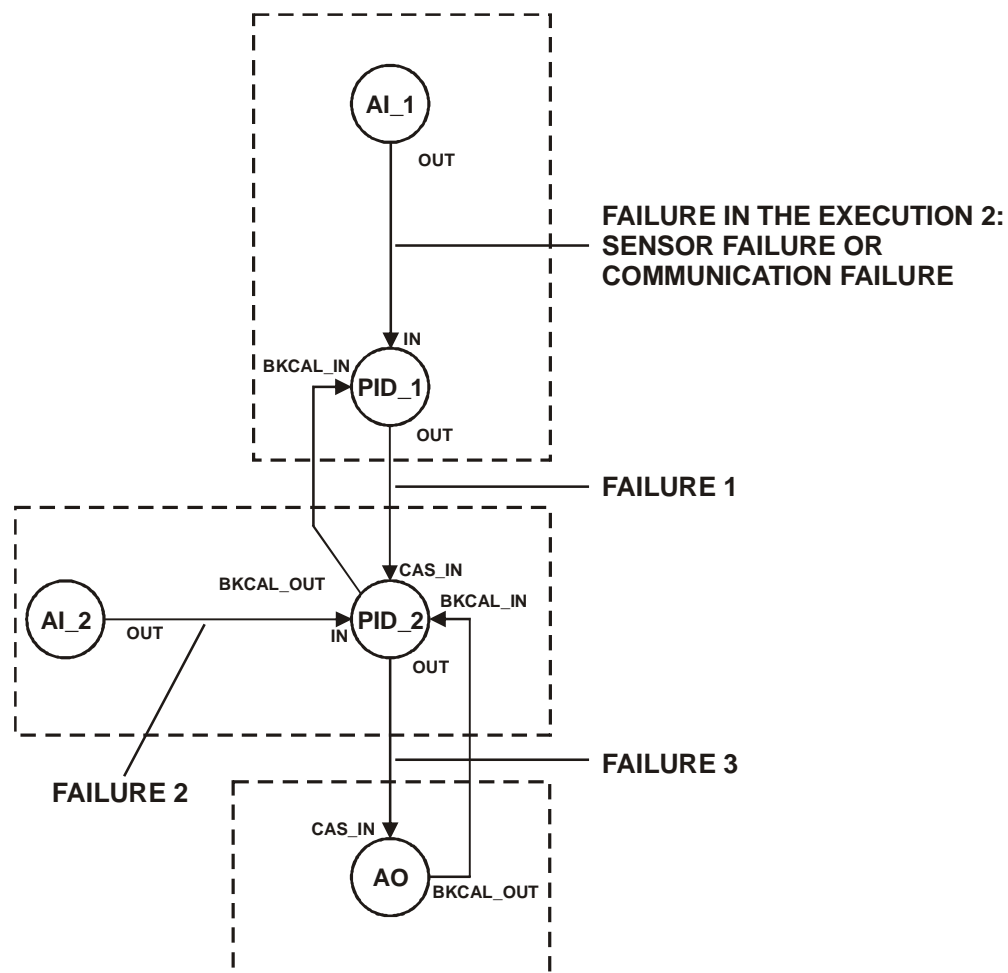
When the output block is in the Fault State the output may retain the last value or goes to the preset value determined by the FSTATE\_VAL. The default is retaining the last value. The output goes to the preset value if the bit “Fault State to value” in the IO\_OPTS is true.

When the Fault State is active then the actual mode of the output block goes to Local Override (LO). In the backward path, the block sends the Not Invited (NI) status to indicate that the block is in “Fault State”.

Optionally, the target mode of the output block will be changed to Manual by the block algorithm when the Fault State is active. In order to set this characteristic, the “Target to Man if Fault State activated” bit in IO\_OPTS needs to be true.

**Examples**

The following control loop must be considered for the examples below. The below tables show the sequence of status exchange between the PID\_1



The Master, PID\_2 Slave and AO blocks are in a fault and normal condition.

The status of PID\_1.IN becomes bad in the execution 2, which may be due to a Bad-Sensor Failure detected by the AI\_1 or a communication failure between AI\_1.OUT and PID\_1.IN. The PID\_2 Slave only propagates the status forward. The AO block receives the IFS status and active the Fault State. In the safety condition the preset value of the FSTATE\_VAL is used to the output of the AO block. After the execution 6, the bad status in the IN of the PID master is cleared and the loop returns to normal operation.

#### PID 1 – Master

STATUS\_OPTS = "IFS if Bad IN"

| Execution | 1    | 2   | ... | 4   | 5   | 6    | 7    | 8    | 9    | 10   | 11   |
|-----------|------|-----|-----|-----|-----|------|------|------|------|------|------|
| Target    | Auto |     |     |     |     |      |      |      |      |      |      |
| IN        | GNC  | Bad | ... | Bad | Bad | Bad  | Bad  | GNC  | GNC  | GNC  | GNC  |
| BKCAL_IN  | GC   | GC  | ... | GC  | GC  | NI   | NI   | NI   | NI   | NI   | IR   |
| Actual    | Auto | Man | ... | Man | Man | Iman | Iman | Iman | Iman | Iman | Iman |
| OUT       | GC   | IFS | ... | IFS | IFS | IFS  | IFS  | GC   | GC   | GC   | IA   |

**PID 2 – Slave**

| Execution | 1   | 2   | ... | 4   | 5          | 6          | 7         | 8         | 9    | 10  | 11  |
|-----------|-----|-----|-----|-----|------------|------------|-----------|-----------|------|-----|-----|
| Target    | Cas |     |     |     |            |            |           |           |      |     |     |
| CAS_IN    | GC  | IFS | ... | IFS | IFS        | IFS        | GC        | GC        | GC   | IA  | GC  |
| BKCAL_IN  | GC  | GC  | ... | GC  | NI         | NI         | NI        | IR        | GC   | GC  | GC  |
| Actual    | Cas | Cas | ... | Cas | Iman       | Iman       | Iman      | Iman      | Auto | Cas | Cas |
| OUT       | GC  | IFS | ... | IFS | IFS<br>100 | IFS<br>100 | GC<br>100 | IA<br>100 | GC   | GC  | GC  |
| BKCAL_OUT | GC  | GC  | ... | GC  | NI         | NI         | NI        | NI        | IR   | GC  | GC  |

**AO**

FSTATE\_VAL = 100

FSTATE\_TIME = 2 sec

IO\_OPTS = "Fault State to value"

PV\_SCALE (E0%-E100%)=0-100

XD\_SCALE (E0%-E100%)= 4-20

2 sec

|-----|

| Execution | 1   | 2   | 3   | 4         | 5          | 6          | 7          | 8         | 9         | 10  | 11  |
|-----------|-----|-----|-----|-----------|------------|------------|------------|-----------|-----------|-----|-----|
| Target    | Cas |     |     |           |            |            |            |           |           |     |     |
| CAS_IN    | GC  | IFS | IFS | IFS       | IFS<br>100 | IFS<br>100 | IFS<br>100 | GC<br>100 | IA<br>100 | GC  | GC  |
| BKCAL_IN  | GC  | GC  | GC  | GC<br>100 | GC<br>100  | GC<br>100  | GC<br>100  | GC<br>100 | IA<br>100 | GC  | GC  |
| Actual    | Cas | Cas | Cas | LO        | LO         | LO         | LO         | Auto      | Cas       | Cas | Cas |
| OUT       | GC  | GC  | GC  | GC<br>20  | GC<br>20   | GC<br>20   | GC<br>20   | GC<br>20  | GC<br>20  | GC  | GC  |
| BKCAL_OUT | GC  | GC  | GC  | NI        | NI         | NI         | NI         | IR        | GC        | GC  | GC  |

Other link failures in the forward path could force the AO into Fault State since the configuration is as it follows:

Failure 1) **PID\_1.OUT to PID\_2.CAS\_IN**: PID\_2.STATUS\_OPTS = "IFS if Bad CAS\_IN"

Failure 2) **AI\_2.OUT to PID\_2.IN**: PID\_2.STATUS\_OPTS = "IFS if Bad IN"

Failure 3) **PID\_2.OUT to AO.CAS\_IN**: No configuration is required for the AO block to force it to Fault State.

*Note:*

- FSTATE\_VAL has the same engineering unit of SP. Therefore it is using the PV\_SCALE, not the XD\_SCALE.
- When the Resource block forces all output blocks to Fault State, the FSTATE\_TIME is not used.

**Alarms and Events – Alert Processing**

Alarms and events, known as alerts, represent state changes within function block applications. In detection of a significant event, a function block may report this event to an interface device or other field devices.

Alarms refer not only to the comparison between a variable and a limit, but also what is called block alarm, that is used to report errors found in the software and/or hardware during block execution.

For alarms, both entering and exiting alarm condition are defined as an alert state, which will cause a report message to be published onto the network. The time at which the alert state was detected is included as a time stamp in the alert message. The reporting of alerts may be individually suppressed by setting the corresponding priority.

Update event is a mechanism used to inform an interface device that a static parameter was changed so the parameter is read only in this case. It is a much optimized way to keep track of such kind of parameters without doing polling, because these parameters are changed very rarely compared with dynamic parameters.

**a) Alarm parameter (X\_ALM parameter)**

The alarm parameter is provided in a block to capture the dynamic information associated with an alarm. The information contained in the alarm parameter is transferred to an alert object when the alarm is reported. The following fields form the alarm parameter:

- Unacknowledged
- Alarm state
- Time stamp
- Subcode
- Value

Each one of these fields is explained below.

**Unacknowledged**

When it is detected a positive transition of alarm state (entering in alarm active), it will be set to Unacknowledged. This field is set to Acknowledged when the plant operator acknowledges the alarm through the interface device, which is responsible for alarm management.

It is possible to configure the auto-acknowledgement feature for each type of alarm for the block through the ACK\_OPTION parameter. If a positive transition of alarm type happens and the corresponding bit in the ACK\_OPTION is true, then it will not be required an operator plant acknowledgement.

The other form to auto-acknowledge the alarm notice is configuring the alert-priority of the respective alarm to 0, 1 or 2. The alert-priority will be discussed later.

Unacknowledged will have the following enumerations:

- 0 = Undefined
- 1 = Acknowledged
- 2 = Unacknowledged

**Alarm state**

This field gives an indication of whether the alert is active and whether it has been reported. The Alarm State will have the following bit enumeration:

- 1 – Clear-Reported
- 2 – Clear-Not Reported
- 3 – Active-Reported
- 4 – Active-Not Reported

The alarm state is cleared when the block goes to Out of service mode.

**Time stamp**

Time stamp is the time when the change in alarm state was detected that is unreported. This value will be maintained constant until alert confirmation has been received.

**Sub code**

This field contains an enumeration specifying the cause of the alert to be reported.

**Value**

The value of the associated parameter at the time the alert was detected.

**b) Alarm limit (X\_LIM parameter)**

An analog alarm occurs when a value meets or exceeds a limit. For a high alarm, an alarm is true when the analog value is greater than the limit. The status of the alarm remains true until the value drops below the limit minus the alarm hysteresis.

The alarm type can be disabled setting its respective alarm limit parameter to +/- infinity, which is the default of all alarm limits.

The analog parameter compared to alarm limit depends on the block type:  
PID: PV and (PV-SP), regardless of CONTROL\_OPTS.Direct acting  
Analog alarm: PV

AI: OUT  
Setpoint generator: BKCAL\_IN – OUT

**c) Alarm hysteresis (ALARM\_HYS parameter)**

Amount the PV or OUT must return within the alarm limits before the alarm condition clears. Alarm Hysteresis is expressed as a percent of the PV/OUT span.

The span used depends on the block type:  
PID: PV\_SCALE  
AI, Setpoint generator, Analog alarm: OUT\_SCALE

**d) Alert Priority (X\_PRI parameter)**

The alert priority is a parameter, which may be used to configure the priority value assigned to an alarm or event.

The Alert Priority can be:

- 0-1 – The associated alert is not sent as a notification. The priority is above 1, and then the alert must be reported. This priority is auto-acknowledged.
- 2 – Reserved for alerts that do not require the attention of a plant operator. Block alarm and update event have this priority. This priority is auto-acknowledged.
- 3-7 – Advisory Alarms. In this priority is necessary to send an acknowledged.
- 8-15 – Critical Alarms. In this priority is necessary to send an acknowledged.

**e) Alert key (ALERT\_KEY parameter)**

It is an identification number of the plant unit. This information may be used in the host for sorting alarms, etc.

**f) Alarm Summary (ALM\_SUM parameter)**

The parameter Alarm Summary summarizes the status of up to 16 process alarms of the same block. It has four attributes:

- Current Alarms - the Active status of each alarm.
- Unacknowledged - the Unacknowledged status of each alarm.
- Unreported - the Unreported status of each alarm.
- Disabled – it allows Enable/Disable each type of alarm.

**g) ACK\_OPTION parameter**

Selection of whether alarms associated with the block will be automatically acknowledged.

Bit = 0 -> auto-acknowledgment disabled

Bit = 1 -> auto-acknowledgment enabled

| Bit | Description   | Meaning              |                |
|-----|---------------|----------------------|----------------|
|     |               | Function Blocks      | Resource Block |
| 0   | Unack Alarm1  | Discrete alarm       | Write Alm      |
| 1   | Unack Alarm2  | High High alarm      |                |
| 2   | Unack Alarm3  | High alarm           |                |
| 3   | Unack Alarm4  | Low Low alarm        |                |
| 4   | Unack Alarm5  | Low alarm            |                |
| 5   | Unack Alarm6  | Deviation High alarm |                |
| 6   | Unack Alarm7  | Deviation Low alarm  |                |
| 7   | Unack Alarm8  | Block alarm          | Block alarm    |
| 8   | Unack Alarm9  | Not used             |                |
| 9   | Unack Alarm10 | Not used             |                |
| 10  | Unack Alarm11 | Not used             |                |
| 11  | Unack Alarm12 | Not used             |                |
| 12  | Unack Alarm13 | Not used             |                |
| 13  | Unack Alarm14 | Not used             |                |
| 14  | Unack Alarm15 | Not used             |                |
| 15  | Unack Alarm16 | Not used             |                |

#### Association of bit number to alarm type

The correspondent bits for each type of alarm in the ACK\_OPTION are the same of that defined for ALARM\_SUM except for the Resource block. The ACK\_OPTION of the Resource block has a different association of bit number to alarm type from the previous table; its special meaning is described below:

Unack Alarm1 – Writes have been enabled (WRITE\_ALM)

Unack Alarm8 – Block Alarm (BLOCK\_ALM)

#### h) FEATURES\_SEL parameter

It is a resource block parameter that has an element to enable/disable alert report for the whole resource, "report supported".

#### i) CONFIRM\_TIME parameter

A reply is required that confirms receipt of the alert notification. If the reply is not received within a time-out period (CONFIRM\_TIME), the alert will be re-transmitted. It is a resource block parameter, so it is valid for all alert of that resource.

Therefore the alarm parameter is a structured object, which is defined in conjunction with other parameters:

- Enable / disable alarm evaluation:

ALARM\_SUM: enable/disable each alarm type for a specific block.

X\_LIM: it is possible to disable the alarm evaluation by setting the limit to +INF or -INF.

- Enable / disable alarm report:

FEATURES\_SEL: setting the bit "report supported" enables alarm report for the whole resource

X\_PRI: besides configuring the FEATURES\_SEL, it is necessary to set the priority of alarm greater than or equal to 2 to enable the alarm report.

- Auto-acknowledgment:

X\_PRI: the corresponding alarm will be auto-acknowledged if the alarm priority is 0, 1 or 2

ACK\_OPTION: this parameter makes possible to enable/disable auto-acknowledgment for each alarm type, regardless the configuration of X\_PRI.

The information contained in the alarm parameter is transferred to an alert object when the alarm is reported (if it is enabled) to an interface device. As an example, the following parameters are used to configure the Low Low Alarm of the PID block: LO\_LO\_PRI (alarm priority), LO\_LO\_LIM (limit parameter) and LO\_LO\_ALM (alarm parameter), ALARM\_SUM, ACK\_OPTION.



Summarizing, the function block detects the alarm condition. The communication stack is responsible to send the alert report to the interface device, which must reply to device, otherwise the alert report will be re-transmitted after a timeout defined by CONFIRM\_TIME. The alarm configuration (ACK\_OPTION and priority) may require that a plant operator acknowledge it even though the condition has cleared.

#### **j) Example of analog alarm**

It follows an example of AI block alarm configuration, which illustrates how is the alarm processing as well the corresponding alert report.

RS parameter:

FEATURES\_SEL = Reports supported

CONFIRM\_TIME = 640 000 (20 seconds, multiple of 1/32 milliseconds)

AI parameters:

ALERT\_KEY = 12 (this value could be related to a boiler, for example, therefore any alert received by interface device with this alert key means an alert in that boiler)

OUT\_SCALE.EU at 100% = 200

OUT\_SCALE.EU at 0% = 0

HI\_LIM = 190

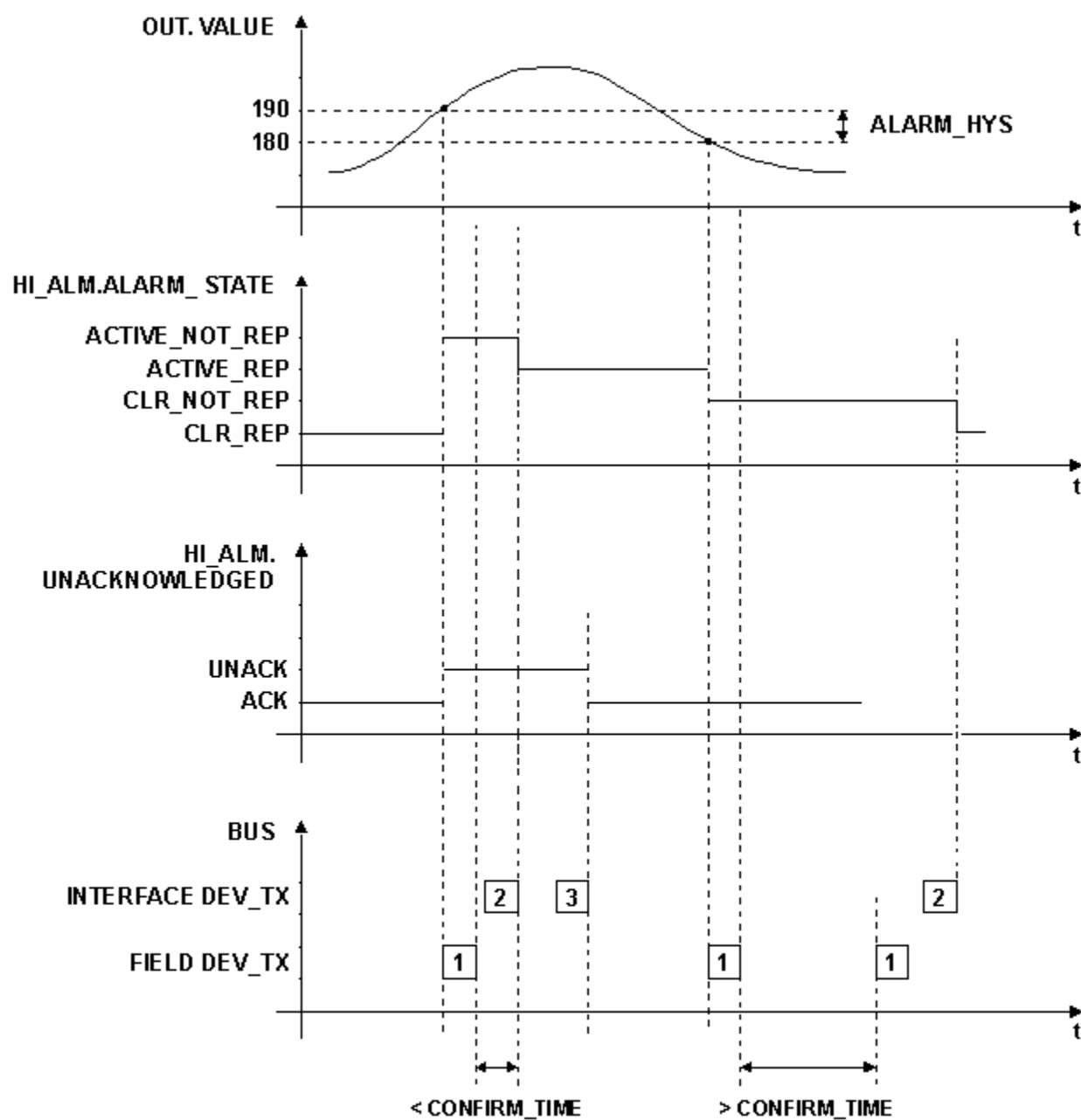
HI\_PRI = 4

ALARM\_HYS = 5%

ALARM\_SUM.DISABLED = Discrete, HiHi, LoLo, Lo, DevHi, DevLo, BlockAlm

ACK\_OPTION = 0x00

Only the high alarm is enabled in ALARM\_SUM.DISABLED and it is disabled the auto-acknowledgement (HI\_PRI=4 and bit reset in ACK\_OPTION), therefore it is necessary an operator plant intervention.



MESSAGE 1: ALERT NOTIFY

MESSAGE 2: ALERT NOTIFY - CONFIRMATION

MESSAGE 3: ALERT ACK (OPERATOR PLANT INTERVENTION)

Fig. 2 – Alert processing

**k) Block alarm (BLOCK\_ALM parameter)**

The block alarm is used for all configuration, hardware, connection failure or system problems in the block. These problems detected by the block algorithm are registered in the BLOCK\_ERR parameter, which is a bit string, so that multiple errors may be shown.

Block error conditions is defined (0= inactive, 1 = active) as follows:

| Bit | Description                                    |
|-----|--|
| 0   | Other (LSB)                                    |
| 1   | Block Configuration Error                      |
| 2   | Link Configuration Error                       |
| 3   | Simulate Active                                |
| 4   | Local Override                                 |
| 5   | Device Fail Safe Set                           |
| 6   | Device Needs Maintenance Soon                  |
| 7   | Input Failure/ process variable has BAD status |
| 8   | Output Failure                                 |
| 9   | Memory Failure                                 |
| 10  | Lost Static Data                               |
| 11  | Lost NV Data                                   |
| 12  | Readback Check Failed                          |
| 13  | Device Needs Maintenance Now                   |
| 14  | Power-up                                       |
| 15  | Out-of-Service (MSB)                           |

The cause of the alert is entered in the Subcode field of BLOCK\_ALM, for example, value 3 means Simulate Active. The first condition to become active will set to Active the Alarm State attribute, if other conditions also become active, no alert will be sent. When all conditions become inactive, then an alert with Clear will be reported.

Block alarm has a fixed priority of 2, therefore it is auto-acknowledged (no operator plant intervention is required).

**l) Update Event (UPDATE\_EVT parameter)**

The update event parameter is provided in a block to capture the dynamic information associated with a write to a static parameter within the block. An update alert object transfers the information contained in the update event parameter when the alert is reported.

The index of the changed parameter (relative to the start of the function block in the OD) and the new static revision level (ST\_REV) are also included in the alert message.

Update event has a fixed priority of 2, therefore it is auto-acknowledged (no operator plant intervention is required).

## Simulation

All input and output class function blocks have a SIMULATE or SIMULATE\_D or SIMULATE\_P parameter, which has a pair of status and values, and an enable switch. This parameter acts as a switch at the interface between the I/O function block and the associated transducer block or hardware channel.

**Enable simulation**

The simulate jumper must be ON to enable simulation in the SIMULATE (SIMULATE\_D or SIMULATE\_P) parameter.

The BLOCK\_ALM and BLOCK\_ERR parameters will show the simulation condition (enable or disable). Such parameters in the Resource Block (RS) will indicate the condition of simulate jumper, while in the Input/Output Function Blocks they will indicate the enable switch condition in the SIMULATE (SIMULATE\_D or SIMULATE\_P) parameter.

**Simulation disabled**

When disabled, the SIMULATE. Simulate Value and Status will track SIMULATE. Transducer Value and Status in order to provide a bumpless transfer from disabled to enabled. The parameter will always initialize to disabled and will be stored in dynamic memory.

**Input Function Block (AI, DI, PUL)**

The SIMULATE. Transducer Status and SIMULATE.Transducer Value come from the transducer block or input channel, and contain what will be sent to the input block if the switch is off (disabled). The SIMULATE.Simulate Value and SIMULATE.Simulate Status are presented to the input block when the enable switch is on, and the transducer block or input channel is ignored. The status can be used to simulate transducer faults. The transducer value and status will always be written with transducer data at each evaluation of the input function block.

| Simulation condition | Action   |
|----------------------|--|
| <b>Enable</b>        | SIMULATE.Simulate Value and Status -> PV (after scaling, linearization and filtering)  |
| <b>Disable</b>       | SIMULATE.Transducer Value and Status -> PV (after scaling, linearization and filtering) and SIMULATE.Simulate Value and Status |

**Output Function Block (AO, DO)**

The SIMULATE.Simulate Value and SIMULATE.Simulate status become the READBACK value and status when the enable switch is on, and the transducer block is ignored. The status can be used to simulate transducer faults. The transducer attribute value and status reflect the transducer readback value and status when simulation is enabled and the transducer maintains last output and ignores the OUT of the Output block.

| Simulation condition | Action  |
|----------------------|---|
| <b>Enable</b>        | SIMULATE.Simulate Value and Status -> READBACK  |
| <b>Disable</b>       | SIMULATE.Transducer Value and Status -> READBACK and SIMULATE.Simulate Value and Status |

## I/O Mapping Forms

The Smar HSE controllers may be classified according to I/O mapping form:

- Numeric Mapping that uses the HC block and CHANNEL parameter.
- ROM Mapping (Remote Operations Management) that uses the ABC block, VAR\_NAME and CHANNEL\_TAG.

### Numeric Mapping

**HC Block (Hardware Configuration Transducer)**

Hardware configuration (definition of I/O module types in the corresponding rack/slot positions) through the HC block (Hardware Configuration Transducer).

There is status indication in the MODULE\_STATUS\_Rx parameters if the I/O module was found during the scan of I/O points.

**CHANNEL parameter configuration**

Addressing of group or I/O point by the transducer block, input function block or output function block through CHANNEL parameter, which has a numeric value.

The CHANNEL parameter configuration depends on the device features as it follows:

**a) Fixed I/O device:** This type of device has a fixed number of I/O. All Smar field devices belong to this class.

The channel is numbered from 1 to the maximum number of I/O.

The DC302 has specific rules to set the CHANNEL parameter as it follows:

- DI and DO Blocks: group A has inputs numbered from 1 to 8 and group B has inputs numbered from 9 to 16.

- MDI and MDO Blocks: the whole group A is selected setting CHANNEL to 1, and group B to 2.

**b) Configurable I/O device:** The user may configure the number of I/O modules as well the I/O type (input or output, discrete, analog, pulse...). The DFI302 is the only device classified as a configurable I/O device.

All I/O modules have the I/O points arranged as it follows:

| Level     | Range                  |
|-----------|------------------------|
| Rack (R)  | 0-14                   |
| Slot (S)  | 0-3                    |
| Group (G) | 0-1                    |
| Point (P) | 0-7<br>9 – whole group |

- Rack (R): Each rack has four slots. The rack is numbered from 0 (first rack) till 14 (last rack). Therefore a single I/O point in the DFI302 may be identified by specifying the rack (R), slot (S), group (G) and point (P). As the CHANNEL parameter in the multiple I/O blocks (MIO) must specify the whole group (8 points), the point must be 9, which mean the whole group.

- Slot (S): One slot supports one I/O module, and it is numbered from 0 (first slot in the rack) till 3 (last slot in the rack).

- Group (G): Ordinal number of group in the specified I/O module, it is numbered from 0 (first group) till number of groups minus 1. The I/O points are arranged in groups of 8 points, regardless how they are grouped for electrical isolation.

If the I/O type is “8- discrete input/4- discrete output”, the inputs belong to the group 0 and the outputs belong to the group 1.

- Point (P): Ordinal number of I/O point in a group, it is numbered from 0 (first point) to 7(last point in the group), and 9 means the whole group of points. The whole group may have 4 or 8 points of I/O.

The value in the CHANNEL parameter is composed by those elements in the following form: RRS GP.

For example, a CHANNEL parameter equals to 1203, it means rack 1, slot 2, group 0 and point 3. If CHANNEL parameter of a MAI block is 10119, it means rack 10, slot 1, group 1 and point 9 (whole group).

Before setting the CHANNEL parameter, it is recommended to configure the hardware in the HC block. Because the write check will verify if the I/O type configured in the HC block is suitable for block type. Therefore setting the CHANNEL parameter of AI block to access an I/O type different of analog input will be rejected.

## ROM Mapping

### Configuration through the ABC block (Component Association Transducer)

In the ROM mapping, all I/O modules require a type of specific transducer block, for example, the DF46 module requires the configuration of its transducer block.

In the hardware configuration through the ABC transducer block, the user defines the expected I/O module type and the expected tag of the transducer block corresponding for each rack/slot, which will be verified by the controller when scanning the I/O points.

The expected I/O module type will be compared with the module ID, while the expected transducer block tag will be compared with transducer blocks' tags of configuration.

The result of this verification will be indicated in the ELEM\_STAT\_x.Element Mismatch parameter. When the ABC block in the ELEM\_STAT\_x.Element Mismatch parameter indicates “Matched”, it means that the corresponding transducer block is reflecting the configuration and the I/O points' values.

Example:

If the configuration is:

ABC.ELEM\_EXPD\_R1\_S1.Expected Tag = “WH99\_SN1-FY100”

ABC.ELEM\_EXPD\_R1\_S1.Expected Type = DF46

Tag of DF46 transducer block = “WH99\_SN1-FY100”

In this case the ABC.ELEM\_STAT\_R1\_S1.Element Mismatch will indicate “Matched” only if:

- The Expected Tag of rack 1 and slot 1 matches with transducer block tag, as in the example above.
- The expected type, the transducer block type found and the module ID match.

If the indication is “Matched”, the DF46 transducer block whose tag is “WH99\_SN1-FY100” will reflect the configuration and the output point values of the module that is in the rack 1 and slot 1.

### **Point mapping through the VAR\_NAME and CHANNEL\_TAG**

The addressing of I/O point by the transducer block or function block is done through the VAR\_NAME and CHANNEL\_TAG.

The I/O transducer blocks have VAR\_NAME (tag) for each point and the function blocks of input or output also have CHANNEL\_TAG (tag) for each point.

When the VAR\_NAME and CHANNEL\_TAG match, thus this I/O point will be reflected in the corresponding function block.

Example:

Continuing the previous example and if the configuration is:

DF46 transducer block whose tag is “WH99\_SN1-FY100” with parameter VAR\_NAMES4[2] = “FY110”

ROMAO.CHANNEL\_TAG = “FY110”

As the VAR\_NAME matches with the CHANNEL\_TAG, so this ROMAO block will can execute in mode different from O/S and will reflect the configuration and the output point value that is in the rack 1 slot 1 on the second point of DF46 module.

### **Configuration of VAR\_NAME parameter**

The following verifications and procedures are performed when the VAR\_NAME parameter is configured:

- 1- Check if the VAR\_NAME value, that will be configured, is unique in the controller scope.
- 2- Reject the writing if the current VAR\_NAME is associated to a function block and the point is Auto.
- 3- Accept the writing if the current VAR\_NAME is associated to a function block, but the point is in manual and undone the association with the CHANNEL\_TAG.
- 4- After writing the new value in the VAR\_NAME, it will look for a CHANNEL\_TAG equals to this new value. If it is an input point it will look for up to 2 equal CHANNEL\_TAGS and with types compatible.

### **Configuration of CHANNEL\_TAG parameter**

The following verifications and procedures are performed when the CHANNEL\_TAG parameter is configured:

- 1- If the current CHANNEL\_TAG is associated to some VAR\_NAME, the association is undone.
- 2- After writing the new value in the CHANNEL\_TAG, automatically is verified if there is an equal, compatible type and available VAR\_NAME (the physical I/O point can be associated to up to two function blocks, and the physical output point can be associated to just one function block).

### **Mapping in the WirelessHART Gateway**

In case of the DF99 and DF100 controllers, which are *WirelessHART* gateway, the TBHG block (Transducer for HART Gateway) has live list function, indicating the *WirelessHART* field devices that are configured for the gateway network.

In addition, the TBHG block indicates if the device on the live list is associated to some TBWH (Transducer for *WirelessHART* device) block that occurs when the device tag and TBWH tag matches. Occuring this association, the corresponding TBWH block will reflect the configuration and the I/O point values of that device.

As well as the transducer blocks of I/O modules, the TBWH block also has a VAR\_NAME for each point, which can be associated to a function block through the procedure described above for VAR\_NAME and CHANNEL\_TAG.

Example:

Tag of TT400 *WirelessHART* = “TIT2020”

Tag of TBWH transducer block = “TIT2020”

In this case the TBHG block will indicate in the LIVE\_LIST\_ST parameter the status “Commissioned”.

```
TBWH.VAR_NAMES9[1] = "TIT2020_1"  
MAI16.CHANNEL_TAG[3] = "TIT2020_1"
```

As the VAR\_NAME matches with the CHANNEL\_TAG, so this MAI16 block will indicate zero in MAP\_MM\_16.Point 3, it means that the CHANNEL\_TAG[3] is associated to a VAR\_NAME. In addition, this input value will be indicated in MAI16.OUTM\_16.Status 3 and MAI16.OUTM\_16.Value 3.

## **Block Instantiation**

Before explaining block instantiation it is better to clarify some concepts:

**Block type:** It is an algorithm to process the input parameters based on the configuration in the contained parameters, then it generates the outputs. It includes also method to write/read the parameters, DD and others. Indeed all this information are stored in Flash memory of device, therefore one device type has a predefined set of block type available in its firmware.

**Block (block instance):** It is a block type associated to a database where the block parameters are stored (RAM and non-volatile memory).

**Action Object:** Through the action object, a block may be instantiated (created) or deleted. Before instantiating a block, it will be checked if the device supports the specified block type as well if there is available RAM and non-volatile memory to store the parameters.

All Smar devices support block instantiation and the Block Library (set of block type) for each type of device is shown in the item "Block type availability and initial block set".

## **Order of Parameters during Download**

Some block parameters have a write check based on the value of others parameters. Such relationships are shown in the block parameter table of each block type in the columns "Valid Range" and "Store/Mode".

It follows the most common parameter relationship used in write check:

- It is required a mode to write the parameter.
- The valid range depends on a scale parameter
- For configurable I/O device, the CHANNEL parameter depends on the hardware configuration in the HC block.

Due to these relationships between parameters in the write check, some times it is necessary to take care about the order of parameters during a download of configuration.

The Smar configuration tool **Syscon** always sends the command to write in the MODE\_BLK parameter as last one for that block, it is enough to avoid a lot of problems in the download. The user must observe the other cases and may change the parameter order easily using drag and drop in **Syscon**.

## Data Type and Data Structure Definition

In this section are defined every data structure and data types used in the system.

| Object Index | Data Type      | Description   |
|--------------|----------------|---|
| 1            | Boolean        | True or false   |
| 2            | Integer8       | 1 byte  |
| 3            | Integer16      | 2 bytes   |
| 4            | Integer32      | 4 bytes   |
| 5            | Unsigned8      | 1 byte  |
| 6            | Unsigned16     | 2 bytes   |
| 7            | Unsigned32     | 4 bytes   |
| 8            | FloatingPoint  | *   |
| 9            | VisibleString  | They are one byte per character, and include the 7 bit ASCII character set. |
| 10           | OctetString    | Octet strings are binary.   |
| 11           | Date           | Date and hour – 7 bytes   |
| 12           | TimeofDay      | Time in millisecond elapsed in the day – 6 bytes                            |
| 13           | TimeDifference | Time difference – 6 bytes   |
| 14           | BitString      | *   |
| 21           | TimeValue      | Integer – 8 bytes<br>It represents the date/hour to synchronize the clock.  |

- Date:
  - Date (3 bytes) – Format#Year#Month#(Week Day and Month Day)
  - Hour (4 bytes) – Format: HH#MM#MSEG
- TimeofDay:
  - Time (4 bytes) – It counted in milliseconds from the Zero hour of the day.
  - Date (3 bytes) – It counted in days related to the January, 1, 1984.
- TimeDifference: The structure is the same of the TimeofDay.
- TimeValue: It used to represent Date and Hour to synchronize the clock. It is an integer with 8 bytes in the base of 1/32 milliseconds.

### Block Structure – DS-64

This data structure consists of the attributes of a block.

| E | Element Name         | Data Type     | Size |
|---|----------------------|---------------|------|
| 1 | Block Tag            | VisibleString | 32   |
| 2 | DD MemberId          | Unsigned32    | 4    |
| 3 | DD ItemId            | Unsigned32    | 4    |
| 4 | DD Revision          | Unsigned16    | 2    |
| 5 | Profile              | Unsigned16    | 2    |
| 6 | Profile Revision     | Unsigned16    | 2    |
| 7 | Execution Time       | Unsigned32    | 4    |
| 8 | Period of Execution  | Unsigned32    | 4    |
| 9 | Number of Parameters | Unsigned16    | 2    |



| E  | Element Name            | Data Type  | Size |
|----|-------------------------|------------|------|
| 10 | Next FB to Execute      | Unsigned16 | 2    |
| 11 | Starting Index of Views | Unsigned16 | 2    |
| 12 | NumberofVIEW_3          | Unsigned8  | 1    |
| 13 | NumberofVIEW_4          | Unsigned8  | 1    |

### Value & Status - Floating Point Structure – DS-65

This data structure consists of the value and status of floating point parameters that are Inputs or Outputs.

| E | Element Name | Data Type | Size |
|---|--------------|-----------|------|
| 1 | Status       | Unsigned8 | 1    |
| 2 | Value        | Float     | 4    |

### Value & Status - Discrete Structure – DS-66

This data structure consists of the value and status of discrete value parameters.

| E | Element Name | Data Type | Size |
|---|--------------|-----------|------|
| 1 | Status       | Unsigned8 | 1    |
| 2 | Value        | Unsigned8 | 1    |

### Scaling Structure – DS-68

This data structure consists of the static data used to scale floating point values for display purposes.

| E | Element Name  | Data Type  | Size |
|---|---------------|------------|------|
| 1 | EU at 100%    | Float      | 4    |
| 2 | EU at 0%      | Float      | 4    |
| 3 | Units Index   | Unsigned16 | 2    |
| 4 | Decimal Point | Integer8   | 1    |

### Mode Structure – DS-69

This data structure consists of bit strings for target, actual, permitted, and normal modes.

| E | Element Name | Data Type | Size |
|---|--------------|-----------|------|
| 1 | Target       | Bitstring | 1    |
| 2 | Actual       | Bitstring | 1    |
| 3 | Permitted    | Bitstring | 1    |
| 4 | Normal       | Bitstring | 1    |

### Access Permissions – DS-70

This data structure consists of access control flags for access to block parameters.

| E | Element Name | Data Type  | Size |
|---|--------------|------------|------|
| 1 | Grant        | Bit String | 1    |
| 2 | Deny         | Bit String | 1    |

**Alarm Float Structure – DS-71**

This data structure consists of data that describes floating point alarms.

| E | Element Name   | Data Type  | Size |
|---|----------------|------------|------|
| 1 | Unacknowledged | Unsigned8  | 1    |
| 2 | Alarm State    | Unsigned8  | 1    |
| 3 | Time Stamp     | Time Value | 8    |
| 4 | Subcode        | Unsigned16 | 2    |
| 5 | Value          | Float      | 4    |

**Alarm Discrete Structure – DS-72**

This data structure consists of data that describes discrete alarms.

| E | Element Name   | Data Type  | Size |
|---|----------------|------------|------|
| 1 | Unacknowledged | Unsigned8  | 1    |
| 2 | Alarm State    | Unsigned8  | 1    |
| 3 | Time Stamp     | Time Value | 8    |
| 4 | Subcode        | Unsigned16 | 2    |
| 5 | Value          | Unsigned8  | 1    |

**Event Update Structure – DS-73**

This data structure consists of data that describes a static revision alarm.

| E | Element Name    | Data Type  | Size |
|---|-----------------|------------|------|
| 1 | Unacknowledged  | Unsigned8  | 1    |
| 2 | Update State    | Unsigned8  | 1    |
| 3 | Time Stamp      | Time Value | 8    |
| 4 | Static Revision | Unsigned16 | 2    |
| 5 | Relative Index  | Unsigned16 | 2    |

**Alarm Summary Structure – DS-74**

This data structure consists of data that summarizes 16 alerts.

| E | Element Name   | Data Type  | Size |
|---|----------------|------------|------|
| 1 | Current        | Bit String | 2    |
| 2 | Unacknowledged | Bit String | 2    |
| 3 | Unreported     | Bit String | 2    |
| 4 | Disabled       | Bit String | 2    |

**Simulate - Floating Point Structure – DS-82**

This data structure consists of simulate and transducer floating point value and status and a simulate enable/disable discrete.

| E | Element Name        | Data Type | Size |
|---|---------------------|-----------|------|
| 1 | Simulate Status     | Unsigned8 | 1    |
| 2 | Simulate Value      | Float     | 4    |
| 3 | Transducer Status   | Unsigned8 | 1    |
| 4 | Transducer Value    | Float     | 4    |
| 5 | Simulate En/Disable | Unsigned8 | 1    |

**Simulate - Discrete Structure – DS-83**

This data structure consists of a simulator, and discrete value and status transducer, and a discrete enable/disable simulator.

| E | Element Name        | Data Type | Size |
|---|---------------------|-----------|------|
| 1 | Simulate Status     | Unsigned8 | 1    |
| 2 | Simulate Value      | Unsigned8 | 4    |
| 3 | Transducer Status   | Unsigned8 | 1    |
| 4 | Transducer Value    | Unsigned8 | 4    |
| 5 | Simulate En/Disable | Unsigned8 | 1    |

### Test Structure – DS-85

This data structure consists of function block test read/write data.

| E  | Element Name | Data Type       | Size |
|----|--------------|-----------------|------|
| 1  | Value1       | Boolean         | 1    |
| 2  | Value2       | Integer8        | 1    |
| 3  | Value3       | Integer16       | 2    |
| 4  | Value4       | Integer32       | 4    |
| 5  | Value5       | Unsigned8       | 1    |
| 6  | Value6       | Unsigned16      | 2    |
| 7  | Value7       | Unsigned32      | 4    |
| 8  | Value8       | FloatingPoint   | 4    |
| 9  | Value9       | VisibleString   | 32   |
| 10 | Value10      | OctetString     | 32   |
| 11 | Value11      | Date            | 7    |
| 12 | Value12      | Time of Day     | 6    |
| 13 | Value13      | Time Difference | 6    |
| 14 | Value14      | Bitstring       | 2    |
| 15 | Value15      | Time Value      | 8    |

### Bitstring32 Alarm - DS-87

| E | Element Name   | Data Type  | Size | Description  |
|---|----------------|------------|------|--|
| 1 | Unacknowledged | Unsigned8  | 1    |  |
| 2 | Alarm State    | Unsigned8  | 1    |  |
| 3 | Time Stamp     | Time Value | 8    |  |
| 4 | Subcode        | Unsigned32 | 4    | Bitstring value of the associated *_Active parameter or'ed with the complement of the associated *_MASK parameter. |
| 5 | Value          | Unsigned8  | 1    | Bit Number from *_ACTIVE   |

### Simulation – Field Diagnostics - DS-89

| E | Element Name              | Data Type | Size | Description   |
|---|---------------------------|-----------|------|---|
| 1 | Diagnostic Simulate Value | Bitstring | 4    | Value used for diagnostic when the simulation is enabled. |
| 2 | Diagnostic Value          | Bitstring | 4    | Diagnostic detected by equipment                          |
| 3 | Enable                    | Unsigned8 | 1    | Enable/disable the simulation                             |

### BitString Structure – DS-158

This data structure consists of the one status and one parameter of 64 discrete value bits.

| E | Element Name | Data Type   | Size |
|---|--------------|-------------|------|
| 1 | Reserved     | OctetString | 3    |
| 2 | Status       | Unsigned8   | 1    |
| 3 | Value        | Bitstring   | 8    |

**Discrete Structure – DS-159**

This data structure consists of one status and eight discrete value parameters.

| E | Element Name | Data Type | Size |
|---|--------------|-----------|------|
| 1 | Status       | Unsigned8 | 1    |
| 2 | Value1       | Unsigned8 | 1    |
| 3 | Value2       | Unsigned8 | 1    |
| 4 | Value3       | Unsigned8 | 1    |
| 5 | Value4       | Unsigned8 | 1    |
| 6 | Value5       | Unsigned8 | 1    |
| 7 | Value6       | Unsigned8 | 1    |
| 8 | Value7       | Unsigned8 | 1    |
| 9 | Value8       | Unsigned8 | 1    |

**Discrete Structure – DS-160**

This data structure consists of one status and sixteen discrete value parameters.

| E  | Element Name | Data Type | Size |
|----|--------------|-----------|------|
| 1  | Status       | Unsigned8 | 1    |
| 2  | Value1       | Unsigned8 | 1    |
| 3  | Value2       | Unsigned8 | 1    |
| 4  | Value3       | Unsigned8 | 1    |
| 5  | Value4       | Unsigned8 | 1    |
| 6  | Value5       | Unsigned8 | 1    |
| 7  | Value6       | Unsigned8 | 1    |
| 8  | Value7       | Unsigned8 | 1    |
| 9  | Value8       | Unsigned8 | 1    |
| 10 | Value9       | Unsigned8 | 1    |
| 11 | Value10      | Unsigned8 | 1    |
| 12 | Value11      | Unsigned8 | 1    |
| 13 | Value12      | Unsigned8 | 1    |
| 14 | Value13      | Unsigned8 | 1    |
| 15 | Value14      | Unsigned8 | 1    |
| 16 | Value15      | Unsigned8 | 1    |
| 17 | Value16      | Unsigned8 | 1    |

**Float Structure – DS-174**

This data structure consists of status and float value of sixteen analog parameters and one general status.

| E     | Element Name  | Data Type   | Size |
|-------|---------------|-------------|------|
| 1     | Reserved      | OctetString | 3    |
| 2     | OverallStatus | Unsigned8   | 1    |
| 3-18  | Status        | Unsigned8   | 1    |
| 19-34 | Value         | Float       | 4    |

**HART Equipment Information - DS-175**

| E | Element Name              | Data Type        | Size | Range              | Description  |
|---|---------------------------|------------------|------|--------------------|--|
| 1 | HART Version              | Unsigned8 – (5)  | 1    |                    |  |
| 2 | NumOfProcessVariables (*) | Unsigned8 – (5)  | 1    | 0 to 8             | Maximum number of digital variables accessed by equipment, from HART_PV up to HART_8V. |
| 3 | Reserved                  | Unsigned8 – (5)  | 1    |                    |  |
| 4 | HART DeviceRevision       | Unsigned8 – (5)  | 1    |                    |  |
| 5 | HART DeviceType           | Unsigned16- (6)  | 2    |                    |  |
| 6 | HART Manufacturer Id      | Unsigned16 – (6) | 2    |                    |  |
| 7 | HART Distributor ID       | Unsigned16- (6)  | 2    |                    |  |
| 8 | Analog Disable (*)        | Unsigned8 – (5)  | 1    | 0=Used<br>1=Unused |  |
| 9 | HART Device Profile       | Unsigned8 – (5)  | 1    |                    |  |

**Expected Element - DS-176**

| E | Element Name             | Data Type          | Size |    | Description   |
|---|--------------------------|--------------------|------|----|---|
| 1 | Expected Tag             | VisibleString [32] | 32   | RW | Transducer block module tag or HART equipment tag.        |
| 2 | Expected Type            | Unsigned32         | 4    | RW | Expected module type.                                     |
| 3 | Expected Minimum Version | Unsigned8          | 1    | RW | Smar equipment does not use                               |
| 4 | Next Redundancy Position | Unsigned8          | 1    | RW | Smar equipment does not use                               |
| 5 | Expected Disable         | Boolean            | 1    | RW | 0 : mismatch alarm enabled<br>1 : mismatch alarm disabled |

**Actual Element - DS-177**

| E | Element Name                       | Data Type          | Size |    | Description  |
|---|------------------------------------|--------------------|------|----|--|
| 1 | Actual Tag                         | VisibleString [32] | 32   | RO | Transducer block module tag or HART equipment tag. |
| 2 | Actual Type                        | Unsigned32         | 4    | RO | Actual module type.                                |
| 3 | Actual Manufacturer Identification | Unsigned32         | 4    | RO | Smar equipment does not use                        |
| 4 | Actual Version                     | Unsigned8          | 1    | RO | Smar equipment does not use                        |

**Element Status - DS-178**

| E | Element Name      | Data Type | Size |    | Description  |
|---|-------------------|-----------|------|----|--|
| 1 | Element Mismatch  | Unsigned8 | 1    | RO | 0: Unknown<br>1: Matched<br>2: Mismatched  |
| 2 | Element Status    | Unsigned8 | 1    | RO | 0: Not Responding (field device)<br>1: Empty (component)<br>2: Failed (component, field device)<br>3: Off-line<br>4: On-line (component, field device) |
| 3 | Redundancy Status | Unsigned8 | 1    | RW | 0: Unknown<br>1: Off-line<br>2: Synchronizing<br>3: Secondary<br>4: Primary<br>8: Go Off-line<br>9: Go Secondary<br>10: Go Primary                     |
| 4 | Element FD Status | Bitstring | 1    | RO | Bit 0 : Fail<br>Bit 1 : Maint<br>Bit 2 : Off-spec<br>Bit 3 : Check   |

## Threshold Analog I/O - DS-179

| E | Element Name | Data Type | Size | Default | Disable |
|---|--------------|-----------|------|---------|---------|
| 1 | OVERRRNG_BAD | Float     | 4    | 20.6    | +INF    |
| 2 | OVERRRNG_UNC | Float     | 4    | 20.45   | +INF    |
| 3 | UNDERRNG_UNC | Float     | 4    | 3.85    | -INF    |
| 4 | UNDERRNG_BAD | Float     | 4    | 3.75    | -INF    |
| 5 | OVERRRNG_LIM | Float     | 4    | 21.0    | +INF    |
| 6 | UNDERRNG_LIM | Float     | 4    | 3.9     | -INF    |

Elements 1 – 4 : used only by the analog input transducer block to determine the input status

Elements 5 – 6 : used only by the analog output transducer block for limiting the output value

## Information of HART Burst Control - DS-183

| E  | Element Name                   | Data Type  | Size | Description  |
|----|--------------------------------|------------|------|--|
| 1  | Publish Control                | Unsigned8  | 1    | 0 : Off<br>1 : Enable Publish on token-passing data link layer only (wired)<br>2 : Enable Publish on TDMA data-link layer only (wireless)<br>3 : Enable Publish on TDMA and token-passing data-link layers |
| 2  | HART Command Expansion         | Unsigned8  | 1    | 250 : unused slot  |
| 3  | Device variable code 0         | Unsigned8  | 1    | 250 : unused slot  |
| 4  | Device variable code 1         | Unsigned8  | 1    | 250 : unused slot  |
| 5  | Device variable code 2         | Unsigned8  | 1    | 250 : unused slot  |
| 6  | Device variable code 3         | Unsigned8  | 1    | 250 : unused slot  |
| 7  | Device variable code 4         | Unsigned8  | 1    | 250 : unused slot  |
| 8  | Device variable code 5         | Unsigned8  | 1    | 250 : unused slot  |
| 9  | Device variable code 6         | Unsigned8  | 1    | 250 : unused slot  |
| 10 | Device variable code 7         | Unsigned8  | 1    | 250 : unused slot  |
| 11 | Publish data message ref       | Unsigned8  | 1    |  |
| 12 | Max publish count              | Unsigned8  | 1    |  |
| 13 | HART Command Number            | Unsigned16 | 2    |  |
| 14 | Reserved                       | Unsigned16 | 2    |  |
| 15 | Minimum update period (sec)    | Float      | 4    |  |
| 16 | Maximum update period (sec)    | Float      | 4    |  |
| 17 | Trigger mode                   | Unsigned8  | 1    | 0 : Continuous or Normal<br>1 : Window or Delta Save<br>2 : High<br>3 : Low<br>4 : On Change   |
| 18 | Device variable classification | Unsigned8  | 1    |  |
| 19 | HART unit                      | Unsigned8  | 1    |  |
| 20 | Reserved                       | Unsigned8  | 1    |  |
| 21 | Trigger level                  | Float      | 4    |  |

## Indication Alarm of Foreign Protocol - DS-184

| E | Element Name     | Data Type (Index)   | Size      |
|---|------------------|---------------------|-----------|
| 1 | Unacknowledged   | Unsigned8 – (5)     | 1         |
| 2 | Alarm State      | Unsigned8 – (5)     | 1         |
| 3 | Time Stamp       | Time Value – (21)   | 8         |
| 4 | Response Code    | Unsigned16 – (6)    | 2         |
| 5 | Response Message | Octet String – (10) | 256       |
|   |                  |                     | Total 268 |

Unacknowledged =  
0=Undefined  
1=Acknowledged  
2=Unacknowledged

Alarm State=  
 0=Undefined  
 1=Update - reported  
 2=Update – not reported

Response Code  
 0=Undefined  
 1=Successful  
 2=Foreign Protocol Communications Timed-Out  
 3=Foreign Protocol Communications Failure: Tag Not Found  
 4=Foreign Protocol Communications Failure: Unrecoverable Errors  
 5=Foreign Protocol Communications Failure: Interface Not Responding

## Manufacturer Specific Data Structure

In this section are defined manufacturer specific data structure used in the system.

### Scaling Conversion Structure - DS-256

This data structure consists of data used to generate constants A and B in equation  $Y = A \cdot X + B$ .

| E | Element Name | Data Type | Size |
|---|--------------|-----------|------|
| 1 | From EU 0%   | Float     | 4    |
| 2 | From EU 100% | Float     | 4    |
| 3 | To EU 0%     | Float     | 4    |
| 4 | To EU 100%   | Float     | 4    |
| 5 | Data Type    | Unsigned8 | 1    |

### Scaling Conversion Structure with Status - DS-257

This data structure consists of data used to generate constants A and B in equation  $Y = A \cdot X + B$  plus the output status.

| E | Element Name  | Data Type | Size |
|---|---------------|-----------|------|
| 1 | From EU 0%    | Float     | 4    |
| 2 | From EU 100%  | Float     | 4    |
| 3 | To EU 0%      | Float     | 4    |
| 4 | To EU 100%    | Float     | 4    |
| 5 | Data Type     | Unsigned8 | 1    |
| 6 | Output Status | Unsigned8 | 1    |

- Output Status: This status can be set in two ways: by Modbus master or by status chosen by user. In the first case the Modbus master sends the value and status (i.e., two different Modbus variables). In the second case the master sends only the value.

For more details consult the description in the MBCS block (Section 2).

### Scaling Locator Structure - DS-258

This data structure consists of data used to generate constants A and B in equation  $Y = A \cdot X + B$  plus the addresses in a slave device.

| E | Element Name            | Data Type  | Size |
|---|-------------------------|------------|------|
| 1 | From EU 0%              | Float      | 4    |
| 2 | From EU 100%            | Float      | 4    |
| 3 | To EU 0%                | Float      | 4    |
| 4 | To EU 100%              | Float      | 4    |
| 5 | Data Type               | Unsigned8  | 1    |
| 6 | Slave Address           | Unsigned8  | 1    |
| 7 | Modbus Address of Value | Unsigned16 | 2    |

- Slave Address: It informs the slave address which is required to reference to the PVALUEn parameter. For example, it suppose there is one LC700 with device address equal 3 and In this equipment is required to monitor one specific variable. Thus, this Slave Address is 3.

- Modbus Address of Value: It informs the Modbus address of variable which it will be monitored. In the example of the last element, it supposes the Modbus address of the monitored variable was 40032. Thus, element must receive this address.

### Scaling Locator Structure with Status- DS-259

This data structure consists of data used to generate constants A and B in equation  $Y = A * X + B$  plus the addresses in a slave device.

| E | Element Name             | Data Type  | Size |
|---|--------------------------|------------|------|
| 1 | From EU 0%               | Float      | 4    |
| 2 | From EU 100%             | Float      | 4    |
| 3 | To EU 0%                 | Float      | 4    |
| 4 | To EU 100%               | Float      | 4    |
| 5 | Data Type                | Unsigned8  | 1    |
| 6 | Slave Address            | Unsigned8  | 1    |
| 7 | Modbus Address of Value  | Unsigned16 | 2    |
| 8 | Modbus Address of Status | Unsigned16 | 2    |

- Slave Address: It informs the slave address which is required to reference to the PVALUEN parameter. For example, it suppose there is one LC700 with device address equal 3 and In this equipment is required to monitor one specific variable. Thus, this Slave Address is 3;
- Modbus Address of Value: It informs the Modbus address of variable which it will be monitored. In the example of the last element, it supposes the Modbus address of the monitored variable was 40032. Thus, element must receive this address.
- Modbus Address of Status: In this parameter, the user informs the Modbus address which the status will be read or write. Each input and output has one correspondent status. The status interpretation follows the Foundation Fieldbus Default (See the item "Parameter Status" for more details).

### Modbus Variable Locator Structure - DS-260

This data structure consists of data indicating the addresses in a slave device.

| E | Element Name            | Data Type  | Size |
|---|-------------------------|------------|------|
| 1 | Slave Address           | Unsigned8  | 1    |
| 2 | Modbus Address of Value | Unsigned16 | 2    |

- Slave Address: It indicates the Slave Address which the required variable to be monitored is located. For example, if in an application one LC700 was adjusted with Device Address 1, the Slave Address must be 1.
- Modbus Address Value: It writes the Modbus address of the variable which will be monitored in the MBSM block. It supposes the user needs to monitor the variable with MODBUS Address 40001 located in one Slave I/O module with Device Address 1. Thus, the Modbus Address of Value must be 40001.

### Modbus Variable Locator Structure with Status- DS-261

This data structure consists of data indicating the addresses in a slave device.

| E | Element Name             | Data Type  | Size |
|---|--------------------------|------------|------|
| 1 | Slave Address            | Unsigned8  | 1    |
| 2 | Modbus Address of Value  | Unsigned16 | 2    |
| 3 | Modbus Address of Status | Unsigned16 | 2    |

### FF Parameter ID Structure - DS-262

This data structure consists of data informing the position of the FF parameter requested.

| E | Element Name   | Data Type         | Size |
|---|----------------|-------------------|------|
| 1 | Block Tag      | VisibleString(32) | 32   |
| 2 | Relative Index | Unsigned16        | 2    |
| 3 | Sub Index      | Unsigned8         | 1    |



- **Block Tag:** It informs the Block Tag that contains the variable which is required to visualization. For example, the user needs to monitor the gain value from the PID block. Thus, it inserts the PID Tag Block which contained the gain parameter required to be visualized in the Modbus Master.
- **Relative Index:** It is the parameter index of a function block which it desired to monitor. (See the function block parameter tables). Thus, it inserts the relative index to the desired parameter to be monitored. In the case above, to monitor the gain parameter from the ID relative block, the relative index is 23.
- **Sub Index:** The Sub Index is used for parameters which posses structure. In this case, it is necessary to indicate which structure element is being referred.

## Slave Address Structure - DS-263

This data structure consists of data informing the IP address and the Modbus address of the slaves.

| E  | Element Name   | Data Type         | Size |
|----|----------------|-------------------|------|
| 1  | IP Slave1      | VisibleString(16) | 16   |
| 2  | IP Slave2      | VisibleString(16) | 16   |
| 3  | IP Slave3      | VisibleString(16) | 16   |
| 4  | IP Slave4      | VisibleString(16) | 16   |
| 5  | IP Slave5      | VisibleString(16) | 16   |
| 6  | IP Slave6      | VisibleString(16) | 16   |
| 7  | IP Slave7      | VisibleString(16) | 16   |
| 8  | IP Slave8      | VisibleString(16) | 16   |
| 9  | Slave Address1 | Unsigned8         | 1    |
| 10 | Slave Address2 | Unsigned8         | 1    |
| 11 | Slave Address3 | Unsigned8         | 1    |
| 12 | Slave Address4 | Unsigned8         | 1    |
| 13 | Slave Address5 | Unsigned8         | 1    |
| 14 | Slave Address6 | Unsigned8         | 1    |
| 15 | Slave Address7 | Unsigned8         | 1    |
| 16 | Slave Address8 | Unsigned8         | 1    |



## BLOCK LIBRARY

This chapter describes in detail the library of Smar FOUNDATION fieldbus functional blocks, structured as described below.

First two tables are shown, which describe the general characteristics of each block grouped according to the class:

- Description of Block Types: Describes each block available in blocks library.
- Block Type Availability and Initial Block Set: Shows the supported blocks for each device Smar and the pre-instantiated values.

Subsequently, each function block is detailed. For each block, the following information are supplied:

- Overview: general description of the block;
- Description: describes all the characteristics of the block;
- BLOCK\_ERR: indicates the possible signals of the BLOCK\_ERR parameter and the possible causes. For more details see the section "Alarms and Events - Alert Processing and Simulation" in the Chapter 1;
- Supported Modes: indicates the block supported modes. For more details see the section "Mode Parameter" in the Chapter 1;
- Parameters: table with all the parameters of the block. For each parameter the following characteristics are detailed:

|                            |   |
|----------------------------|---|
| <b>Idx</b>                 | Relative index of the parameter.  |
| <b>Parameter</b>           | Parameter tag (the same of the block device description (DD)).  |
| <b>Data Type (Length)</b>  | Supported data type:<br><b>Name</b> – Simple variable or array.<br><b>DS-n</b> – Data structure (Record) of index n.<br>The value between parentheses represents the size in bytes of the parameter. For more details see the section "Data Type and Data Structure Definition" in the Chapter 1.   |
| <b>Valid Range/Options</b> | Determine the valid range of values or the options that can be written in the parameter.  |
| <b>Default Value</b>       | Parameter value when the block is initialized the first time.   |
| <b>Units</b>               | Represents the parameter unit. All parameters with the same unit have the same relation within the block. The units can be PV, OUT, FF, TRK and XD. For example, the SP and PV parameters of PID block have the same PV unit, i.e., both have the same range of PV_SCALE. The units Sec (seconds), % (percent) and On/Off (discrete state) are constant units. "None" is used for numbers that don't have specific units, and "na" is used for bit strings where units don't apply.   |
| <b>Store/ Mode</b>         | This column has two information: <ul style="list-style-type: none"> <li>• The "store" where the parameter is stored: Static, Non-volatile or Dynamic. See Parameter Storage in chapter 1; and</li> <li>• The "mode" minimum to change the parameter. It can be: <ul style="list-style-type: none"> <li>○ "No mode" - indicates that the parameter don't depend on the mode to be change;</li> <li>○ "RO" (Read-Only) – indicates that the parameter is read only;</li> <li>○ "Auto", "Man" or "OOS" – indicates the minimum Target mode that the block must be to change the parameter. The minimum means that if the block is in a mode with priority higher than the minimum, it is possible to change. For example, the OOS mode is the mode with higher priority. Therefore in OOS mode is possible to change all the parameters. The MAN mode has priority in relation to the Auto mode, and so forth. For more information about the priorities of the modes, see the table in the section "Mode Parameter" item c, chapter 1.</li> </ul> </li> </ul> |
| <b>Description</b>         | A brief description of each parameter of the block and its use.   |

## Description of Block Types

| BLOCK CLASS                             | BLOCK TYPE | MAPPING [1] | DESCRIPTION  |
|---|------------|-------------|--|
| RESOURCE                                | RS         | -           | <b>RESOURCE</b> – This block contains data that is specific to the hardware that is associated to the resource.  |
|   | RS2        | -           | <b>RESOURCE</b> – This block contains data that is specific to the hardware that is associated to the resource with Field Diagnostics parameters.  |
| TRANSDUCER BLOCKS                       | DIAG       | -           | <b>DIAGNOSTICS TRANSDUCER</b> – It provides online measurement of block execution time, check of links between blocks and other features   |
|   | TEMP       | NUM         | <b>DF45 TEMPERATURE TRANSDUCER</b> – This is the transducer block for the module DF45, an eight low level input module for RTD, TC, mV and Ohm.  |
|   | HC         | NUM         | <b>HARDWARE CONFIGURATION TRANSDUCER</b> – It configures the module type for each slot in the DFI302.  |
|   | TRDRED     | -           | <b>REDUNDANCY TRANSDUCER</b> - This block provides information about redundant devices. It allows verify the actual status and data related to the devices that work in redundancy.  |
|   | ABC        | ROM         | <b>ASSOCIATION BLOCK FOR COMPONENT</b> – It configures the module type for each slot using the ROM technology.   |
|   | TB8BI      | ROM         | <b>TB8BI</b> –Generic term for all transducers of modules with 8 digital inputs.   |
|   | TB8BO      | ROM         | <b>TB8BO</b> –Generic term for all transducers of modules with 8 digital outputs.  |
|   | TB8AI      | ROM         | <b>TB8AI</b> –Generic term for all transducers of modules with 8 analog inputs.  |
|   | TB4AO      | ROM         | <b>TB4AO</b> –Generic term for all transducers of modules with analog outputs.   |
|   | TBHG       | ROM         | <b>TRANSDUCER BLOCK FOR HART GATEWAY</b> – Transducer block for <i>WirelessHART</i> gateway.   |
|   | TBWH       | ROM         | <b>TRANSDUCER BLOCK FOR WIRELESSHART</b> – HSE WIO transducer block used to map each HART and <i>WirelessHART</i> field device.  |
| INPUT FUNCTION BLOCKS                   | AI         | NUM         | <b>ANALOG INPUT</b> – This block takes the analog input data from the analog input signal and it makes available to other function blocks. It has scaling conversion, filtering, square root, low cut and alarm processing.                        |
|   | EAI        | NUM         | <b>ENHANCED ANALOG INPUT</b> – This block has all the parameters of the AI block. It has an additional feature for alarm output (OUT_ALM parameter) to be used in control logic.   |
|   | DI         | NUM         | <b>DISCRETE INPUT</b> – This block takes the discrete input data from the discrete input signal, and it makes available to other function blocks. It has option to invert, filtering and alarm processing.   |
|   | MAI        | NUM         | <b>MULTIPLE ANALOG INPUT</b> – It provides a way to receive 8 analog variables from other modules or physical inputs.  |
|   | MDI        | NUM         | <b>MULTIPLE DISCRETE INPUT</b> – It provides a way to receive 8 discrete variables from other modules or physical inputs.  |
|   | PUL        | NUM         | <b>PULSE INPUT</b> – It provides an analog value that represents a totalization of pulses in a physical discrete input.  |
|   | ROMAI      | ROM         | <b>ROMAI ANALOG INPUT</b> - This block has all characteristics of the standard AI block in addition the features related to alarm and IO mapping via CHANNEL_TAG.  |
|   | MAI16      | ROM         | <b>MULTIPLE ANALOG INPUTS 16</b> – It provides to the FOUNDATION fieldbus network 16 analog input variables in a single output parameter.  |
|   | MBI64      | ROM         | <b>MULTIPLE BINARY INPUTS 64</b> – It provides to the FOUNDATION fieldbus network 64 input binary variables in a single output parameter.  |
| CONTROL AND CALCULATION FUNCTION BLOCKS | PID        | -           | <b>PID CONTROL</b> – This standard block has a lot of valuables features as setpoint treatment (value and rate limiting), filtering and alarm on PV, feedforward, output tracking and others.  |
|   | EPID       | -           | <b>ENHANCED PID</b> – It has all the standard features plus: bumpless or hard transfer from a “manual” mode to an “automatic” mode and bias.   |
|   | APID       | -           | <b>ADVANCED PID</b> – It has all the standard features plus: bumpless or hard transfer from a “manual” mode to an “automatic” mode, bias. Adaptive gain, PI sampling, dead band for error, special treatment for error, ISA or parallel algorithm. |

|  |       |   |   |
|--|-------|---|---|
|  | ARTH  | - | <b>ARITHMETIC</b> – This calculation block provides some pre-defined equations ready for use in applications as flow compensation, HTG, ratio control and others.   |
|  | SPLT  | - | <b>SPLITTER</b> – This block is used in two typical applications: split ranging and sequencing. It receives the output of PID block, which is processed according to the selected algorithm, and then it generates the values for the two analog output blocks.                         |
|  | CHAR  | - | <b>SIGNAL CHARACTERIZER</b> – It has capability for two signal characterization based on the same curve. The second input has an option for swapping “x” to “y”, providing an easy way to use the inverse function, which may be used in signal characterization of readback variables. |
|  | ECHAR | - | <b>ENHANCED SIGNAL CHARACTERIZER</b> – It has all the characteristics of the CHAR block, and allows cascade operation.  |
|  | INTG  | - | <b>INTEGRATOR</b> – It integrates a variable in function of the time. There is a second flow input that may be used for the following applications: net flow totalization, volume/mass variation in vessels and precise flow ratio control.   |
|  | AALM  | - | <b>ANALOG ALARM</b> – This alarm block has dynamic or static alarm limits, hysteresis, and temporary expansion of alarm limits on step setpoint changes to avoid nuisance alarms, two levels of alarm limits and delay for alarm detection.   |
|  | EAALM | - | <b>ENHANCED ANALOG ALARM</b> – It has all characteristics of the AALM block, and also provides one output when the input has not usable status and it allows the inversion of the outputs.  |
|  | ISEL  | - | <b>INPUT SELECTOR</b> – This block has four analog inputs that may be selected by an input parameter or according to a criterion as first good, maximum, minimum, middle and average.   |
|  | SPG   | - | <b>SETPOINT RAMP GENERATOR</b> – This block generates setpoint following a profile in function of the time. Typical applications are temperature control, batch reactors, etc.  |
|  | ESPG  | - | <b>ENHANCED SETPOINT RAMP GENERATOR</b> – It has an extra parameter to identify the step or segment of the profile in the float format.   |
|  | TIME  | - | <b>TIMER and Logic</b> – This block has four discrete inputs that are processed by combination logic. The selected timer processing type operates on the combined input signal to produce a measurement, delay, extension, pulse or debounce.   |
|  | LLAG  | - | <b>LEAD-LAG</b> – This block provides dynamic compensation of a variable. It is used normally in a feed forward control.  |
|  | OSDL  | - | <b>OUTPUT SELECTOR / DYNAMIC LIMITER</b> – It has two algorithms:<br>Output selector – selection of output by a discrete input<br>Dynamic limiter – this algorithm was developed specially for double cross limit in combustion control.  |
|  | DENS  | - | <b>DENSITY</b> – This block has a special algorithm to calculate the density in different types of engineering units: plato degree, INPM and others.  |
|  | CTRW  | - | <b>CONSTANT AND CONTAINED RW</b> - The Constant function block generates constant values to use in input parameters of other blocks. It can also read/write in contained parameters of other blocks into the same device.   |
|  | FFET  | - | <b>FLIP-FLOP AND EDGE TRIGGER</b> – It can be configured to work as SR flip-flop, RS flip-flop, D-LATCH and EDGE TRIGGER (rising, falling or bi-directional)  |
|  | AEQU  | - | <b>ADVANCED EQUATIONS</b> - This block was specially designed to support specific calculations.   |
|  | PRED  | - | <b>SMITH PREDITOR</b> –This block becomes available the following functions as Delay, Bypass and Smith Predictor.   |
|  | TF    | - | <b>TRANSFER FUNCTION</b> – This block is intended to represent systems up to 2nd order, using the coefficients A, B, C, D, E and F.   |
|  | LCF   | - | <b>LIQUID CORRECTION FACTORS</b> – This block calculates the correction factors (CTL, CPL and BSW in operation temperature) for liquid measurement.   |
|  | FMTH  | - | <b>FLEXIBLE MATHEMATICAL</b> – This block allows execute mathematical expressions generated by the user with inputs, outputs and auxiliary variables, and also including conditional expressions.   |

|                                     |                  |     |  |
|-------------------------------------|------------------|-----|--|
|                                     | <b>ADT</b>       | -   | <b>ANALOG DATA TRANSFER</b> - This block obtains the analog values and status from the inputs and makes available for the outputs. Optionally the block can be configured to generate in its outputs constant values to be used in the inputs of other blocks.   |
|                                     | <b>DDT</b>       | -   | <b>DISCRETE DATA TRANSFER</b> - This block obtains the discrete values from the inputs and makes available for the outputs. Optionally the block can be used to make available the constants for the outputs.  |
| <b>FLEXIBLE FUNCTION BLOCK 1131</b> | <b>FFB_1131</b>  | -   | <b>FFB_1131</b> – It has as main purpose to interconnect the ladder logic (usually used for discrete control strategies) and continuous control system, configured through function blocks.  |
|                                     | <b>FFB2_1131</b> | -   | <b>FFB2_1131</b> – It has as main purpose to interconnect the ladder logic (usually used for discrete control strategies) and continuous control system, configured through function blocks.   |
| <b>MODBUS FUNCTION BLOCKS</b>       | <b>MBCF</b>      | -   | <b>MODBUS CONFIGURATION</b> – This transducer block is used to configure general features related to Modbus gateway.   |
|                                     | <b>MBCS</b>      | -   | <b>MODBUS CONTROL SLAVE</b> – When the device is working as gateway between Foundation Fieldbus and Modbus (slave device), this block may be used to exchange control data between both protocols.   |
|                                     | <b>MBSS</b>      | -   | <b>MODBUS SUPERVISION SLAVE</b> – When the device is working as gateway between Foundation Fieldbus and Modbus (slave device), this block may be used to convert Foundation Fieldbus parameters into Modbus variables. Such variables will be available to the supervisory with a Modbus driver.                       |
|                                     | <b>MBCM</b>      | -   | <b>MODBUS CONTROL MASTER</b> – When the device is working as gateway between Foundation Fieldbus and Modbus (master device), this block may be used to exchange control data between both protocols.   |
|                                     | <b>MBSM</b>      | -   | <b>MODBUS SUPERVISION MASTER</b> – When the device is working as gateway between Foundation Fieldbus and Modbus (master device), this block may be used to convert Modbus variables into Foundation Fieldbus parameters. Such parameters will be available to the supervisory with a Foundation Fieldbus driver (OPC). |
| <b>OUTPUT FUNCTION BLOCKS</b>       | <b>AO</b>        | NUM | <b>ANALOG OUTPUT</b> – The AO block provides an analog value to generate an analog output signal. It provides value and rate limiting, scaling conversion, fault state mechanism and other features.   |
|                                     | <b>DO</b>        | NUM | <b>DISCRETE OUTPUT</b> – The DO block provides a discrete value to generate a discrete output signal. There is option to invert the discrete value, fault state mechanism and other features.  |
|                                     | <b>MAO</b>       | NUM | <b>MULTIPLE ANALOG OUTPUT</b> – It provides a way to send 8 analog variables to other modules or physical outputs.   |
|                                     | <b>MDO</b>       | NUM | <b>MULTIPLE DISCRETE OUTPUT</b> – It provides a way to send 8 discrete variables to other modules or physical outputs.   |
|                                     | <b>STEP</b>      | NUM | <b>STEP OUTPUT PID</b> – It is used when the final control element has an actuator driven by an electric motor.  |
|                                     | <b>ROMAO</b>     | ROM | <b>ROMAO ANALOG OUTPUT</b> - This block has all characteristics of the standard AO block in addition the features related to alarm and IO mapping via CHANNEL_TAG.   |
|                                     | <b>MAO16</b>     | ROM | <b>MULTIPLE ANALOG OUTPUT 16</b> – It receives from the FOUNDATION fieldbus network 16 analog output variables in a single input parameter.  |
|                                     | <b>MBO64</b>     | ROM | <b>MULTIPLE BINARY OUTPUT 64</b> - It receives from the FOUNDATION fieldbus network 64 binary output variables in a single input parameter   |

[1] Legend:

NUM : exclusive block for controller with Numeric Mapping

ROM : exclusive block for controller with ROM Mapping

- : block supported by controller with Numeric or ROM Mapping

## Block Type Availability and Initial Block Set

The table below shows the list of supported blocks for each product. Read carefully the notes in order to fully understand the information in this table.

| Block Class                             | Block Type | DF62 | DF63 | DF73 | DF75 | DF79 | DF81 | DF89 | DF95 | DF97 | DF99 | DF100 |
|---|------------|------|------|------|------|------|------|------|------|------|------|-------|
| Resource                                | RS (1)     | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |      |       |
|   | RS2 (1)    |      |      |      |      |      |      |      |      |      | 1    | 1     |
| Transducer Blocks                       | DIAG (1)   | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1     |
|   | TEMP       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | HC (1)     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | TRDRED (1) | 1    | 1    | 1    | 1    |      |      | 1    |      |      |      |       |
|   | ABC        |      |      |      |      |      |      |      |      |      | 1    |       |
|   | TB8BI      |      |      |      |      |      |      |      |      |      | 0    |       |
|   | TB8BO      |      |      |      |      |      |      |      |      |      | 0    |       |
|   | TB8AI      |      |      |      |      |      |      |      |      |      | 0    |       |
|   | TB4AO      |      |      |      |      |      |      |      |      |      | 0    |       |
|   | TBHG (1)   |      |      |      |      |      |      |      |      |      | 1    | 1     |
|   | TBWH       |      |      |      |      |      |      |      |      |      | 0    | 0     |
| Input Function Blocks                   | AI         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | EAI        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | DI         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | MAI        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | MDI        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | PUL        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | MAI16      |      |      |      |      |      |      |      |      |      | 0    | 0     |
|   | ROMAI      |      |      |      |      |      |      |      |      |      | 0    | 0     |
|   | MBI64      |      |      |      |      |      |      |      |      |      | 0    |       |
| Control and Calculation Function Blocks | PID        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | EPID       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | APID       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | ARTH       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | SPLT       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | CHAR       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | ECHAR      | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | INTG       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | AALM       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | EAALM      | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | ISEL       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | SPG        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | ESPG       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | TIME       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | LLAG       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | OSDL       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | DENS       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | CTRW       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|   | FFET       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | AEQU       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|   | PRED       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |

| Block Class                  | Block Type    | DF62 | DF63 | DF73 | DF75 | DF79 | DF81 | DF89 | DF95 | DF97 | DF99 | DF100 |
|------------------------------|---------------|------|------|------|------|------|------|------|------|------|------|-------|
|                              | TF            | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | LCF           | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | FMTH          | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | ADT           | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|                              | DDT           | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| Flexible Function Block 1131 | FFB_1131 (1)  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | FFB2_1131 (1) | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
| Modbus Function Blocks       | MBCF (1)      | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
|                              | MBCS (16)     | 0    | 0    |      |      |      |      |      |      |      | 0    | 0     |
|                              | MBCM (16)     | 0    | 0    |      |      |      |      |      |      |      | 0    | 0     |
|                              | MBSM (16)     | 0    | 0    |      |      |      |      |      |      |      | 0    | 0     |
| Output Function Blocks       | AO            | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | DO            | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | MAO           | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | MDO           | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | STEP          | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |       |
|                              | ROMAO         |      |      |      |      |      |      |      |      |      | 0    |       |
|                              | MAO16         |      |      |      |      |      |      |      |      |      | 0    |       |
|                              | MBO64         |      |      |      |      |      |      |      |      |      | 0    |       |

**Note 1** – The column “Block type” indicates which block type is available for each type of device.

**Note 2** – The number associated to the block type and the device type is the number of instantiated blocks during the factory initialization.

**Note 3** – If the function block type is not available to the device type, it will be indicated by blank space.

**Note 4** – The column Block type shows the mnemonics, if it is followed by a number between parentheses, it indicates the maximum number of block instances.

**Note 5** – To know the firmware versions of controllers which support the FFB\_1131 and FFB2\_1131 blocks, consult the description of block in “Flexible Function Block 1131 – Chapter 2”.

**Note 6** – The following transducer blocks represents generically a set of specific transducer blocks for each type of I/O module.

TB8BI represents the transducer of DF16.

TB8BO represents the transducer of DF25.

TB8AI represents the transducer of DF44.

TB4AO represents the transducer of DF46.



## Resource

### RS – Resource Block

#### Description

This block contains data that is specific to the hardware that is associated with the resource. All data is modeled 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 part of their respective transducer blocks.

The mode is used to control major states of the resource. O/S mode stops all function block execution. The actual mode of the function blocks will be changed to O/S, but the target mode will not be changed. Auto mode allows normal operation of the resource. IMan shows that the resource is initializing or receiving a software download.

Parameters MANUFAC\_ID, DEV\_TYPE, DEV\_REV, DD\_REV, and DD\_RESOURCE are required to identify and locate the DD so that Device Description Services can select the correct DD for use with the resource.

The parameter HARD\_TYPES is a read only BitString that indicates the types of hardware that are available to this resource. If an I/O block is configured that requires a type of hardware that is not available, the result will be a block alarm for a configuration error.

The RS\_STATE parameter contains the operational state of the Function Block Application for the resource containing this resource block.

#### RESTART parameter

The RESTART parameter allows degrees of initialization of the resource. They are:

- 1 - Run: it is the passive state of the parameter
- 2 - Restart resource: it is intended to clear up problems like garbage collection
- 3 - Restart with defaults: it is intended to wipe configuration memory; it works like a factory initialization.
- 4 - Restart processor: it provides a way to hit the reset button on the processor associated with the resource

This parameter does not appear in a view because it returns to 1 shortly after being written.

#### Non-volatile parameters

The Smar devices do not support cyclic saving of non-volatile parameters to a non-volatile memory, therefore NV\_CYCLE\_T parameter will always be zero, which means not supported feature.

On the other hand, the Smar devices have a mechanism to save non-volatile parameters into a non-volatile memory during the power down, and they will be recovered in the power up.

#### Timeout for remote cascade modes

SHED\_RCAS and SHED\_ROUT set the time limit for loss of communication from a remote device. These constants are used by all function blocks that support a remote cascade mode. The effect of a timeout is described in Mode Calculation. Shedding from RCAS/ROUT shall not happen when SHED\_RCAS or SHED\_ROUT is set to zero.

#### Alert Notification

The MAX\_NOTIFY parameter value is the maximum number of alert reports that this resource can have sent without getting a confirmation, corresponding to the amount of buffer space available for alert messages. A user can set the number lower than that, to control alert flooding, by adjusting the LIM\_NOTIFY parameter value. If LIM\_NOTIFY is set to zero, then no alerts are reported. The CONFIRM\_TIME parameter is the time for the resource to wait for confirmation of receipt of a report before trying again. If the CONFIRM\_TIME = 0 the device shall not retry.

#### FEATURES / FEATURE\_SEL parameters

The bit strings FEATURES and FEATURE\_SEL determine optional behavior of the resource. The first defines the available features, and is read only. The second is used to turn on an available

feature by configuration. If a bit is set in FEATURE\_SEL that is not set in FEATURES, the result will be a block alarm for a configuration error.

Smar devices support the following features: Reports supported, Fault State supported, Soft Write lock supported.

#### **Fault state for the whole resource**

If the user sets the SET\_FSTATE parameter, the FAULT\_STATE parameter will indicate active and it will cause **all output function blocks** in the resource to go immediately to the condition chosen by the fault state Type I/O option. It may be cleared by setting the CLR\_FSTATE parameter. The set and clear parameters do not appear in a view because they are momentary.

#### **Write lock by software**

The WRITE\_LOCK parameter, if set, will prevent any external change to the static or nonvolatile data base in the Function Block Application of the resource. Block connections and calculation results will proceed normally, but the configuration will be locked. It is set and cleared by writing to the WRITE\_LOCK parameter. Clearing WRITE\_LOCK will generate the discrete alert WRITE\_ALM, at the WRITE\_PRI priority. Setting WRITE\_LOCK will clear the alert, if it exists.

Before setting WRITE\_LOCK parameter to *Locked*, it is necessary to select the “Soft Write lock supported” option in FEATURE\_SEL.

#### **Schedule Optimization**

The functional blocks optimization is enable when "Sched and Link Optimization" of the RES\_OPTS parameter is set. In this case the blocks Schedule is executed in sequential mode not consider the dowloaded FBSchedule, but complying with the macrocycle calculated by Syscon.

Optionally the user can change the optimization level by OPTIM\_LEVEL parameter, establishing a high (level 1) or low (level 3) optimization. The level indicates the gap size among the block groups.

RES\_STATUS parameter indicates “Schedule Overflow” when the Resource block detects an insufficient Syscon’s macrocycle to execute the block logic. Indicates “Download in Progress” always when to start a configuration download in resource. In this case, when finishing the download the parameter change the status for None. Indicates “CPU Overload” when CPU overload is higher than 80%.

Besides, this parameter can indicate error if the Schedule is not executed within a specified time by macrocycle. This problem occurs when the configuration has many HSE links. In this case the user should increase the macrocycle and the optimization level, to do a new download, until the macrocycle is in conformity with the control logic, in other words, until execute completely the Schedule.

#### **Other Features**

The parameter CYCLE\_TYPE is a BitString that defines the types of cycles that this resource can do. CYCLE\_SEL allows the configurator chooses one of them. If CYCLE\_SEL contains more than one bit, or the bit set is not set in CYCLE\_TYPE, the result will be a block alarm for a configuration error. MIN\_CYCLE\_T is the manufacturer specified minimum time to execute a cycle. It puts a lower limit on the scheduling of the resource.

MEMORY\_SIZE declares the size of the resource for configuration of function blocks, in kilobytes. The parameter FREE\_SPACE shows the percentage of configuration memory that is still available. FREE\_TIME shows the approximate percentage of time that the resource has left for processing new function blocks, should they be configured.

#### **BLOCK\_ERR**

The BLOCK\_ERR of the resource block will reflect the following causes:

- Device Fault State Set – When FAULT\_STATE is active.
- Simulate Active – When the Simulate jumper is ON.
- Out of Service – When the block is in O/S mode.

#### **Supported Modes**

O/S, IMAN and AUTO

## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/Options  | Default Value | Units         | Store/Mode | Description  |
|-----|-------------|--------------------|--|---------------|---------------|------------|--|
| 1   | ST_REV      | Unsigned16         |  | 0             | None          | S/RO       |  |
| 2   | TAG_DESC    | OctString(32)      |  | Spaces        | Na            | S          |  |
| 3   | STRATEGY    | Unsigned16         |  | 0             | None          | S          |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255   | 0             | None          | S          |  |
| 5   | MODE_BLK    | DS-69              |  | O/S           | Na            | S          | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)       |  |               | E             | D / RO     |  |
| 7   | RS_STATE    | Unsigned8          |  |               | E             | D / RO     | State of the function block application state machine.   |
| 8   | TEST_RW     | DS-85              |  |               | None          | D          | Read/write test parameter - used only for conformance testing.   |
| 9   | DD_RESOURCE | VisibletString(32) |  | Spaces        | Na            | S / RO     | String identifying the tag of the resource which contains the Device Description for this resource.                              |
| 10  | MANUFAC_ID  | Unsigned32         | Enumeration; controlled by FF  | 0x00000302    | None          | S / RO     | Manufacturer identification number - used by an interface device to locate the DD file for the resource.                         |
| 11  | DEV_TYPE    | Unsigned16         | Set by mfgr  |               | None          | S / RO     | Manufacturer's model number associated with the resource - used by interface devices to locate the DD file for the resource.     |
| 12  | DEV_REV     | Unsigned8          | Set by mfgr  |               | None          | S / RO     | Manufacturer revision number associated with the resource - used by an interface device to locate the DD file for the resource.  |
| 13  | DD_REV      | Unsigned8          | Set by mfgr  |               | None          | S / RO     | Revision of the DD associated with the resource - used by an interface device to locate the DD file for the resource.            |
| 14  | GRANT_DENY  | DS-70              | See Block Options  | 0             | Na            | D          | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 15  | HARD_TYPES  | Bitstring(2)       | Set by mfgr  |               | Na            | S / RO     | The types of hardware available as channel numbers.  |
| 16  | RESTART     | Unsigned8          | 1: Run,<br>2: Restart resource,<br>3: Restart with defaults,<br>4: Restart processor |               | E             | D          | Allows a manual restart to be initiated. Several degrees of restart are possible.  |
| 17  | FEATURES    | Bitstring(2)       | Set by mfgr  |               | Na            | S / RO     | Used to show supported resource block options.   |
| 18  | FEATURE_SEL | Bitstring(2)       |  | 0             | Na            | S          | Used to select resource block options.   |
| 19  | CYCLE_TYPE  | Bitstring(2)       | Set by mfgr  |               | Na            | S / RO     | Identifies the block execution methods available for this resource.  |
| 20  | CYCLE_SEL   | Bitstring(2)       |  | 0             | Na            | S          | Used to select the block execution method for this resource.   |
| 21  | MIN_CYCLE_T | Unsigned32         | Set by mfgr  |               | 1/32 millisec | S / RO     | Time duration of the shortest cycle interval of which the resource is capable.   |
| 22  | MEMORY_SIZE | Unsigned16         | Set by mfgr  |               | kbytes        | S / RO     | Available configuration memory in the empty resource. To be checked before attempting a download.                                |
| 23  | NV_CYCLE_T  | Unsigned32         |  |               | 1/32 millisec | S / RO     | Interval between writing copies of NV parameters to non-volatile memory. Zero means never.                                       |
| 24  | FREE_SPACE  | Float              | 0 to 100 %   |               | %             | D / RO     | Percent of memory available for further configuration. Zero in a preconfigured resource.   |
| 25  | FREE_TIME   | Float              | 0 to 100%  |               | %             | D / RO     | Percent of the block processing time that is free to process additional blocks.  |
| 26  | SHED_RCAS   | Unsigned32         |  | 640000        | 1/32millisec  | S          | Time duration at which to give up on computer writes to function block RCAs locations.   |

| Idx | Parameter      | Data Type (length) | Valid Range/Options                       | Default Value | Units         | Store/Mode | Description  |
|-----|----------------|--------------------|---|---------------|---------------|------------|--|
| 27  | SHED_ROUT      | Unsigned32         |   | 640000        | 1/32 millisec | S          | Time duration at which to give up on computer writes to function block ROut locations.   |
| 28  | FAULT_STATE    | Unsigned8          | 1: Clear, 2: Active                       |               | E             | D          | Condition set by loss of communication to an output block, failure promoted to an output block or a physical contact. When Fault State condition is set, and then output function blocks will perform their FSAFE actions.   |
| 29  | SET_FSTATE     | Unsigned8          | 1: Off, 2: Set                            | 1             | E             | D          | Allows the fault state condition to be manually initiated by selecting Set.  |
| 30  | CLR_FSTATE     | Unsigned8          | 1: Off, 2: Clear                          | 1             | E             | D          | Writing a Clear to this parameter will clear the device fault state if the field condition, if any, has cleared.   |
| 31  | MAX_NOTIFY     | Unsigned8          | Set by mfg                                |               | None          | S / RO     | Maximum number of unconfirmed notifies messages possible.  |
| 32  | LIM_NOTIFY     | Unsigned8          | 0 to MAX_NOTIFY                           | MAX_NOTIFY    | None          | S          | Maximum number of unconfirmed alert notifies messages allowed.   |
| 33  | CONFIRM_TIME   | Unsigned32         |   | 640000        | 1/32 millisec | S          | The minimum time between retries of alert reports.   |
| 34  | WRITE_LOCK     | Unsigned8          | 1: Unlocked, 2: Locked                    | 1             | E             | S          | If set, no writes from anywhere are allowed, except to clear WRITE_LOCK. Block inputs will continue to be updated.   |
| 35  | UPDATE_EVT     | DS-73              |   |               | Na            | D          | This alert is generated by any change to the static data.  |
| 36  | BLOCK_ALM      | DS-72              |   |               | Na            | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 37  | ALARM_SUM      | DS-74              |   |               | Na            | S          | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 38  | ACK_OPTION     | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable | 0             | Na            | S          | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 39  | WRITE_PRI      | Unsigned8          | 0 to 15                                   | 0             | None          | S          | Priority of the alarm generated by clearing the write lock.  |
| 40  | WRITE_ALM      | DS-72              |   |               | None          | D          | This alert is generated if the write lock parameter is cleared.  |
| 41  | ITK_VER        | Unsigned16         |   |               | Na            | S/RO       | <b>This parameter informs which ITK version is the device (only for certified devices).</b>  |
| 42  | RES_OPTS       | BitString (2)      |   | None          |               | RW / OOS   | Options of Resource block. See description below.  |
| 43  | OPTIM_LEVEL    | Unsigned8          | 1: High, 2: Average, 3: Low               | 1             |               | RW / OOS   | If the optimization is enable indicates the level of the optimization: (1) is best optimization up to; (2) average optimization; (3) low optimization.   |
| 44  | RES_STATUS     | BitString (2)      |   |               |               | RO         | Status of some resource features.  |
| 45  | RES_MIN_CYCL E | Unsigned32         |   |               |               | RO         | Dynamic minimum macrocycle time recommended.   |

Legend: E – Enumerated Parameter; Na – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static.

Gray Background Line: Default Parameters in Syscon

## RS2 - Resource Block

Additional parameters to the Resource Block adding the Field Diagnostics feature

This block has additional features of FOUNDATION Diagnostic (Field Diagnostics) whose parameters are described below.

### Parameters

| Idx | Parameter         | Data Type (length) | Valid Range/Options | Default Value | Units | Store/Mode | Description   |
|-----|-------------------|--------------------|---------------------|---------------|-------|------------|---|
| 46  | FD_VER            | Unsigned16         |                     |               | Na    | S, RO      | A parameter equal to the value of the major version of the Field Diagnostics specification that this device was designed to.  |
| 47  | FD_FAIL_ACTIVE    | BitString[4]       |                     |               | Na    | D, RO      | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 48  | FD_OFFSPEC_ACTIVE | BitString[4]       |                     |               | Na    | D, RO      | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 49  | FD_MAINT_ACTIVE   | BitString[4]       |                     |               | Na    | D, RO      | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 50  | FD_CHECK_ACTIVE   | BitString[4]       |                     |               | Na    | D, RO      | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown. See section 2.9   |
| 51  | FD_FAIL_MAP       | BitString[4]       |                     |               | Na    | S          | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 52  | FD_OFFSPEC_MAP    | BitString[4]       |                     |               | Na    | S          | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 53  | FD_MAINT_MAP      | BitString[4]       |                     |               | Na    | S          | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 54  | FD_CHECK_MAP      | BitString[4]       |                     |               | Na    | S          | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 55  | FD_FAIL_MASK      | BitString[4]       |                     |               | Na    | S          | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |

|    |                 |              |      |          |    |   |   |
|----|-----------------|--------------|------|----------|----|---|---|
| 56 | FD_OFFSPEC_MASK | BitString[4] |      |          | Na | S | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition.   |
| 57 | FD_MAINT_MASK   | BitString[4] |      |          | Na | S | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition.   |
| 58 | FD_CHECK_MASK   | BitString[4] |      |          | Na | S | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition.   |
| 59 | FD_FAIL_ALM     | DS-87        |      |          | Na | D | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 60 | FD_OFFSPEC_ALM  | DS-87        |      |          | Na | D | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 61 | FD_MAINT_ALM    | DS-87        |      |          | Na | D | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 62 | FD_CHECK_ALM    | DS-87        |      |          | Na | D | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 63 | FD_FAIL_PRI     | Unsigned8    | 0-15 | 0        | Na | S | This parameter allows the user to specify the priority of this alarm category.  |
| 64 | FD_OFFSPEC_PRI  | Unsigned8    | 0-15 | 0        | Na | S | This parameter allows the user to specify the priority of this alarm category.  |
| 65 | FD_MAINT_PRI    | Unsigned8    | 0-15 | 0        | Na | S | This parameter allows the user to specify the priority of this alarm category.  |
| 66 | FD_CHECK_PRI    | Unsigned8    | 0-15 | 0        | Na | S | This parameter allows the user to specify the priority of this alarm category.  |
| 67 | FD_SIMULATE     | DS-89        |      | disabled | Na | D | This parameter allows the conditions to be manually supplied when simulation is enabled. When simulation is disabled both the diagnostic simulate value and the diagnostic value tracks the actual conditions. The simulate jumper is required for simulation to be enabled and while simulation is enabled the recommended action will show that simulation is active. |

|    |                 |            |   |   |    |       |   |
|----|-----------------|------------|---|---|----|-------|---|
| 68 | FD_RECOMMEN_ACT | Unsigned16 | 0-Not Initialized<br>1-No Action Required<br>28-Replace the battery in the CPU module<br>29-RS block in O/S<br>31-Check the power supply system, including the power supply module<br>32-Replace CPU module | 0 | Na | D, RO | This parameter is a device enumerated summarization of the most severe condition or conditions detected. The DD help should describe by enumerated action, what should be done to alleviate the condition or conditions. 0 is defined as Not Initialized, 1 is defined as No Action Required, all others defined by manufacturer. |
|----|-----------------|------------|---|---|----|-------|---|

**Legend:** E – Enumerated Parameter; Na – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static.

Gray Background Line: Default Parameters in Syscon

### Definition of Field Diagnostics

| Bit# | Bit Description            | Default Value |                |              |             |
|------|----------------------------|---------------|----------------|--------------|-------------|
|      |                            | FD_MAINT_MAP  | FD_OFFSPEC_MAP | FD_CHECK_MAP | FD_FAIL_MAP |
| 0    | Check                      |               |                | x            |             |
| 1    | FD_1                       |               |                |              |             |
| 2    | FD_2                       |               |                |              |             |
| 3    | FD_3                       |               |                |              |             |
| 4    | FD_4                       |               |                |              |             |
| 5    | FD_5                       |               |                |              |             |
| 6    | FD_6                       |               |                |              |             |
| 7    | FD_7                       |               |                |              |             |
| 8    | FD_8                       |               |                |              |             |
| 9    | FD_9                       |               |                |              |             |
| 10   | FD_10                      |               |                |              |             |
| 11   | FD_11                      |               |                |              |             |
| 12   | FD_12                      |               |                |              |             |
| 13   | FD_13                      |               |                |              |             |
| 14   | FD_14                      |               |                |              |             |
| 15   | FD_15                      |               |                |              |             |
| 16   | FD_16                      |               |                |              |             |
| 17   | FD_17                      |               |                |              |             |
| 18   | FD_18                      |               |                |              |             |
| 19   | FD_19                      |               |                |              |             |
| 20   | FD_20                      |               |                |              |             |
| 21   | FD_21                      |               |                |              |             |
| 22   | FD_22                      |               |                |              |             |
| 23   | FD_23                      |               |                |              |             |
| 24   | FD_24                      |               |                |              |             |
| 25   | FD_25                      |               |                |              |             |
| 26   | HART communication failure | x             |                |              | x           |
| 27   | Low battery voltage        | x             |                |              |             |
| 28   | Resource in O/S            |               |                | x            |             |
| 29   | FD_29                      |               |                |              |             |
| 30   | Power supply failure       | x             |                |              |             |
| 31   | Bad memory of CPU module   | x             |                |              | x           |

## Transducer Blocks

### DIAG – Diagnostics Transducer Block

#### Description

This transducer block provides the following features:

- Online measurement of block execution time
- Hardware revision
- Firmware revision
- Serial number of device
- Serial number of main board

The parameter BEHAVIOR will define which initial values for parameters will be used after a block instantiation. The option *Adapted* selects a more suitable initial value set; it will avoid invalid values for parameters. It is still possible to have the initial values defined by specification by selecting the option *Spec*.

#### Supported modes

O/S and AUTO.

#### Parameters

| Idx | Parameter      | Data Type (length)  | Valid Range/ Options            | Default Value | Units | Store / Mode | Description   |
|-----|----------------|---------------------|---------------------------------|---------------|-------|--------------|---|
| 1   | ST_REV         | Unsigned16          |                                 | 0             | None  | S/RO         |   |
| 2   | TAG_DESC       | OctString(32)       |                                 | Spaces        | Na    | S            |   |
| 3   | STRATEGY       | Unsigned16          |                                 | 0             | None  | S            |   |
| 4   | ALERT_KEY      | Unsigned8           | 1 to 255                        | 0             | None  | S            |   |
| 5   | MODE_BLK       | DS-69               |                                 | O/S           | Na    | S            | See Mode Parameter.   |
| 6   | BLOCK_ERR      | Bitstring(2)        |                                 |               | E     | D/RO         |   |
| 7   | EXE_TIME_TAG   | Visible String (32) |                                 | spaces        | Na    | D            | Block tag of the selected block to measure the execution time.                    |
| 8   | MIN_EXE_TIME   | Float               |                                 | +INF          | ms    | D / RO       | Minimum execution time of the selected block.                                     |
| 9   | CUR_EXE_TIME   | Float               |                                 | 0             | ms    | D / RO       | Current execution time of the selected block.                                     |
| 10  | MAX_EXE_TIME   | Float               |                                 | 0             | ms    | D / RO       | Maximum execution time of the selected block.                                     |
| 11  | HW_REV         | Visible String(5)   |                                 |               |       | S / RO       | Hardware revision.  |
| 12  | FIRMWARE_REV   | Visible String(5)   |                                 |               |       | S / RO       | Firmware revision.  |
| 13  | DEV_SN         | Unsigned32          |                                 |               |       | S / RO       | Device serial number.   |
| 14  | MAIN_BOARD_SN  | Unsigned32          |                                 |               |       | S / RO       | Main board serial number.   |
| 15  | BEHAVIOR       | Unsigned8           | 0:Adapted<br>1:Spec             | 0             | E     | S            | Select the initial values for parameters, there are two options Adapted and Spec. |
| 16  | PUB_SUB_STATUS | Unsigned8           | 0-good<br>1-bad                 |               | E     | D / RO       | Indicate if all external links is good or if at least one is bad.                 |
| 17  | LINK_SELECTION | Unsigned8           | 0-first<br>1-next<br>2-previous | 0             | E     | D            | Select an external link.  |
| 18  | LINK_NUMBER    | Unsigned16          |                                 |               |       | D / RO       | Number of the external link selected.   |
| 19  | LINK_STATUS    | Unsigned8           |                                 |               |       | D / RO       | Status of the external link selected (see table below)                            |
| 20  | LINK_RECOVER   | Unsigned8           | 0-no action<br>1-action         | No action     | E     | D            | Command a recovery process to the external link selected.                         |



| Idx | Parameter     | Data Type<br>(length) | Valid Range/<br>Options         | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|---------------|-----------------------|---------------------------------|------------------|-------|-----------------|--|
| 21  | BLOCK_ALM     | DS-72                 |                                 |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 22  | SAVING_CONFIG | Unsigned 8            | 0 – NOT<br>SAVING<br>1 - SAVING | 0                | E     | D               | It indicates if the device is saving the configuration in a non-volatile memory.   |

**Legend:** E – Enumerated Parameter; Na – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static.

*Gray Background Line: Default Parameters in Syscon*

#### Description of the values given by the LINK\_STATUS parameter

| Link Status | General<br>Status | Publisher/<br>Subscriber | Connection<br>Status | Sending/Receiving     | Block Update |
|-------------|-------------------|--------------------------|----------------------|-----------------------|--------------|
| 0X00        | Good              | Publisher                |                      |                       |              |
| 0X40        | Good              | Subscriber               |                      |                       |              |
| 0X84        | Bad               | Publisher                | Established          | Sending/Receiving     | Not updating |
| 0X88        | Bad               | Publisher                | Established          | Not sending/receiving | Updating     |
| 0X8C        | Bad               | Publisher                | Established          | Not sending/receiving | Not updating |
| 0X98        | Bad               | Publisher                | Not established      | Not sending/receiving | Updating     |
| 0X9C        | Bad               | Publisher                | Not established      | Not sending/receiving | Not updating |
| 0XA8        | Bad               | Publisher                | Pending              | Not sending/receiving | Updating     |
| 0XAC        | Bad               | Publisher                | Pending              | Not sending/receiving | Not updating |
| 0XBC        | Bad               | Publisher                | Not configured       | Not sending/receiving | Not updating |
| 0XC4        | Bad               | Subscriber               | Established          | Sending/Receiving     | Not updating |
| 0XCC        | Bad               | Subscriber               | Established          | Not sending/receiving | Not updating |
| 0XDC        | Bad               | Subscriber               | Not established      | Not sending/receiving | Not updating |
| 0XEC        | Bad               | Subscriber               | Pending              | Not sending/receiving | Not updating |
| 0XFC        | Bad               | Subscriber               | Not configured       | Not sending/receiving | Not updating |

## TEMP – DF45 Temperature Transducer

### Overview

This is the transducer block for the module DF45, an eight low level input module for RTD, TC, mV and Ohm.

#### IMPORTANT

If the temperature module is used for blocks and for Ladder, the temperature module configuration must be the same Ladder configuration. In this case, it will not be allowed an online change (after a download) of block channel if the Ladder is using this module. In this case, the online change of channel can be made only if the rack, slot is not being used by the Ladder.

### Description

This transducer block has parameters to configure the eight inputs of low signal, as well an individual status and value in engineering units for each input. Therefore it is enough to configure only the TEMP block if the purpose is to monitor variables.

If the application is a control loop or calculation, it is also necessary to configure an AI or MAI block to address these variables. One important difference for the TEMP block, when using an AI block to access an input : write to VALUE\_RANGE\_x parameter is disabled. The user must configure the scale in the XD\_SCALE parameter of the AI block, that will be copied to the corresponding VALUE\_RANGE\_x parameter.

### BLOCK\_ERR

The BLOCK\_ERR will reflect the following causes:

- Block Configuration Error - When it is not compatible the CHANNEL parameter and HC configuration (DFI302);
- Input Failure – At least one input is in failure (DFI302);
- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S and AUTO.

### Parameters

| Idx | Parameter     | Data Type (length) | Valid Range/Options | Default Value | Units | Store / Mode | Description   |
|-----|---------------|--------------------|---------------------|---------------|-------|--------------|---|
| 1   | ST_REV        | Unsigned16         |                     | 0             | None  | S/RO         |   |
| 2   | TAG_DESC      | OctString(32)      |                     | Spaces        | Na    | S            |   |
| 3   | STRATEGY      | Unsigned16         |                     | 0             | None  | S            |   |
| 4   | ALERT_KEY     | Unsigned8          | 1 to 255            | 0             | None  | S            |   |
| 5   | MODE_BLK      | DS-69              |                     | O/S           | Na    | S            | See Mode Parameter  |
| 6   | BLOCK_ERR     | Bitstring(2)       |                     |               | E     | D / RO       |   |
| 7   | CHANNEL       | Unsigned16         |                     |               | None  | S / O/S      | The rack and slot number of the associated DF-45 module coded as RRSXX.   |
| 8   | TEMP_0        | DS-65              |                     |               |       | D            | Temperature of point 0.   |
| 9   | TEMP_1        | DS-65              |                     |               |       | D            | Temperature of point 1.   |
| 10  | TEMP_2        | DS-65              |                     |               |       | D            | Temperature of point 2.   |
| 11  | TEMP_3        | DS-65              |                     |               |       | D            | Temperature of point 3.   |
| 12  | TEMP_4        | DS-65              |                     |               |       | D            | Temperature of point 4.   |
| 13  | TEMP_5        | DS-65              |                     |               |       | D            | Temperature of point 5.   |
| 14  | TEMP_6        | DS-65              |                     |               |       | D            | Temperature of point 6.   |
| 15  | TEMP_7        | DS-65              |                     |               |       | D            | Temperature of point 7.   |
| 16  | VALUE_RANGE_0 | DS-68              |                     | 0-100%        | VR0   | S / O/S      | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |

| Idx | Parameter            | DataType<br>(length) | Valid Range/<br>Options                      | Default<br>Value  | Units    | Store /<br>Mode | Description   |
|-----|----------------------|----------------------|--|-------------------|----------|-----------------|---|
| 17  | SENSOR_CONNECTION_0  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 0.   |
| 18  | <b>SENSOR_TYPE_0</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 0.</b>  |
| 19  | VALUE_RANGE_1        | DS-68                |  | 0-100%            | VR1      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 20  | SENSOR_CONNECTION_1  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 1.   |
| 21  | <b>SENSOR_TYPE_1</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 1.</b>  |
| 22  | VALUE_RANGE_2        | DS-68                |  | 0-100%            | VR2      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 23  | SENSOR_CONNECTION_2  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 2.   |
| 24  | <b>SENSOR_TYPE_2</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 2.</b>  |
| 25  | VALUE_RANGE_3        | DS-68                |  | 0-100%            | VR3      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 26  | SENSOR_CONNECTION_3  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 3.   |
| 27  | <b>SENSOR_TYPE_3</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 3.</b>  |
| 28  | VALUE_RANGE_4        | DS-68                |  | 0-100%            | VR4      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 29  | SENSOR_CONNECTION_4  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 4.   |
| 30  | SENSOR_TYPE_4        | Unsigned8            | See table below                              | Pt 100 IEC        | E        | S / O/S         | Type of sensor 4.   |
| 31  | VALUE_RANGE_5        | DS-68                |  | 0-100%            | VR5      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 32  | SENSOR_CONNECTION_5  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 5.   |
| 33  | <b>SENSOR_TYPE_5</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 5.</b>  |
| 34  | VALUE_RANGE_6        | DS-68                |  | 0-100%            | VR6      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |
| 35  | SENSOR_CONNECTION_6  | Unsigned8            | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3                 | E        | S / O/S         | Connection of the sensor 6.   |
| 36  | <b>SENSOR_TYPE_6</b> | <b>Unsigned8</b>     | <b>See table below</b>                       | <b>Pt 100 IEC</b> | <b>E</b> | <b>S / O/S</b>  | <b>Type of sensor 6.</b>  |
| 37  | VALUE_RANGE_7        | DS-68                |  | 0-100%            | VR7      | S / O/S         | If it is connected to AI block, it is a copy of XD_SCALE. Otherwise the user may write in this scaling parameter. |

| Idx | Parameter           | DataType (length) | Valid Range/ Options                         | Default Value | Units | Store / Mode | Description  |
|-----|---------------------|-------------------|--|---------------|-------|--------------|--|
| 38  | SENSOR_CONNECTION_7 | Unsigned8         | 1 : differential<br>2 : 2-wire<br>3 : 3-wire | 3             | E     | S / O/S      | Connection of the sensor 7.  |
| 39  | SENSOR_TYPE_7       | Unsigned8         | See table below                              | Pt 100 IEC    | E     | S / O/S      | Type of sensor 7.  |
| 40  | UPDATE_EVT          | DS-73             |  |               | Na    | D            | This alert is generated by any change to the static data.  |
| 41  | BLOCK_ALM           | DS-72             |  |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static  
Gray Background Line: Default Parameters of Syscon

| Code | Sensor Type | Class | Sensor range – Differential – (Celsius) | Sensor range – 2-wire (Celsius) | Sensor range – 3-wire (Celsius) |
|------|-------------|-------|---|---------------------------------|---------------------------------|
| 1    | Cu 10 GE    | RTD   | -270 to 270                             | -20 to 250                      | -20 to 250                      |
| 2    | Ni 120 DIN  |       | -320 to 320                             | -50 to 270                      | -50 to 270                      |
| 3    | Pt 50 IEC   |       | -1050 to 1050                           | -200 to 850                     | -200 to 850                     |
| 4    | Pt 100 IEC  |       | -1050 to 1050                           | -200 to 850                     | -200 to 850                     |
| 5    | Pt 500 IEC  |       | -270 to 270                             | -200 to 450                     | -200 to 450                     |
| 6    | Pt 50 JIS   |       | -850 to 850                             | -200 to 600                     | -200 to 600                     |
| 7    | Pt 100 JIS  |       | -800 to 800                             | -200 to 600                     | -200 to 600                     |
| 51   | 0 to 100    | Ohm   |   | 0 to 100                        | 0 to 100                        |
| 52   | 0 to 400    |       |   | 0 to 400                        | 0 to 400                        |
| 53   | 0 to 2000   |       |   | 0 to 2000                       | 0 to 2000                       |
| 151  | B NBS       | TC    | -1600 to 1600                           | 100 to 1800                     |                                 |
| 152  | E NBS       |       | -1100 to 1100                           | -100 to 1000                    |                                 |
| 153  | J NBS       |       | 900 to 900                              | -150 to 750                     |                                 |
| 154  | K NBS       |       | -1550 to 1550                           | -200 to 1350                    |                                 |
| 155  | N NBS       |       | -1400 to 1400                           | -100 to 1300                    |                                 |
| 156  | R NBS       |       | -1750 to 1750                           | 0 to 1750                       |                                 |
| 157  | S NBS       |       | -1750 to 1750                           | 0 to 1750                       |                                 |
| 158  | T NBS       |       | -600 to 600                             | -200 to 400                     |                                 |
| 159  | L DIN       |       | -1100 to 1100                           | -200 to 900                     |                                 |
| 160  | U DIN       |       | -800 to 800                             | -200 to 600                     |                                 |
| 201  | -6 to 22    | MV    |   | -6 to 22                        |                                 |
| 202  | -10 to 100  |       |   | -10 to 100                      |                                 |
| 203  | -50 to 500  |       |   | -50 to 500                      |                                 |

**If DIAG.BEHAVIOR parameter is “Adapted”:**

When the configuration of sensor type means a different class, the connection is automatically changed to default (RTD and Ohm – 3-wire, TC and mV – 2-wire).

## HC – Hardware Configuration Transducer

### Overview

It configures the module type for each slot in the **DFI302**.

#### IMPORTANT

For configuration of the same rack, slot for blocks (in HC block) and for Ladder, the module must be of the same type. It will not be allowed an online change (after a download) of an IO module in the HC block if the Ladder is using this module. In this case, the online change of IO module can be made only if the rack, slot is not being used by the Ladder.

### Description

The following table shows the available module types.

| Code | Description  | I/O Type                                 |
|------|--|--|
|      | Available slot                                       | No I/O                                   |
| DF51 | DFI302 Processor 1x10Mbps, 4xH1                      | No I/O                                   |
| DF50 | Power Supply 90-264VAC                               | No I/O                                   |
| DF56 | Power Supply for Backplane 20-30VDC                  | No I/O                                   |
| DF52 | Power Supply for Fieldbus                            | No I/O                                   |
| DF49 | 2-channel Power Supply Impedance                     | No I/O                                   |
| DF53 | 4-channel Power Supply Impedance                     | No I/O                                   |
| DF11 | 2 Groups of 8 24VDC Inputs (Isolated)                | 16-discrete input                        |
| DF12 | 2 Groups of 8 48VDC Inputs (Isolated)                | 16- discrete input                       |
| DF13 | 2 Groups of 8 60VDC Inputs (Isolated)                | 16- discrete input                       |
| DF14 | 2 Groups of 8 125VDC Inputs (Isolated)               | 16- discrete input                       |
| DF15 | 2 Groups of 8 24VDC Inputs (Sink)(Isolated)          | 16- discrete input                       |
| DF16 | 2 Groups of 4 120VAC Inputs (Isolated)               | 8- discrete input                        |
| DF17 | 2 Groups of 4 240VAC Inputs (Isolated)               | 8- discrete input                        |
| DF18 | 2 Groups of 8 120VAC Inputs (Isolated)               | 16- discrete input                       |
| DF19 | 2 Groups of 8 240VAC Inputs (Isolated)               | 16- discrete input                       |
| DF20 | 1 Group of 8 On/Off Switches                         | 8- discrete input                        |
| DF21 | 1 Group of 16 Open Collector Outputs                 | 16- discrete output                      |
| DF22 | 2 Group of 8 Transistor Outputs (source) (Isolated)  | 16- discrete output                      |
| DF23 | 2 Groups of 4 120/240VAC Outputs                     | 8- discrete output                       |
| DF24 | 2 Groups of 8 120/240VAC Outputs                     | 16- discrete output                      |
| DF25 | 2 Groups of 4 NO Relays Outputs                      | 8- discrete output                       |
| DF26 | 2 Groups of 4 NC Relays Outputs                      | 8- discrete output                       |
| DF27 | 1 Group of 4 NO and 4 NC Relay Outputs               | 8- discrete output                       |
| DF28 | 2 Groups of 8 NO Relays Outputs                      | 16- discrete output                      |
| DF29 | 2 Groups of 4 NO Relays Outputs (W/o RC)             | 8- discrete output                       |
| DF30 | 2 Groups of 4 NC Relays Outputs (W/o RC)             | 8- discrete output                       |
| DF31 | 1 Group of 4 NO and 4 NC Relay Outputs (W/o RC)      | 8- discrete output                       |
| DF32 | 1 Group of 8 24VDC Inputs and 1 Group of 4 NO Relays | 8- discrete input/<br>4- discrete output |
| DF33 | 1 Group of 8 48VDC Inputs and 1 Group of 4 NO Relays | 8- discrete input/<br>4- discrete output |
| DF34 | 1 Group of 8 60VDC Inputs and 1 Group of 4 NO Relays | 8- discrete input/<br>4- discrete output |
| DF35 | 1 Group of 8 24VDC Inputs and 1 Group of 4 NC Relays | 8- discrete input/<br>4- discrete output |

| Code | Description  | I/O Type                                 |
|------|--|--|
| DF36 | 1 Group of 8 48VDC Inputs and 1 Group of 4 NC Relays         | 8- discrete input/<br>4- discrete output |
| DF37 | 1 Group of 8 60VDC Inputs and 1 Group of 4 NC Relays         | 8- discrete input/<br>4- discrete output |
| DF38 | 1 Group of 8 24VDC Inputs ,1 Group of 2 NO and 2 NC Relays   | 8- discrete input/<br>4- discrete output |
| DF39 | 1 Group of 8 48VDC Inputs , 1 Group of 2 NO and 2 NC Relays  | 8- discrete input/<br>4- discrete output |
| DF40 | 1 Group of 8 60VDC Inputs , 1 Group of 2 NO and 2 NC Relays  | 8- discrete input/<br>4- discrete output |
| DF41 | 2 Groups of 8 pulse inputs – low frequency                   | 16-pulse input                           |
| DF42 | 2 Groups of 8 pulse inputs – high frequency                  | 16-pulse input                           |
| DF44 | 1 Group of 8 analog inputs with shunt resistors              | 8-analog input                           |
| DF57 | 1 Group of 8 differential analog inputs with shunt resistors | 8-analog input                           |
| DF45 | 1 Group of 8 temperature Inputs                              | 8-temperature                            |
| DF46 | 1 Group of 4 analog output                                   | 4-analog output                          |

The execution method of this transducer block will write to all output modules and it will read all the input modules. If any I/O module has failed in this scan, it will be indicated in BLOCK\_ERR as well in the MODULE\_STATUS\_x. It makes easy to find the module in failure or even the sensor.

All the I/O modules in the previous table may be accessed directly using Input/Output Function Blocks without a transducer block, except for the DF45 that requires the TEMP block.

#### IMPORTANT

In order to have a safe startup, is necessary to set in Offline mode the MODE\_BLK parameter to OOS to download the configuration. Change the MODE\_BLK parameter to Auto, only after the download finished.

#### BLOCK\_ERR

The BLOCK\_ERR of the HC block will reflect the following causes:

- Lost static data – Low battery voltage indication
- Device needs maintenance now – High temperature in the CPU
- Input Failure – a physical input point in failure
- Output Failure – a physical output point in failure
- Out of Service – When the block is in O/S mode.

#### Supported Modes

O/S and AUTO.

#### Parameters

| Idx | Parameter | DataType (length) | Valid Range/ Options | Default Value | Units | Store/ Mode | Description        |
|-----|-----------|-------------------|----------------------|---------------|-------|-------------|--------------------|
| 1   | ST_REV    | Unsigned16        |                      | 0             | None  | S           |                    |
| 2   | TAG_DESC  | OctString(32)     |                      | Spaces        | Na    | S           |                    |
| 3   | STRATEGY  | Unsigned16        |                      | 0             | None  | S           |                    |
| 4   | ALERT_KEY | Unsigned8         | 1 to 255             | 0             | None  | S           |                    |
| 5   | MODE_BLK  | DS-69             |                      | O/S           | Na    | S           | See Mode Parameter |
| 6   | BLOCK_ERR | Bitstring(2)      |                      |               | E     | D / RO      |                    |

| Idx | Parameter                | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|--------------------------|-----------------------|---|------------------|-------|----------------|--|
| 7   | REMOTE_IO                | Unsigned8             | 0 : Remote I/O Master<br>1 : Remote I/O Slave 1<br>2 : Remote I/O Slave 2<br>3 : Remote I/O Slave 3<br>4 : Remote I/O Slave 4<br>5 : Remote I/O Slave 5<br>6 : Remote I/O Slave 6 | 0                | E     | RO             | Identification for master remote I/O. It supports only the I/O Master option.  |
| 8   | IO_TYPE_R0               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 0  |
| 9   | IO_TYPE_R1               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 1  |
| 10  | IO_TYPE_R2               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 2  |
| 11  | IO_TYPE_R3               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 3  |
| 12  | IO_TYPE_R4               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 4  |
| 13  | IO_TYPE_R5               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 5  |
| 14  | IO_TYPE_R6               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 6  |
| 15  | IO_TYPE_R7               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 7  |
| 16  | IO_TYPE_R8               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 8  |
| 17  | IO_TYPE_R9               | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 9  |
| 18  | IO_TYPE_R10              | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 10   |
| 19  | IO_TYPE_R11              | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 11   |
| 20  | IO_TYPE_R12              | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 12   |
| 21  | IO_TYPE_R13              | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 13   |
| 22  | IO_TYPE_R14              | 4 Unsigned8           |   | 0                | E     | S / O/S        | Select module type for the rack 14   |
| 23  | MODULE_STATUS_R<br>0_3   | Bitstring(2)          |   |                  |       | D / RO         | Status of modules in rack 0-3.   |
| 24  | MODULE_STATUS_R<br>4_7   | Bitstring(2)          |   |                  |       | D / RO         | Status of modules in rack 4-7.   |
| 25  | MODULE_STATUS_R<br>8_11  | Bitstring(2)          |   |                  |       | D / RO         | Status of modules in rack 8-11.  |
| 26  | MODULE_STATUS_R<br>12_14 | Bitstring(2)          |   |                  |       | D / RO         | Status of modules in rack 12-14.   |
| 27  | UPDATE_EVT               | DS-73                 |   |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 28  | BLOCK_ALM                | DS-72                 |   |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated Parameter; Na – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static.

**Gray Background Line: Default Parameters in Syscon**

## TRDRED - Redundancy Transducer

### Description

This block provides information about redundant devices. It allows verify the actual status and data related to the devices that work in redundancy.

### Supported modes

O/S and AUTO.

### Parameters

| Idx | Parameter            | Data Type (length) | Valid Range/ Options | Default Value | Units | Store/ Mode | Description  |
|-----|----------------------|--------------------|----------------------|---------------|-------|-------------|--|
| 1   | ST_REV               | Unsigned int       | 0 to 2 <sup>16</sup> | 0             | -     | S/RO        |  |
| 2   | TAG_DESC             | OctString(32)      |                      | Blank         | NA    | RW          |  |
| 3   | STRATEGY             | Unsigned int       | 0 to 2 <sup>16</sup> | 0             | -     | RW          |  |
| 4   | ALERT_KEY            | Unsigned char      | 1 to 255             | 1             | -     | RW          |  |
| 5   | MODE_BLK             | DS-69              |                      | O/S           | NA    |             | See Mode Parameter.  |
| 6   | BLOCK_ERR            | BitString(2)       |                      |               | E     | D/RO        |  |
| 7   | UPDATE_EVT           | EventUpdate        |                      |               |       |             | This alert is generated by any change to the static data.  |
| 8   | BLOCK_ALM            | AlarmDiscrete      |                      |               |       |             | The block alarm is used for all configuration, hardware, and connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | TRANSDUCER_DIRECTORY | Unsigned int       | 0 to 2 <sup>16</sup> |               |       | RO          | A directory that specifies the number and the starting indices of the transducers in the transducer block.   |
| 10  | TRANSDUCER_TYPE      | Unsigned int       | 0 to 2 <sup>16</sup> |               |       | RO          | Identities the transducer that follows.  |
| 11  | XD_ERROR             | Unsigned char      | 1 to 255             |               |       | RO          | Define one of the error code.  |
| 12  | COLLECTION_DIRECTORY | Unsigned long      | 0 to 2 <sup>32</sup> |               |       | RO          | A directory that specifies the number, the starting indices, and DD Item IDs of data collections in each transducers in the transducer block.  |
| 13  | SOFTWARE_NAME        | VisibleString      |                      |               |       | S/RO        | Name of the last software downloaded to the bridge.  |
| 14  | RED_PRIMARY_SN       | Unsigned int       | 0 to 65535           | 0             | NA    | D/RO        | Indicates the Serial Number of the Primary controller.   |
| 15  | RED_SECONDARY_SN     | Unsigned int       | 0 to 65535           | 0             | NA    | D/RO        | Indicates the Serial Number of the Secondary controller.   |
| 16  | RED_PRIMARY_IP       | VisibleString(16)  |                      | Blank         | NA    | D/RO        | Indicates the IP of the Redundant Primary module.  |
| 17  | RED_SECONDARY_IP     | VisibleString(16)  |                      | Blank         | NA    | D/RO        | Indicates the IP of the Redundant Secondary module.  |



| Idx | Parameter                    | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store/<br>Mode | Description   |
|-----|------------------------------|-----------------------|---|------------------|-------|----------------|---|
| 18  | RED_SYNC_STATUS              | Unsigned char         | 0: Not defined<br>1: Stand Alone<br>2: Synchronizing<br>3: Updating Secondary<br>4: Synchronized<br>5: WARNING: Role Conflict<br>6: WARNING: Sync Cable Fail<br>7: WARNING: Updating Secondary Fail | 0                | E     | D / RO         | Indicates the Synchronism status of the controller pair.<br>0: Default value just after start up.<br>1: Stand alone operation (no redundancy).<br>2: Checking configuration for synchronization.<br>3: <i>Primary</i> transferring configuration to the <i>Secondary</i> .<br>4: The <i>Primary</i> continuously updates the <i>Secondary</i> dynamic databases.<br>5: It was not possible to solve autonomously the Role. ( <i>Primary</i> / <i>Secondary</i> ).<br>6: Fail on all the synchronism channels (redundancy unavailable).<br>7: Primary fail before synchronism be completed (redundancy unavailable). |
| 19  | RED_PRIMARY_BAD_CONDITIONS   | Bitstring(2)          | 0. Modbus<br>1. H1-1<br>2. H1-2<br>3. H1-3<br>4. H1-4<br>5. Live List<br>6. Eth1<br>7. HSE link<br>8. Eth2<br>9. Serial Sync Cable<br>10. Unable to Sync  | <None>           | E     | D / RO         | Bad conditions for the Primary controller.  |
| 20  | RED_SECONDARY_BAD_CONDITIONS | Bitstring(2)          | 0. Modbus<br>1. H1-1<br>2. H1-2<br>3. H1-3<br>4. H1-4<br>5. Live List<br>6. Eth1<br>7. HSE link<br>8. Eth2<br>9. Serial Sync Cable<br>10. Unable to Sync  | <None>           | E     | D / RO         | Bad conditions for the Secondary controller.  |
| 21  | RED_PRIMARY_WDG              | Unsigned char         | 0 ~ 255   | 0                | NA    | D / RO         | Watchdog indicating OPC communication status with the Primary controller.   |
| 22  | RED_SECONDARY_WDG            | Unsigned char         | 0 ~ 255   | 0                | NA    | D / RO         | Watchdog indicating OPC communication status with the Secondary controller.   |
| 23  | RED_RESERVED1                | Unsigned long         | 0 to $2^{32}$   | 0                | NA    | D / RW         | Reserved for future use.  |
| 24  | RED_RESERVED2                | Unsigned long         | 0 to $2^{32}$   | 0                | NA    | D / RW         | Reserved for future use.  |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static

Gray Background Line: Custom Parameters

## ABC – Association Block for Components

### Overview

This block was developed using ROM (Remote Operations Management) technology of Fieldbus Foundation. It has similar functionality of the HC block, i.e., allows the user to configure the hardware defining the I/O module type for each position (rack/slot).

### Description

- The execution method performs the I/O scan and calls the execution methods of corresponding transducer blocks.
- The I/O module scanning occurs only if there is compatibility between the I/O module type obtained from Module ID and from the Expected Type (part number). The execution method is invoked only if ABC.ELEM\_STAT\_Rx\_Sx.Element mismatch = Matched, i.e., the Expected tag matches with the transducer block tag and the Expected type, Module ID and transducer block tag match.

ELEM\_EXPD\_Rx\_Sx: user configures the tag and the expected module type for each rack/slot.

ELEM\_ACTL\_Rx\_Sx : indicates the tag and module type found in each rack/slot

*Actual Tag:* If the tags match and there is compatibility between Expected type and transducer block type, thus it is the very Expected tag. Otherwise indicates null. So, the Actual Tag reflects the matching between Expected tag and type with the transducer block.

*Actual Type:* It indicates the module transducer block found in each rack/slot based on Module ID, regardless if it was configured in Expected Tag or Expected Type. If the Module ID is not supported, Actual Type will indicate the Module ID instead of a possible corresponding DD Item ID.

ELEM\_STAT\_Rx\_Sx.Element mismatch: It will indicate "Matched" only if tag and type match.

**Note:** This block was defined to support 32 I/O modules, but it already has parameters (ELEM\_MISMAT\_BS\_2 and FD\_EXTENDED\_ACTIVE\_2) for an expansion up to 64 modules, where only the triads for additional 32 modules would be added.

### Field Diagnostics Definition

| Bit# | Bit Description        | Default Value |                |              |             |
|------|------------------------|---------------|----------------|--------------|-------------|
|      |                        | FD_MAINT_MAP  | FD_OFFSPEC_MAP | FD_CHECK_MAP | FD_FAIL_MAP |
| 0    | Module Failure Rack#0  |               |                |              | x           |
| 1    | Module Failure Rack#1  |               |                |              | x           |
| 2    | Module Failure Rack#2  |               |                |              | x           |
| 3    | Module Failure Rack#3  |               |                |              | x           |
| 4    | Module Failure Rack#4  |               |                |              | x           |
| 5    | Module Failure Rack#5  |               |                |              | x           |
| 6    | Module Failure Rack#6  |               |                |              | x           |
| 7    | Module Failure Rack#7  |               |                |              | x           |
| 8    | Module Failure Rack#8  |               |                |              | x           |
| 9    | Module Failure Rack#9  |               |                |              | x           |
| 10   | Module Failure Rack#10 |               |                |              | x           |
| 11   | Module Failure Rack#11 |               |                |              | x           |
| 12   | Module Failure Rack#12 |               |                |              | x           |
| 13   | Module Failure Rack#13 |               |                |              | x           |
| 14   | Module Failure Rack#14 |               |                |              | x           |
| 15   | Module Failure Rack#15 |               |                |              | x           |
| 16   | FD_16                  |               |                |              |             |
| 17   | FD_17                  |               |                |              |             |
| 18   | FD_18                  |               |                |              |             |
| 19   | FD_19                  |               |                |              |             |
| 20   | FD_20                  |               |                |              |             |
| 21   | FD_21                  |               |                |              |             |
| 22   | FD_22                  |               |                |              |             |
| 23   | FD_23                  |               |                |              |             |
| 24   | FD_24                  |               |                |              |             |
| 25   | FD_25                  |               |                |              |             |
| 26   | FD_26                  |               |                |              |             |
| 27   | FD_27                  |               |                |              |             |
| 28   | FD_28                  |               |                |              |             |
| 29   | FD_29                  |               |                |              | x           |
| 30   | FD_30                  |               |                |              | x           |
| 31   | FD_31                  |               |                |              | x           |
|      |                        | 0x00000000    | 0x00000000     | 0x00000000   | 0x0000ffff  |

**FD\_EXTENDED\_ACTIVE\_1** – The four bits of each slot in the rack are “OR’ed” to represent just one bit in the Field Diagnostics.

| Bit# | Bit Description      |
|------|----------------------|
| 0    | Module Failure R0 S0 |
| 1    | Module Failure R0 S1 |
| 2    | Module Failure R0 S2 |
| 3    | Module Failure R0 S3 |
| 4    | Module Failure R1 S0 |
| 5    | Module Failure R1 S1 |
| 6    | Module Failure R1 S2 |
| 7    | Module Failure R1 S3 |
| 8    | Module Failure R2 S0 |
| 9    | Module Failure R2 S1 |
| 10   | Module Failure R2 S2 |
| 11   | Module Failure R2 S3 |
| 12   | Module Failure R3 S0 |
| 13   | Module Failure R3 S1 |
| 14   | Module Failure R3 S2 |
| 15   | Module Failure R3 S3 |
| 16   | Module Failure R4 S0 |
| 17   | Module Failure R4 S1 |
| 18   | Module Failure R4 S2 |
| 19   | Module Failure R4 S3 |
| 20   | Module Failure R5 S0 |
| 21   | Module Failure R5 S1 |
| 22   | Module Failure R5 S2 |
| 23   | Module Failure R5 S3 |
| 24   | Module Failure R6 S0 |
| 25   | Module Failure R6 S1 |
| 26   | Module Failure R6 S2 |
| 27   | Module Failure R6 S3 |
| 28   | Module Failure R7 S0 |
| 29   | Module Failure R7 S1 |
| 30   | Module Failure R7 S2 |
| 31   | Module Failure R7 S3 |

**FD\_EXTENDED\_ACTIVE\_2** – The four bits of each slot in the rack are “OR’ed” to represent just one bit in the Field Diagnostics.

| Bit# | Bit Description       |
|------|-----------------------|
| 0    | Module Failure R8 S0  |
| 1    | Module Failure R8 S1  |
| 2    | Module Failure R8 S2  |
| 3    | Module Failure R8 S3  |
| 4    | Module Failure R9 S0  |
| 5    | Module Failure R9 S1  |
| 6    | Module Failure R9 S2  |
| 7    | Module Failure R9 S3  |
| 8    | Module Failure R10 S0 |
| 9    | Module Failure R10 S1 |
| 10   | Module Failure R10 S2 |
| 11   | Module Failure R10 S3 |
| 12   | Module Failure R11 S0 |
| 13   | Module Failure R11 S1 |
| 14   | Module Failure R11 S2 |
| 15   | Module Failure R11 S3 |
| 16   | Module Failure R12 S0 |
| 17   | Module Failure R12 S1 |
| 18   | Module Failure R12 S2 |
| 19   | Module Failure R12 S3 |
| 20   | Module Failure R13 S0 |
| 21   | Module Failure R13 S1 |
| 22   | Module Failure R13 S2 |
| 23   | Module Failure R13 S3 |
| 24   | Module Failure R14 S0 |
| 25   | Module Failure R14 S1 |
| 26   | Module Failure R14 S2 |
| 27   | Module Failure R14 S3 |

|    |                       |
|----|-----------------------|
| 28 | Module Failure R15 S0 |
| 29 | Module Failure R15 S1 |
| 30 | Module Failure R15 S2 |
| 31 | Module Failure R15 S3 |

**Diagnostic and troubleshooting**

BLOCK\_ERR. Block configuration: this indication may occur, but does not force the mode block to OS, due to the following problem: ELEM\_MISMAT\_NUM is different from zero.

| Idx | Type/<br>View | Parameter        | Data type     | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|------------------|---------------|-------------|---------------|-------|-------------|--|
| 1   | 1,2,3,4       | ST_REV           | Unsigned16    |             | 0             | None  | S / RO      |  |
| 2   |               | TAG_DESC         | OctString(32) |             | Spaces        | Na    | S           |  |
| 3   | 4             | STRATEGY         | Unsigned16    |             | 255           | None  | S           |  |
| 4   | 4             | ALERT_KEY        | Unsigned8     |             | 1             | None  | S           |  |
| 5   | 1,3           | MODE_BLK         | DS-69         |             | Auto          | Na    | S           |  |
| 6   | 1,3           | BLOCK_ERR        | BitString(2)  |             |               | E     | D / RO      |  |
| 7   |               | UPDATE_EVT       | DS-73         |             |               | Na    | D           | This alert is generated by any change to the static data.  |
| 8   |               | BLOCK_ALM        | DS-72         |             |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | 2             | NUM_POSNS        | Unsigned16    | 0 to 64     | 0             |       | S           | This is the number of positions provided by the current configuration (i.e., number of slots, positions, or addresses physically provided, whether used or not).   |
| 10  | 2             | MAX_NUM_POSNS    | Unsigned16    |             | 64            |       | S / RO      | This is the maximum number of positions supportable by this set of association block parameters (i.e., maximum number of slots, positions, or addresses that could be provided by physical extensions).  |
| 11  | 1,3           | ELEM_MISMAT_NUM  | Unsigned16    | 0 to 64     | 0             | Na    | D / RO      | This is the number of elements whose off-line configuration and on-line presence disagree.   |
| 12  | 1,3           | ELEM_MISMAT_BS_1 | BitString[4]  |             | 0             | Na    | D / RO      | This is a 32-bit bitstring indication which of the first set of 32 elements, if any, has a mismatch of an off-line configuration and on-line presence. The description of the element position effected is provided in the manufacturer's DD for the association block.  |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 13  | 4             | ELEM_EXPD_R0_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 0) |
| 14  |               | ELEM_ACTL_R0_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 0)        |
| 15  | 1,3           | ELEM_STAT_R0_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 0)  |
| 16  | 4             | ELEM_EXPD_R0_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 1) |
| 17  |               | ELEM_ACTL_R0_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 1)        |
| 18  | 1,3           | ELEM_STAT_R0_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 1)  |
| 19  | 4             | ELEM_EXPD_R0_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 2) |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 20  |               | ELEM_ACTL_R0_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 2)        |
| 21  | 1,3           | ELEM_STAT_R0_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 2)  |
| 22  | 4             | ELEM_EXPD_R0_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 3) |
| 23  |               | ELEM_ACTL_R0_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 3)        |
| 24  | 1,3           | ELEM_STAT_R0_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 0 slot 3)  |
| 25  | 4             | ELEM_EXPD_R1_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 0) |
| 26  |               | ELEM_ACTL_R1_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 0)        |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 27  | 1,3           | ELEM_STAT_R1_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 0)  |
| 28  | 4             | ELEM_EXPD_R1_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 1) |
| 29  |               | ELEM_ACTL_R1_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 1)        |
| 30  | 1,3           | ELEM_STAT_R1_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 1)  |
| 31  | 4             | ELEM_EXPD_R1_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 2) |
| 32  |               | ELEM_ACTL_R1_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 2)        |
| 33  | 1,3           | ELEM_STAT_R1_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 2)  |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 34  | 4             | ELEM_EXPD_R1_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 3) |
| 35  |               | ELEM_ACTL_R1_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 3)        |
| 36  | 1,3           | ELEM_STAT_R1_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 1 slot 3)  |
| 37  | 4             | ELEM_EXPD_R2_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 0) |
| 38  |               | ELEM_ACTL_R2_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 0)        |
| 39  | 1,3           | ELEM_STAT_R2_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 0)  |
| 40  | 4             | ELEM_EXPD_R2_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 1) |



| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 41  |               | ELEM_ACTL_R2_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 1)        |
| 42  | 1,3           | ELEM_STAT_R2_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 1)  |
| 43  | 4             | ELEM_EXPD_R2_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 2) |
| 44  |               | ELEM_ACTL_R2_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 2)        |
| 45  | 1,3           | ELEM_STAT_R2_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 2)  |
| 46  | 4             | ELEM_EXPD_R2_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 3) |
| 47  |               | ELEM_ACTL_R2_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 3)        |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 48  | 1,3           | ELEM_STAT_R2_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 2 slot 3)  |
| 49  | 4             | ELEM_EXPD_R3_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 0) |
| 50  |               | ELEM_ACTL_R3_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 0)        |
| 51  | 1,3           | ELEM_STAT_R3_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 0)  |
| 52  | 4             | ELEM_EXPD_R3_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 1) |
| 53  |               | ELEM_ACTL_R3_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 1)        |
| 54  | 1,3           | ELEM_STAT_R3_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 1)  |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 55  | 4             | ELEM_EXPD_R3_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 2) |
| 56  |               | ELEM_ACTL_R3_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 2)        |
| 57  | 1,3           | ELEM_STAT_R3_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 2)  |
| 58  | 4             | ELEM_EXPD_R3_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 3) |
| 59  |               | ELEM_ACTL_R3_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 3)        |
| 60  | 1,3           | ELEM_STAT_R3_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 3 slot 3)  |
| 61  |               | ELEM_EXPD_R4_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 0) |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 62  |               | ELEM_ACTL_R4_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 0)        |
| 63  | 1,3           | ELEM_STAT_R4_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 0)  |
| 64  |               | ELEM_EXPD_R4_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 1) |
| 65  |               | ELEM_ACTL_R4_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 1)        |
| 66  | 1,3           | ELEM_STAT_R4_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 1)  |
| 67  |               | ELEM_EXPD_R4_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 2) |
| 68  |               | ELEM_ACTL_R4_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 2)        |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 69  | 1,3           | ELEM_STAT_R4_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 2)  |
| 70  |               | ELEM_EXPD_R4_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 3) |
| 71  |               | ELEM_ACTL_R4_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 3)        |
| 72  | 1,3           | ELEM_STAT_R4_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 4 slot 3)  |
| 73  |               | ELEM_EXPD_R5_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 0) |
| 74  |               | ELEM_ACTL_R5_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 0)        |
| 75  | 1,3           | ELEM_STAT_R5_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 0)  |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 76  |               | ELEM_EXPD_R5_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 1) |
| 77  |               | ELEM_ACTL_R5_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 1)        |
| 78  | 1,3           | ELEM_STAT_R5_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 1)  |
| 79  |               | ELEM_EXPD_R5_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 2) |
| 80  |               | ELEM_ACTL_R5_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 2)        |
| 81  | 1,3           | ELEM_STAT_R5_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 2)  |
| 82  |               | ELEM_EXPD_R5_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 3) |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 83  |               | ELEM_ACTL_R5_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 3)        |
| 84  | 1,3           | ELEM_STAT_R5_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 5 slot 3)  |
| 85  |               | ELEM_EXPD_R6_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 0) |
| 86  |               | ELEM_ACTL_R6_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 0)        |
| 87  | 1,3           | ELEM_STAT_R6_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 0)  |
| 88  |               | ELEM_EXPD_R6_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 1) |
| 89  |               | ELEM_ACTL_R6_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 1)        |

| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 90  | 1,3           | ELEM_STAT_R6_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 1)  |
| 91  |               | ELEM_EXPD_R6_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 2) |
| 92  |               | ELEM_ACTL_R6_S2 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 2)        |
| 93  | 1,3           | ELEM_STAT_R6_S2 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 2)  |
| 94  |               | ELEM_EXPD_R6_S3 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 3) |
| 95  |               | ELEM_ACTL_R6_S3 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 3)        |
| 96  | 1,3           | ELEM_STAT_R6_S3 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 6 slot 3)  |



| Idx | Type/<br>View | Parameter       | Data type | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-----------------|-----------|-------------|---------------|-------|-------------|--|
| 97  |               | ELEM_EXPD_R7_S0 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 0) |
| 98  |               | ELEM_ACTL_R7_S0 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 0)        |
| 99  | 1,3           | ELEM_STAT_R7_S0 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 0)  |
| 100 |               | ELEM_EXPD_R7_S1 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 1) |
| 101 |               | ELEM_ACTL_R7_S1 | DS-177    |             |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 1)        |
| 102 | 1,3           | ELEM_STAT_R7_S1 | DS-178    |             |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 1)  |
| 103 |               | ELEM_EXPD_R7_S2 | DS-176    |             |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 2) |

| Idx | Type/<br>View | Parameter         | Data type    | Valid Range                               | Initial Value | Units | Store/Other | Description  |
|-----|---------------|-------------------|--------------|---|---------------|-------|-------------|--|
| 104 |               | ELEM_ACTL_R7_S2   | DS-177       |   |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 2)        |
| 105 | 1,3           | ELEM_STAT_R7_S2   | DS-178       |   |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 2)  |
| 106 |               | ELEM_EXPD_R7_S3   | DS-176       |   |               |       | S           | This structured parameter indicates the expected configuration (per off-line configuration) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 3) |
| 107 |               | ELEM_ACTL_R7_S3   | DS-177       |   |               |       | D / RO      | This structured parameter indicates the actual configuration (per element detection) of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 3)        |
| 108 | 1,3           | ELEM_STAT_R7_S3   | DS-178       |   |               |       | D / mixed   | This structured parameter indicates the status of an element beneath the association block hierarchically. The posn suffix indicates the vendor's nomenclature for the particular position or address (rack 7 slot 3)  |
| 109 | 4             | ACK_OPTION        | Bitstring(2) | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |               | E     | S           | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 110 | 4             | FD_VER            | Unsigned16   |   |               | Na    | S, RO       | A parameter equal to the value of the major version of the Field Diagnostics specification that this device was designed to.   |
| 111 | 1,3           | FD_FAIL_ACTIVE    | BitString[4] |   |               | Na    | D, RO       | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.  |
| 112 | 1,3           | FD_OFFSPEC_ACTIVE | BitString[4] |   |               | Na    | D, RO       | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.  |

| Idx | Type/<br>View | Parameter       | Data type    | Valid Range | Initial Value | Units | Store/Other | Description   |
|-----|---------------|-----------------|--------------|-------------|---------------|-------|-------------|---|
| 113 | 1,3           | FD_MAINT_ACTIVE | BitString[4] |             |               | Na    | D, RO       | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 114 | 1,3           | FD_CHECK_ACTIVE | BitString[4] |             |               | Na    | D, RO       | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 115 | 4             | FD_FAIL_MAP     | BitString[4] |             |               | Na    | S           | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 116 | 4             | FD_OFFSPEC_MAP  | BitString[4] |             |               | Na    | S           | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 117 | 4             | FD_MAINT_MAP    | BitString[4] |             |               | Na    | S           | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 118 | 4             | FD_CHECK_MAP    | BitString[4] |             |               | Na    | S           | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 119 | 4             | FD_FAIL_MASK    | BitString[4] |             |               | Na    | S           | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |
| 120 | 4             | FD_OFFSPEC_MASK | BitString[4] |             |               | Na    | S           | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |

| Idx | Type/<br>View | Parameter      | Data type    | Valid Range | Initial Value | Units | Store/Other | Description   |
|-----|---------------|----------------|--------------|-------------|---------------|-------|-------------|---|
| 121 | 4             | FD_MAINT_MASK  | BitString[4] |             |               | Na    | S           | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |
| 122 | 4             | FD_CHECK_MASK  | BitString[4] |             |               | Na    | S           | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |
| 123 |               | FD_FAIL_ALM    | DS-87        |             |               | Na    | D           | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 124 |               | FD_OFFSPEC_ALM | DS-87        |             |               | Na    | D           | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 125 |               | FD_MAINT_ALM   | DS-87        |             |               | Na    | D           | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 126 |               | FD_CHECK_ALM   | DS-87        |             |               | Na    | D           | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 127 | 4             | FD_FAIL_PRI    | Unsigned8    | 0-15        | 0             | Na    | S           | This parameter allows the user to specify the priority of this alarm category.  |
| 128 | 4             | FD_OFFSPEC_PRI | Unsigned8    | 0-15        | 0             | Na    | S           | This parameter allows the user to specify the priority of this alarm category.  |
| 129 | 4             | FD_MAINT_PRI   | Unsigned8    | 0-15        | 0             | Na    | S           | This parameter allows the user to specify the priority of this alarm category.  |
| 130 | 4             | FD_CHECK_PRI   | Unsigned8    | 0-15        | 0             | Na    | S           | This parameter allows the user to specify the priority of this alarm category.  |

| Idx | Type/<br>View | Parameter       | Data type  | Valid Range                               | Initial Value | Units | Store/Other | Description   |
|-----|---------------|-----------------|------------|---|---------------|-------|-------------|---|
| 131 | 3             | FD_SIMULATE     | DS-89      |   | disabled      | Na    | D           | This parameter allows the conditions to be manually supplied when simulation is enabled. When simulation is disabled both the diagnostic simulate value and the diagnostic value tracks the actual conditions. The simulate jumper is required for simulation to be enabled and while simulation is enabled the recommended action will show that simulation is active. |
| 132 | 1,3           | FD_RECOMMEN_ACT | Unsigned16 | 0-Not Initialized<br>1-No Action Required | 0             | Na    | D, RO       | This parameter is a device enumerated summarization of the most severe condition or conditions detected. The DD help should describe by enumerated action, what should be done to alleviate the condition or conditions. 0 is defined as Not Initialized, 1 is defined as No Action Required, all others defined by manuf.  |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static**  
**Gray Background Line: Custom Parameters**

## TB8BI – 8 Binary Inputs Transducer Block

### Description

This transducer block represents generically the transducers of all module types with 8 binary inputs (digitals).

### Diagnostic and troubleshooting

BLOCK\_ERR. Block configuration: this indication may occur due to the following problem:

- It exceeded the maximum number of the system I/O points, so it was not possible to allocate VAR\_NAME to this block points.

| Idx | Type/View | Parameter            | Data Type     | Valid Range                               | Initial Value | Units | Store/Other | Description  |
|-----|-----------|----------------------|---------------|---|---------------|-------|-------------|--|
| 1   | 1,2,3,4   | ST_REV               | Unsigned16    |   | 0             | None  | S / RO      |  |
| 2   |           | TAG_DESC             | OctString(32) |   | Spaces        | Na    | S           |  |
| 3   | 4         | STRATEGY             | Unsigned16    |   | 255           | None  | S           |  |
| 4   | 4         | ALERT_KEY            | Unsigned8     |   | 1             | None  | S           |  |
| 5   | 1,3       | MODE_BLK             | DS-69         |   | Auto          | Na    | S           |  |
| 6   | 1,3       | BLOCK_ERR            | Bitstring(2)  |   |               | E     | D / RO      |  |
| 7   |           | UPDATE_EVT           | DS-73         |   |               | Na    | D           | This alert is generated by any change to the static data.  |
| 8   |           | BLOCK_ALM            | DS-72         |   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | 4         | ACK_OPTION           | Bitstring(2)  | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |               | E     | S           | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 10  |           | TRANSDUCER_DIRECTORY | Unsigned16[1] |   | 0             |       | N / RO      | A directory that specifies the number and starting indices of the transducers in the transducer block.   |
| 11  | 1,2,3,4   | TRANSDUCER_TYPE      | Unsigned16    | FF-131                                    | 200           | E     | N / RO      | Identifies the transducer that follows.  |
| 12  |           | TRANSDUCER_TYPE_VER  | Unsigned16    |   |               |       | N / RO      | The version of the transducer identified by TRANSDUCER_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device.   |
| 13  | 1,3       | XD_ERROR             | Unsigned8     | Ver enumerações                           |               | E     | D / RO      | Error code for transducer.   |

| Idx | Type/<br>View | Parameter            | Data Type            | Valid Range                        | Initial Value        | Units | Store/<br>Other | Description   |
|-----|---------------|----------------------|----------------------|------------------------------------|----------------------|-------|-----------------|---|
| 14  |               | COLLECTION_DIRECTORY | Unsigned32[1]        |                                    | 0                    |       | N / RO          | A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.  |
| 15  |               | VAR_NAMES8           | VisibleString[8][32] |                                    | Manufacturer defined | Na    | S               | Array of 8 variable names to address each individual I/O.   |
| 16  |               | VAR_DIR8             | Unsigned8[8]         | 0:Undefined<br>1:Input<br>2:Output | 1's                  |       | S, RO           | Direction of the I/O: 0=undefined, 1=input, 2=output. If the I/O direction is fixed, this parameter is read only. If the module allows configurable I/O selection, the parameter is r/w and may be configured. The default is then 0. |
| 17  | 1,3           | PROCESS_VALUE_B_1    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 18  | 1,3           | PROCESS_VALUE_B_2    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 19  | 1,3           | PROCESS_VALUE_B_3    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 20  | 1,3           | PROCESS_VALUE_B_4    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 21  | 1,3           | PROCESS_VALUE_B_5    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 22  | 1,3           | PROCESS_VALUE_B_6    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 23  | 1,3           | PROCESS_VALUE_B_7    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 24  | 1,3           | PROCESS_VALUE_B_8    | DS-66                |                                    |                      |       | D / RO          | The measured Boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 25  | 1             | RACK                 | Unsigned8            | 0 to 15                            | 0                    | Na    | S / RO          | Rack number where the module is plugged in.   |
| 26  | 1             | SLOT                 | Unsigned8            | 0 to 3                             | 0                    | Na    | S / RO          | Slot number where the module is plugged in.   |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static  
 Gray Background Line: Custom Parameters

## TB8BO – 8 Binary Outputs Transducer Block

### Description

This transducer block represents generically the transducers of all module types with 8 binary outputs (digitals).

### Diagnostic and troubleshooting

BLOCK\_ERR. Block configuration: this indication may occur due to the following problem:

- It exceeded the maximum number of the system I/O points, so it was not possible to allocate VAR\_NAME to this block points.

| Idx | Type/View | Parameter            | Data Type     | Valid Range                               | Initial Value | Units | Store/Other | Description  |
|-----|-----------|----------------------|---------------|---|---------------|-------|-------------|--|
| 1   | 1,2,3,4   | ST_REV               | Unsigned16    |   | 0             | None  | S / RO      |  |
| 2   |           | TAG_DESC             | OctString(32) |   | Spaces        | Na    | S           |  |
| 3   | 4         | STRATEGY             | Unsigned16    |   | 255           | None  | S           |  |
| 4   | 4         | ALERT_KEY            | Unsigned8     |   | 1             | None  | S           |  |
| 5   | 1,3       | MODE_BLK             | DS-69         |   | Auto          | Na    | S           |  |
| 6   | 1,3       | BLOCK_ERR            | Bitstring(2)  |   |               | E     | D / RO      |  |
| 7   |           | UPDATE_EVT           | DS-73         |   |               | Na    | D           | This alert is generated by any change to the static data.  |
| 8   |           | BLOCK_ALM            | DS-72         |   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | 4         | ACK_OPTION           | Bitstring(2)  | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |               | E     | S           | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 10  |           | TRANSDUCER_DIRECTORY | Unsigned16[1] |   | 0             |       | N / RO      | A directory that specifies the number and starting indices of the transducers in the transducer block.   |
| 11  | 1,2,3,4   | TRANSDUCER_TYPE      | Unsigned16    | FF-131                                    | 200           | E     | N / RO      | Identifies the transducer that follows.  |
| 12  |           | TRANSDUCER_TYPE_VER  | Unsigned16    |   |               |       | N / RO      | The version of the transducer identified by TRANSDUCER_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device.   |
| 13  | 1,3       | XD_ERROR             | Unsigned8     | See enumerations                          |               | E     | D / RO      | Error code for transducer.   |



| Idx | Type/<br>View | Parameter            | Data Type            | Valid Range                        | Initial Value        | Units | Store/<br>Other | Description   |
|-----|---------------|----------------------|----------------------|------------------------------------|----------------------|-------|-----------------|---|
| 14  |               | COLLECTION_DIRECTORY | Unsigned32[1]        |                                    | 0                    |       | N / RO          | A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.  |
| 15  |               | VAR_NAMES8           | VisibleString[8][32] |                                    | Manufacturer defined | Na    | S               | Array of 8 variable names to address each individual I/O.   |
| 16  |               | VAR_DIR8             | Unsigned8[8]         | 0:Undefined<br>1:Input<br>2:Output | 2's                  |       | S, RO           | Direction of the I/O: 0=undefined, 1=input, 2=output. If the I/O direction is fixed, this parameter is read-only. If the module allows configurable I/O selection, the parameter is r/w and may be configured. The default is then 0. |
| 17  | 1,3           | PROCESS_VALUE_B_1    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 18  | 1,3           | PROCESS_VALUE_B_2    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status provided by the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.  |
| 19  | 1,3           | PROCESS_VALUE_B_3    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 20  | 1,3           | PROCESS_VALUE_B_4    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 21  | 1,3           | PROCESS_VALUE_B_5    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 22  | 1,3           | PROCESS_VALUE_B_6    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 23  | 1,3           | PROCESS_VALUE_B_7    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 24  | 1,3           | PROCESS_VALUE_B_8    | DS-66                |                                    |                      |       | D / RO          | The measured boolean value and status available to the Function Block or the final Boolean output value (FINAL_VALUE) and status available from the Function Block.   |
| 25  | 1             | RACK                 | Unsigned8            | 0 to 15                            | 0                    | Na    | N / RO          | Rack number where the module is plugged in.   |
| 26  | 1             | SLOT                 | Unsigned8            | 0 to 3                             | 0                    | Na    | N / RO          | Slot number where the module is plugged in.   |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static**  
**Gray Background Line: Custom Parameters**

## TB8AI – 8 Analog Inputs Transducer Block

### Description

This transducer block represents generically the transducers of all module types with 8 analog inputs.

This block performs:

- Reading the points of I/O scan database in percentage.
- Convert to engineering unit configured by user in PROCESS\_VALUE\_RANGE\_A\_x.

From A/D converter to % of 4-20mA

From % to EU\_0 to EU\_100 of PROCESS\_VALUE\_RANGE\_A\_x

| Scale and unit              | PROCESS_VALUE_RANGE_A_x (*) |
|-----------------------------|-----------------------------|
| 4-20mA or 0-20mA            | EU_0=4 or 0<br>EU_100=20    |
| % of 4-20mA                 | EU_0=0<br>EU_100=100        |
| % of 0-20mA                 | EU_0=20<br>EU_100=100       |
| 4mA → EU_0<br>20mA → EU_100 | EU_0 and EU_100 set         |

(\*)UNITS\_INDEX is not used in the conversion, only for indication

### Diagnostic and troubleshooting

BLOCK\_ERR. Block configuration: this indication may occur due to the following problem:

- It exceeded the maximum number of the system I/O points, so it was not possible to allocate VAR\_NAME to this block points.

| Idx | Type/View | Parameter  | Data type     | Valid Range                               | Initial Value | Units | Store/Other | Description  |
|-----|-----------|------------|---------------|---|---------------|-------|-------------|--|
| 1   | 1,2,3,4   | ST_REV     | Unsigned16    |   | 0             | None  | S / RO      |  |
| 2   |           | TAG_DESC   | OctString(32) |   | Spaces        | Na    | S           |  |
| 3   | 4         | STRATEGY   | Unsigned16    |   | 255           | None  | S           |  |
| 4   | 4         | ALERT_KEY  | Unsigned8     |   | 1             | None  | S           |  |
| 5   | 1,3       | MODE_BLK   | DS-69         |   | Auto          | Na    | S           |  |
| 6   | 1,3       | BLOCK_ERR  | Bitstring(2)  |   |               | E     | D / RO      |  |
| 7   |           | UPDATE_EVT | DS-73         |   |               | Na    | D           | This alert is generated by any change to the static data.  |
| 8   |           | BLOCK_ALM  | DS-72         |   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | 4         | ACK_OPTION | Bitstring(2)  | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |               | E     | S           | Selection of whether alarms associated with the block will be automatically acknowledged.  |

| Idx | Type/<br>View | Parameter            | Data type            | Valid Range                        | Initial Value        | Units | Store/<br>Other | Description  |
|-----|---------------|----------------------|----------------------|------------------------------------|----------------------|-------|-----------------|--|
| 10  |               | TRANSDUCER_DIRECTORY | Unsigned16[1]        |                                    | 0                    |       | N / RO          | A directory that specifies the number and starting indices of the transducers in the transducer block.   |
| 11  | 1,2,3,4       | TRANSDUCER_TYPE      | Unsigned16           | FF-131                             | 200                  | E     | N / RO          | Identifies the transducer that follows.  |
| 12  |               | TRANSDUCER_TYPE_VER  | Unsigned16           |                                    |                      |       | N / RO          | The version of the transducer identified by TRANSDUCER_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device. |
| 13  | 1,3           | XD_ERROR             | Unsigned8            | See enumerations                   |                      | E     | D / RO          | Error code for transducer.   |
| 14  |               | COLLECTION_DIRECTORY | Unsigned32[1]        |                                    | 0                    |       | N / RO          | A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.   |
| 15  |               | VAR_NAMES8           | VisibleString[8][32] |                                    | Manufacturer defined | Na    | S               | Array of 8 variable names to address each individual I/O.  |
| 16  |               | VAR_DIR8             | Unsigned8[8]         | 0:Undefined<br>1:Input<br>2:Output | 1's                  |       | S, RO           | Direction of the I/O: 0=undefined, 1=input, 2=output. If the I/O direction is fixed, this parameter is read-only. If the module allows configurable I/O selection, the parameter is r/w and may be configured. The default is then 0.                                  |
| 17  | 1,3           | PROCESS_VALUE_A_1    | DS-65                |                                    |                      | PVR1  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 18  | 1,3           | PROCESS_VALUE_A_2    | DS-65                |                                    |                      | PVR2  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 19  | 1,3           | PROCESS_VALUE_A_3    | DS-65                |                                    |                      | PVR3  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |

| Idx | Type/<br>View | Parameter               | Data type | Valid Range | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------------|-----------|-------------|---------------|-------|-----------------|--|
| 20  | 1,3           | PROCESS_VALUE_A_4       | DS-65     |             |               | PVR4  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 21  | 1,3           | PROCESS_VALUE_A_5       | DS-65     |             |               | PVR5  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 22  | 1,3           | PROCESS_VALUE_A_6       | DS-65     |             |               | PVR6  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 23  | 1,3           | PROCESS_VALUE_A_7       | DS-65     |             |               | PVR7  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 24  | 1,3           | PROCESS_VALUE_A_8       | DS-65     |             |               | PVR8  | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 25  | 2             | PROCESS_VALUE_RANGE_A_1 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |

| Idx | Type/<br>View | Parameter               | Data type | Valid Range | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------------|-----------|-------------|---------------|-------|-----------------|--|
| 26  | 2             | PROCESS_VALUE_RANGE_A_2 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 27  | 2             | PROCESS_VALUE_RANGE_A_3 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 28  | 2             | PROCESS_VALUE_RANGE_A_4 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 29  | 2             | PROCESS_VALUE_RANGE_A_5 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |

| Idx | Type/<br>View | Parameter               | Data type | Valid Range | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------------|-----------|-------------|---------------|-------|-----------------|--|
| 30  | 2             | PROCESS_VALUE_RANGE_A_6 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 31  | 2             | PROCESS_VALUE_RANGE_A_7 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 32  | 2             | PROCESS_VALUE_RANGE_A_8 | DS-68     |             | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_n defines the ends of the PROCESS_VALUE, the units of the PROCESS_VALUE, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 33  | 2             | IO_THRESHOLD_A_1        | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |
| 34  | 2             | IO_THRESHOLD_A_2        | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |

| Idx | Type/<br>View | Parameter        | Data type | Valid Range | Initial Value | Units | Store/<br>Other | Description   |
|-----|---------------|------------------|-----------|-------------|---------------|-------|-----------------|---|
| 35  | 2             | IO_THRESHOLD_A_3 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 36  | 2             | IO_THRESHOLD_A_4 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 37  | 2             | IO_THRESHOLD_A_5 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 38  | 2             | IO_THRESHOLD_A_6 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 39  | 2             | IO_THRESHOLD_A_7 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 40  | 2             | IO_THRESHOLD_A_8 | DS-179    |             |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device. |
| 41  | 1             | RACK             | Unsigned8 | 0 to 15     | 0             | Na    | N / RO          | Rack number where the module is plugged in.   |
| 42  | 1             | SLOT             | Unsigned8 | 0 to 3      | 0             | Na    | N / RO          | Slot number where the module is plugged in.   |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static**  
**Gray Background Line: Custom Parameters**

## TB4AO - 4 Analog Outputs Transducer Block

### Description

This transducer block represents generically the transducers of all module types with 4 analog outputs.

The TB4AO block expects to receive the function block values in percentage, then it converts the percentage to an unit appropriated for the module type (4-20 mA for DF46), which is not the value written in the transducer (D/A converter). The converted value will be used for indication in the PROCESS\_VALUE\_x parameters.

- It is still necessary to convert the value to be written in the sensor (D/A converter).

- The conversion to engineering unit is configured by user in PROCESS\_VALUE\_RANGE\_A\_x:

From 0-100% to EU\_0 - EU\_100 in mA, that is indicated in PROCESS\_VALUE\_x.

From EU\_0 - EU\_100 in mA to D/A converter

| Scale and unit | PROCESS_VALUE_RANGE_A_x (*) |
|----------------|-----------------------------|
| 4-20mA         | EU_0=4<br>EU_100=20         |
| 0-20mA         | EU_0=0<br>EU_100=20         |

(\*)UNITS\_INDEX fixed in mA, thus the indication in PROCESS\_VALUE\_RANGE\_A\_x always refers to current output of DF46.

### IO\_THRESHOLD\_A\_n

The elements OVERRNG\_LIM and UNDERRNG\_LIM are used to limit the value that will be sent to the D/A converter. The other elements of this structure are not used.

### PROCESS\_VALUE\_A\_n

These parameters indicate the value in mA of value assigned to the sensor, so now including the limitation imposed by IO\_THRESHOLD\_A\_n parameter, despite the description of the standard does not mention this.

### Diagnostic and troubleshooting

BLOCK\_ERR. Block configuration: this indication may occur due to the following problem:

- It exceeded the maximum number of the system I/O points, so it was not possible to allocate VAR\_NAME to this block points.

| Idx | Type/View | Parameter  | Data type     | Valid Range | Initial Value | Units | Store/Other | Description   |
|-----|-----------|------------|---------------|-------------|---------------|-------|-------------|---|
| 1   | 1,2,3,4   | ST_REV     | Unsigned16    |             | 0             | None  | S / RO      |   |
| 2   |           | TAG_DESC   | OctString(32) |             | Spaces        | Na    | S           |   |
| 3   | 4         | STRATEGY   | Unsigned16    |             | 255           | None  | S           |   |
| 4   | 4         | ALERT_KEY  | Unsigned8     |             | 1             | None  | S           |   |
| 5   | 1,3       | MODE_BLK   | DS-69         |             | Auto          | Na    | S           |   |
| 6   | 1,3       | BLOCK_ERR  | Bitstring(2)  |             |               | E     | D / RO      |   |
| 7   |           | UPDATE_EVT | DS-73         |             |               | Na    | D           | This alert is generated by any change to the static data. |



| Idx | Type/<br>View | Parameter            | Data type            | Valid Range                               | Initial<br>Value     | Units | Store/<br>Other | Description  |
|-----|---------------|----------------------|----------------------|---|----------------------|-------|-----------------|--|
| 8   |               | BLOCK_ALM            | DS-72                |   |                      | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   | 4             | ACK_OPTION           | Bitstring(2)         | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |                      | E     | S               | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 10  |               | TRANSDUCER_DIRECTORY | Unsigned16[1]        |   | 0                    |       | N / RO          | A directory that specifies the number and starting indices of the transducers in the transducer block.   |
| 11  | 1,2,3,4       | TRANSDUCER_TYPE      | Unsigned16           | FF-131                                    | 200                  | E     | N / RO          | Identifies the transducer that follows.  |
| 12  |               | TRANSDUCER_TYPE_VER  | Unsigned16           |   |                      |       | N / RO          | The version of the transducer identified by TRANSDUCER_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device.   |
| 13  | 1,3           | XD_ERROR             | Unsigned8            | See enumeration                           |                      | E     | D / RO          | Error code for transducer.   |
| 14  |               | COLLECTION_DIRECTORY | Unsigned32[1]        |   | 0                    |       | N / RO          | A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.   |
| 15  |               | VAR_NAMES4           | VisibleString[4][32] |   | Manufacturer defined | Na    | S               | Array of 4 variable names to address each individual I/O.  |
| 16  |               | VAR_DIR4             | Unsigned8[4]         | 0:Undefined<br>1:Input<br>2:Output        | 2's                  |       | S, RO           | Direction of the I/O: 0=undefined, 1=input, 2=output. If the I/O direction is fixed, this parameter is read-only. If the module allows configurable I/O selection, the parameter is r/w and may be configured. The default is then 0.  |
| 17  | 1,3           | PROCESS_VALUE_A_1    | DS-65                |   |                      | mA    | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |

| Idx | Type/<br>View | Parameter               | Data type | Valid Range                             | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------------|-----------|---|---------------|-------|-----------------|--|
| 18  | 1,3           | PROCESS_VALUE_A _2      | DS-65     |   |               | mA    | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 19  | 1,3           | PROCESS_VALUE_A _3      | DS-65     |   |               | mA    | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 20  | 1,3           | PROCESS_VALUE_A _4      | DS-65     |   |               | mA    | D / RO          | The measured analog value and status available to the Function Block (PRIMARY_VALUE) or the final analog output value (FINAL_VALUE) and status available to the Function Block.  |
| 21  | 2             | PROCESS_VALUE_RANGE_A_1 | DS-68     | EU_0:0-20<br>EU_100:0-20<br>EU_UNITS=mA | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_1 defines the ends of the PROCESS_VALUE_A_1, the units of the PROCESS_VALUE_A_1, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 22  | 2             | PROCESS_VALUE_RANGE_A_2 | DS-68     | EU_0:0-20<br>EU_100:0-20<br>EU_UNITS=mA | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_2 defines the ends of the PROCESS_VALUE_A_2, the units of the PROCESS_VALUE_A_2, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 23  | 2             | PROCESS_VALUE_RANGE_A_3 | DS-68     | EU_0:0-20<br>EU_100:0-20<br>EU_UNITS=mA | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_3 defines the ends of the PROCESS_VALUE_A_3, the units of the PROCESS_VALUE_A_3, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |

| Idx | Type/<br>View | Parameter               | Data type | Valid Range                             | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------------|-----------|---|---------------|-------|-----------------|--|
| 24  | 2             | PROCESS_VALUE_RANGE_A_4 | DS-68     | EU_0:0-20<br>EU_100:0-20<br>EU_UNITS=mA | 4..20mA       |       | S               | PROCESS_VALUE_RANGE_A_4 defines the ends of the PROCESS_VALUE_A_4, the units of the PROCESS_VALUE_A_4, and the decimal point position (number of significant digits to the right of the point). The engineering units must match the units selected in the parameter XD_SCALE of the Analog Input Block that reads the channel with this value, if a block AI is used. |
| 25  | 2             | IO_THRESHOLD_A_1        | DS-179    |   |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE_A_1 status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |
| 26  | 2             | IO_THRESHOLD_A_2        | DS-179    |   |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE_A_2 status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |
| 27  | 2             | IO_THRESHOLD_A_3        | DS-179    |   |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE_A_3 status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |
| 28  | 2             | IO_THRESHOLD_A_4        | DS-179    |   |               |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (PROCESS_VALUE_A_4 status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.  |
| 29  | 1             | RACK                    | Unsigned8 | 0 to 15                                 | 0             | Na    | N / RO          | Rack number where the module is plugged in.  |
| 30  | 1             | SLOT                    | Unsigned8 | 0 to 3                                  | 0             | Na    | N / RO          | Slot number where the module is plugged in.  |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static**  
**Gray Background Line: Custom Parameters**

## TBHG - Transducer Block for HART Gateway

This transducer block has the following characteristics:

- Specific and unique HSE WIO transducer block for the *WirelessHART Gateway*;
- Number<sup>1</sup> of HART and *WirelessHART* field devices supported;
- Status<sup>2</sup> about commissioning of HART and *WirelessHART* field devices;
- Live List<sup>3</sup> of HART and *WirelessHART* field devices.

### Description

- If the field device Long Tag is equal to the **Block tag** of some **TBWH**<sup>4</sup> block, then the device will be commissioned and it will be represented in the application by the corresponding **TBWH** block. If there is not equality mentioned between tags, then the device Long Tag is indicated, but it will be stated as "NotCommissioned."
- The execution of **TBHG** primarily triggers the execution of commissioned **TBWHs** for treatment of commands in the area of bypass and timeout in the updating of digital variables.

See in the following table the parameters description.

| Idx | Type/View | Parameter            | Data Type     | Valid Range | Initial Value | Units | Store/Other | Description  |
|-----|-----------|----------------------|---------------|-------------|---------------|-------|-------------|--|
| 1   | 1,2,3,4   | ST_REV               | Unsigned16    |             | 0             | None  | S / RO      |  |
| 2   |           | TAG_DESC             | OctString(32) |             | Spaces        | Na    | S           |  |
| 3   | 4         | STRATEGY             | Unsigned16    |             | 255           | None  | S           |  |
| 4   | 4         | ALERT_KEY            | Unsigned8     |             | 1             | None  | S           |  |
| 5   | 1,3       | MODE_BLK             | DS-69         |             | Auto          | Na    | S           |  |
| 6   | 1,3       | BLOCK_ERR            | Bitstring(2)  |             |               | E     | D / RO      |  |
| 7   |           | UPDATE_EVT           | DS-73         |             |               | Na    | D           | This alert is generated by any change to the static data.  |
| 8   |           | BLOCK_ALM            | DS-72         |             |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 9   |           | TRANSDUCER_DIRECTORY | Unsigned16[1] |             | 0             |       | N / RO      | A directory that specifies the number and starting indices of the transducers in the transducer block.   |
| 10  | 1,2,3,4   | TRANSDUCER_TYPE      | Unsigned16    | FF-131      | 203           | E     | N / RO      | Identifies the transducer that follows.  |
| 11  | 2,4       | TRANSDUCER_TYPE_VER  | Unsigned16    |             |               |       | N / RO      | The version of the transducer identified by TRANSDUCER_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device.   |

<sup>1</sup> See N\_DEV\_SUPPORTED parameter.

<sup>2</sup> See LIVE\_LIST\_ST parameter.

<sup>3</sup> See LIVE\_LIST\_TAG\_A, LIVE\_LIST\_TAG\_B, LIVE\_LIST\_TAG\_C and LIVE\_LIST\_TAG\_D parameters. Each one of the three first mentioned parameters inform up to 32 field devices in the Live List. The last parameter informs just four.

<sup>4</sup> See Transducer Block for *WirelessHART* (TBWH), also specific for the *WirelessHART* gateway.

| Idx | Type/<br>View | Parameter            | Data Type             | Valid Range   | Initial<br>Value   | Units | Store/<br>Other | Description  |
|-----|---------------|----------------------|-----------------------|---|--------------------|-------|-----------------|--|
| 12  | 1,3           | XD_ERROR             | Unsigned8             | See enumerations  |                    | E     | D / RO          | Error code for transducer.   |
| 13  |               | COLLECTION_DIRECTORY | Unsigned32[1]         |   | 0                  |       | N / RO          | A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.   |
| 14  | 2             | HART_EXPDP_DEV_INFO  | DS-175                |   |                    |       | S               | Expected HART Device information for use by asset managing host and configuration host for DD location and display validity in off-line mode   |
| 15  | 2             | HART_ACTL_DEV_INFO   | DS-175                |   |                    |       | S, RO           | Actual HART Device information for use by asset managing host and configuration host for DD location and display validity in online mode.  |
| 16  |               | HART_CMD             | OctetString [256]     |   | Null               | Na    | D               | HART digital protocol command buffer for use by asset managing host. No need for user access.  |
| 17  |               | HART_RESP            | OctetString [256]     |   | Null               | Na    | D, RO           | HART digital protocol response buffer for use by asset managing host. No need for user access.   |
| 18  |               | HART_IND             | DS-184                |   |                    |       | D, RO           | HART response available indicator. An alert object for notifying an asset managing host.   |
| 19  |               | HART_COM_STAT        | Unsigned8             | 0:Idle<br>1:Writing<br>2:Awaiting Response<br>3:Receiving<br>4-Timed-out<br>5-Completed | 1                  |       | D, RO           | HART communication status (idle, busy)   |
| 20  | 4             | HART_IND_PRI         | Unsigned8             | 0 to 2  | 0                  | Na    | S               | Priority of response indication.   |
| 21  | 4             | ACK_OPTION           | Bitstring(2)          | 0:Auto Ack Disabled<br>1:Auto Ack Enabled   |                    | E     | S               | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 22  | 4             | N_DEV_SUPPORTED      | Unsigned16            |   | 100                | Na    | S, RO           | Number of devices supported by the gateway.  |
| 23  | 3             | LIVE_LIST_ST         | Unsigned8[100]        | 0: NotCommissioned<br>1: Commissioned   | 0: NotCommissioned | E     | D, RO           | Commissioned status of the devices alive in the network connected to the gateway. (0 = Not Commissioned; 1= Commissioned)  |
| 24  |               | LIVE_LIST_TAG_A      | VisibleString[32][32] |   | Blanks             | Na    | D, RO           | TAGs of the devices alive in the network connected to the gateway. LIVE_LIST_TAG_A holds the TAGs of the first 32 devices. If N_DEV_SUPPORTED is greater than 32, then multiples LIVE_LIST_TAG_x (where x = A, B, C...) should be included in order to list all N_DEV_SUPPORTED possible devices in the live list.           |
| 25  |               | LIVE_LIST_TAG_B      | VisibleString[32][32] |   | Blanks             | Na    | D, RO           | TAGs of the devices alive in the network connected to the gateway. LIVE_LIST_TAG_B holds the TAGs of the second group of 32 devices. If N_DEV_SUPPORTED is greater than 32, then multiples LIVE_LIST_TAG_x (where x = A, B, C...) should be included in order to list all N_DEV_SUPPORTED possible devices in the live list. |

| Idx | Type/<br>View | Parameter       | Data Type                 | Valid Range | Initial<br>Value | Units | Store/<br>Other | Description  |
|-----|---------------|-----------------|---------------------------|-------------|------------------|-------|-----------------|--|
| 26  |               | LIVE_LIST_TAG_C | VisibleString[32]<br>[32] |             | Blanks           | Na    | D, RO           | TAGs of the devices alive in the network connected to the gateway.<br>LIVE_LIST_TAG_C holds the TAGs of the third group of 32 devices. If N_DEV_SUPPORTED is greater than 32, then multiples LIVE_LIST_TAG_x (where x = A, B, C...) should be included in order to list all N_DEV_SUPPORTED possible devices in the live list. |
| 27  |               | LIVE_LIST_TAG_D | VisibleString[4]<br>[32]  |             | Blanks           | Na    | D, RO           | TAGs of the devices alive in the network connected to the gateway.<br>LIVE_LIST_TAG_D holds the TAGs of the fourth group of 4 devices. If N_DEV_SUPPORTED is greater than 32, then multiples LIVE_LIST_TAG_x (where x = A, B, C...) should be included in order to list all N_DEV_SUPPORTED possible devices in the live list. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static  
**Gray Background Line: Custom Parameters**

## TBWH - Transducer Block for *WirelessHART*

This transducer block has the main characteristics:

- HART<sup>5</sup> digital variables read from the field device;
- Names (variable names) to identify the HART digital variables.
- Bypass of HART command;
- Standard Field Diagnostics (FD).

### Description

The TBWH is a HSE WIO transducer block used to map each one of the HART and *WirelessHART* field devices. The tag of the **TBWH** block is very important and should be configured with the tag6 of the field device that will be mapped. Can be instantiated as many **TBWH** blocks, as there are the HART and *WirelessHART*<sup>7</sup> field devices.

### Update mechanism of the field device configuration parameter

When the *WirelessHART* gateway detects the field device configuration changes and the changed variable is mapped in this block, the **ST\_REV** parameter value is automatically incremented and can generate an event<sup>8</sup>.

### Mechanism of the HART Bypass command

The bypass area can be used to send a HART command to the field device. A client, for example an Asset Management application, can benefit from this mechanism.

Since the status of **HART\_COM\_STAT** parameter is **Idle**, the HART command can be written in the **HART\_CMD** parameter. After writing, the status of **HART\_COM\_STAT** parameter will change to **Busy**<sup>9</sup> and the command will be sent to device that will process it and will return a corresponding response. The response to the HART command will be updated in the **HART\_RESP** parameter. In addition, an event<sup>10</sup> will be reported on the HSE network and the status of **HART\_COM\_STAT** parameter will return to **Idle**.

To access a field device response, the client may choose one of the methods below:

- a) Read the available response directly from the **HART\_RESP** parameter. Given the described bypass mechanism, we may conclude that this is only possible after the transition of **HART\_COM\_STAT** parameter value from **Busy** to **Idle**
- b) Subscription of the event that will be automatically reported on the HSE network.

### Diagnostic and troubleshooting

The **BLOCK\_ERR** parameter may indicate **Block configuration** due to the following problems:

- A field device there is not in the Live List of the **TBHG** block with Long Tag equals to the tag of the **TBWH** block.
- The **HART\_EXPD\_DEV\_INFO** parameter is not parameterized as indicated by the **HART\_ACTL\_DEV\_INFO** parameter.

The status of the digital HART variables indicates events, for example, failure in the communication with the field device and failure in the device sensor. The table below lists four conditions and their hierarchical status<sup>11</sup> to the HART variables

<sup>5</sup> HART\_PV, HART\_SV, HART\_TV, HART\_QV, HART\_5V, HART\_6V, HART\_7V, HART\_8V and PRIMARY\_VALUE parameters.

<sup>6</sup> The Long Tag has been introduced in the HART protocol from version 6. Knowing this, the field device tag is the Long Tag of HART field device for all devices that implement version 6 or 7 of the HART protocol. Otherwise the field device tag will be considered equal to the Message of the HART device. The Message is defined by the HART protocol and is 32 characters long.

<sup>7</sup> See Section 11 – Technical Specifications of DFI302 manual for further about number of supported devices.

<sup>8</sup> See UPDATE\_EVT parameter.

<sup>9</sup> This prevents the client writing another command before receiving the response for the newly written command.

<sup>10</sup> See HART\_IND parameter. This parameter also has the received response for the HART command.

<sup>11</sup> If two or more conditions are present simultaneously, the lower hierarchical condition is used to determine the status. For example, if conditions 2 and 3 are present, the status is determined by the condition 2.

| HIERARCHY OF CONDITION | DESCRIPTION OF THE CONDITION  | STATUS (HART_PV A HART_8V)              |
|------------------------|---|---|
| 1                      | Not commissioned  | Bad: Out of Service                     |
| 2                      | -Commissioned<br>-HART_BAD_TMOU different from zero<br>-Elapsed time since the last update greater than HART_BAD_TMOU   | Bad: No Communication Last Usable Value |
| 3                      | - Commissioned<br>-HART_UNC_TMOU different from zero<br>- Elapsed time since the last update greater than HART_UNC_TMOU | Uncertain: Last Usable Value            |
| 4                      | - Commissioned<br>-There was no timeout.  | Status conversion from HART to FF       |

### Indication of Mismatch in the Field Diagnostics

The indication of *Mismatch* in the *Field Diagnostics* may occur due to the following situations:

- Mismatch between field device tag and **TBWH** block tag, that is done regardless of mode;
- Mismatch between **HART\_EXPD\_DEV\_INFO** and **HART\_ACTL\_DEV\_INFO** that is done only in Auto (Automatic) mode.

### Definition of Field Diagnostics for the *WirelessHART* Gateway

| FD  |                 | HART_DIAG_MAP |       |                                    | Default      |                |              |             |
|-----|-----------------|---------------|-------|------------------------------------|--------------|----------------|--------------|-------------|
| Bit | Bit Description | Index         | Value | Enumeration String                 | FD_MAINT_MAP | FD_OFFSPEC_MAP | FD_CHECK_MAP | FD_FAIL_MAP |
| 0   | FD_0            | 1             | 253   | "Maintenance worker is checking"   |              |                | x            |             |
| 1   | FD_1            | 2             | 1     | "Primary Variable Out-of-Limits"   |              | x              |              |             |
| 2   | FD_2            | 3             | 2     | "Non-Prim Variable Out-of-Limits"  |              | x              |              |             |
| 3   | FD_3            | 4             | 6     | "HART Cold Start"                  |              |                | x            |             |
| 4   | FD_4            | 5             | 7     | "HART Configuration Changed"       |              |                |              |             |
| 5   | FD_5            | 6             | 8     | "HART Device Malfuction"           |              |                |              | x           |
| 6   | FD_6            | 7             | 57    | "HART Maintenance Required"        | x            |                |              |             |
| 7   | FD_7            | 8             | 59    | "Critical Power Failure"           | x            |                | x            |             |
| 8   | FD_8            | 9             | 73    | "Simulation Active"                |              |                | x            |             |
| 9   | FD_9            | 10            | 77    | "Voltage Conditions out of Range"  |              | x              |              |             |
| 10  | FD_10           | 11            | 78    | "Environmental Cond. out of Range" |              | x              |              |             |
| 11  | FD_11           | 12            | 105   | "Capacity denied"                  |              | x              |              |             |
| 12  | FD_12           | 13            | 240   | "Electronics or memory Defect"     |              |                |              | x           |
| 13  | FD_13           | 14            | 252   | "Mismatch"                         |              |                |              |             |
| 14  | FD_14           | 15            | 255   | "Reserved for FF use"              |              |                |              |             |
| 15  | FD_15           | 16            | 255   | "Reserved for FF use"              |              |                |              |             |
| 16  | FD_16           | 17            | 0     | "Unassigned"                       |              |                |              |             |
| 17  | FD_17           | 18            | 0     | "Unassigned"                       |              |                |              |             |
| 18  | FD_18           | 19            | 0     | "Unassigned"                       |              |                |              |             |
| 19  | FD_19           | 20            | 0     | "Unassigned"                       |              |                |              |             |
| 20  | FD_20           | 21            | 0     | "Unassigned"                       |              |                |              |             |
| 21  | FD_21           | 22            | 0     | "Unassigned"                       |              |                |              |             |
| 22  | FD_22           | 23            | 0     | "Unassigned"                       |              |                |              |             |
| 23  | FD_23           | 24            | 0     | "Unassigned"                       |              |                |              |             |
| 24  | FD_24           | 25            | 0     | "Unassigned"                       |              |                |              |             |
| 25  | FD_25           | 26            | 0     | "Unassigned"                       |              |                |              |             |
| 26  | FD_26           | 27            | 0     | "Unassigned"                       |              |                |              |             |
| 27  | FD_27           | 28            | 0     | "Unassigned"                       |              |                |              |             |
| 28  | FD_28           | 29            | 0     | "Unassigned"                       |              |                |              |             |
| 29  | FD_29           | 30            | 0     | "Unassigned"                       |              |                |              |             |
| 30  | FD_30           | 31            | 0     | "Unassigned"                       |              |                |              |             |
| 31  | FD_31           | 32            | 0     | "Unassigned"                       |              |                |              |             |
|     |                 |               |       | Valor (Hex)                        | 0x00000040   | 0x00000e06     | 0x00000189   | 0x00001020  |

### Diagnostic and troubleshooting

1. **BLOCK\_ERR**. Block configuration: this indication may occur due to the following problems:

- Block is not associated to a position in the TBHG.
- It was not possible to allocate VAR\_NAMES to this block.

See in the following table the description of the **TBWH** block parameters.



| Idx | Type/<br>View | Parameter                | Data type            | Valid Range      | Initial<br>Value | Units | Store/<br>Other | Description   |
|-----|---------------|--------------------------|----------------------|------------------|------------------|-------|-----------------|---|
| 1   | 1,2,3,4       | ST_REV                   | Unsigned16           |                  | 0                | None  | S / RO          |   |
| 2   |               | TAG_DESC                 | OctString(32)        |                  | Spaces           | Na    | S               |   |
| 3   | 4             | STRATEGY                 | Unsigned16           |                  | 255              | None  | S               |   |
| 4   | 4             | ALERT_KEY                | Unsigned8            |                  | 1                | None  | S               |   |
| 5   | 1,3           | MODE_BLK                 | DS-69                |                  | Auto             | Na    | S               |   |
| 6   | 1,3           | BLOCK_ERR                | Bitstring(2)         |                  |                  | E     | D / RO          |   |
| 7   |               | UPDATE_EVT               | DS-73                |                  |                  |       | D               |   |
| 8   |               | BLOCK_ALM                | DS-72                |                  |                  |       | D               |   |
| 9   |               | TRANSDUCER_DIRECT<br>ORY | Unsigned16           |                  |                  |       | N / RO          | It is used to select several Transducer Blocks.   |
| 10  | 1,2,3,4       | TRANSDUCER_TYPE          | Unsigned16           |                  | 202              |       | N / RO          | Indicates the type of Transducer according to its class.  |
| 11  | 2,4           | TRANSDUCER_TYPE_V<br>ER  | Unsigned16           |                  |                  |       | N / RO          | The version of the transducer identified by TRANSDUCERT_TYPE in the form 0xAABB where AA is the major revision of the transducer specification on which the transducer is based, and BB is a revision number assigned and controlled by the manufacturer of the device. |
| 12  | 1,3           | XD_ERROR                 | Unsigned8            |                  |                  |       | D / RO          | It is used to indicate calibration status.  |
| 13  |               | COLLECTION_DIRECTO<br>RY | Unsigned16           |                  |                  |       | N / RO          | Specifies the number of transducer index into Transducer Block.   |
| 14  | 2             | HART_EXPD_DEV_INFO       | DS-175               |                  |                  |       | S               | Expected HART Device information for use by asset managing host and configuration host for DD location and display validity in off-line mode  |
| 15  | 2             | HART_ACTL_DEV_INFO       | DS-175               |                  |                  |       | S, RO           | Actual HART Device information for use by asset managing host and configuration host for DD location and display validity in online mode.   |
| 16  |               | HART_CMD                 | OctetString<br>[256] |                  | Null             | Na    | D               | HART digital protocol command buffer for use by asset managing host. No need for user access.   |
| 17  |               | HART_RESP                | OctetString<br>[256] |                  | Null             | Na    | D, RO           | HART digital protocol response buffer for use by asset managing host. No need for user access.  |
| 18  |               | HART_IND                 | DS-184               |                  |                  |       | D, RO           | HART response available indicator. An alert object for notifying an asset managing host.  |
| 19  |               | HART_COM_STAT            | Unsigned8            | 0:Idle<br>1:Busy | 1                |       | D, RO           | HART communication status (0:Idle, 1:Busy)  |
| 20  | 4             | HART_IND_PRI             | Unsigned8            | 0 to 2           | 0                | Na    | S               | Priority of response indication.  |
| 21  | 2             | HART_DIAG_MAP            | Unsigned8[3<br>2]    |                  |                  |       | S               | An array of 1-octet enumeration with 32 elements, one for each bit, index 1 for bit-0 through index 32 for bit-31, defining the mapping between HART diagnostics status flags and the FF Field Diagnostics parameters specified in FF-912.                              |

| Idx | Type/<br>View | Parameter       | Data type    | Valid Range | Initial<br>Value              | Units            | Store/<br>Other | Description  |
|-----|---------------|-----------------|--------------|-------------|-------------------------------|------------------|-----------------|--|
| 22  | 1             | HART_TSTAMP     | Time Value   |             |                               |                  | D, RO           | Time stamp of the last update of digital HART variables.   |
| 23  | 4             | HART_BAD_TMOUT  | Unsigned16   |             | 0                             | s                | S               | Time interval, in seconds, to wait for an update of the digital HART variables before changing the status to Bad/No_Comm. A zero value means “do not timeout”.   |
| 24  | 4             | HART_UNC_TMOUT  | Unsigned16   |             | 0                             | s                | S               | Time interval, in seconds, to wait for an update of the digital HART variables before changing the status to Unc/Last Usable Value. A zero value means “do not timeout”.   |
| 25  | 4             | HART_VAR_CODES8 | Unsigned8[8] |             | 255                           |                  | S               | HART Device Variable Code for process data variables (HART_PV, HART_SV, HART_TV, HART_4V, HART_5V, HART_6V, HART_7V, HART_8V). Value 250 for an element from this array means no HART variable is associated with the related process data variable. |
| 26  | 1             | HART_PV         | DS-65        |             |                               | VAR_U<br>NITS9.1 | D, RO           | For wired HART, this is the converted 4-20mA value. For <i>WirelessHART™</i> , this is not applicable.   |
| 27  | 1             | HART_SV         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.2 | D, RO           | Digital form of the HART secondary variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 28  | 1             | HART_TV         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.3 | D, RO           | Digital form of the HART tertiary variable.<br>Status of “Bad/Not_Connected” if unused.  |
| 29  | 1             | HART_QV         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.4 | D, RO           | Digital form of the HART 4th variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 30  | 1             | HART_5V         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.5 | D, RO           | Digital form of the HART 5th variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 31  | 1             | HART_6V         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.6 | D, RO           | Digital form of the HART 6th variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 32  | 1             | HART_7V         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.7 | D, RO           | Digital form of the HART 7th variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 33  | 1             | HART_8V         | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.8 | D, RO           | Digital form of the HART 8th variable.<br>Status of “Bad/Not_Connected” if unused.   |
| 34  | 1             | ANALOG_VALUE    | DS-65        |             | 0<br>Bad:Not<br>connect<br>ed | VAR_U<br>NITS9.9 | D, RO           | For wired HART, this is the converted 4-20mA value to engineering unit based on HART_LRL and HART_URL. For <i>WirelessHART™</i> , this is not applicable.  |

| Idx | Type/<br>View | Parameter         | Data type            | Valid Range                               | Initial<br>Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------|----------------------|---|------------------|-------|-----------------|--|
| 35  | 4             | VAR_UNITS9        | Unsigned16[9]        |   |                  | E     | S, RO           | This is an array of the Fieldbus Foundation engineering units' codes corresponding to the HART engineering units' codes for each of up to 9 variables. VAR_UNITS9.1 through VAR_UNITS9.8 are the variable units for HART_PV through HART_8V, and VAR_UNITS9.9 is the variable unit for ANALOG_VALUE. |
| 36  |               | VAR_NAMES9        | VisibleString[9][32] |   | blanks           |       | S               | This is an array of the names of the HART device variables. VAR_NAMES9.1 through VAR_NAMES9.8 are the variable names for HART_PV through HART_8V, and VAR_NAMES9.9 is the variable name for ANALOG_VALUE.  |
| 37  | 4             | VAR_DIR9          | Unsigned8[9]         | 0:Undefined<br>1:Input<br>2:Output        | 0                |       | S               | This is an array of 1-octet enumeration with 9 elements, one for each HART device variables, defining the direction of each variable.  |
| 38  | 2             | HART_URL          | Float                |   |                  |       | S, RO           | HART device upper range limit value obtained from the device. Used for the 20mA value of 4 to 20mA conversion. Read-only in transducer block. User must change via HART configuration tool.  |
| 39  | 2             | HART_LRL          | Float                |   |                  |       | S, RO           | HART device lower range limit value obtained from the device. Used for the 4mA value of 4 to 20mA conversion. Read-only in transducer block. User must change via HART configuration tool.   |
| 40  | 2             | IO_THRESHOLD      | DS-179               |   |                  |       | S               | Thresholds of over-range and under-range bad and uncertain status indications on input (ANALOG_VALUE status) and over-range and under-range limits on output for the 4-20mA signal of the HART device.   |
| 41  | 4             | ACK_OPTION        | Bitstring(2)         | 0:Auto Ack Disabled<br>1:Auto Ack Enabled |                  | E     | S               | Selection of whether alarms associated with the block will be automatically acknowledged.  |
| 42  | 4             | FD_VER            | Unsigned16           |   |                  | Na    | S, RO           | A parameter equal to the value of the major version of the Field Diagnostics specification that this device was designed to.   |
| 43  | 1,3           | FD_FAIL_ACTIVE    | BitString[4]         |   |                  | Na    | D, RO           | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.  |
| 44  | 1,3           | FD_OFFSPEC_ACTIVE | BitString[4]         |   |                  | Na    | D, RO           | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.  |

| Idx | Type/<br>View | Parameter       | Data type    | Valid Range | Initial Value | Units | Store/<br>Other | Description   |
|-----|---------------|-----------------|--------------|-------------|---------------|-------|-----------------|---|
| 45  | 1,3           | FD_MAINT_ACTIVE | BitString[4] |             |               | Na    | D, RO           | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 46  | 1,3           | FD_CHECK_ACTIVE | BitString[4] |             |               | Na    | D, RO           | This parameter reflects the error conditions that are being detected as active as selected for this category. It is a bit string, so that multiple conditions may be shown.   |
| 47  | 4             | FD_FAIL_MAP     | BitString[4] |             |               | Na    | S               | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 48  | 4             | FD_OFFSPEC_MAP  | BitString[4] |             |               | Na    | S               | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 49  | 4             | FD_MAINT_MAP    | BitString[4] |             |               | Na    | S               | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 50  | 4             | FD_CHECK_MAP    | BitString[4] |             |               | Na    | S               | This parameter maps conditions to be detected as active for this alarm category. Thus the same condition may be active in all, some, or none of the 4 alarm categories.   |
| 51  | 4             | FD_FAIL_MASK    | BitString[4] |             |               | Na    | S               | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |
| 52  | 4             | FD_OFFSPEC_MASK | BitString[4] |             |               | Na    | S               | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |
| 53  | 4             | FD_MAINT_MASK   | BitString[4] |             |               | Na    | S               | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition. |

| Idx | Type/<br>View | Parameter      | Data type    | Valid Range | Initial<br>Value | Units | Store/<br>Other | Description   |
|-----|---------------|----------------|--------------|-------------|------------------|-------|-----------------|---|
| 54  | 4             | FD_CHECK_MASK  | BitString[4] |             |                  | Na    | S               | This parameter allows the user to suppress any single or multiple conditions that are active, in this category, from being broadcast to the host through the alarm parameter. A bit equal to '1' will mask i.e. inhibit the broadcast of a condition, and a bit equal to '0' will unmask i.e. allow broadcast of a condition.   |
| 55  |               | FD_FAIL_ALM    | DS-87        |             |                  | Na    | D               | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 56  |               | FD_OFFSPEC_ALM | DS-87        |             |                  | Na    | D               | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 57  |               | FD_MAINT_ALM   | DS-87        |             |                  | Na    | D               | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 58  |               | FD_CHECK_ALM   | DS-87        |             |                  | Na    | D               | This parameter is used primarily to broadcast a change in the associated active conditions, which are not masked, for this alarm category to a Host System.   |
| 59  | 4             | FD_FAIL_PRI    | Unsigned8    | 0-15        | 0                | Na    | S               | This parameter allows the user to specify the priority of this alarm category.  |
| 60  | 4             | FD_OFFSPEC_PRI | Unsigned8    | 0-15        | 0                | Na    | S               | This parameter allows the user to specify the priority of this alarm category.  |
| 61  | 4             | FD_MAINT_PRI   | Unsigned8    | 0-15        | 0                | Na    | S               | This parameter allows the user to specify the priority of this alarm category.  |
| 62  | 4             | FD_CHECK_PRI   | Unsigned8    | 0-15        | 0                | Na    | S               | This parameter allows the user to specify the priority of this alarm category.  |
| 63  | 3             | FD_SIMULATE    | DS-89        |             | disabled         | Na    | D               | This parameter allows the conditions to be manually supplied when simulation is enabled. When simulation is disabled both the diagnostic simulate value and the diagnostic value tracks the actual conditions. The simulate jumper is required for simulation to be enabled and while simulation is enabled the recommended action will show that simulation is active. |

| Idx | Type/<br>View | Parameter         | Data type  | Valid Range   | Initial Value | Units | Store/<br>Other | Description  |
|-----|---------------|-------------------|------------|---|---------------|-------|-----------------|--|
| 64  | 1,3           | FD_RECOMMEN_ACT   | Unsigned16 | 0-Not Initialized<br>1-No Action Required<br>2-Check primary sensor range<br>3- Check non-primary sensor range<br>4--No Action Required<br>5--No Action Required<br>6- Check primary sensor<br>7-HART device requires maintenance<br>8-Check battery charge<br>9--No Action Required<br>10-Check power supply voltage<br>11-Check environment conditions<br>12-Check HART device specification<br>13-Replace the electronics of HART device | 0             | Na    | D, RO           | This parameter is a device enumerated summarization of the most severe condition or conditions detected. The DD help should describe by enumerated action, what should be done to alleviate the condition or conditions. 0 is defined as Not Initialized, 1 is defined as No Action Required, all others defined by manuf. |
| 65  | 4             | HART_BURST_CTRL_1 | DS-183     |   |               |       | S, RO           | Data structure that describes burst control information configured in a HART device.   |
| 66  | 4             | HART_BURST_CTRL_2 | DS-183     |   |               |       | S, RO           | Data structure that describes burst control information configured in a HART device.   |
| 67  | 4             | HART_BURST_CTRL_3 | DS-183     |   |               |       | S, RO           | Data structure that describes burst control information configured in a HART device.   |
| 68  | 4             | HART_BURST_CTRL_4 | DS-183     |   |               |       | S, RO           | Data structure that describes burst control information configured in a HART device.   |
| 69  |               | LOCAL_MOD_MAP     | Unsigned8  | 0 to 99,<br>255-non mapped  | 255           | Na    | S               | Define the Modbus addresses.   |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - Static**  
**Gray Background Line: Custom Parameters**

## Input Function Blocks

### AI - Analog Input

#### Overview

The Analog Input block takes the input data from the Transducer block, selected by channel number, and makes it available to other function blocks at its output.

#### Description

The AI block is connected to the transducer block through the CHANNEL parameter that must match with the following parameter in the transducer block:

- SENSOR\_TRANSDUCER\_NUMBER parameter for the TT302
- TERMINAL\_NUMBER parameter for the IF302

The CHANNEL parameter must be set to 1 (one) if the AI block is running in the LD302, and no configuration is necessary in the transducer block to connect it to the AI block.

Transducer scaling (XD\_SCALE) is applied to the value from the channel to produce the FIELD\_VAL in percent. The XD\_SCALE engineering units code and range must be suitable to the sensor of transducer block connected to the AI block, otherwise a block alarm indicating configuration error will be generated.

The L\_TYPE parameter determines how the values passed by the transducer block will be used into the block. The options are:

- Direct - the transducer value is passed directly to the PV. Therefore OUT\_SCALE is useless.
- Indirect - the PV value is the FIELD\_VAL value converted to the OUT\_SCALE.
- Indirect with Square Root - the PV value is square root of the FIELD\_VAL converted to the OUT\_SCALE.

PV and OUT parameters always have identical scaling based on OUT\_SCALE.

The LOW\_CUT parameter is an optional characteristic that may be used to eliminate noise near zero for a flow sensor. The LOW\_CUT parameter has a corresponding "Low cutoff" option in the IO\_OPTS bit string. If the option bit is true, any calculated output below the low cutoff value (LOW\_CUT) will be changed to zero.

#### BLOCK\_ERR

The BLOCK\_ERR of the AI block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when one or more of the following situations occur:
  - When the CHANNEL or L\_TYPE parameters have an invalid value;
  - When the XD\_SCALE does not have a suitable engineering unit or range for the sensor of transducer block.
  - When it is not compatible the CHANNEL parameter and HC configuration (DFI302).
- Simulate Active – When the Simulate is active.
- Input Failure – I/O module failure (DFI302)
- Out of Service – When the block is in O/S mode.

#### Supported Modes

O/S, Man and Auto.

#### Status Handling

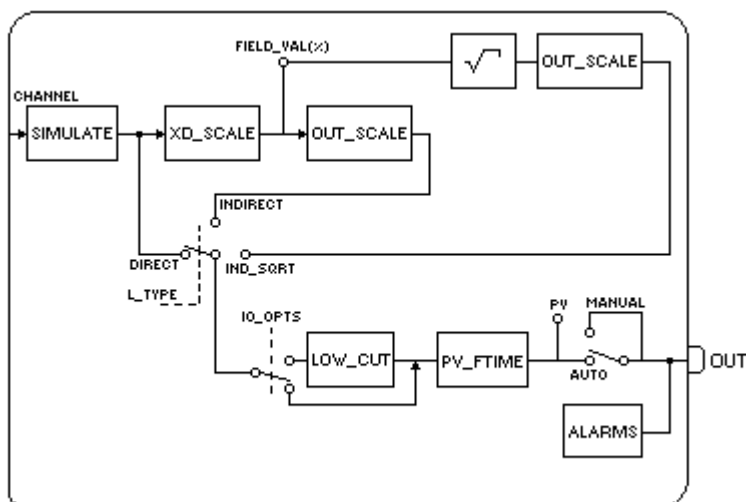
The AI block does not support cascade path. Then, the output status has not a cascade sub-status.

When the OUT value exceeds the OUT\_SCALE range and no worse condition exists in the block then the OUT status will be "uncertain, EU Range Violation".

The following options from STATUS\_OPTS apply, where Limited refers to the sensor limits: (see in the Function block options to more details about each option)

- Propagate Fault Forward
- Uncertain if Limited
- BAD if Limited
- Uncertain if Man mode

## Schematic



## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/Options   | Default Value   | Units | Store/Mode | Description  |
|-----|-------------|--------------------|---|---|-------|------------|--|
| 1   | ST_REV      | Unsigned16         |   | 0   | None  | S/RO       |  |
| 2   | TAG_DESC    | OctString(32)      |   | Spaces  | Na    | S          |  |
| 3   | STRATEGY    | Unsigned16         |   | 0   | None  | S          |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255  | 0   | None  | S          |  |
| 5   | MODE_BLK    | DS-69              |   | O/S   | Na    | S          | See Mode Parameter.  |
| 6   | BLOCK_ERR   | Bitstring(2)       |   |   | E     | D / RO     |  |
| 7   | PV          | DS-65              |   |   | PV    | D / RO     | Process analog value for use in executing the function.  |
| 8   | OUT         | DS-65              | OUT_SCALE +/- 10%   |   | OUT   | D / Man    | The analog value calculated as a result of executing the function.   |
| 9   | SIMULATE    | DS-82              | 1: Disable ;<br>2: Active<br>are the Enable/Disable options.                          | Disable   |       | D          | Allows the input value to be manually supplied when simulate is enabled. In this case, the simulate value and status will be the PV value.   |
| 10  | XD_SCALE    | DS-68              | Depends on the device type. See the manual. See the corresponding manual for details. | Depends on the device type. See description item for details. | XD    | S / Man    | The high and low scale values, to transducer for a specified channel.<br>The Default value for each Smar device is showed below:<br>DFI302: 0 to 100 [%]   |
| 11  | OUT_SCALE   | DS-68              |   | 0-100%  | OUT   | S / Man    | The high and low scale values to the OUT parameter.  |
| 12  | GRANT_DENY  | DS-70              |   | 0   | na    | D          |  |
| 13  | IO_OPTS     | Bitstring(2)       | See Block Options   | 0   | na    | S / O/S    | See Block Options  |
| 14  | STATUS_OPTS | Bitstring(2)       | See Block Options   | 0   | Na    | S / O/S    | See Block Options  |
| 15  | CHANNEL     | Unsigned16         |   | 0   | None  | S / O/S    | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 16  | L_TYPE      | Unsigned8          | 1: Direct<br>2: Indirect<br>3: Indirect Square Root                                   | 0   | E     | S / Man    | Determines how the values passed by the transducer block may be used:<br>Direct: Directly<br>Indirect: with a percent<br>Indirect Sq Root: with a percent and with square root                       |
| 17  | LOW_CUT     | Float              | Non-Negative  | 0   | OUT   | S          | A value of zero percent of scale is used in block processing if the transducer value falls below this limit, in % of scale. This feature may be used to eliminate noise near zero for a flow sensor. |



| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options                   | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|------------|-----------------------|---|------------------|-------|----------------|--|
| 18  | PV_FTIME   | Float                 | Non-Negative                              | 0                | Sec   | S              | Time constant of a single exponential filter for the PV, in seconds.   |
| 19  | FIELD_VAL  | DS-65                 |   |                  | %     | D / RO         | Raw value of the field device in percent of the PV range, with a status reflecting the Transducer condition, before signal characterization (L_TYPE) or filtering (PV_FTIME).  |
| 20  | UPDATE_EVT | DS-73                 |   |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 21  | BLOCK_ALM  | DS-72                 |   |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 22  | ALARM_SUM  | DS-74                 | See Block Options                         |                  | Na    | S              | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 23  | ACK_OPTION | Bitstring(2)          | 0: Auto ACK Disable<br>1: Auto ACK Enable | 0                | Na    | S              | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 24  | ALARM_HYS  | Float                 | 0 to 50 %                                 | 0.5%             | %     | S              | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis.  |
| 25  | HI_HI_PRI  | Unsigned8             | 0 to 15                                   |                  |       | S              | Priority of the high high alarm.   |
| 26  | HI_HI_LIM  | Float                 | OUT_SCALE, +INF                           | +INF             | OUT   | S              | The setting for high high alarm in engineering units.  |
| 27  | HI_PRI     | Unsigned8             | 0 to 15                                   |                  |       | S              | Priority of the high alarm.  |
| 28  | HI_LIM     | Float                 | OUT_SCALE, +INF                           | +INF             | OUT   | S              | The setting for high alarm in engineering units.   |
| 29  | LO_PRI     | Unsigned8             | 0 to 15                                   |                  |       | S              | Priority of the low alarm.   |
| 30  | LO_LIM     | Float                 | OUT_SCALE, -INF                           | -INF             | OUT   | S              | The setting for low alarm in engineering units.  |
| 31  | LO_LO_PRI  | Unsigned8             | 0 to 15                                   |                  |       | S              | Priority of the low low alarm.   |
| 32  | LO_LO_LIM  | Float                 | OUT_SCALE, -INF                           | -INF             | OUT   | S              | The setting for low low alarm in engineering units.  |
| 33  | HI_HI_ALM  | DS-71                 |   |                  | OUT   | D              | The status for high high alarm and its associated time stamp.  |
| 34  | HI_ALM     | DS-71                 |   |                  | OUT   | D              | The status for high alarm and its associated time stamp.   |
| 35  | LO_ALM     | DS-71                 |   |                  | OUT   | D              | The status for low alarm and its associated time stamp.  |
| 36  | LO_LO_ALM  | DS-71                 |   |                  | OUT   | D              | The status for low low alarm and its associated time stamp.  |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – static  
Gray Background Line: Default Parameters of Syscon

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of CHANNEL is the lowest available number.

The default value of L\_TYPE is direct.

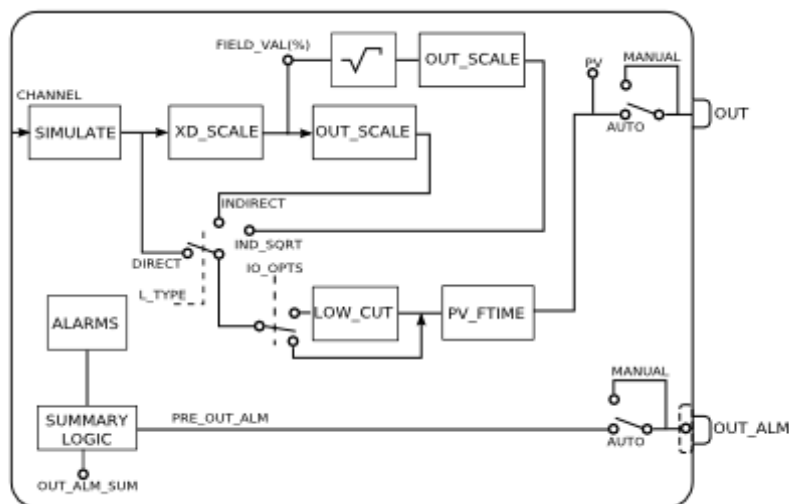
The required mode for writing is the Actual mode for the OUT parameter, regardless the Target mode.

## EAI – Enhanced Analog Input

### Overview

The Enhanced Analog Input block has all the parameters of the AI block. It has an additional feature for alarm output (OUT\_ALM parameter) to be used in control logic.

### Schematic



### Description

The EAI block has the additional feature of OUT\_ALM alarm output that indicates some alarm condition established. The block has 4 static alarms configured for xx\_LIM parameters (where xx indicates one of the four types of block alarm: HI, LO, HI\_HI and LO\_LO).

The alarm type generated for OUT\_ALM output is chosen through OUT\_ALM\_SUM parameter. In this case it is allowed to select the following alarm options:

| OUT_ALM_SUM | ALARM CONDITIONS |        |        |           |
|-------------|------------------|--------|--------|-----------|
|             | HI_HI_ALM        | HI_ALM | LO_ALM | LO_LO_ALM |
| ANY         | ✓                | ✓      | ✓      | ✓         |
| LOWs        |                  |        | ✓      | ✓         |
| HIGHs       | ✓                | ✓      |        |           |
| LEVEL1      |                  | ✓      | ✓      |           |
| LEVEL2      | ✓                |        |        | ✓         |
| LO_LO       |                  |        |        | ✓         |
| LO          |                  |        | ✓      |           |
| HI          |                  | ✓      |        |           |
| HI_HI       | ✓                |        |        |           |
| NONE        |                  |        |        |           |

The PRE\_OUT\_ALM parameter generates internally a 1 logic value when the alarm condition chosen in OUT\_ALM\_SUM is established.

Optionally the INVERT\_OPTS parameter allows inverting the TRUE logic value of the OUT\_ALM output from 1 to 0.

### BLOCK\_ERR

The BLOCK\_ERR of the block will reflect the following causes:  
Out of Service – occurs when the block is in O/S mode.

### Supported Modes

O/S, MAN and AUTO

## Modes

Only in Out of Service mode is possible to configure INVERT\_OPTS in two operation modes.

The manual mode disconnects the OUT\_ALM output of the algorithm and it allows manual substitution of the alarm. In automatic mode the OUT\_ALM output is PRE\_OUT\_ALM, verifying the option of TRUE logic value configured in INVERT\_OPTS.

## Parameters

| Index | Parameter   | Data Type    | Valid Range  | Initial Value | Storage / Access | Views | Description  |
|-------|-------------|--------------|--|---------------|------------------|-------|--|
| 37    | PRE_OUT_ALM | DS_66        |  |               | D/RO             | 1, 3  | This parameter indicates if exist alarm and it is available when the block is in <i>Auto</i> mode in OUT_ALM output.   |
| 38    | OUT_ALM     | DS_66        |  | 0             | D/MAN            | 1, 3  | Discrete output that indicates if the block is in alarm condition (value 1) or it is not in alarm condition (value 0). |
| 39    | OUT_ALM_SUM | Unsigned 8   | 0: NONE<br>1: LO_LO<br>2: LO<br>3: LOWs<br>4: HI<br>6: LEVEL1<br>8: HI_HI<br>9: LEVEL2<br>12: HIGHs<br>15: ANY | 0             | S//MAN           | 4     | Specify the alarm conditions that should assume the <i>True</i> value for the OUT_ALM output assume <i>True</i> value. |
| 40    | INVERT_OPTS | Bitstring(2) |  | 0             | S//MAN           | 4     | Parameter to invert the logic value corresponding to output of <i>True</i> value.                                      |

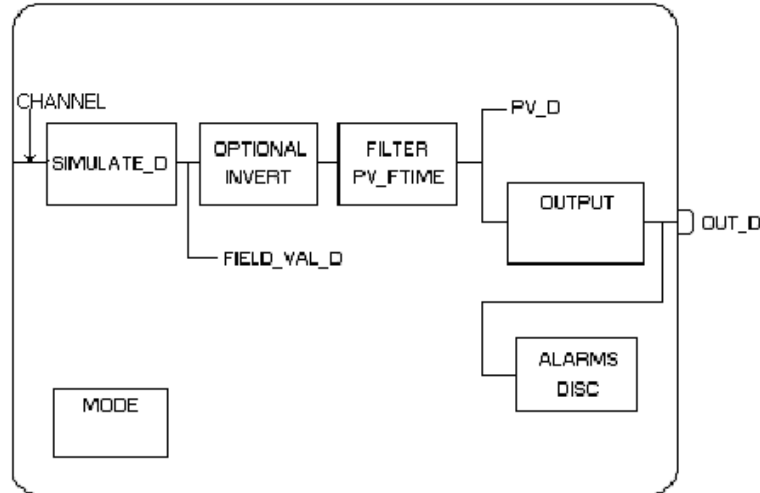
**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – Dynamic; *N* – non-volatile; *S* – static

## DI - Discrete Input

### Overview

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

### Schematic



### Description

The FIELD\_VAL\_D shows the true on/off state of the hardware, using XD\_STATE.

The block output can be inverted using the “Invert” option of the IO\_OPTS parameter. If “Invert” is selected, then a non-zero field value would result in an output equal to zero (0). And, a zero field value would result in an output equal to one (1).

PV\_FTIME may be used to set the time that the hardware must be in one state before it gets passed to the PV\_D. The PV\_D is always the value that the block will place in OUT\_D if the mode is Auto. If Man is allowed, someone may write a value to OUT\_D. The PV\_D and the OUT\_D always have identical scaling. OUT\_STATE provides scaling for PV\_D.

### BLOCK\_ERR

The BLOCK\_ERR of the DI block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when one or more of the following situations occur:
  - When the CHANNEL parameter has an invalid value;
  - When it is not compatible the CHANNEL parameter and HC configuration (DFI302).
- Simulate Active – When the Simulate is active;
- Input Failure – I/O module failure (DFI302);
- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S, Man, and Auto.

### Status Handling

The DI block does not support cascade path. Then, the output status has not a cascade sub-status. The following options from STATUS\_OPTS apply: Propagate Fault Forward

## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/ Options   | Default Value | Units  | Store/ Mode | Description  |
|-----|-------------|--------------------|--|---------------|--------|-------------|--|
| 1   | ST_REV      | Unsigned16         |  | 0             | None   | S/RO        |  |
| 2   | TAG_DESC    | OctString(32)      |  | Spaces        | Na     | S           |  |
| 3   | STRATEGY    | Unsigned16         |  | 0             | None   | S           |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255   | 0             | None   | S           |  |
| 5   | MODE_BLK    | DS-69              |  | O/S           | Na     | S           | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)       |  |               | E      | D / RO      |  |
| 7   | PV_D        | DS-66              |  |               | PV     | D / RO      | The primary discrete value for use in executing the function, or a process value associated with it.   |
| 8   | OUT_D       | DS-66              | OUT_STATE  |               | OUT    | D / Man     | The primary discrete value calculated as a result of executing the function.   |
| 9   | SIMULATE_D  | DS-83              | 1: Disable ;<br>2: Active<br>are the Enable/Disable options. | Disable       |        | D           | Allows the transducer discrete input or output to the block to be manually supplied when simulate is enabled. When simulation is disabled, the simulate value and status track the actual value and status.  |
| 10  | XD_STATE    | Unsigned16         |  | 0             | XD     | S           | Index to the text describing the states of a discrete for the value obtained from the transducer.  |
| 11  | OUT_STATE   | Unsigned16         |  | 0             | OUT    | S           | Index to the text describing the states of a discrete output.  |
| 12  | GRANT_DENY  | DS-70              |  | 0             | Na     | D           |  |
| 13  | IO_OPTS     | Bitstring(2)       | See Block Options  | 0             | Na     | S / O/S     | See Block Options  |
| 14  | STATUS_OPTS | Bitstring(2)       | See Block Options  | 0             | Na     | S / O/S     | See Block Options  |
| 15  | CHANNEL     | Unsigned16         |  | 0             | None   | S / O/S     | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 16  | PV_FTIME    | Float              | Non-Negative   | 0             | Sec    | S           | Time constant of a single exponential filter for the PV, in seconds.   |
| 17  | FIELD_VAL_D | DS-66              |  |               | On/Off | D / RO      | Raw value of the field device discrete input, with a status reflecting the Transducer condition.   |
| 18  | UPDATE_EVT  | DS-73              |  |               | Na     | D           | This alert is generated by any change to the static data.  |
| 19  | BLOCK_ALM   | DS-72              |  |               | Na     | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 20  | ALARM_SUM   | DS-74              | See Block Options  |               | Na     | S           | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 21  | ACK_OPTION  | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable                    | 0             | Na     | S           | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 22  | DISC_PRI    | Unsigned8          | 0 to 15  | 0             |        | S           | Priority of the discrete alarm.  |
| 23  | DISC_LIM    | Unsigned8          | PV_STATE   | 0             | PV     | S           | State of discrete input which will generate an alarm.  |
| 24  | DISC_ALM    | DS-72              |  |               | PV     | D           | The status and time stamp associated with the discrete alarm.  |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

## MAI - Multiple Analog Input

### Description

The MAI block makes available for the FOUNDATION Fieldbus network eight analog variables of the I/O subsystem through its eight output parameters OUT\_1 through OUT\_8.

For DFI working with Analog Input, the 4-20mA or 1-5V range must be worked. In this case, the output values are in percent from 0 to 100%. If it is necessary to work in another configuration, the AI block must be used.

Status indication in the output parameters OUT\_x depends on the I/O subsystem and the transducer block, that is manufacturer specific. For example, if there is individual detection of sensor failure, it will be indicated in the status of related OUT\_x parameter. Problem in the interface to the I/O subsystem will be indicated in the status of all OUT\_x as BAD – Device Failure.

### BLOCK\_ERR

The BLOCK\_ERR of the MAI block will reflect the following causes:

- Other – the number of MDI, MDO, MAI and MAO blocks or the device tag in FB700 is different from LC700;
- Block Configuration Error – the configuration error occurs when the configuration of the CHANNEL parameter is not compatible with the HC (DFI302);
- Out of Service – When the block is in O/S mode.

### Status Handling

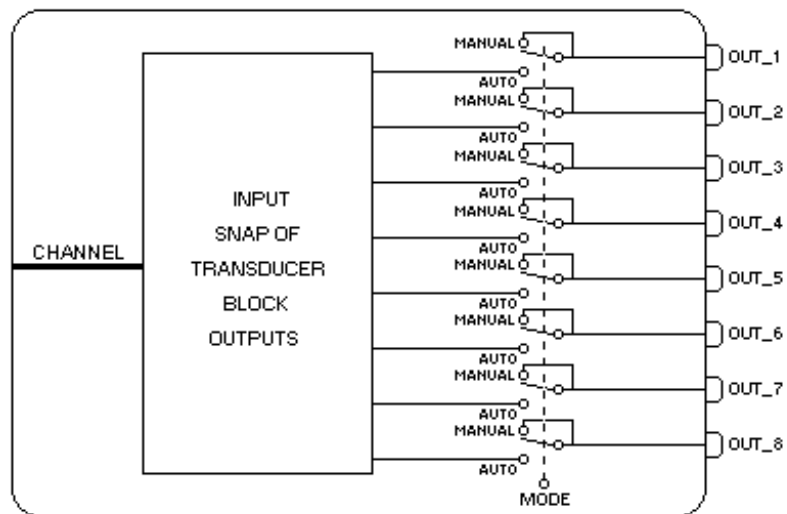
The status of OUT\_x will be the following if the BLOCK\_ERR indicates:

- Other – Bad : Configuration Error
- Input failure – Bad : Device Failure
- Power up – Bad : Device Failure

### Supported Modes

O/S, Man and Auto.

### Schematic



## Parameters

| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|------------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 1   | ST_REV     | Unsigned16            |                         | 0                | None  | S/RO            |  |
| 2   | TAG_DESC   | OctString(32)         |                         | Spaces           | Na    | S               |  |
| 3   | STRATEGY   | Unsigned16            |                         | 0                | None  | S               |  |
| 4   | ALERT_KEY  | Unsigned8             | 1 to 255                | 0                | None  | S               |  |
| 5   | MODE_BLK   | DS-69                 |                         | O/S              | Na    | S               | See Mode Parameter   |
| 6   | BLOCK_ERR  | Bitstring(2)          |                         |                  | E     | D / RO          |  |
| 7   | CHANNEL    | Unsigned16            |                         | 0                | None  | S / O/S         | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration". |
| 8   | OUT_1      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 1.   |
| 9   | OUT_2      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 2.   |
| 10  | OUT_3      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 3.   |
| 11  | OUT_4      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 4.   |
| 12  | OUT_5      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 5.   |
| 13  | OUT_6      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 6.   |
| 14  | OUT_7      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 7.   |
| 15  | OUT_8      | DS-65                 |                         |                  | %     | D / Man         | Numbered analog input 8.   |
| 16  | UPDATE_EVT | DS-73                 |                         |                  | Na    | D               |  |
| 17  | BLOCK_ALM  | DS-72                 |                         |                  | Na    | D               |  |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static  
Gray Background Line: Default Parameters of Syscon

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The required mode for writing is the Actual mode for the parameters from OUT\_1 to OUT\_8, regardless the Target mode.

## MDI - Multiple Discrete Input

### Description

The MDI block makes available for the FOUNDATION Fieldbus network eight discrete variables of the I/O subsystem through its eight output parameters OUT\_D1 through OUT\_D8. Status indication in the output parameters OUT\_Dx depends on the I/O subsystem and the transducer block, that is manufacturer specific. For example, if there is individual detection of sensor failure, it will be indicated in the status of related OUT\_Dx parameter. Problem in the interface to the I/O subsystem will be indicated in the status of all OUT\_Dx as BAD – Device Failure.

### BLOCK\_ERR

The BLOCK\_ERR of the MDI block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the configuration of CHANNEL parameter and HC block is not compatible (DFI302);
- Out of Service – When the block is in O/S mode.

### Status Handling

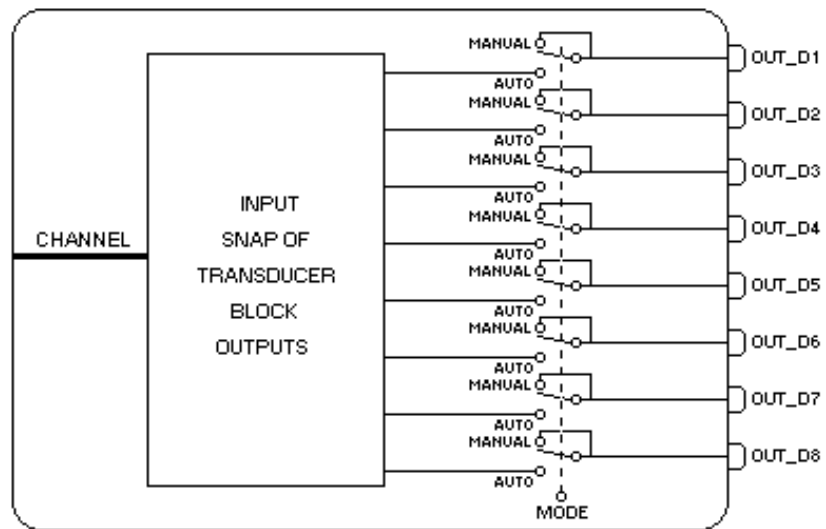
The status of OUT\_Dx will be the following if the BLOCK\_ERR indicates:

- Other – Bad : Configuration Error
- Input failure – Bad : Device Failure
- Power up – Bad : Device Failure

### Supported Modes

O/S, Man and Auto.

### Schematic





## Parameters

| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|------------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 1   | ST_REV     | Unsigned16            |                         | 0                | None  | S/RO            |  |
| 2   | TAG_DESC   | OctString(32)         |                         | Spaces           | Na    | S               |  |
| 3   | STRATEGY   | Unsigned16            |                         | 0                | None  | S               |  |
| 4   | ALERT_KEY  | Unsigned8             | 1 to 255                | 0                | None  | S               |  |
| 5   | MODE_BLK   | DS-69                 |                         | O/S              | Na    | S               | See Mode Parameter   |
| 6   | BLOCK_ERR  | Bitstring(2)          |                         |                  | E     | D / RO          |  |
| 7   | CHANNEL    | Unsigned16            |                         | 0                | None  | S / O/S         | For more details about the configuration of this parameter, see Chapter 1 “CHANNEL Configuration”. |
| 8   | OUT_D1     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 1.   |
| 9   | OUT_D2     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 2.   |
| 10  | OUT_D3     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 3.   |
| 11  | OUT_D4     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 4.   |
| 12  | OUT_D5     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 5.   |
| 13  | OUT_D6     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 6.   |
| 14  | OUT_D7     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 7.   |
| 15  | OUT_D8     | DS-66                 |                         |                  |       | D / Man         | Numbered discrete input 8.   |
| 16  | UPDATE_EVT | DS-73                 |                         |                  | Na    | D               |  |
| 17  | BLOCK_ALM  | DS-72                 |                         |                  | Na    | D               |  |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static  
Gray Background Line: Default Parameters of Syscon

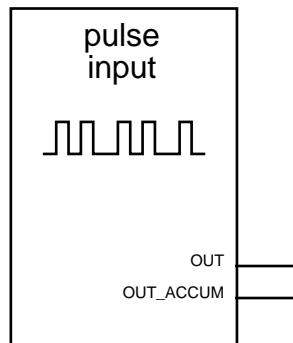
**If DIAG.BEHAVIOR parameter is “Adapted”:**

The required mode for writing is the Actual mode for the parameters from OUT\_1 to OUT\_8, regardless the Target mode.

## PUL – Pulse Input

### Overview

The Pulse Input Block provides analog values based on a pulse (counter) transducer input. There are two primary outputs available. An accumulation output is intended to be connected to an integrator block for differencing, conversion, and integration. This is most useful when the count rate is low relative to the block execution rate. For high count rates, the accumulated count of pulses per block execution can be interpreted as an analog rate (vs. accumulation) value and can be alarmed. (Alarm conditions include high, high-high, low, and low-low alarms.)



### Description

OUT is a connectable bipolar (signed) analog value output of the PI block. It is determined by taking the number of counts accumulated since the last execution of the block, multiplying by the value of each pulse (PULSE\_VAL), dividing by the block's execution rate in seconds, converting to units/minute, units/hour, or units/day in accordance with the TIME\_UNITS enumeration, and filtering using PV\_FTIME. PV\_FTIME is the time constant for a filter. Alarming is performed on this filtered value. Reverse flow may be detected from the transducer and indicated via a negative value of OUT.

Pre-filtered value = (CHANGE\_IN\_COUNTS \* PULSE\_VAL / EXEC\_PERIOD) \* time\_unit\_factor

Where,

**CHANGE\_IN\_COUNTS** is the number of counts received since last execution

**PULSE\_VAL** is the value in engineering units of each pulse

**EXEC\_PERIOD** is the execution period of the block in seconds

**Time Unit Factor** is 1 sec/sec, 60 sec/min, 3600 sec/hour, or 86400 sec/day, per the TIME\_UNITS index.

For example, 70 pulse counts are received by the transducer in the 0.5 second execution period of the PI block. The metering device manufacturer specifies that each pulse represents 0.1 gallons of flow. The user wants the flow rate expressed in "gallons per minute". PULSE\_VAL should be set to 0.1. TIME\_UNITS should be set to "units/minute". The related time\_unit\_factor will be "60 sec/min". Using the above equation, the pre-filtered rate value will then be computed as:

$$\begin{aligned} & ((70 \text{ pulses} * 0.1 \text{ gallons/pulse}) / 0.5 \text{ sec}) * 60 \text{ sec/min} = \\ & (7.0 \text{ gallons} / 0.5 \text{ sec}) * 60 \text{ sec/min} = \\ & (14.0 \text{ gallons/sec}) * 60 \text{ sec/min} = \\ & 840 \text{ gallons/min} \end{aligned}$$

OUT\_ACCUM is a connectable float output of the PI block. It is intended to be connected to the Integrator block for totalization, so it only accumulates enough to avoid rollover between executions of the Integrator block. It represents a continuous accumulation of counts from the transducer, limited to the range of values from 0 to 999,999. It can count either up or down. An increment of the accumulation 999,999 by 1 will result in the accumulation 0 and a decrement of the accumulation 0 by 1 will result in the accumulation 999,999. A maximum change to the accumulation of  $\pm 499,999$  counts is permitted to be reflected in OUT\_ACCUM in a single execution of the block. If a change of counts greater in magnitude than 499,999 occurs at the transducer:

- The change in OUT\_ACCUM is limited to 499,999 of the proper sign,
- The OUT\_ACCUM status's quality is set to Uncertain,
- The OUT\_ACCUM status's sub-status is set to "Engr. Units Range Violation",
- The OUT\_ACCUM status's limits = low (if negative) or limits = high (if positive) indicator is set
- A BLOCK\_ALM must be issued.

CHANNEL is used to associate the block with the hardware that is connected to this block. It defines the transducer to be used coming from the physical world.

### Supported Modes

O/S, Manual and Auto.

### Mode Handling

Manual mode "disconnects" the input from the output and permits manual substitution of the value. OUT is the alarmed value and the value which normally would be substituted, but OUT\_ACCUM may also be substituted.

On transition from Manual to Auto, the PV filter will be initialized to the value of OUT, and the accumulated total will be set to the value of OUT\_ACCUM.

### Status Handling

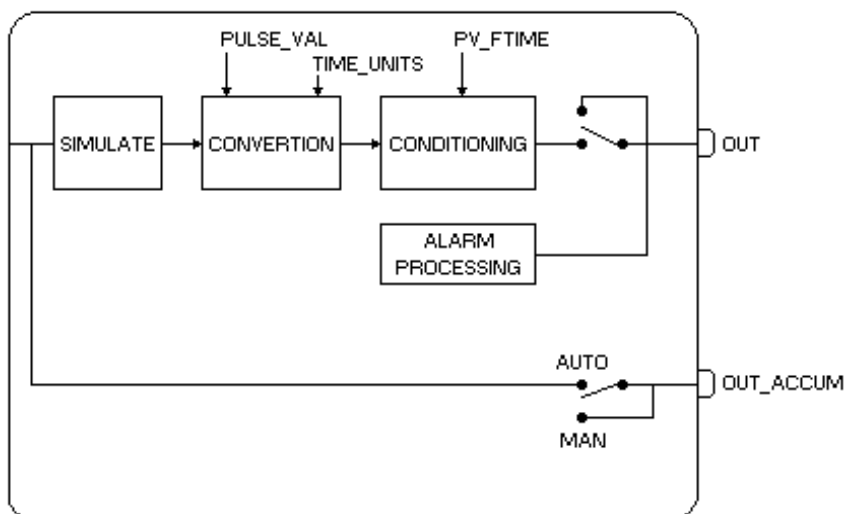
This block has no inputs from other blocks and therefore does not react to status of other blocks. Both the OUT and the OUT\_ACCUM outputs have status and will reflect the status of the transducer (e.g. hardware failure) and the mode of the block (e.g., out-of-service, manual, etc.) using the conventional rules of status.

An unusable status (bad) for OUT will cause the alarm processing to be suspended. Current alarms will not be cleared and new alarms will not be generated until the status returns to a usable status.

### Simulation

The SIMULATE\_P parameter is provided to simulate pulse input as a rate in pulses/second, rather than the actual transducer value. The value entered in the SIMULATE\_P record is considered to be the signed change in accumulation per second. The quality of the status entered is passed to the status of OUT and OUT\_ACCUM.

### Schematic



## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/Options  | Default Value | Units | Store / Mode | Description   |
|-----|-------------|--------------------|--|---------------|-------|--------------|---|
| 1   | ST_REV      | Unsigned16         |  | 0             | None  | S/RO         |   |
| 2   | TAG_DESC    | OctString(32)      |  | Spaces        | Na    | S            |   |
| 3   | STRATEGY    | Unsigned16         |  | 0             | None  | S            |   |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255   | 0             | None  | S            |   |
| 5   | MODE_BLK    | DS-69              |  | O/S           | Na    | S            | See Mode Parameter  |
| 6   | BLOCK_ERR   | Bitstring(2)       |  |               | E     | D / RO       |   |
| 7   | PV          | DS-65              |  |               | PV    | D / RO       | Process analog value for use in executing the function.   |
| 8   | OUT         | DS-65              | OUT_SCALE +/- 10%  |               | OUT   | D / Man      | The analog value calculated as a result of executing the function.  |
| 9   | OUT_ACCUM   | DS-65              |  |               | None  | N / Man      | This parameter is the number of counts accumulated in an on-going basis. It is not normally reset except that it wraps around to zero after reaching 999,999 counts. The value is based on the transducer input in <i>Auto</i> mode and is the last transducer value of the value specified by the operator/engineer in <i>Man</i> mode. (Its meaning is most useful when the number of counts received between executions of the block is small.) It is intended to be connected to the counter input of an integrator block. The OUT_ACCUM value may increase or decrease by a maximum of 499,999 counts per execution. |
| 10  | SIMULATE_P  | DS-82              | 1: Disable ;<br>2: Active<br>are the Enable/Disable options.                     | Disable       |       | D            | Allows the transducer input to the Pulse Input block to be manually supplied when simulate is enabled. When simulation is disabled, the simulate value and status track the actual value and status. The value is the rate of change of the transducer count in counts per second, not the accumulation.  |
| 11  | PULSE_VAL   | Float              |  | 0             | None  |              | Value of each metered pulse in engineering units. Used only to calculate PV and OUT. Not used for OUT_ACCUM calculation.  |
| 12  | TIME_UNITS  | Unsigned8          | 1: seconds<br>2: minutes<br>3: hours<br>4: days<br>5: [day-<br>[hr:[min[:sec]]]] | 0             | E     | S            | Time units factor to be used in the conversion of the output.   |
| 13  | OUT_SCALE   | DS-68              |  | 0-100%        | OUT   | S / Man      | The high and low scale values to the OUT parameter.   |
| 14  | GRANT_DENY  | DS-70              |  | 0             | na    | D            |   |
| 15  | IO_OPTS     | Bitstring(2)       | See Block Options  | 0             | na    | S / O/S      | See Block Options   |
| 16  | STATUS_OPTS | Bitstring(2)       | See Block Options  | 0             | Na    | S / O/S      | See Block Options   |
| 17  | CHANNEL     | Unsigned16         |  | 0             | None  | S / O/S      | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".  |

| Idx | Parameter  | Data Type (length) | Valid Range/Options                       | Default Value | Units | Store / Mode | Description  |
|-----|------------|--------------------|---|---------------|-------|--------------|--|
| 18  | PV_FTIME   | Float              | Non-Negative                              | 0             | Sec   | S            | Time constant of a single exponential filter for the PV, in seconds.   |
| 19  | FIELD_VAL  | DS-65              |   |               | %     | D / RO       | Raw value of the field device with a status reflecting the Transducer condition, before filtering (PV_FTIME).  |
| 20  | UPDATE_EVT | DS-73              |   |               | Na    | D            | This alert is generated by any change to the static data.  |
| 21  | BLOCK_ALM  | DS-72              |   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 22  | ALARM_SUM  | DS-74              | See Block Options                         |               | Na    | S            | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 23  | ACK_OPTION | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable | 0             | Na    | S            | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 24  | ALARM_HYS  | Float              | 0 to 50 %                                 | 0.5%          | %     | S            | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis (percentage of OUT_SCALE).  |
| 25  | HI_HI_PRI  | Unsigned8          | 0 to 15                                   |               |       | S            | Priority of the high high alarm.   |
| 26  | HI_HI_LIM  | Float              | OUT_SCALE, +INF                           | +INF          | OUT   | S            | The setting for high high alarm in engineering units.  |
| 27  | HI_PRI     | Unsigned8          | 0 to 15                                   |               |       | S            | Priority of the high alarm.  |
| 28  | HI_LIM     | Float              | OUT_SCALE, +INF                           | +INF          | OUT   | S            | The setting for high alarm in engineering units.   |
| 29  | LO_PRI     | Unsigned8          | 0 to 15                                   |               |       | S            | Priority of the low alarm.   |
| 30  | LO_LIM     | Float              | OUT_SCALE, -INF                           | -INF          | OUT   | S            | The setting for low alarm in engineering units.  |
| 31  | LO_LO_PRI  | Unsigned8          | 0 to 15                                   |               |       | S            | Priority of the low low alarm.   |
| 32  | LO_LO_LIM  | Float              | OUT_SCALE, -INF                           | -INF          | OUT   | S            | The setting for low low alarm in engineering units.  |
| 33  | HI_HI_ALM  | DS-71              |   |               | OUT   | D            | The status for high high alarm and its associated time stamp.  |
| 34  | HI_ALM     | DS-71              |   |               | OUT   | D            | The status for high alarm and its associated time stamp.   |
| 35  | LO_ALM     | DS-71              |   |               | OUT   | D            | The status for low alarm and its associated time stamp.  |
| 36  | LO_LO_ALM  | DS-71              |   |               | OUT   | D            | The status for low low alarm and its associated time stamp.  |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static  
Gray Background Line: Default Parameters of Syscon

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of CHANNEL is the lowest available number.

The required mode for writing is the Actual mode for the OUT parameter, regardless the Target mode.

## ROMAI – ROM Analog Input

### Description

This block has all characteristics of the standard AI block in addition the features related to alarms and I/O mapping via CHANNEL\_TAG.

### CHANNEL\_TAG Parameter

This parameter allows association with the physical points of analog input, which occurs when the value of this parameter coincides with a compatible VAR\_NAME in terms of type, ie, analog input. The same input physical point may be associated with up to 2 CHANNEL\_TAGS in different function blocks.

The configuration of the CHANNEL\_TAG parameter indirectly implies in the configuration of the CHANNEL parameter, which depends on the match with a valid VAR\_NAME. So, the CHANNEL parameter becomes a read-only parameter to the user.

### BLOCK\_ERR

It indicates a problem at some configured points, which can be: Configuration Error and Input Failure.

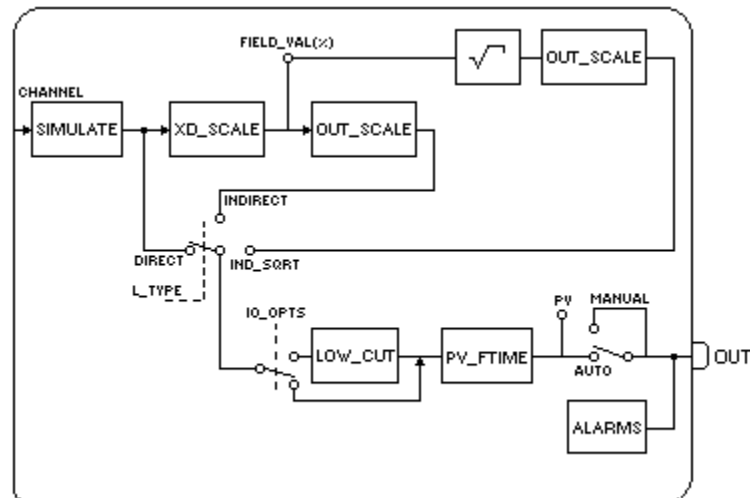
### Diagnostic and troubleshooting

- Block configuration : possible causes of error are :
  - if CHANNEL\_TAG specified here is not found in a transducer block
  - if an output Tag is specified in more than one CHANNEL\_TAG parameter.
- Sensor (Input) failure : It indicates that a configured point has a bad status or a TRD is in O/S;
- Out-of-service: It indicates that Actual Mode is O/S.

### Supported Modes

Auto, Man, O/S.

### Schematic



## Parameters

| Idx | Parameter   | Data type         | Valid Range  | Initial Value | Units | Store/ Other | Description   |
|-----|-------------|-------------------|--|---------------|-------|--------------|---|
| 37  | PRE_OUT_ALM | DS-66             |  | 0             |       | D/RO         | This parameter indicates if exist alarm and it is available when the block is in Auto mode in OUT_ALM output.                     |
| 38  | OUT_ALM     | DS-66             |  | 0             |       | D/Man        | Discrete output that indicates if the block is in alarm condition (value 1) or it is not in alarm condition (value 0).            |
| 39  | OUT_ALM_SUM | Unsigned8         | 0:NONE<br>1:LO_LO<br>2:LO<br>3:LOWs<br>4:HI<br>6:LEVEL1<br>8:HI_HI<br>9:LEVEL2<br>12:HIGhs<br>15:ANY | 0             |       | S/MAN        | Specify the alarm conditions that should assume the <i>True</i> value for the OUT_ALM output assume <i>True</i> value.            |
| 40  | INVERT_OPTS | Bistring[2]       |  | 0             |       | S / MAN      | Parameter to invert the logic value corresponding to output of <i>True</i> value.   |
| 41  | CHANNEL_TAG | VisibleString[32] |  | Blanks        |       | S            | TAG of the I/O point to be mapped with TAG in the I/O transducer block. Blank Tag means no association with I/O transducer block. |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

## MAI16 –16 Multiple Analog Input

### Overview

The block MAI16 provides to the FOUNDATION fieldbus network 16 I/O analog variables through its 16 **OUT\_16** parameters whose values are expressed in engineering units.

### CHANNEL\_TAG\_16 Parameter

This parameter allows association with the physical points of analog input, which occurs when the value of this parameter coincides with a compatible VAR\_NAME in terms of type, ie, analog input. The same input physical point may be associated with up to 2 CHANNEL\_TAGS in different function blocks.

### FORCE\_ENB\_16 Parameter

This parameter enables/disables the point update in the output parameter (OUT\_16) of this block, but the scan of the physical point and the transducer block update remains to be done regardless of this parameter.

### BLOCK\_ERR:

It indicates a problem at some configured points, which can be:

- Block configuration: It indicates that a configured point has a configuration error;
- Sensor (input) failure: It indicates that a configured point has a bad status or a TRD is in O/S;
- Out-of-service: It indicates that Actual Mode is O/S.

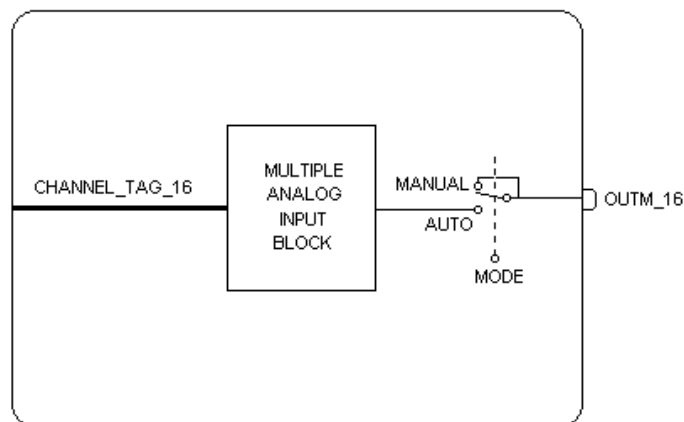
Point's Status: It indicates a problem at specific point.

Overall status: It does not indicate problem of the point individually.

### Supported Modes

Auto, Man, O/S.

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/Options | Default Value | Units | Store/Mode | Description |
|-----|-----------|--------------------|---------------------|---------------|-------|------------|-------------|
| 1   | ST_REV    | Unsigned16         |                     | 0             | None  | S / RO     |             |
| 2   | TAG_DESC  | OctString(32)      |                     | Spaces        | Na    | S          |             |
| 3   | STRATEGY  | Unsigned16         |                     | 255           | None  | S          |             |
| 4   | ALERT_KEY | Unsigned8          |                     | 1             | None  | S          |             |
| 5   | MODE_BLK  | DS-69              |                     | Auto          | Na    | S          |             |



| Idx | Parameter      | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|----------------|-----------------------|-------------------------|------------------|-------|----------------|--|
| 6   | BLOCK_ERR      | Bitstring(2)          |                         |                  | E     | D / RO         |  |
| 7   | FORCE_ENB_16   | Bit String[2]         |                         | 0                |       | S              | Controls the updating of each of the 16 analog points when the block is in AUTO mode. A value of 1 will stop the updating of the corresponding input/output point.   |
| 8   | CHANNEL_TAG_16 | VisibleString[16][32] |                         | Blanks           |       | S              | TAGs of the I/O point to be mapped with TAGs in the I/O transducer block. Blank Tag means no association with I/O transducer block. The block's MAP_MM_* parameter indicates if any Tags specified here are not found in a transducer block or if an output Tag is specified in more than one CHANNEL_TAG_* parameter.   |
| 9   | OUTM_16        | DS-174                |                         |                  | Na    | N / Man        | 16 float (and status) values output parameter for multiple input and output function blocks.   |
| 10  | MAP_MM_16      | BitString[2]          |                         | 0                | Na    | D / RO         | Shows which of 16 function block points are in Tag mismatch mapping error. A "1" for point "n" indicates that the Tag for function block point "n" either (1) cannot be found in any transducer block, (2) is associated with a transducer block providing an incompatible data type, or (3) references a transducer block output that is also referenced by the same tag in another output function block.                            |
| 11  | UPDATE_EVT     | DS-73                 |                         |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 12  | BLOCK_ALM      | DS-72                 |                         |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static**

## MBI64 – Multiple Binary Inputs 64

### Overview

The MBI64 block provides to the FOUNDATION fieldbus network 64 binary inputs variables, each one with status, through the OUTM\_B64 output parameter. Thus the transfer of these 64 binary variables for control is performed by a single HSE link quite optimized.

### CHANNEL\_TAG\_64A/B Parameter

These parameters allow association with the physical points of binary input, which occurs when the value of this parameter coincides with a VAR\_NAME with type compatible, i.e., binary input.

A single physical input point may be associated up to 2 CHANNEL\_TAGs in different function blocks.

### MAP\_MM\_64 Parameter

It indicates which of 64 points configured in CHANNEL\_TAG\_64A/B is not associated to input point in transducer block, which can be for the following reasons:

- (1) VAR\_NAME equals to CHANNEL\_TAG configured was not found,
- (2) Type is incompatible with the analog input.
- (3) Input point is already associated to two function blocks.

### FORCE\_ENB\_B64 Parameter

This parameter enables/disables the point updating in the block output parameter (OUTM\_B64) in Auto mode. But the physical point scanning and the transducer block updating continue to be performed regardless this parameter. So this parameter allows similar functionality to the Auto/Man mode individually for each point.

### OUTM\_B64 Status

The status of OUTM\_B64 output only reflects the block actual mode, i.e., the status of any individual configured points has no influence.

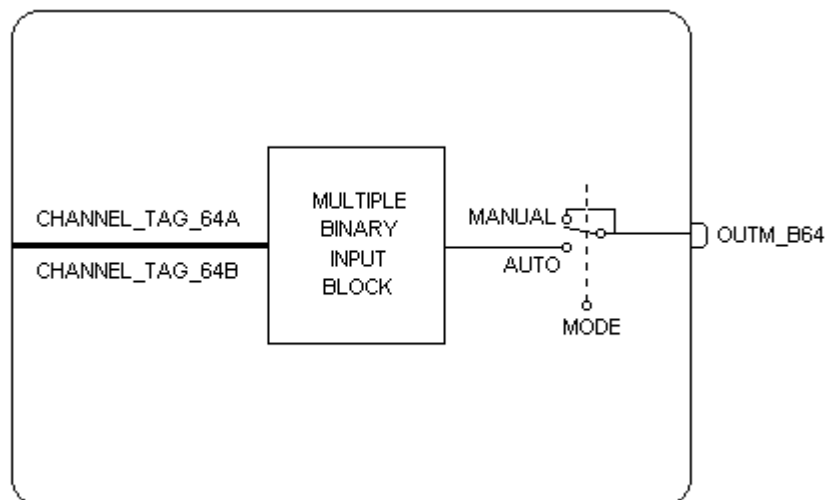
### Diagnostic and troubleshooting

- Block configuration: it indicates if the configured point has some configuration error.
- Sensor (Input) failure: It indicates that the configured point has a bad status or a TRD is in O/S;
- Out-of-service: It indicates that Actual Mode is O/S.

### Supported Modes

Auto, Man, O/S.

### Schematic



## Parameters

| Idx | Type/<br>View | Parameter       | Data Type             | Valid Range | Initial Value | Unit | Store/<br>Mode | Description  |
|-----|---------------|-----------------|-----------------------|-------------|---------------|------|----------------|--|
| 1   | 1,2,3,4       | ST_REV          | Unsigned16            |             | 0             | None | S / RO         |  |
| 2   |               | TAG_DESC        | OctString(32)         |             | Spaces        | Na   | S              |  |
| 3   | 4             | STRATEGY        | Unsigned16            |             | 255           | None | S              |  |
| 4   | 4             | ALERT_KEY       | Unsigned8             |             | 1             | None | S              |  |
| 5   | 1,3           | MODE_BLK        | DS-69                 |             | Auto          | Na   | S              |  |
| 6   | 1,3           | BLOCK_ERR       | Bitstring(2)          |             |               | E    | D / RO         |  |
| 7   | 2,4           | FORCE_ENB_B64   | Bit String[8]         |             | 0             |      | S              | It controls the updating of each of the 64 binary points when the block is in Auto.<br><br>A value of 1 in bit stops updating the input point in OUTM_B64.   |
| 8   |               | CHANNEL_TAG_64A | VisibleString[32][32] |             | Blanks        |      | S              | Tags of the first group of 32 input points that must be mapped to points on the transducer block .<br>Blank tag means that the input point is not configured.<br>The MAP_MM_64 parameter indicates if some configured tag does not have corresponding in transducer block. |
| 9   |               | CHANNEL_TAG_64B | VisibleString[32][32] |             | Blanks        |      | S              | Tags of the second group of 32 input points must be mapped to points on the transducer block.<br>Blank tag means that the input point is not configured.<br>The MAP_MM_64 parameter indicates if some configured tag does not have corresponding in transducer block.      |
| 10  | O,1,3         | OUTM_B64        | DS-158                |             |               | Na   | N / Man        | Output parameter with 64 binary values and status.   |
| 11  | 1,3           | MAP_MM_64       | BitString[8]          |             | 0             | Na   | D / RO         | Indicates which of 64 points configured in CHANNEL_TAG_64A/B is not associated to an input point on transducer block: (1) tag was not found on transducer block, (2) incompatible type.  |
| 12  |               | UPDATE_EVT      | DS-73                 |             |               | Na   | D              | This alert is generated by any change to the static data.  |

| Idx | Type/<br>View | Parameter | Data Type | Valid Range | Initial Value | Unit | Store/<br>Mode | Description  |
|-----|---------------|-----------|-----------|-------------|---------------|------|----------------|--|
| 13  |               | BLOCK_ALM | DS-72     |             |               | Na   | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static**

## Control and Calculation Function Blocks

### PID - PID Control

#### Overview

The PID block offers a lot of control algorithms that use the Proportional, integral and derivative terms.

#### Description

The algorithm of the PID is the non-iterative or ISA. In this algorithm, the GAIN is applied to all terms of the PID, and the Proportional and the Integral actuate over the error, and the derivative actuates over the PV value. Therefore user changes of SP will not cause bump in the output due to the derivative term when the block is in Auto.

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.

See the PV calculation and SP calculation section for details.

#### Direct and Reverse Acting

It is possible to choose the direct or reverse action of control that is made through the "Direct Acting" bit in the CONTROL\_OPTS parameter:

- If the "Direct acting" bit is true then the error is obtained subtracting the SP from the PV:  

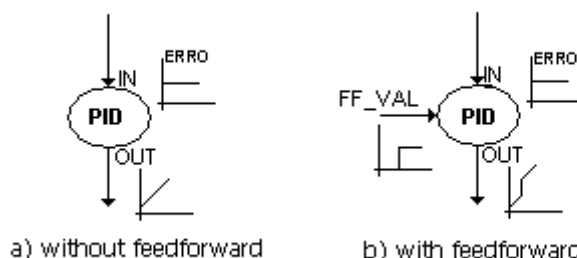
$$\text{Error} = (\text{PV} - \text{SP})$$
- If the "Direct acting" bit is false (clear), the choice is "Reverse acting" then the error is obtained subtracting the PV from the SP:  

$$\text{Error} = (\text{SP} - \text{PV})$$

The Default value of the "Direct acting" bit is false, it means "reverse action".

#### Feedforward Control

The PID block supports the feedforward algorithm. The FF\_VAL input is supplied by an external value, which is proportional to some disturbance in the control loop. The value is converted to output scale using the FF\_SCALE and OUT\_SCALE parameters. This value is multiplied by the FF\_GAIN and added to the output of the PID algorithm.



If the status of FF\_VAL is Bad, the last usable value will be used. When the status returns to good, the difference of FF\_VAL values will be subtracted from BIAS\_A/M in order to avoid bump in the output.

#### PID Constants

GAIN (Kp), RESET (Tr), and RATE (Td) 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.

#### Bypass

When bypass is active the SP value will be transferred to the OUT without the calculation of PID terms. Bypass is used in secondary cascade controller when the PV is bad.

Conditions to turn the Bypass on:

- The “Bypass Enable” bit in the CONTROL\_OPTS must be true.
- BYPASS parameter is changed to ON.
- The BYPASS parameter is the ON/OFF switch that activates the bypass. By default, it can be changed only when the block mode is Man or O/S. Optionally, when the “Change of Bypass in an automatic mode” bit in the FEATURES\_SEL parameter in Resource block is true, then the block permits that the BYPASS switch changes in automatic modes too.

There is special treatment when the Bypass parameter changes ON/OFF in order to avoid bump in the output. When the bypass is switched to ON, the SP receives the OUT value in percent of the OUT\_SCALE. And when the bypass is switched to OFF, the SP receives the PV value.

| Transition in BYPASS | Action                           |
|----------------------|----------------------------------|
| OFF → ON             | OUT → SP with scaling conversion |
| ON → OFF             | PV → SP                          |

Below, there is an example of the bypass in the PID block working as a PID slave in cascade control.

- Step 1 – the status of IN is bad, therefore the actual mode of PID is Man  
 Step 2 - the Target mode is changed to Man in order to write BYPASS  
 Step 3 – the user sets BYPASS to ON, and OUT is transferred to SP with scaling conversion  
 Step 4 – the user changes the Target mode to Cas  
 Step 5 – the PID block reaches the Cas mode, despite of IN.Status.  
 Step 7 – the status of IN becomes good  
 Step 8 - the target mode is changed to Man in order to write BYPASS  
 Step 9 – the user sets BYPASS to OFF, and PV is transferred to SP

CONTROL\_OPTS = “Bypass Enable”

| Steps     | 1        | 2        | 3        | 4        | 5        | 6        | 7         | 8         | 9         | 10        | 11        |
|-----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
| Target    | Cas      | Man      |          | Cas      |          |          |           | Man       |           | Cas       |           |
| Bypass    | Off      |          | On       |          |          |          |           |           | Off       |           |           |
| IN        | Bad      | Bad      | Bad      | Bad      | Bad      | Bad      | GNC<br>80 | GNC<br>80 | GNC<br>80 | GNC<br>80 | GNC<br>80 |
| SP        | GC<br>50 | GC<br>50 | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20  | GC<br>20  | GC<br>80  | GC<br>80  | GC<br>80  |
| Actual    | Man      | Man      | Man      | Man      | Cas      | Cas      | Cas       | Man       | Man       | Man       | Cas       |
| BKCAL_OUT | NI       | NI       | NI       | IR       | GC       | GC       | GC        | NI        | NI        | IR        | GC        |
| OUT       | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20 | GC<br>20  | GC<br>20  | GC<br>20  | GC<br>20  | GC<br>20  |

**Legend: GNC-Good Non Cascade status; GC-Good Cascade status**

**Output Tracking**

The PID block supports the output track algorithm, which allows the output to be forced to a tracking value when the tracking switch is on.

In order to activate the output tracking, the block should attend the following conditions:

- bit “Track Enable” in the CONTROL\_OPTS parameter must be true;
- the target mode is an automatic mode (Auto, Cas and Rcas) or Rout;
- status TRK\_VAL and TRK\_IN\_D are usable, it means the status is good or uncertain and the bit STATUS\_OPTS.”Use Uncertain as good” is true;
- the value of TRK\_IN\_D is active;
- If the target mode is Man, it is necessary, besides the above conditions, the bit “Track in Manual” in the CONTROL\_OPTS parameter must be true.

When the tracking output is active, the OUT will be transferred to TRK\_VAL converted in OUT\_SCALE. The limit status becomes constant and the actual mode goes to LO.

If the TRK\_IN\_D or TRK\_VAL status is unusable, the tracking output will be disabled and PID will return to the normal operation.

### BLOCK\_ERR

The BLOCK\_ERR of the PID block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the BYPASS and SHED\_OPT parameters have an invalid value;
- Out of Service – it occurs when the block is in OOS mode.

### Supported Modes

O/S, IMAN, LO, MAN, AUTO, CAS, RCAS and ROUT.

### Control Algorithm

$$\text{OUT} = \text{GAIN} * \left[ E + \frac{\text{RATE} * S}{1 + \alpha * \text{RATE} * S} * \text{PV} + \frac{E}{\text{RESET} * S} \right] + \text{BIAS\_A/M} + \text{FEEDFORWARD}$$

NOTE: ① BIAS\_A/M: Internal BIAS Calculated on changing to automatic modes (RCAS, CAS, AUTO).

- $\alpha$ : Pseudo - Derivative Gain Equals to 0.13

### Example for Control Algorithm

Below there is an example of how the BIAS\_A/M internal variable works in order to have bumpless output.

Initially it considers the values:

PV=10  
 SP=60  
 GAIN=1  
 RATE=0  
 RESET=+Inf  
 FEEDFORWARD=0  
 CONTROL\_OPTS=None (reverse direction, SP-PV)

For PV equals to 10 and SP equals to 60, the calculated Error will be 50 (SP-PV). In t1, a transition from Automatic mode to Manual mode occurs and the value 40 is attributed to the OUT.

So, according to the Control Algorithm equation, the BIAS\_A/M variable will be -10

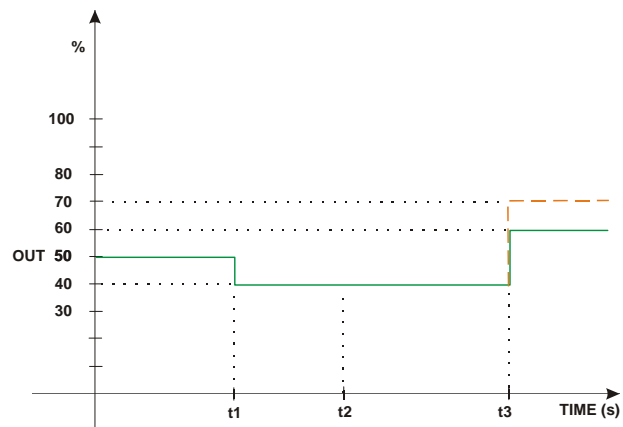
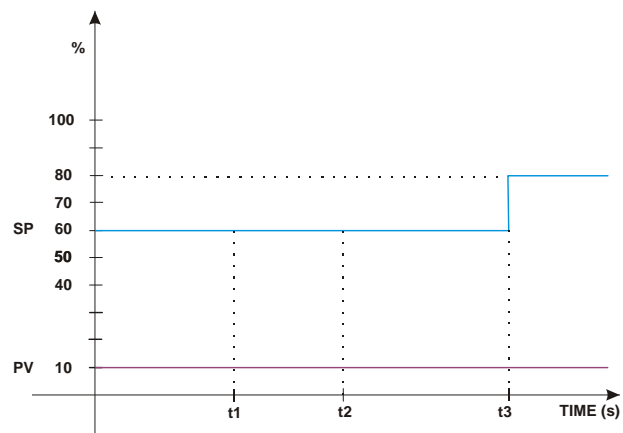
$$\text{OUT} = \text{GAIN} * \left[ E + \frac{\text{RATE} * S}{1 + \alpha * \text{RATE} * S} * \text{PV} + \frac{E}{\text{RESET} * S} \right] + \text{BIAS\_A/M} + \text{FEEDFORWARD}$$

$$\text{BIAS\_A/M} = \text{OUT} - \text{GAIN} * E$$

$$\text{BIAS\_A/M} = 40 - 1 * 50$$

$$\text{BIAS\_A/M} = -10$$

In t2, block returns to the Automatic mode. In order to have a bumpless output, BIAS\_A/M is applied to the Error, so the output is kept in the last value. In t3, SP receives a step equals to 20, it goes from 60 to 80, then the calculated Error would be 70 (SP-PV), and how the BIAS\_A/M is applied to the Error, so the step in the output will be 20.

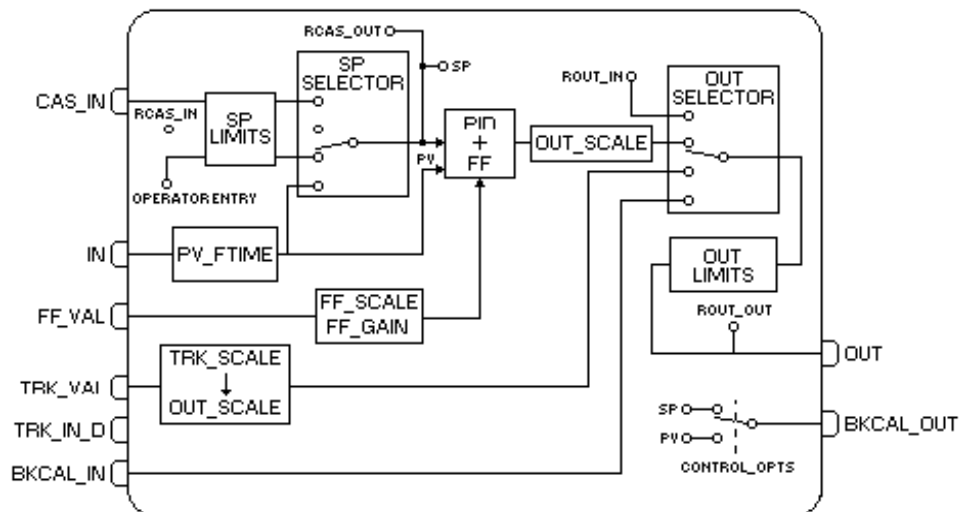


t1 – Automatic to Manual transition (time that the value 40 is attributed to the OUT)

t2 – Manual to Automatic transition

t3 – SP receives a step of 20, then the OUT will also receive a step equals to 20

### Schematic





## Parameters

| Idx | Parameter    | Data Type (length) | Valid Range/Options | Default Value | Units  | Store / Mode | Description  |
|-----|--------------|--------------------|---------------------|---------------|--------|--------------|--|
| 1   | ST_REV       | Unsigned16         |                     | 0             | None   | S/RO         |  |
| 2   | TAG_DESC     | OctString(32)      |                     | Spaces        | Na     | S            |  |
| 3   | STRATEGY     | Unsigned16         |                     | 0             | None   | S            |  |
| 4   | ALERT_KEY    | Unsigned8          | 1 to 255            | 0             | None   | S            |  |
| 5   | MODE_BLK     | DS-69              |                     | O/S           | Na     | S            | See Mode Parameter   |
| 6   | BLOCK_ERR    | Bitstring(2)       |                     |               | E      | D / RO       |  |
| 7   | PV           | DS-65              |                     |               | PV     | D / RO       | Process analog value. This is the IN value after pass over the PV filter.  |
| 8   | SP           | DS-65              | PV_SCALE +/- 10%    |               | PV     | N / Auto     | The analog set point. Can be set manually, automatically through the interface device or another field device.                                     |
| 9   | OUT          | DS-65              | OUT_SCALE +/- 10%   |               | OUT    | N / Man      | The output value result of the PID calculation.  |
| 10  | PV_SCALE     | DS-68              |                     | 0-100%        | PV     | S / Man      | The high and low scale values to the PV and SP parameter.  |
| 11  | OUT_SCALE    | DS-68              |                     | 0-100%        | OUT    | S / Man      | The high and low scale values to the OUT parameter.  |
| 12  | GRANT_DENY   | DS-70              |                     | 0             | na     | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.                   |
| 13  | CONTROL_OPTS | Bitstring(2)       | See Block Options   | 0             | na     | S / O/S      | See Block Options  |
| 14  | STATUS_OPTS  | Bitstring(2)       | See Block Options   | 0             | Na     | S / O/S      | See Block Options  |
| 15  | IN           | DS-65              |                     |               | PV     | D            | The primary input value of the block, or PV value.   |
| 16  | PV_FTIME     | Float              | Non-Negative        | 0             | Sec    | S            | Time constant of a single exponential filter for the PV, in seconds.   |
| 17  | BYPASS       | Unsigned8          | 1:Off<br>2:On       | 0             | E      | S / Man      | When bypass is set, the setpoint value (in percent) will be directly transferred to the output.  |
| 18  | CAS_IN       | DS-65              |                     |               |        | D            | This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link.                   |
| 19  | SP_RATE_DN   | Float              | Positive            | +INF          | PV/Sec | S            | Ramp rate at which upward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply only in AUTO mode.   |
| 20  | SP_RATE_UP   | Float              | Positive            | +INF          | PV/Sec | S            | Ramp rate at which downward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply only in AUTO mode. |
| 21  | SP_HI_LIM    | Float              | PV_SCALE +/- 10%    | 100           | PV     | S            | The setpoint high limit is the highest setpoint operator entry that can be used for the block.   |
| 22  | SP_LO_LIM    | Float              | PV_SCALE +/- 10%    | 0             | PV     | S            | The setpoint low limit is the lowest setpoint operator entry that can be used for the block.   |
| 23  | GAIN         | Float              |                     | 0             | None   | S            | Proportional term of the PID. It is the Kp value.  |
| 24  | RESET        | Float              | Positive            | +INF          | sec    | S            | Integral term of the PID. It is the Tr value.  |

| Idx | Parameter  | Data Type (length) | Valid Range/ Options   | Default Value | Units | Store / Mode | Description   |
|-----|------------|--------------------|--|---------------|-------|--------------|---|
| 25  | BAL_TIME   | Float              | Positive   | 0             | sec   | S            | This specifies the time for the internal working value of bias or ratio to return to the operator set bias or ratio, in seconds.<br><br>In the PID block, it may be 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. |
| 26  | RATE       | Float              | Positive   | 0             | sec   | S            | <b>Derivative term of the PID. It is the Td value.</b>  |
| 27  | BKCAL_IN   | DS-65              |  |               | OUT   | N            | <b>The value and status from a lower block's BKCAL_OUT that is used to prevent reset windup and to initialize the control loop.</b>   |
| 28  | OUT_HI_LIM | Float              | OUT_SCALE +/- 10%  | 100           | OUT   | S            | Limits the maximum output value.  |
| 29  | OUT_LO_LIM | Float              | OUT_SCALE +/- 10%  | 0             | OUT   | S            | Limits the minimum output value.  |
| 30  | BKCAL_HYS  | Float              | 0 to 50%   | 0.5%          | %     | S            | The amount that the output must change away from its output limit before the limit status is turned off, expressed as a percent of the span of the output.  |
| 31  | BKCAL_OUT  | DS-65              |  |               | PV    | D / RO       | <b>The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control.</b>  |
| 32  | RCAS_IN    | DS-65              |  |               | PV    | D            | Target setpoint and status provided by a supervisory Host to a analog control or output block.  |
| 33  | ROUT_IN    | DS-65              |  |               | OUT   | D            | Target output and status provided by a Host to the control block for use as the output (Rout mode).   |
| 34  | SHED_OPT   | Unsigned8          | 1: NormalShed, NormalReturn<br>2: NormalShed, NoReturn<br>3: ShedToAuto, NormalReturn<br>4: ShedToAuto, NoReturn<br>5: ShedToMan, NormalReturn<br>6: ShedToMan, NoReturn<br>7: ShedToRetainedTarget, NormalReturn<br>8: ShedToRetainedTarget, NoReturn | 0             |       | S            | Defines action to be taken on remote control device timeout.  |
| 35  | RCAS_OUT   | DS-65              |  |               | PV    | D / RO       | Block setpoint and status after ramping - provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change.   |
| 36  | ROUT_OUT   | DS-65              |  |               | OUT   | D / RO       | Block output and status - provided to a Host for back calculation in ROut mode and to allow action to be taken under limited conditions or mode change  |

| Idx | Parameter  | Data Type (length) | Valid Range/ Options                      | Default Value | Units  | Store / Mode | Description  |
|-----|------------|--------------------|---|---------------|--------|--------------|--|
| 37  | TRK_SCALE  | DS-68              |   | 0-100%        | TRK    | S / Man      | 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   | DS-66              |   |               | On/Off | D            | <b>This discrete input is used to initiate external tracking of the block output to the value specified by TRK_VAL.</b>  |
| 39  | TRK_VAL    | DS-65              |   |               | TRK    | D            | <b>This input is used as the track value when external tracking is enabled by TRK_IN_D.</b>  |
| 40  | FF_VAL     | DS-65              |   |               | FF     | D            | The feed forward value and status.   |
| 41  | FF_SCALE   | DS-68              |   | 0-100%        | FF     | S            | The feedforward input high and low scale values, engineering units code, and number of digits to the right of the decimal point.   |
| 42  | FF_GAIN    | Float              |   | 0             | none   | S/Man        | The gain that the feed forward input is multiplied by before it is added to the calculated control output.   |
| 43  | UPDATE_EVT | DS-73              |   |               | Na     | D            | This alert is generated by any change to the static data.  |
| 44  | BLOCK_ALM  | DS-72              |   |               | Na     | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 45  | ALARM_SUM  | DS-74              | See Block Options                         |               | Na     | S            | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 46  | ACK_OPTION | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable | 0             | Na     | S            | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 47  | ALARM_HYS  | Float              | 0 to 50 %                                 | 0.5%          | %      | S            | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis.  |
| 48  | HI_HI_PRI  | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the high high alarm.   |
| 49  | HI_HI_LIM  | Float              | PV_SCALE, +INF                            | +INF          | PV     | S            | The setting for high high alarm in engineering units.  |
| 50  | HI_PRI     | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the high alarm.  |
| 51  | HI_LIM     | Float              | PV_SCALE, +INF                            | +INF          | PV     | S            | The setting for high alarm in engineering units.   |
| 52  | LO_PRI     | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the low alarm.   |
| 53  | LO_LIM     | Float              | PV_SCALE, +INF                            | -INF          | PV     | S            | The setting for low alarm in engineering units.  |
| 54  | LO_LO_PRI  | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the low low alarm.   |
| 55  | LO_LO_LIM  | Float              | PV_SCALE, +INF                            | -INF          | PV     | S            | The setting for low low alarm in engineering units.  |
| 56  | DV_HI_PRI  | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the deviation high alarm.  |
| 57  | DV_HI_LIM  | Float              | 0 to PV span, +INF                        | +INF          | PV     | S            | The setting for deviation high alarm in engineering units.   |
| 58  | DV_LO_PRI  | Unsigned8          | 0 to 15                                   | 0             |        | S            | Priority of the deviation low alarm.   |
| 59  | DV_LO_LIM  | Float              | -INF, -PV span to 0                       | -INF          | PV     | S            | The setting for deviation low alarm in engineering units.  |
| 60  | HI_HI_ALM  | DS-71              |   |               | PV     | D            | The status for high high alarm and its associated time stamp.  |

| Idx | Parameter | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 61  | HI_ALM    | DS-71                 |                         |                  | PV    | D               | The status for high alarm and its associated time stamp.           |
| 62  | LO_ALM    | DS-71                 |                         |                  | PV    | D               | The status for low alarm and its associated time stamp.            |
| 63  | LO_LO_ALM | DS-71                 |                         |                  | PV    | D               | The status for low low alarm and its associated time stamp.        |
| 64  | DV_HI_ALM | DS-71                 |                         |                  | PV    | D               | The status for deviation high alarm and its associated time stamp. |
| 65  | DV_LO_ALM | DS-71                 |                         |                  | PV    | D               | The status for deviation low alarm and its associated time stamp.  |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

**Gray Background Line:** Default Parameters of Syscon

## EPID – Enhanced PID Control

### Description

The EPID block has all parameters of the PID block. Additionally it provides 4 types for bumpless transference from Manual mode to Auto mode, and also a special treatment for tracking output

### Additional Parameters

#### 1- Different type of transfer from a “manual” mode to an “automatic” mode.

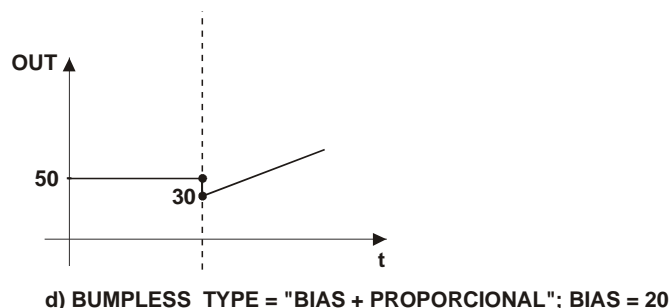
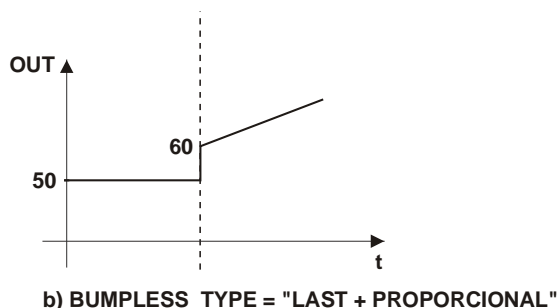
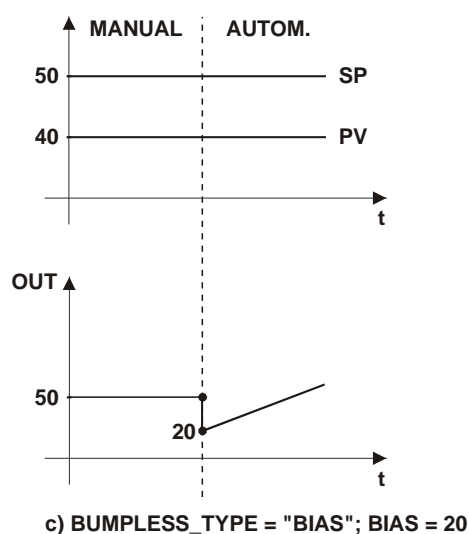
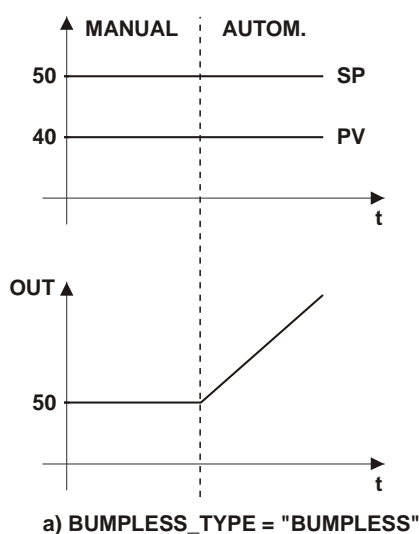
The BUMPLESS\_TYPE parameter provides four types of transfer from a “manual” mode to an “automatic” mode:

**a.bumpless** : This is the default value and the behavior of the standard PID block. The block starts to calculate from the last value in the “manual” mode.

**b.Last + proportional**: The block starts to calculate from the last value in the “manual” mode plus the proportional term

**c. Bias**: The block starts to calculate from the BIAS parameter.

**d. Bias + proportional**: The block starts to calculate from the BIAS parameter plus the proportional term.



#### 2-Special treatment to Output Tracking

Special treatment is made when the output tracking is enabled:

The algorithm generates an IFS status in the output in the following situations:

- When TRK\_IN\_D has an unusable status and the “IFS if Bad TRK\_IN\_D” bit in PID\_OPTS is true.
- When TRK\_VAL has an unusable status and the “IFS if Bad TRK\_VAL” bit in PID\_OPTS is true.

The mode is changed to Man when the tracking inputs are not usable in the following ways:

- When the TRK\_IN\_D is not usable and the “Man if Bad TRK\_IN\_D” bit in PID\_OPTS is true then the mode will be Man and the OUT will be the last value. Optionally, if the “target to Man if Bad TRK\_IN\_D” bit in the PID\_OPTS is true, then the target mode will be changed to Man too.

- When the TRK\_VAL is not usable and the “Man if Bad TRK\_VAL” bit in PID\_OPTS is true, then the mode will be Man and the OUT will be the last usable value. Optionally, if the “target to Man if Bad TRK\_VAL” bit in the PID\_OPTS is true, then the target mode will be changed to Man too.

Optionally, the block target mode will be changed to Manual by the block algorithm when the “tracking” is active. To set this feature, the “Target to Man if tracking active” bit in the PID\_OPTS parameter needs to be true.

The required actions are summarized in the following table:

| Situation  | PID_OPTS  | Mode   |                | Algorithm Action   |
|--|---|--------|----------------|--|
|  |   | Target | Actual         |  |
| TRK_IN_D is not usable   | 0x00  |        | “auto”         | . Output tracking is not active.<br>. The algorithm continues the normal calculation.  |
|  | IFS if Bad TRK_IN_D                                     |        | “auto” -> Iman | . Output tracking is not active.<br>. The algorithm continues the normal calculation.<br>. OUT.Status is GoodC-IFS.<br>. When the output block goes to fault state, the upper blocks go to Iman. |
|  | Man if Bad TRK_IN_D                                     |        | Man            | . Output tracking is not active.<br>. The algorithm stops the calculation.   |
|  | “Target to Man if Bad TRK_IN_D” ; “Man if Bad TRK_IN_D” | Man    | Man            | . Output tracking is not active. The target mode is changed to Man.  |
| TRK_VAL is not usable  | 0x00  |        | “auto”         | . Output tracking is not active.<br>. The algorithm continues the normal calculation.  |
|  | IFS if Bad TRK_VAL                                      |        | “auto” -> Iman | . Output tracking is not active.<br>. The algorithm continues the normal calculation.<br>OUT.Status is GoodC-IFS.<br>. When the output block goes to fault state, the upper blocks go to Iman.   |
|  | Man if Bad TRK_VAL                                      |        | Man            | . Output tracking is not active.<br>. The algorithm stops the calculation.   |
|  | “Target to Man if Bad TRK_VAL” ; “Man if Bad TRK_VAL”   | Man    | Man            | . Output tracking is not active. the target mode is changed to Man.  |
| TRK_IN_D and TRK_VAL is usable, TRK_IN_D is active, output tracking is enabled |   |        | LO             | Output Tracking is active.   |

Optionally, the block target mode will be changed to Manual when the Device is in “Power Up”. To set this feature, the “Target to Man if Power Up” bit in the PID\_OPTS parameter needs to be true.

If the additional parameters of EPID block are configured with the default values, the block works as the standard PID block.

## Parameters

| Idx | Parameter     | Type         | Valid Range/<br>Options  | Default<br>Value | Units | Mode To<br>Change | Description  |
|-----|---------------|--------------|--|------------------|-------|-------------------|--|
| 66  | BUMPLESS_TYPE | Unsigned8    | 0: Bumpless<br>1: Last+Proportional<br>2: Bias<br>3: Bias+Proportional | 0                | E     | S / Man           | Options that defines the algorithm action to start the output when the block transfer from “manual” to “automatic” mode. |
| 67  | BIAS          | Float        |  | 0                | OUT   | S                 | The bias value to use in the PID algorithm when the BUMPLESS type is “Bias” or “Bias+Proportional”.                      |
| 68  | PID_OPTS      | Bitstring(2) | See block options  | 0                |       | S / O/S           | The options for handling the additional features of the output tracking.   |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of BYPASS is OFF.

The default value of SHED\_OPT is NormalShed/NormalReturn.

The required mode for writing is the Actual mode for the parameters SP and OUT, regardless the Target mode.

## APID – Advanced PID Control

### Overview

The advanced PID function block provides the following additional features comparing to the standard PID algorithm and the enhanced PID:

- Selection of the terms (proportional, integral, derivative) calculated on error or process variable
- PI Sampling algorithm
- Adaptive gain
- Configurable Limits of anti reset wind-up
- Special treatment for the error
- Discrete output to indicate the actual mode

The standard features as well the enhanced ones are described in the PID and EPID blocks, respectively, therefore they will not be repeated here.

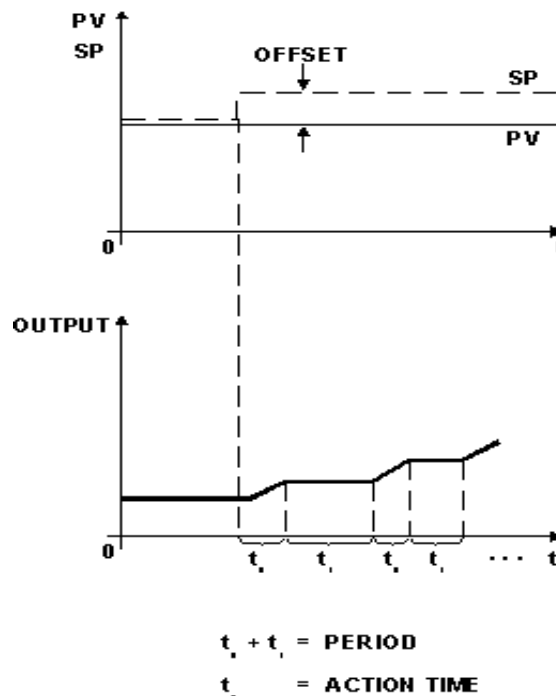
### Description

Selection of the terms (proportional, integral, derivative) calculated on error or process variable.

The control algorithm can be ideal parallel or not iterative (ISA). For each algorithm can choose the terms (proportional, integral, and derivative) calculated on error or process variable by setting the PID\_TYPE parameter. Where:

- PI.D - The P and I terms are calculated based on the error and the D term on the PV.
- PID - The P, I and D terms are calculated based on the error.
- I.PD - The I is calculated based on the error and the P and D terms on the PV.

### PI Sampling algorithm



The output is calculated based on the PI algorithm during time  $t_0$ . After that, the algorithm stops calculating and it holds the last value during time  $t_1$ . The time  $t_0$  is adjusted by SAMP\_ON, and  $t_1$  by (SAMP\_PER – SAMP\_ON).

If the SAMP\_PER is less than SAMP\_ON or SAMP\_ON is zero, then the algorithm works as an ordinary PI controller.

### Adaptive gain

The adaptive gain permits to change the algorithm PID terms by a factor obtained in a curve established by CURVE\_X and CURVE\_Y parameters. This curve is based in SP, PV, Error, OUT or another value set in AD\_GAIN\_IN parameter. The algorithm actions that will be changed are defined



by the AD\_GAIN\_ACTION parameter. The AD\_GAIN\_IN\_SEL parameter selects the input value to enter into the curve in order to get the adaptive gain.

The CURVE\_X points of the curve are in the same engineering units of the selected variable. The CURVE\_Y points are the adaptive gain. The adaptive gain (G) changes the PID constants GAIN, RESET and RATE to:

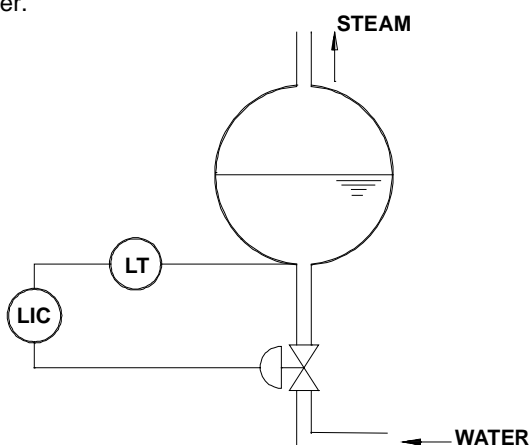
$$\text{GAIN}' = G * \text{GAIN}$$

$$\text{RESET}' = \text{RESET} / G$$

$$\text{RATE}' = G * \text{RATE}$$

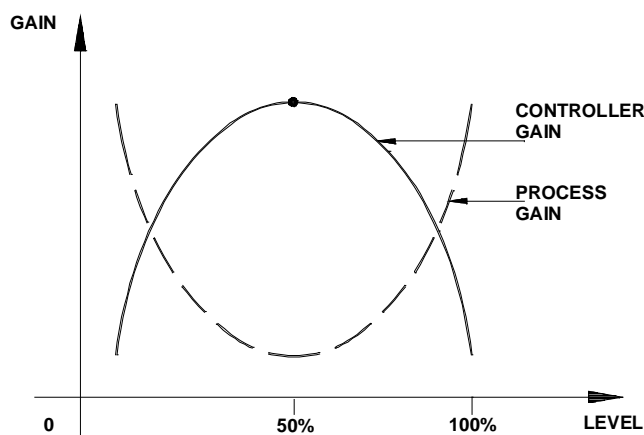
If the curve has less than 20 configured points, the non-configured points shall be set with +INF. The curve must have crescent values in the X axis. Any configuration error will be indicated by the BLOCK\_ALM parameter. If the curve has a configuration error, and thus the adaptive gain value will be the CURVE\_Y corresponding to the highest CURVE\_X point.

If the AD\_GAIN\_IN is selected and it has a bad status, the algorithm uses the last usable value to provide bumpless transfer.



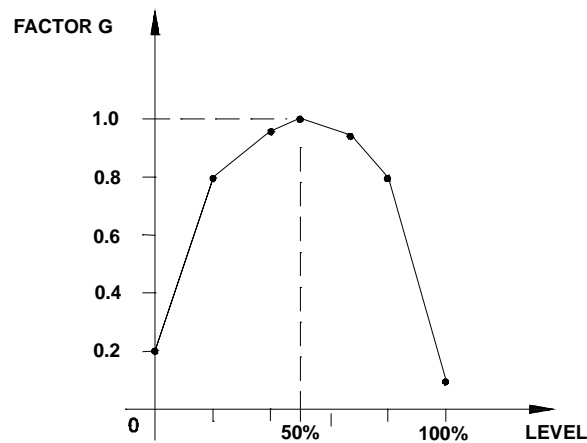
The volume variations are nonlinear with the level variations. The dotted line of figure below shows the volume gain with the level. Note that the volume varies slowly (low gain), around 50% level and varies very fast (high gain) around the level extremes.

The control action must have a gain that is the inverse of the process gain. This is shown by the continuous line of the figure below.



**Process and Controller Gain**

The adaptive gain characteristic can be configured as shown in Fig 4.9.7. This curve can be represented by the following points of Curve 1: (X1 = 0 - Y1 = 0.2; X2 = 20 - Y2 = 0.8; X3 = 40 - Y3 = 0.96; etc.).



**Gain Curve as a Function of PV**

While planning the configuration, observe the following:

1. It is not necessary to use all 13 points of the curve.
2. It is fundamental to use the 0% and the 100% of the determining variable (-100 and +100% for the Error).
3. It is recommendable to program the variable up to 102%, since the variable may be above 100%.
4. Tuning is normally done for  $G = 1$ . In the example, the control becomes slower above or below 50% of the level.

Adaptative Gain is also very useful for pH control.

#### Configurable Limits of anti reset wind-up

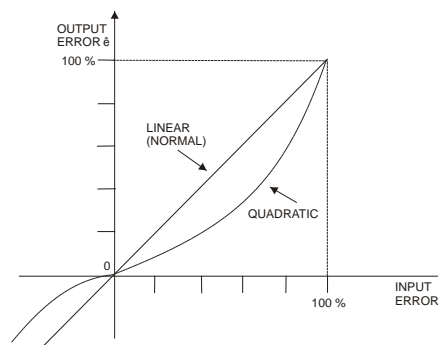
The saturation limits to integral term can be changed by the ARW\_LOW and ARW\_UP parameters. Then the control algorithm stops the integral calculation when the output signal reaches the anti-reset wind-up limits. The proportional and derivative calculations are not affected.

The Anti Reset Wind-up will not be stopped to the output limits, i.e., when the ARW\_UP limit is greater than OUT\_HI\_LIM, the OUT is clamped in the OUT\_HI\_LIM value, but internally the algorithm continue the integral calculation until the ARW\_UP limit. The user can avoid this case configuring the ARW\_UP less or equal to OUT\_HI\_LIM. The same idea applies to the low limit.

#### Special treatment for the error

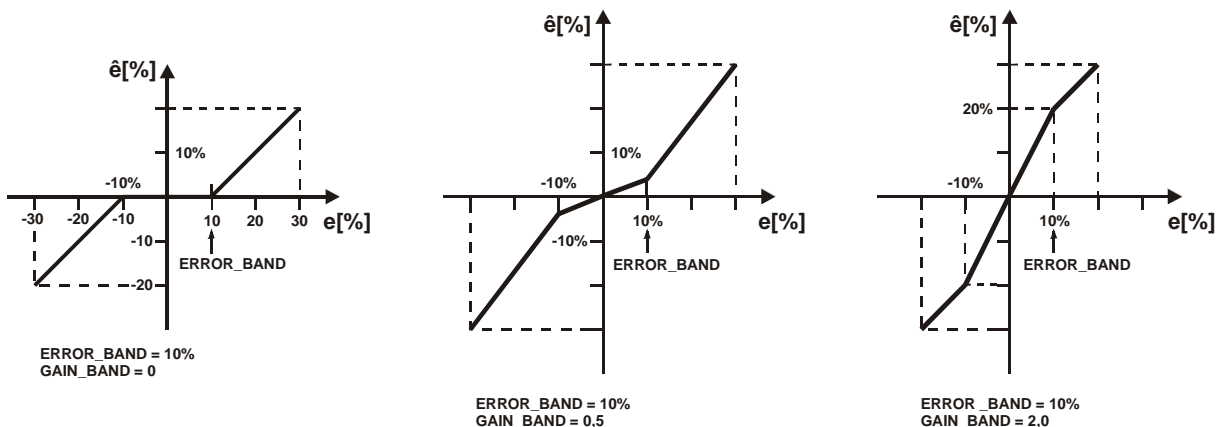
The treatment of the error in the control process can be chosen by the ERROR\_TYPE parameter. The quadratic error can be applied on only integral term or on all PID terms. In the quadratic error, the considered error for the calculation will be:

$$\hat{e} = \frac{e * |e|}{100}$$



In order to use the GAP control to consider the cases where the control is unstable over a band around SP due the dead band of the actuator or due the noise or other reasons, there is a special gain in the error type.

In order to use the ERROR\_TYPE as special gain it is necessary to define the ERROR\_BAND where it will be applied the special gain GAIN\_BAND parameter on the error. If the ERROR\_BAND is zero the algorithm will not apply the special gain.



### Mode indication

The MODE\_IND parameter is used to configure which mode(s) types in the Actual mode will be indicated by a TRUE value in the discrete output MODE\_OUT. If more than one mode type is chosen, then it will be used an OR logic.

### Working as a standard PID

If the additional parameters of APID block is configured with the default values, then it works as the standard PID block.

### BLOCK\_ERR

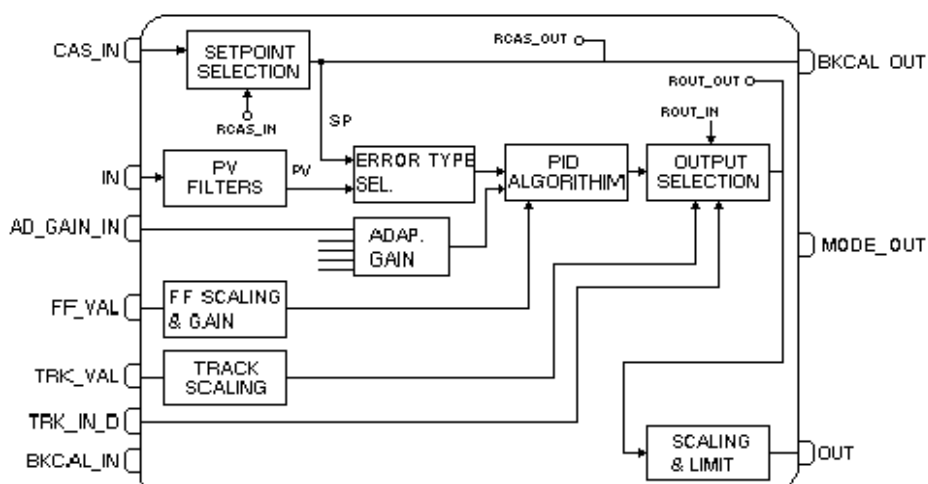
The BLOCK\_ERR will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the BYPASS or SHED\_OPT parameter has an invalid value or the curve has any of the following problems:
  - CURVE\_X[i] > CURVE\_X[i+1]
  - If the curve is not using **effectively** 20 points and any non-configured point is different from +INFINITY.
- Out of Service – it occurs when the block is in O/S mode.

### Supported modes

O/S, IMAN, LO, MAN, AUTO, CAS, RCAS and ROUT.

### Schematic



## Parameters

| Idx | Parameter      | Data Type (length) | Valid Range/ Options   | Default Value | Units                                     | Store / Mode | Description   |
|-----|----------------|--------------------|--|---------------|---|--------------|---|
| 66  | MODE_OUT       | DS-66              |  |               |   | D / RO       | This output parameter indicates if one of selected modes in the parameter "MODE_IND" is the actual mode                                   |
| 67  | MODE_IND       | Bitstring          | Actual mode bitstring  | 0             |   | D            | Selection of mode(s) to be indicated in the output parameter "MODE_OUT".  |
| 68  | AD_GAIN_ACTION | Unsigned8          | 0: Disable<br>1: PID<br>2: PI<br>3: P<br>4: I<br>5: D  | 0             | E   | S / Man      | It chooses the term(s) of PID algorithm multiplied by the adaptive gain.  |
| 69  | AD_GAIN_IN_SEL | Unsigned8          | 0: SP<br>1: PV<br>2: Error<br>3: OUT<br>4: AD_GAIN_IN  | 0             | E   | S / Man      | It selects the input to enter into the curve in order to get the adaptive gain. The option Error may be selected if ERROR_TYPE is Normal. |
| 70  | AD_GAIN_IN     | DS-65              |  |               |   | D            | Input parameter to enter into the curve in order to get the adaptive gain.  |
| 71  | CURVE_X        | 20 Floats          |  | 0's           | EU of variable selected by AD_GAIN_IN_SEL | S            | Curve input points. The $x_i$ points of the curve are defined by an array of twenty points.   |
| 72  | CURVE_Y        | 20 Floats          |  | 0's           | Na  | S            | Curve output points. The $y_i$ points of the curve are defined by an array of twenty points.  |
| 73  | ERROR_TYPE     | Unsigned8          | 0: Normal<br>1: Quadratic (Integral)<br>2: Quadratic (all terms)<br>3: Special gain  | 0             | E   | S / Man      | Type of error used by the PID algorithm. The options Quadratic and Special gain may be selected if AD_GAIN_IN_SEL is different of Error.  |
| 74  | ERROR_BAND     | Float              | 0-300%   | 0             | %   | S            | It is applied a special treatment for error within the "ERROR_BAND".  |
| 75  | GAIN_BAND      | Float              | 0-10   | 0             | Na  | S            | Special gain applied to the error, if it is selected in the ERROR_BAND.   |
| 76  | PID_TYPE       | Unsigned8          | 0:PI.D + ISA<br>1:PID + ISA<br>2:I.PD + ISA<br>3:PI Sampling + ISA<br>4:PI.D + Parallel<br>5:PID + Parallel<br>6:I.PD + Parallel<br>7:PI Sampling+Parallel | 0             | E   | S            | Type of PID algorithm.  |
| 77  | SAMP_ON        | Float              | 0-10800  | 0             | Sec                                       | S            | Time interval of the PID algorithm activity, therefore (SAMP_PER – SAMP_ON) means the hold time.  |

| Idx | Parameter     | Data Type<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store /<br>Mode | Description   |
|-----|---------------|-----------------------|--|------------------|-------|-----------------|---|
| 78  | SAMP_PER      | Float                 | 0-10800  | 0                | Sec   | S               | Period of the PI sampling algorithm.  |
| 79  | BUMPLESS_TYPE | Unsigned8             | 0: Bumpless<br>1: Last+Proportional<br>2: Bias<br>3: Bias+Proportional | 0                | E     | S / Man         | This parameter defines the type of transfer from a “manual” mode to an “automatic” mode.            |
| 80  | BIAS          | Float                 |  | 0                | OUT   | S               | The bias value to use in the PID algorithm when the BUMPLESS type is “Bias” or “Bias+Proportional”. |
| 81  | ARW_UP        | Float                 |  | + INF            | OUT   | S               | High limit for anti reset windup.   |
| 82  | ARW_LOW       | Float                 |  | - INF            | OUT   | S               | Low limit for anti reset windup.  |
| 83  | PID_OPTS      | Bitstring(2)          |  | 0                |       | S / O/S         | A bit string for handling the additional characteristics of the output tracking.                    |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

## ARTH - Arithmetic

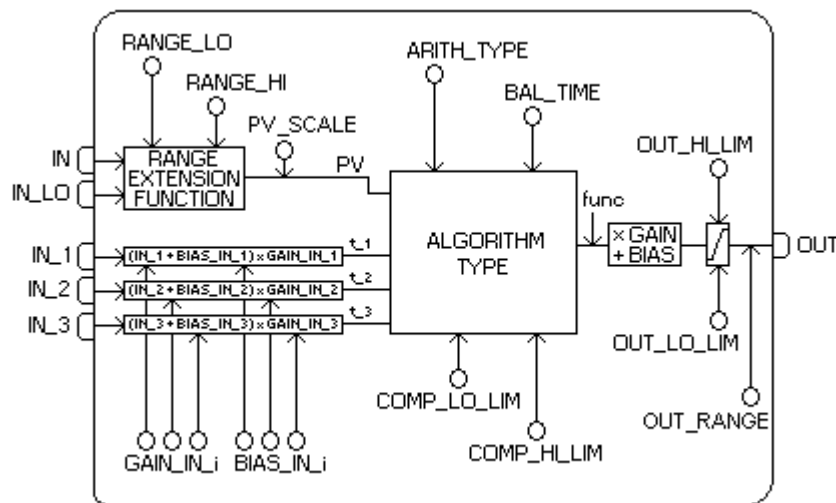
### Description

The ARTH block can be used 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 cascades or back calculation. It does no conversions to percent, so scaling is not supported. 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 must 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.

Below are detailed the steps of the calculations according to schematic.

### Schematic



### PV Treatment

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 0 to 1 over the range of RANGE\_LO to RANGE\_HI.

The equation for PV follows:

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

if :

$IN < RANGE\_LO$  or

$IN\_LO < RANGE\_HI$  and status of IN is unusable and status of IN\_LO is usable:

$g = 0$

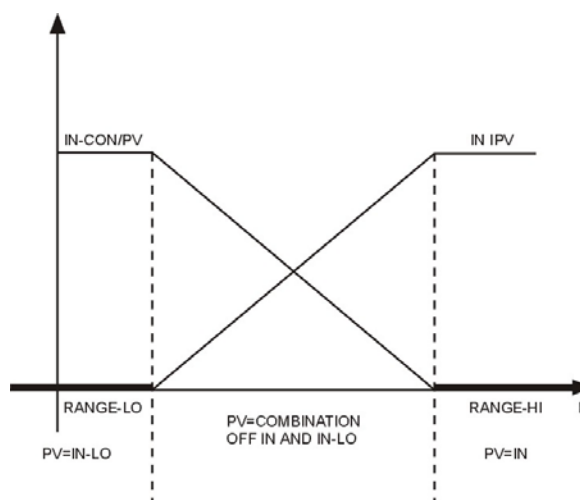
$IN > RANGE\_HI$  or

$IN > RANGE\_LO$  and status of IN is usable and status of IN\_LO is unusable:

$g = 1$

$RANGE\_LO \leq IN \leq RANGE\_HI$ :

$$g = \frac{IN - RANGE\_LO}{RANGE\_HI - RANGE\_LO}$$



If the status of IN\_LO is unusable and IN is usable and greater than RANGE\_LO, then g will be set to one. If the status of IN is unusable, and IN\_LO is usable and less than RANGE\_HI, then g will be set to zero. In each case the PV will 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.

#### Auxiliary Inputs Treatment

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 GAIN static constant. For the inputs, the bias is added and the gain is applied to the sum. The result is an internal value called t\_i in the function equations.

$$t_i = (IN_i + BIAS\_IN_i) * GAIN\_IN_i$$

#### Algorithm Calculation

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 following equations have the compensation factor limited to COMP\_HI\_LIM and COMP\_LO\_LIM:

- flow compensation, linear
- flow compensation, square root
- flow compensation, approximate
- BTU flow
- Traditional multiply divide

Arithmetic exceptions:

- a) Division by zero will produce a value equals to OUT\_HI\_LIM or OUT\_LO\_LIM, it depends on the sign of PV.
- b) Roots of negative numbers will produce the root of the absolute value, with a negative sign.

#### Output Calculation

At the end of the algorithm calculation, the GAIN and the BIAS are applied at the final value. The output still can be limited using the absolute limits of high and low (OUT\_HI\_LIM and OUT\_LO\_LIM).

#### MINIMUM CONFIGURATION

RANGE\_HI and RANGE\_LO: If the range extension function is not used, these two parameters must be set to +INF and -INF, respectively. The inputs IN\_1, IN\_2 and IN\_3 must be configured according to the equation type selected (see in the table the available equation types), or using INPUT\_OPTS to disable determined input. Therefore the PV will be a copy of IN.

If the ARITH\_TYPE is one of five first equations, the COMP\_HI\_LIM and COMP\_LO\_LIM parameters must be set properly. The default value of COMP\_HI\_LIM parameter is zero.

As the default value of GAIN parameter is zero, it is necessary to set a suitable value.

### Manual to Automatic Transition

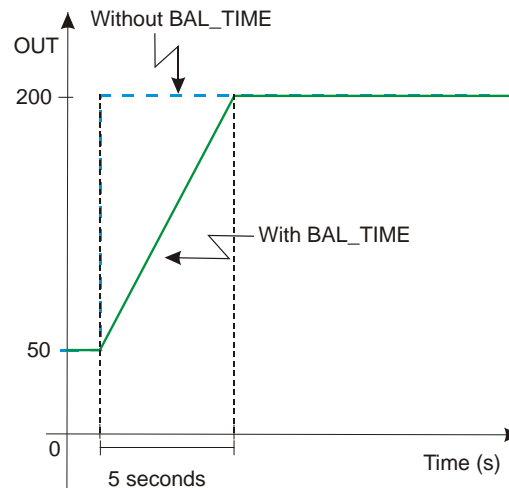
The writing in the OUT is enabled when the block is in Man mode. During the Manual to Automatic transition, if the BAL\_TIME parameter does not have configured, i.e., with the value equals to 0, the transition of the output from the written value to the calculated value will be performed with bump, such as a pulse (positive or negative). BAL\_TIME parameter allows a bumpless transition for the output. It is set in seconds. See the example below:

Consider the values:

IN\_1=10  
 GAIN\_IN\_1=10  
 BIAS\_IN\_1=5  
 IN\_2=10  
 GAIN\_IN\_2=10  
 BIAS\_IN\_2=5  
 ARITH\_TYPE=Traditional summer  
 BIAS=0  
 GAIN=2  
 BAL\_TIME=5 (seconds)

OUT (calculated by block)=200

OUT(written by user)=50



### BLOCK\_ERR

The BLOCK\_ERR of the Arithmetic block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the ARITH\_TYPE has an invalid value;
- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S, Man and Auto.

### Status Handling

The status of PV depends on the factor “g”, if it is less than 0.5, then it will be used the status of IN\_LO, and otherwise it will be the status of IN.

The INPUT\_OPTS parameter allows the usage of auxiliary inputs with less than good status. The status of unused inputs is ignored.

The status of the output will be that of the PV, except for when the status of PV is good and the status of a used auxiliary input is not good and INPUT\_OPTS is not configured to use it. In this case, the status of OUT will be Uncertain.



## Parameters

| Idx | Parameter   | DataType<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|-------------|----------------------|-------------------------|------------------|-------|----------------|--|
| 1   | ST_REV      | Unsigned16           |                         | 0                | None  | S/RO           |  |
| 2   | TAG_DESC    | OctString(32)        |                         | Spaces           | Na    | S              |  |
| 3   | STRATEGY    | Unsigned16           |                         | 0                | None  | S              |  |
| 4   | ALERT_KEY   | Unsigned8            | 1 to 255                | 0                | None  | S              |  |
| 5   | MODE_BLK    | DS-69                |                         | O/S              | Na    | S              | See Mode Parameter.  |
| 6   | BLOCK_ERR   | Bitstring(2)         |                         |                  | E     | D / RO         |  |
| 7   | PV          | DS-65                |                         |                  | PV    | D / RO         | Process analog value for use in executing the function.  |
| 8   | OUT         | DS-65                |                         |                  | OUT   | D / Man        | The analog value calculated as a result of executing the function.   |
| 9   | PRE_OUT     | DS-65                |                         |                  | OUT   | D / RO         | Displays what would be the OUT value and status if the mode was Auto or lower.   |
| 10  | PV_SCALE    | DS-68                |                         | 0                | PV    | S              | The engineering units index for display.   |
| 11  | OUT_RANGE   | DS-68                |                         | 0                | OUT   | S              | The engineering units of the output for display.   |
| 12  | GRANT_DENY  | DS-70                |                         | 0                | na    | D              | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 13  | INPUT_OPTS  | Bitstring(2)         |                         | 0                | na    | S / OOS        | Option bit string for handling the status of the auxiliary inputs.   |
| 14  | IN          | DS-65                |                         |                  | PV    | D              | The primary input of the block   |
| 15  | IN_LO       | DS-65                |                         |                  | PV    | D              | Input for the low range transmitter, in a range extension application.   |
| 16  | IN_1        | DS-65                |                         |                  | None  | D              | Numbered input 1.  |
| 17  | IN_2        | DS-65                |                         |                  | None  | D              | Numbered input 2.  |
| 18  | IN_3        | DS-65                |                         |                  | None  | D              | Numbered input 3.  |
| 19  | RANGE_HI    | Float                |                         | 0                | PV    | S              | Constant value above which the range extension has switched to the high range transmitter.                                       |
| 20  | RANGE_LO    | Float                |                         | 0                | PV    | S              | Constant value below which the range extension has switched to the low range transmitter.  |
| 21  | BIAS_IN_1   | Float                |                         | 0                | None  | S              | The constant to be added to IN_1.  |
| 22  | GAIN_IN_1   | Float                |                         | 0                | None  | S              | The constant to be multiplied times (IN_1 + bias).   |
| 23  | BIAS_IN_2   | Float                |                         | 0                | None  | S              | The constant to be added to IN_2.  |
| 24  | GAIN_IN_2   | Float                |                         | 0                | None  | S              | The constant to be multiplied times (IN_2 + bias).   |
| 25  | BIAS_IN_3   | Float                |                         |                  | None  | S              | The constant to be added to IN_3.  |
| 26  | GAIN_IN_3   | Float                |                         | 0                | None  | S              | The constant to be multiplied times (IN_3 + bias).   |
| 27  | COMP_HI_LIM | Float                |                         | 0                | None  | S              | The high limit imposed on the PV compensation term.  |
| 28  | COMP_LO_LIM | Float                |                         | 0                | None  | S              | The low limit imposed on the PV compensation term.   |

| Idx | Parameter  | DataType<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|------------|----------------------|--|------------------|-------|----------------|--|
| 29  | ARITH_TYPE | Unsigned8            | 1= Flow comp.<br>linear<br>2= Flow comp.<br>square root<br>3= Flow comp.<br>approx.<br>4= BTU flow<br>5= Traditional<br>mult. div.<br>6= Average<br>7= Traditional<br>summer<br>8= Fourth order<br>polynomial<br>9= HTG comp.<br>level | 0                | E     | S              | It identifies the equation that will be used.  |
| 30  | BAL_TIME   | Float                | Positive   | 0                | Sec   | S              | This parameter specifies the time in order to have a bumpless transition in the output during the Manual to Automatic transition.  |
| 31  | BIAS       | Float                |  | 0                | OUT   | S              | The bias value used in computing the function block output, expressed in engineering units.  |
| 32  | GAIN       | Float                |  | 0                | None  | S              | Dimensionless value used by the block algorithm in calculating the block output.   |
| 33  | OUT_HI_LIM | Float                |  | 100              | OUT   | S              | Limits the maximum output value.   |
| 34  | OUT_LO_LIM | Float                |  | 0                | OUT   | S              | Limits the minimum output value  |
| 35  | UPDATE_EVT | DS-73                |  |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 36  | BLOCK_ALM  | DS-72                |  |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of ARITH\_TYPE is the Gas flow compensation for linear transmitters, equation type 1.

The required mode for writing is the Actual mode for the OUT parameter, regardless the Target mode.

## Equation Types

| ARITH_TYPE                   | Equation  |
|------------------------------|---|
| 1<br>Flow comp. Linear       | $OUT = PV * f * GAIN + BIAS$ <p>where <math>f = \left[ \frac{T1}{T2} \right] \quad (*)</math></p>                                       |
| 2<br>Flow comp. Square root  | $OUT = PV * f * GAIN + BIAS$ <p>where <math>f = \left[ \sqrt{\frac{T1}{T2 * T3}} \right] \quad (*)</math></p>                           |
| 3<br>Flow comp. Approx.      | $OUT = PV * f * GAIN + BIAS$ <p>where <math>f = \left[ \sqrt{T1 * T2 * T3^2} \right] \quad (*)</math></p>                               |
| 4<br>BTU flow                | $OUT = PV * f * GAIN + BIAS$ <p>where <math>f = [T1 - T2] \quad (*)</math></p>  |
| 5<br>Traditional mult. div.  | $OUT = PV * f * GAIN + BIAS$ <p>where <math>f = \left[ \frac{T1}{T2} + T3 \right] \quad (*)</math></p>                                  |
| 6<br>Average                 | $OUT = \frac{PV + T1 + T2 + T3}{f} * GAIN + BIAS$ <p>Where f is number of inputs used in computation (unusable inputs are not used)</p> |
| 7<br>Traditional summer      | $OUT = (PV + T1 + T2 + T3) * GAIN + BIAS$   |
| 8<br>Fourth order polynomial | $OUT = (PV + T1^2 + T2^3 + T3^4) * GAIN + BIAS$   |
| 9<br>HTG comp. Level         | $OUT = \frac{PV - T1}{PV - T2} * GAIN + BIAS$   |

(\*) It is limited by COMP\_HI\_LIM and COMP\_LO\_LIM parameters

### Examples

| ARITH_TYPE | Example  | Example equation   | Note   |
|------------|--|--|--|
| 1          | Gas flow compensation for linear transmitters (e.g. turbine) | $Q_b = Q_f * K * \frac{P}{T}$  |  |
| 2          | Gas flow compensation<br>For DP transmitters.                | $Q_b = Q_f * K * \sqrt{\frac{P}{T * Z}}$   | Z may be constant or an input from other block<br>(AGA3) |
| 3          | Approx. liquid & steam flow comp.                            | $Q_b = Q_f * K * \sqrt{(K + K * T + K * T^2)}$<br>$Q_b = Q_f * K * \sqrt{(K + K * P)}$ | Temperature connected to 3 and 4                         |
| 4          | BTU meter (heat flow)  | $Q_{HEAT} = K * Q_{VOL} * (t_1 - t_2)$   |  |
| 5          | Simple "hard" (non-cascade) ratio                            | $Q_{SP} = Q_{WILD} * RATIO$  | output is setpoint to PID block                          |
| 6          | Average of four temperature measurements                     | $t_a = \frac{t_1 + t_2 + t_3 + t_4}{f}$  | .  |
| 7          | Difference in pressure (or level)                            | $P_{bm} = P_b - P_m$   |  |
| 9          | Simple HTG compensated level                                 | $h_{BT} = \frac{P_B - P_T}{P_B - P_M} * h_{BM}$  |  |

**NOTE:** Square root of the third power may be achieved by selecting ARITH\_TYPE = 3 and connecting input to IN and IN\_1. Square root of the fifth power may likewise be achieved by connecting the input to IN, IN\_1 and IN\_3.

## SPLT-Splitter

### Description

The Splitter block provides the capability to drive multiple outputs from a single input, usually a PID. This block would normally be used in split ranging or sequencing of multiple valve applications. Included in the block features are the capability to open valves as part of a predetermined schedule and leave open or closed a given valve after the controller has transitioned off the valve. The splitter supports two outputs. Since this block will participate in the control path after a PID block, back calculation support is included.

The application(s) targeted for the splitter block would be a single output from a controller which is used to control up to two valves in either split ranging fashion or as sequenced operation. Split ranging is that application where two valves are used such as a reactor where heating and cooling must be applied by the same controller. The controller action, direct or reversing, is implicitly reversed owing to the change in slope of the function as the input increases or decreases. Sequencing as applied to this document is when two or more valves are used to manipulate the flow of some material and the controller action is not reversed implicitly or otherwise. An example is pH control where added valves are required to increase the loop rangeability.

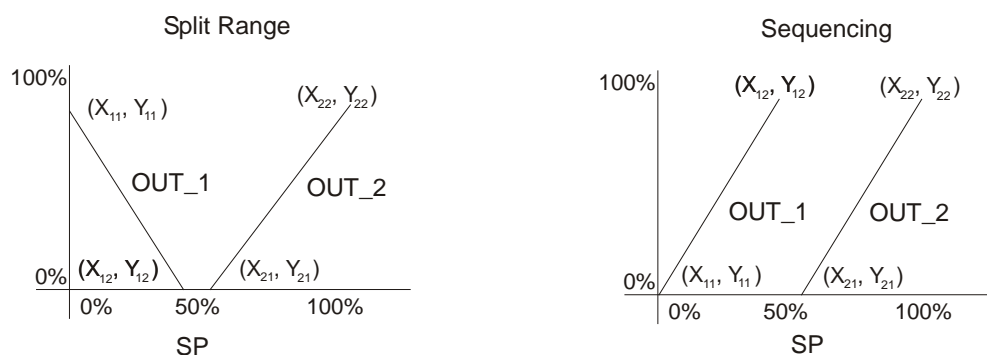
The following parameters are used to specify the signal splitter operation:

X11, Y11 X12, Y12

X21, Y21 X22, Y22

Where  $X_{n,j}$  is the value of CAS\_IN associated with OUT\_n and  $X_{n1}$  and  $X_{n2}$  refers to the 1st and 2nd coordinates respectively. The Y values are defined in the same way.

A graphical representation of OUT\_1 and OUT\_2 vs. SP is shown below. Both a Split range and Sequencing application are shown.

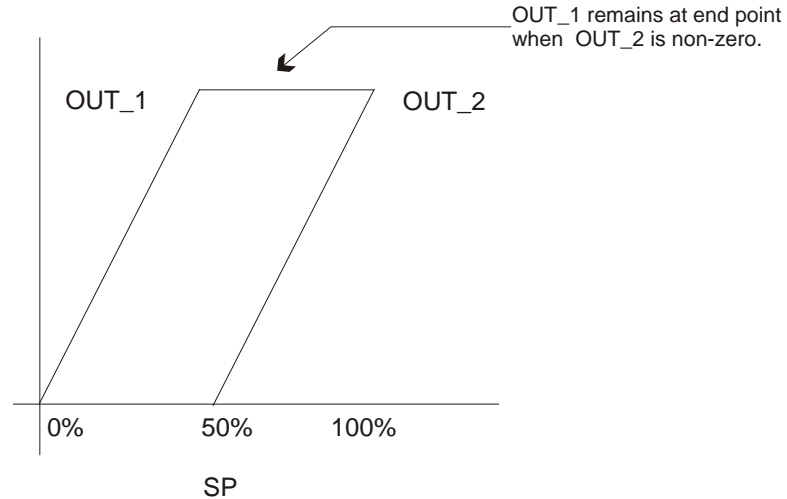


| IN_ARRAY |                 | OUT_ARRAY |                 |
|----------|-----------------|-----------|-----------------|
| 1        | X <sub>11</sub> | 1         | Y <sub>11</sub> |
| 2        | X <sub>12</sub> | 2         | Y <sub>12</sub> |
| 3        | X <sub>21</sub> | 3         | Y <sub>21</sub> |
| 4        | X <sub>22</sub> | 4         | Y <sub>22</sub> |

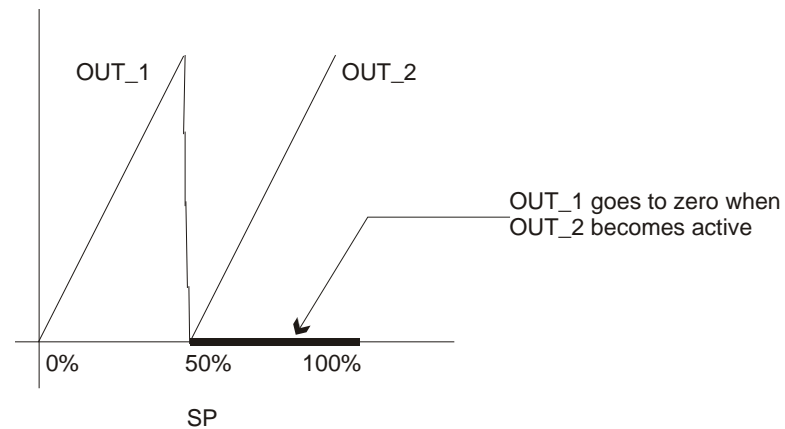
By specifying the coordinates as shown above the endpoints of the lines are defined. The contents of the respective X's and Y's are held in IN\_ARRAY and OUT\_ARRAY parameters. If a set of points is specified such that a region of the input range is not specified, then the block will interpolate to the endpoint of the input value, either high or low.

The parameter LOCKVAL provides an option to specify whether OUT\_1 remains at its ending level when control is switched to OUT\_2, or goes to zero. If LOCKVAL is true, OUT\_1 remains at its current value when OUT\_2 is non-zero. If LOCKVAL is false then as soon as the OUT\_2 becomes non-zero then OUT\_1 goes to zero. The parameter OUT\_1 will be the initial value if LOCKVAL = FALSE and the parameter OUT\_ARRAY must be "Sequencing".

In the situation below, LOCKVAL = TRUE.



In the situation below, LOCKVAL = FALSE.



The HYSTVAL parameter determines the hysteresis value in the switching point that may be required because the output may change by a full stroke of the valve. The OUT may be determined of the following way:

- If  $X \leq X12 - \text{HYSTVAL}$ , OUT\_1 may be determined by the calculated y value.
- If  $X12 - \text{HYSTVAL} < X < X12$  and X has not reached X12 since it was less than X12-HYSTVAL, OUT\_1 may be determined by the calculated y value.
- If  $X12 - \text{HYSTVAL} < X < X12$  and X has reached X12 since it was less than X12-HYSTVAL, OUT\_1 may be determined by the LOCKVAL setting.
- If  $X12 < X$ , OUT\_1 may be determined by the LOCKVAL setting.

#### BLOCK\_ERR

The BLOCK\_ERR of the Splitter block will reflect the following causes:

- Block Configuration Error – there is configuration error when the Actual mode changes to OOS in the cases:
  - When the LOCKVAL parameter has an invalid value;
  - When the IN\_ARRAY value (X axis) has one of the conditions:  $X21 < X11$ ,  $X12 \leq X11$ ,  $X22 \leq X21$ .
- Out of Service – When the block is in O/S mode.

#### Supported Modes

O/S, IMAN, Auto and Cas.

### Status Handling

Sub-status values received from CAS\_IN will be passed to both outputs, except for those used in the cascade handshake.

The STATUS\_OPTS "IFS if Bad CAS\_IN" is supported.

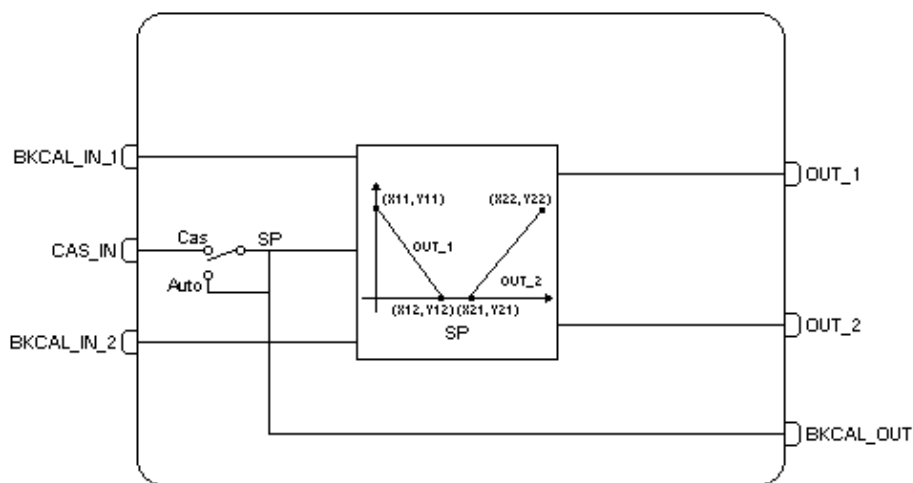
If the STATUS\_OPTS "Propagate Failure Backward" is selected, the block propagates the failure only if both BKCAL\_IN inputs show failure status.

The cascade initialization is also required when there is a Auto to Cas transition. The actions that occur are summarized in the following table:

| Actual mode | BKCAL_IN_1 | BKCAL_IN_2 | BKCAL_OUT | ACTION  |
|-------------|------------|------------|-----------|---|
| Iman        | NI         | NI         | NI        | Not specified                                       |
| Auto or Cas | NI         | OK         | OK        | BKCAL_OUT limited to X21 low and X22 high           |
| Auto or Cas | OK         | NI         | OK        | BKCAL_OUT limited to X11 low and X12 high           |
| Iman        | IR         | NI         | IR        | Initialize cascade to value given by curve X1 vs Y1 |
| Auto or Cas | IR         | OK         | OK        | Initialize OUT_1 using internal offset from Y1      |
| Iman        | NI         | IR         | IR        | Initialize cascade to value given by curve X2 vs Y2 |
| Auto or Cas | OK         | IR         | OK        | Initialize OUT_2 using internal offset from Y2      |

**Legend:** NI-not invited; IR-initialization request; OK-working in cascade

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/ Options | Default Value | Units | Store/ Mode | Description |
|-----|-----------|--------------------|----------------------|---------------|-------|-------------|-------------|
| 1   | ST_REV    | Unsigned16         |                      | 0             | None  | S/RO        |             |
| 2   | TAG_DESC  | OctString(32)      |                      | Spaces        | Na    | S           |             |
| 3   | STRATEGY  | Unsigned16         |                      | 0             | None  | S           |             |
| 4   | ALERT_KEY | Unsigned8          | 1 to 255             | 0             | None  | S           |             |

| Idx | Parameter   | Data Type (length) | Valid Range/ Options    | Default Value     | Units | Store/ Mode | Description  |
|-----|-------------|--------------------|-------------------------|-------------------|-------|-------------|--|
| 5   | MODE_BLK    | DS-69              |                         | O/S               | Na    | S           | See Mode Parameter.  |
| 6   | BLOCK_ERR   | Bitstring(2)       |                         |                   | E     | D/RO        |  |
| 7   | SP          | DS-65              |                         |                   |       | N / Auto    | The analog set point.  |
| 8   | OUT_1       | DS-65              |                         |                   | OUT1  | D / RO      | Numbered output parameter 1.   |
| 9   | OUT_2       | DS-65              |                         |                   | OUT2  | D / RO      | Numbered output parameter 2.   |
| 10  | OUT_1_RANGE | DS-68              |                         | 0                 | E     | S           | Scale range for the corresponding output.  |
| 11  | OUT_2_RANGE | DS-68              |                         | 0                 | E     | S           | Scale range for the corresponding output.  |
| 12  | GRANT_DENY  | DS-70              |                         | 0                 | Na    | D           | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.   |
| 13  | STATUS_OPTS | Bitstring(2)       |                         | 0                 | Na    | S / OOS     | Options which the user may select in the block processing of status.   |
| 14  | CAS_IN      | DS-65              |                         |                   |       | D           | This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link.   |
| 15  | BKCAL_OUT   | DS-65              |                         |                   |       | D / RO      | The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control.  |
| 16  | IN_ARRAY    | 4 Floats           |                         | 0's               |       | S           | An array which contains the values of the input or X variables.  |
| 17  | OUT_ARRAY   | 4 Floats           |                         | 0's               |       | S           | An array which contains the values of the output or Y variables.   |
| 18  | LOCKVAL     | Unsigned8          | 1 - No Lock<br>2 - Lock | 0 - Uninitialized | E     | S           | It allows to keep the OUT_1 when OUT_2 is different of zero.   |
| 19  | BKCAL_IN_1  | DS-65              |                         |                   |       | N           | The back calculated input required to initialize a lower cascade 1.  |
| 20  | BKCAL_IN_2  | DS-65              |                         |                   |       | N           | The back calculated input required to initialize a lower cascade 2.  |
| 21  | BAL_TIME    | Float              |                         | 0                 | Sec   | S           | This specifies the time for the OUT_x (from BKCAL_IN value) reaches the work value after a cascade initialization with the down block in order to have a bumpless transition..   |
| 22  | HYSTVAL     | Float              | Positive or zero        | 0                 | None  | S           | It sets the hysteresis for the operation of the LOCKVAL parameter.   |
| 23  | UPDATE_EVT  | DS-73              |                         |                   | Na    | D           | This alert is generated by any change to the static data.  |
| 24  | BLOCK_ALM   | DS-72              |                         |                   | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

- The required mode for writing is the Actual mode for the SP parameter, regardless the Target mode.
- The Default value of LOCKVAL parameter is “No Lock”.
- The Default value of IN\_ARRAY parameter is [0,50,50,100] and OUT\_ARRAY parameter is [100,0,0,100].



## CHAR - Signal Characterizer

### Description

The block calculates OUT\_1 from IN\_1 and OUT\_2 from IN\_2, according to a curve given by the points:

$[x_1; y_1], [x_2; y_2], \dots, [x_{21}; y_{21}]$

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 path is represented by IN\_1 and OUT\_1, and the path 2 is represented by IN\_2 and OUT\_2. OUT\_1 is related to IN\_1 and OUT\_2 is related to IN\_2 using the same curve, but there is no correlation between IN\_1 and IN\_2 or between OUT\_1 and OUT\_2.

In order to determine the output values, the linear correlation between two points of the curve is calculated according to the input.

The output will be limited when the input is lower than the minor value of the curve ( $x_1; y_1$ ) or upper than the bigger value of the curve ( $x_m; y_m$ ). The limit will be indicated in the output status.

For the correct action of the algorithm, the x values must be crescent.

If the curve has m points,  $m < 21$ , the non-configured points,  $[x_{m+1}; y_{m+1}], [x_{m+2}; y_{m+2}], \dots, [x_{21}; y_{21}]$  shall be set with +INF

The input status is passed to the corresponding output, so the block can be used in control mode

### Inverting Path 2

Optionally, IN\_2 and OUT\_2 meanings can be inverted providing a back path of the value for the same curve. When the SWAP\_2 parameter is set to "Swap", the inversion is done as follows:

- IN\_1 = x and OUT\_1 = y (it does not have changes)
- IN\_2 = y and OUT\_2 = x (the curve is inverted). In this situation, IN\_2 uses the engineering units of Y\_RANGE and OUT\_2 uses the engineering units of X\_RANGE.

When using "Swap" option, the curve must be monotonic in y-axis too.

The curve is called monotonic when y-values always increase or decrease, and x-values increase, that is, there are neither tops nor bottoms in the curve.

### BLOCK\_ERR

The BLOCK\_ERR will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the SWAP\_2 parameter is not initialized or the curve has any of the following problems:
  - $CURVE\_X[i] > CURVE\_X[i+1]$
  - If the curve is not using **effectively** 21 points and any non-configured point is different from +INF.
  - If SWAP\_2 is set to "Swap" and the curve is not monotonic.
- Out of Service – it occurs when the block is in O/S mode.

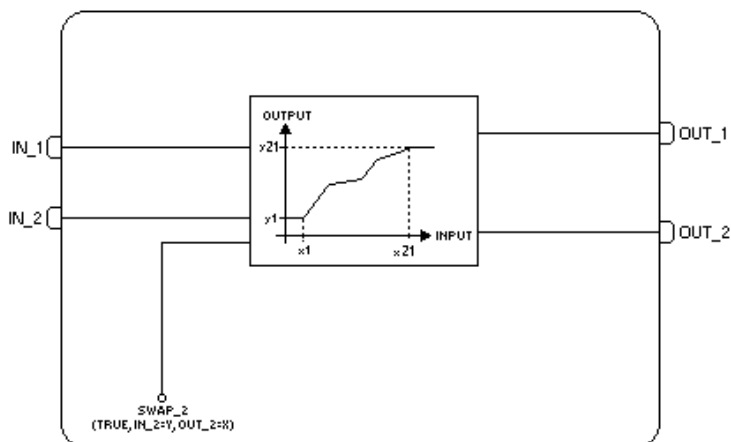
### Supported Modes

O/S and AUTO.

### Status Handling

The quality and sub-status of OUT\_1 and OUT\_2 reflect the status of IN\_1 and IN\_2, respectively. If one of the curve limits is reached, the appropriate limit is indicated. Limits are reversed if the curve slope is negative.

## Schematic



## Parameters

| Idx | Parameter  | Data Type (length) | Valid Range/Options     | Default Value | Units  | Store / Mode | Description  |
|-----|------------|--------------------|-------------------------|---------------|--------|--------------|--|
| 1   | ST_REV     | Unsigned16         |                         | 0             | None   | S/RO         |  |
| 2   | TAG_DESC   | OctString(32)      |                         | Spaces        | Na     | S            |  |
| 3   | STRATEGY   | Unsigned16         |                         | 0             | None   | S            |  |
| 4   | ALERT_KEY  | Unsigned8          | 1 to 255                | 0             | None   | S            |  |
| 5   | MODE_BLK   | DS-69              |                         | O/S           | Na     | S            | See Mode Parameter   |
| 6   | BLOCK_ERR  | Bitstring(2)       |                         |               | E      | D/RO         |  |
| 7   | OUT1       | DS-65              |                         |               | Y      | D / RO       | Numbered output parameter 1.<br>The interpolation result of IN_1.  |
| 8   | OUT2       | DS-65              |                         |               | X or Y | D / RO       | Numbered output parameter 2.<br>The interpolation result of IN_2.  |
| 9   | X_RANGE    | Unsigned16         |                         |               | E      | S            | The engineering unit of the variables corresponding to the x-axis for display.   |
| 10  | Y_RANGE    | Unsigned16         |                         |               | E      | S            | The engineering unit of the variables corresponding to the y-axis for display.   |
| 11  | GRANT_DENY | DS-70              |                         | 0             | Na     | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 12  | IN_1       | DS-65              |                         |               |        | D            | Numbered input parameter 1.  |
| 13  | IN_2       | DS-65              |                         |               |        | D            | Numbered input parameter 2.  |
| 14  | SWAP_2     | Unsigned8          | 1 – No Swap<br>2 - Swap | 0             | Na     | S            | 0 – Uninitialized<br>1 – No Swap – It does not invert the curve to OUT_2<br>2 – Swap – It inverts the curve for OUT_2            |
| 15  | CURVE_X    | 21 Floats          |                         | 0's           | X      | S            | Curve input points. The $x_i$ points of the curve are defined by an array of twenty points.                                      |
| 16  | CURVE_Y    | 21 Floats          |                         | 0's           | Y      | S            | Curve output points. The $y_i$ points of the curve are defined by an array of twenty points.                                     |
| 17  | UPDATE_EVT | DS-73              |                         |               | na     | D            | This alert is generated by any change to the static data.  |

| Idx | Parameter | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 18  | BLOCK_ALM | DS-72                 |                         |                  | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of BYPASS is "No Swap".

The default value of all elements of CURVE\_X is +INF.

The default value of all elements of CURVE\_Y is +INF.

## ECHAR – Enhanced Signal Characterizer

### Description

The ECHAR block has all parameters of the CHAR block. Additionally it provides cascade operation. The CAS\_CONTROL parameter allows the cascade operation for the block (it supports more than 21 points in the curve). To do this, it is necessary to configure in the CAS\_CONTROL parameter as follows:

- Indicate which is the first (FIRST), the intermediate blocks (INTERMEDIATE) and the last block (LAST) of the CHAR series. Thus, the input parameters will be limited by the lower X value of the curve indicated by FIRST (Low Limit) and the upper X value of the curve indicated by LAST (High Limit).
- The first, intermediate and last values are for the two inputs: IN\_1 and IN\_2.
- When the CAS\_CONTROL value is equal 0 (None), the block will work as "Alone".

To work in series, the CURVE\_X parameter must be always crescent in relation to the points inside the block and also in relation to the other blocks of the CHAR cascade. For example, the order of the block configuration of the example below must be followed:

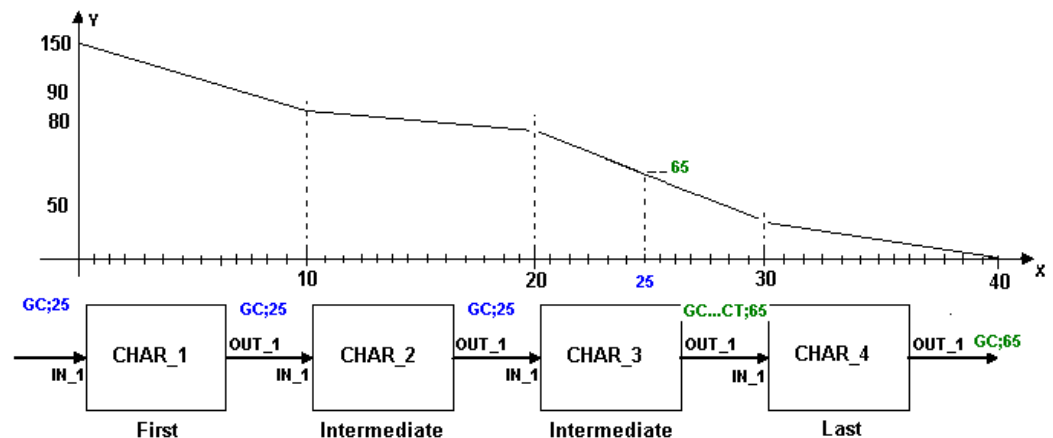
1. CHAR1 – CAS\_CONTROL = "First"
2. CHAR2 – CAS\_CONTROL = "intermediate"
3. CHAR3 – CAS\_CONTROL = "intermediate"
4. CHAR4 – CAS\_CONTROL = "Last"

If the order was not followed correctly, the blocks will not indicate configuration error, but the algorithm will not work correctly.

To work properly with the cascade, the first values X and Y of the block must be equal to the last value of the previous block. If the values are not equal to, the block will calculate the interpolation between the points.

The cascade will work as only one ECHAR block. Therefore, it supports more points than only one block. If a determined input is lower than the low X value of the first block, the last block will indicate the corresponding Y value with the output status "Low Limited". If the input is upper than the bigger X value of the last block, the corresponding Y value will be indicated with the "High Limited" status. The status are inverted if the angular coefficient of the curve is negative.

In the following example, it considers the input value for the "FIRST" block (CHAR\_1) is 25. The value is repassed for the CHAR\_1 and CHAR\_2 blocks because the input value is out of the curve limits of these blocks. The value is "resolved" by the CHAR\_3 block that repasses the Y correspondent value for the following block, as well the "Constant" Status Limit. Thus, the following blocks (CHAR\_4 in the example), when check the input with "Constant" limit, know the value was already "resolved" and repass it for the output.



The Cascade Swap is supported, since the curve was monotonic in the whole extension (always increasing or decreasing). There is no check if the curve is monotonic for all cascade blocks, thus if the curve was not monotonic, the result will be the first Y value found.

The user must certify the curve is monotonic for all cascade blocks. If it is increasing in one part and decreasing in another, the cascade can not work properly.

### Parameters

The ECHAR block has all parameters of the CHAR block, plus the parameter showed below:

| Idx | Parameter   | Data Type<br>(length) | Valid Range/<br>Options                               | Default<br>Value | Units | Store / Mode | Description  |
|-----|-------------|-----------------------|---|------------------|-------|--------------|--|
| 19  | CAS_CONTROL | Unsigned8             | 0 – None<br>1 – First<br>2 – Intermediate<br>3 – Last | 0                | E     | S/O/S        | It allows the cascade block operation:<br>0 – it does not support CHAR cascade<br>1 – First – it is the first block of the curve<br>2 – Intermediate – it is the intermediate block of the curve<br>3 – Last – it is the last block of the curve |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

**Gray Background Line:** Default Parameters of Syscon

## INTG - Integrator

### Description

The Integrator Function Block integrates a variable in function of the time or accumulates the counting of a Pulse Input block. The integrated or accumulated value is compared to pre-trip and trip limits, generating discrete signals when these limits are reached. The integrated value may go up, starting from zero, or down, starting from the trip value (parameter SP). The block has two inputs to calculate flow. This block can be also used to calculate volume or mass variation in tanks.

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

The basic function of the Integrator block is to integrate an analog value over time. It can also accumulate the count pulses coming from Pulse Input blocks or from another Integrator Block. 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. Each input can receive a measurement per unit of time (rate) or an accumulated number of pulses. Each input can receive one of the following types of variables:

**RATE** – it is used when the variable connected is a rate (For example: 10 kg/s, 50 Gal/h, etc.). This input can come from the output OUT of a Pulse Input block or from the output of an Analog Input block.

**ACCUM** – (Accumulation) it is used when the input is an accumulation, it comes from a continuous accumulation of pulses, for example when it is connected to the OUT\_ACCUM of a Pulse Input block. Another application can be an accumulation when it is connected to the Integrator block, in this case can be a relative totalization related to the first one.

The input type is configured in the bit string parameter INTEG\_OPTS. The bits corresponding to IN\_1 and IN\_2 can be set to false when the input type is RATE, or true when the input type is ACCUM

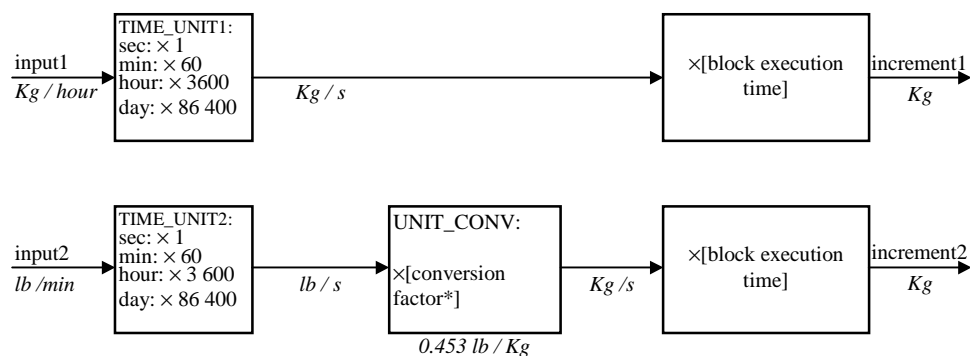
### For RATE input option

Each input needs a parameter to define the flow time unit: [TIME\_UNIT1] or [TIME\_UNIT2]. The time units are used to convert the two flows in units of mass, volume or energy per second.

The second analog input can 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 flow, multiplied by the block execution time, gives the mass, volume or energy increment per block execution. This increment will be added or subtracted in a register, according to some rules defined ahead.

The following diagram is an example of using two inputs of RATE type:



**Fig. 1 - Increment calculation with rate input**

### For ACCUM input option

The counter inputs are originated in OUT\_ACCUM output of the Pulse Input block. It can also be connected to the output of another Integrator block, and it has a relative totalization related to the first one.

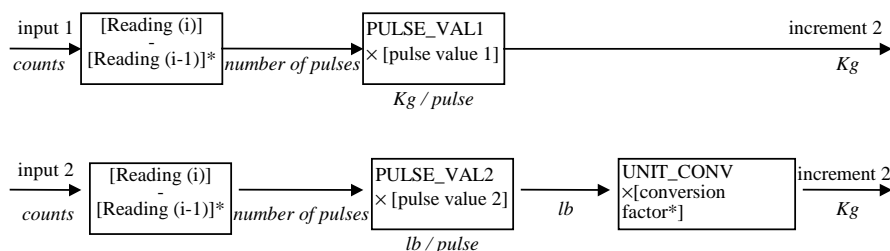
The Integrator block shall determine the variation of the counter input readings since the last execution.

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 variation is determined as follows:

- a) If the difference between the reading in one cycle and the reading in the preceding cycle is smaller than 500,000 or larger than (-500,000), the difference will be taken as variation.
- If the difference between the reading in one cycle and the reading in the preceding cycle is larger or equal to (+500,000), add (-1.000,000) and use the result as the variation.
- c) If the difference between the reading in one cycle and the reading in the preceding cycle is smaller 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 have only increasing counting.

The variation of each input must be multiplied by the value, in engineering units, of each pulse given by: **PULSE\_VAL1** and **PULSE\_VAL2**. The result is the increment in engineering units of, for example, mass, volume or energy per block execution.



**Fig. 2- Increment calculation with counter input**

### Net Flow

In order to distinguish forward and reverse flows, the Integrator block considers a negative sign as an indication of reverse flow. Some flow meters 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, with the following options:

0 = False (not invert signal)

1 = True (invert signal)

Where True will invert the signal of the IN\_i signal.

The net flow is obtained by adding the two increments. The net increment will 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 will 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 will be treated as zero. The option bit Reverse must be set to *true*.

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

### Integration of Inputs

There are two values 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.

The OUT has the value of the TOTAL register, and the RTOTAL has the value of the RTOTAL register.

The status that will be considered as usable for the algorithm are defined in INTEG\_OPTS parameter:

USE\_UNCERTAIN = Use *good* and *uncertain* inputs when this bit is set to *true*.

USE\_BAD = Use the last *good* or *uncertain* input before it went to *bad* when this bit is set to *true*.

### Integration Types

The integration can start from zero and go up or it can start from a Setpoint value (TOTAL\_SP) and go down. The totalization may be reset automatically, periodically or by batch. This is defined by the parameter INTEG\_TYPE:

- UP\_AUTO - Counts up starting from zero with automatic reset when SP is reached
- UP\_DEM - Counts up starting from zero with demand reset
- DN\_AUTO - Counts down starting from SP with auto reset when zero is reached
- DN\_DEM - Counts down starting from SP 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

### Resetting the Totals

The block has a discrete input (RESET\_IN) to reset the totalization. If this parameter is set to 1, the block will be in Reset. It initializes the totalization only after going to false. The operator can send an operator command by making OP\_CMD\_INT = RESET, because it has the same effect that the RESET\_IN parameter.

Block takes a snapshot of TOTAL(OUT), RTOTAL and TOTAL\_SP prior the Reset and keeps the information in the registers STOTAL, SRTOTAL and SSP, respectively. The information is kept at least until the next reset.

The integrator shall reject reset requests for 5 seconds after a reset. This is to guarantee that the snapshots values are exposed to FIELDBUS 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 provides a guarantee that another equipment 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.

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.

### Batch Totalizer Outputs

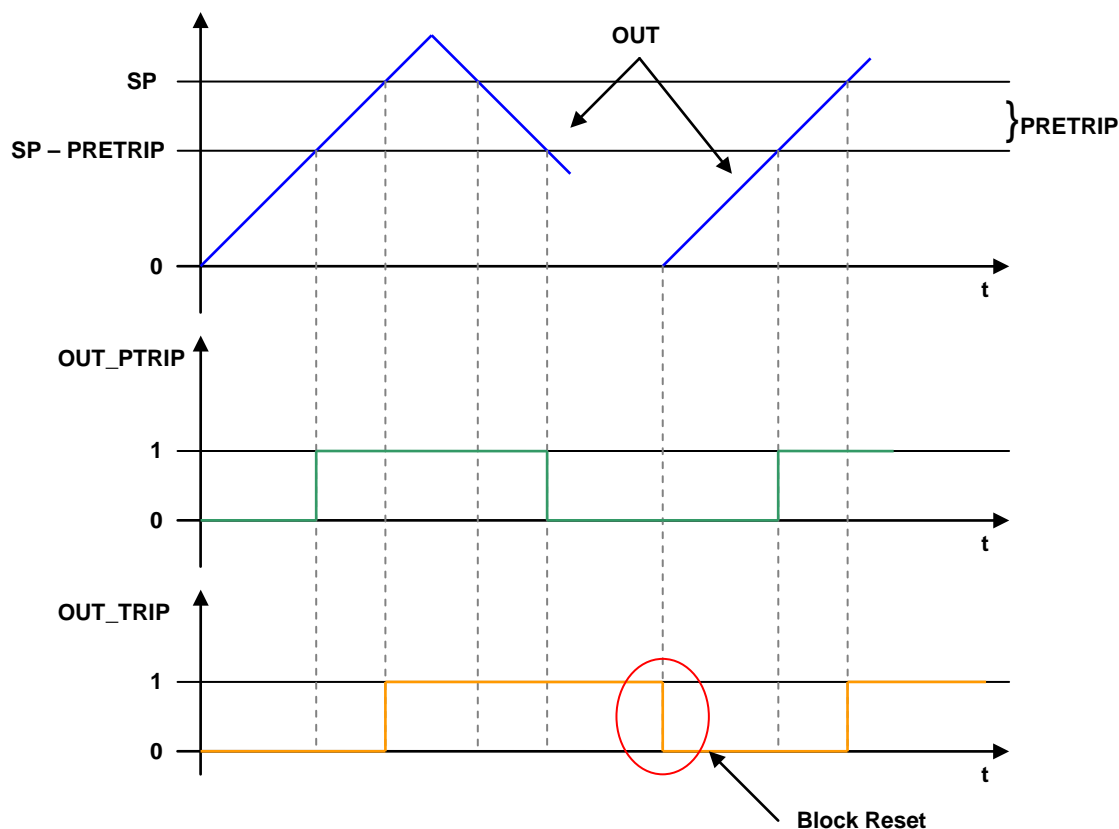
When the integration is counting up (type 1 or 2) and the value of OUT equals or exceeds a value given by TOTAL\_SP minus PRE\_TRIP then the discrete output OUT\_PTRIP is set. When it equals or exceeds a value given by the parameter TOTAL\_SP, the discrete output OUT\_TRIP is set. OUT\_PTRIP remains 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.

### Examples for Positive and Negative Totalizations

#### - Positive Totalization

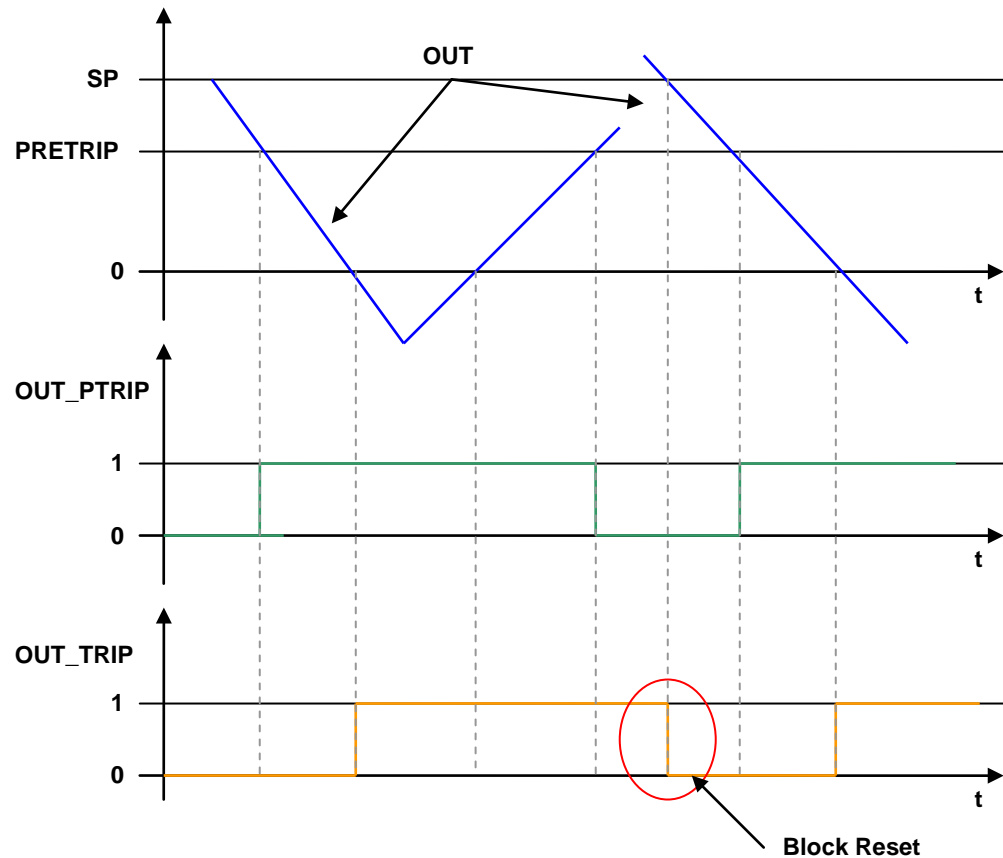
For positive totalization, the options UP\_AUTO, UP\_DEM and PERIODIC must be selected. The parameters OUT\_TRIP and OUT\_PTRIP will be as shown in figure below:



$OUT\_PTRIP = 1$  when  $SP > OUT \geq SP - PRETRIP$   
 $OUT\_PTRIP = 0$  when  $OUT < SP - PRETRIP$  or  $OUT > SP$   
 $OUT\_TRIP = 1$  when  $OUT \geq SP$   
 $OUT\_TRIP = 0$  when block has a Reset ( $OUT = 0$ )

- Negative Totalization

For negative totalization, the options DN\_AUTO and DN\_DEM must be selected. The parameters OUT\_TRIP and OUT\_PTRIP will be as shown in figure below:



OUT\_PTRIP = 1 when  $0 < OUT \leq PRETRIP$

OUT\_PTRIP = 0 when  $OUT > PRETRIP$  or  $OUT \leq 0$

OUT\_TRIP = 1 when  $OUT \leq 0$

OUT\_TRIP = 0 when block has a Reset ( $OUT = SP$ )

### BLOCK\_ERR

The BLOCK\_ERR of the INTG block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when TIME\_UNIT1, TIME\_UNIT2 or INTEG\_TYPE parameters have an invalid value;
- Out of Service – it occurs when the block is in O/S mode.

### Supported Modes

O/S, MAN and AUTO.

### Status

If IN\_1 or IN\_2 is not connected, it will be ignored. The configuration of INTEG\_OPTS (Use Bad / Uncertain) will be applied to the worst status between IN\_1 and IN\_2. And OUT will receive the status determined by GOOD\_LIM and UNCERT\_LIM as explained below:

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 as follows:

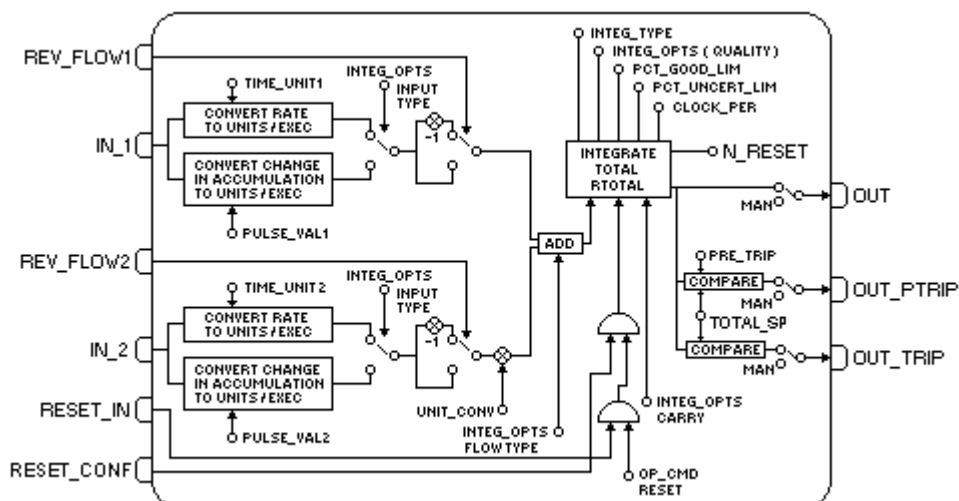
$$PCT\_INCL = 100 * (1 - RTOTAL / ATOTAL)$$

The output status follows the rule:

- The acceptable limit for good status is established by GOOD\_LIM. The acceptable limit for Uncertain status is established by UNCERT\_LIM;

- When the block mode is Automatic, if  $PCT\_INCL \geq UNCERT\_LIM$  the status will be Uncertain, or bad;
- If the block mode is Manual, the output status will be Good (NC), Non Specific, Constant. If the "Uncertain if Man" option in STATUS\_OPTS parameter is set, the output status will be Uncertain, Non Specific.

### Schematic



## Parameters

| Idx | Parameter   | Data Type<br>(length) | Valid Range/<br>Options                     | Default<br>Value | Units  | Store/<br>Mode | Description  |
|-----|-------------|-----------------------|---|------------------|--------|----------------|--|
| 1   | ST_REV      | Unsigned16            |   | 0                | None   | S/RO           |  |
| 2   | TAG_DESC    | OctString(32)         |   | Spaces           | Na     | S              |  |
| 3   | STRATEGY    | Unsigned16            |   | 0                | None   | S              |  |
| 4   | ALERT_KEY   | Unsigned8             | 1 to 255                                    | 0                | None   | S              |  |
| 5   | MODE_BLK    | DS-69                 |   | O/S              | Na     | S              | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)          |   |                  | E      | D / RO         |  |
| 7   | TOTAL_SP    | Float                 |   |                  | OUT    | N / Auto       | The analog set point for trip detection.   |
| 8   | OUT         | DS-65                 |   |                  | OUT    | N / Man        | The output is the result of the corresponding totalization of the RTOTAL register  |
| 9   | OUT_RANGE   | DS-68                 |   | 0                | OUT    | S              | The engineering units of the output for display.   |
| 10  | GRANT_DENY  | DS-70                 |   | 0                | Na     | D              | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 11  | STATUS_OPTS | Bistring(2)           | Bit8: Uncertain if Man                      | 0                | Na     | S/O/S          | See Block Options.   |
| 12  | IN_1        | DS-65                 |   | 0                |        | D              | Input parameter for flow 1.  |
| 13  | IN_2        | DS-65                 |   | 0                |        | D              | Input parameter for flow 2.  |
| 14  | OUT_TRIP    | DS-66                 |   | 0                | On/Off | D              | Output parameter for trip indication.  |
| 15  | OUT_PTRIP   | DS-66                 |   | 0                | On/Off | D              | Output parameter for pre-trip indication.  |
| 16  | TIME_UNIT1  | Unsigned8             | 1:seconds<br>2:minutes<br>3:hours<br>4:days | 0                | E      | S/Man          | Converts the rate time units in seconds.   |

| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|------------|-----------------------|---|------------------|-------|----------------|--|
| 17  | TIME_UNIT2 | Unsigned8             | 1:seconds<br>2:minutes<br>3:hours<br>4:days   | 0                | E     | S/Man          | Converts the rate time units in seconds.   |
| 18  | UNIT_CONV  | Float                 |   | 1                | none  | S/Man          | Factor to convert the engineering units of input 2 into the engineering units of input 1.  |
| 189 | PULSE_VAL1 | Float                 |   | 0                | none  | S/Man          | Determines the mass, volume or energy per pulse. It is used only in accum mode.  |
| 20  | PULSE_VAL2 | Float                 |   | 0                | none  | S/Man          | Determines the mass, volume or energy per pulse. It is used only in accum mode.  |
| 21  | REV_FLOW1  | DS-66                 | 0=FORWARD<br>1=REVERSE  |                  | E     | D              | It indicates reverse flow in IN_1 when true.   |
| 22  | REV_FLOW2  | DS-66                 | 0=FORWARD<br>1=REVERSE  |                  | E     | D              | It indicates reverse flow in IN_2 when true.   |
| 23  | RESET_IN   | DS-66                 | 0:Off<br>1:Reset  |                  | E     | D              | Resets the totalizer.  |
| 24  | STOTAL     | Float                 |   | 0                | OUT   | D              | Indicates the snapshot of the totalizer - OUT parameter just before a reset.   |
| 25  | RTOTAL     | Float                 |   | 0                | OUT   | D              | Indicates the totalization of bad or bad and uncertain inputs, according to INTEG_OPTS.  |
| 26  | SRTOTAL    | Float                 |   | 0                | OUT   | D              | The snapshot of RTOTAL just before a reset.  |
| 27  | SSP        | Float                 |   | 0                | OUT   | D              | The snapshot of SP.  |
| 28  | INTEG_TYPE | Unsigned8             | 1=UP_AUTO<br>2=UP_DEM<br>3=DN_AUTO<br>4=DN_DEM<br>5=PERIODIC<br>6=DEMAND<br>7=PER&DEM | 0                | E     | S              | Defines the type of counting (up or down) and the type of resetting (demand or periodic)   |
| 29  | INTEG_OPTS | Bitstring(2)          |   |                  | none  | 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 others.<br>See Block Options. |
| 30  | CLOCK_PER  | Float                 |   | 0                | Sec   | S              | Establishes the period for periodic reset, in seconds.   |
| 31  | PRE_TRIP   | Float                 |   | 0                | OUT   | S              | Adjusts the amount of mass, volume or energy that will set OUT_PTRIP when the integration reaches (SP-PRE_TRIP) when counting up or PRE_TRIP when counting down.   |
| 32  | N_RESET    | DS-65                 |   |                  | none  | N / RO         | Counts the number of resets. It can not be written nor reset.  |
| 33  | PCT_INCL   | Float                 |   |                  | %     | D / RO         | Indicates the percentage of inputs with good stati compared to the ones with bad or uncertain and bad stati.   |
| 34  | GOOD_LIM   | Float                 | 0 to 100%   | 0.1              | %     | S              | Sets the limit for PCT_INCL. Below this limit OUT receives the status good.  |
| 35  | UNCERT_LIM | Float                 | 0 to 100%   | 0.2              | %     | S              | Sets the limit for PCT_INCL. Below this limit OUT receives the status uncertain.   |
| 36  | OP_CMD_INT | Unsigned8             | 0=Undefined<br>1=Reset  |                  | E     | D              | Operator command. RESET. Resets the totalizer.   |

| Idx | Parameter     | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|---------------|-----------------------|-------------------------|------------------|-------|----------------|--|
| 37  | OUTAGE_LIM    | Float                 | Positive                | 0                | Sec   | S              | The maximum tolerated duration for power failure.<br>This feature is not supported.  |
| 38  | RESET_CONFIRM | DS-66                 | 0:Off]<br>1: Confirm    |                  | E     | D              | Discrete value of the Reset confirmation to enable new Resets. This options is enabled when "Confirm reset" is selected in the INTEG_OPTS parameter.   |
| 39  | UPDATE_EVT    | DS-73                 |                         |                  | na    | D              | This alert is generated by any change to the static data.  |
| 40  | BLOCK_ALM     | DS-72                 |                         |                  | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of TIME\_UNIT1 is seconds.

The default value of TIME\_UNIT2 is seconds.

The default value of INTEG\_TYPE is UP\_DEM.

The required mode for writing is the Actual mode for the SP and OUT parameters, regardless the Target mode.

## AALM - Analog Alarm

### Description

The Analog Alarm Block provides alarm condition for process variable. Alarm conditions include high, high-high, low, and low-low alarms. These limits are computed based on gains and biases from a process setpoint input, thus providing dynamic deviation alarming. An option to temporarily expand alarm limits after a setpoint change is provided. Also, an alarm condition may be ignored for a specified period of time to avoid nuisance alarm reporting.

This block can also be used as a comparator by comparing IN to PSP. And also, it has a discrete output to indicate if there are one or more alarm conditions.

The input value, IN, is filtered according to the PV\_FTIME time-constant, to become PV. PV is repassed by the alarm block.

Alarm limits may be dynamically calculated from process setpoint (PSP) or can be configured based on a fixed limit. The operating limits (same parameter names as limits, but suffixed with an "X") are calculated from specified gains and biases as follows:

$HI\_HI\_LIMX = PSP * HI\_GAIN + HI\_HI\_BIAS + EXPAND\_UP$  (or default to HI\_HI\_LIM if any used parameter is undefined)

$HI\_LIMX = PSP * HI\_GAIN + HI\_BIAS + EXPAND\_UP$  (or default to HI\_LIM if any used parameter is undefined)

$LO\_LIMX = PSP * LO\_GAIN - LO\_BIAS - EXPAND\_DN$  (or default to LO\_LIM if any used parameter is undefined)

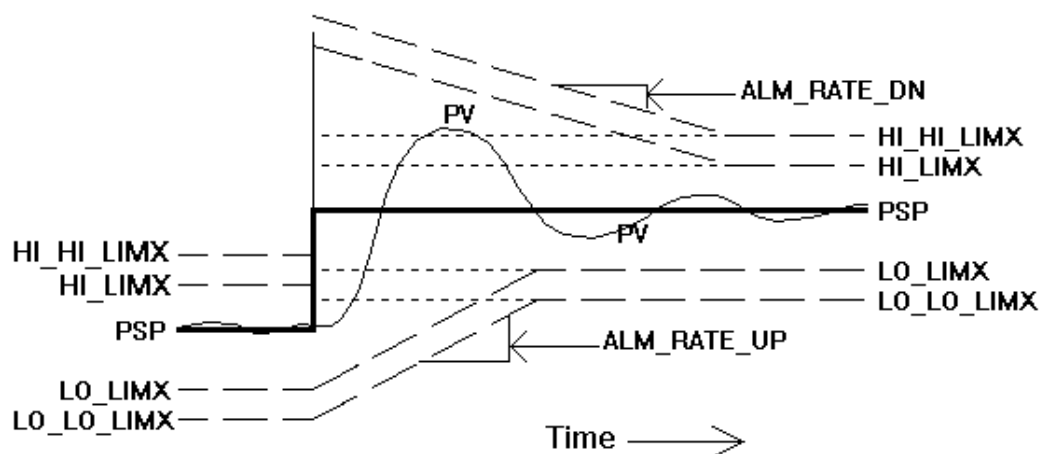
$LO\_LO\_LIMX = PSP * LO\_GAIN - LO\_LO\_BIAS - EXPAND\_DN$  (or default to LO\_LO\_LIM if any used parameter is undefined)

Undefined means:

- HI\_GAIN/HI\_HI\_BIAS =  $\pm$  INF
- PSP\_STATUS = BAD O/S

### Expansion of the Alarm Limits

Effective alarm limits may be temporarily expanded on step setpoint changes to avoid nuisance alarms. The operating high alarm limits are increased by a calculated term, EXPAND\_UP. The operating low alarm limits are decreased by a calculated term, EXPAND\_DN. See the example in the following time chart:



Both the level 1 (advisory) and level 2 (critical) effective alarm limits are expanded after a setpoint change by the absolute value of the change to PSP. The expansions then decay toward the base limits at a rate determined by ALM\_RATE\_UP and ALM\_RATE\_DN parameters. The following rules are applied:

- The four limits initially expand by the same value, the setpoint change;
- The two high limits always expand by the same value, EXPAND\_UP, and decay by the same rate, ALM\_RATE\_DN (which may differ from the low limits);
- The two low limits always expand by the same value, EXPAND\_DN, and decay by the same rate, ALM\_RATE\_UP (which may differ from the high limits);
- The expansion feature may be suppressed in the upward direction by setting ALM\_RATE\_DN to zero or INF. The expansion feature may be suppressed in the downward direction by setting ALM\_RATE\_UP to zero or INF;
- Additional step setpoint changes prior to complete decay of a previous expansion will expand the alarm limits in each direction to the maximum of the remaining expansion value or new expansion value.

### IGNORE\_TIME

The existence of a new alarm condition may be temporarily ignored by setting the IGNORE\_TIME parameter to the number of seconds to disregard the alarm. Both the reporting of the alarm and the possible change to PRE\_OUT\_ALM will be ignored during this time. This parameter does not delay the clearing of the existence of the alarm on return-to-normal. If the alarm condition does not persist for IGNORE\_TIME seconds, it will not be reported.

### OUT\_ALM indication

OUT\_ALM parameter will be the PRE\_OUT\_ALM value when the block is in Auto mode.

PRE\_OUT\_ALM and OUT\_ALM indicate the existence of one or more selected alarm conditions per the specification of the OUT\_ALM\_SUM parameter. Enumerated choices of the OUT\_ALM\_SUM parameter and their included alarm conditions are listed below:

| OUT_ALM_SUM | INCLUDED ALARM CONDITIONS |        |        |           |
|-------------|---------------------------|--------|--------|-----------|
|             | HI_HI_ALM                 | HI_ALM | LO_ALM | LO_LO_ALM |
| ANY         | ✓                         | ✓      | ✓      | ✓         |
| LOWs        |                           |        | ✓      | ✓         |
| HIGHs       | ✓                         | ✓      |        |           |
| LEVEL1      |                           | ✓      | ✓      |           |
| LEVEL2      | ✓                         |        |        | ✓         |
| LO_LO       |                           |        |        | ✓         |
| LO          |                           |        | ✓      |           |
| HI          |                           | ✓      |        |           |
| HI_HI       | ✓                         |        |        |           |
| NONE        |                           |        |        |           |

For example, if LOWs is chosen for OUT\_ALM\_SUM, either a LO\_ALM or LO\_LO\_ALM being *true* will cause OUT\_ALM to be set to *true*. If LEVEL1 is chosen for OUT\_ALM\_SUM, either a LO\_ALM or HI\_ALM being *true* will cause OUT\_ALM to be set to *true*.

In order to the alarm occurs, the alarm conditions must be kept for more than 1 cycle of the block execution. It means that if in one execution cycle occurs one alarm and in the next cycle it stops, the alarm will not be notified, so it is possible to avoid nuisances.

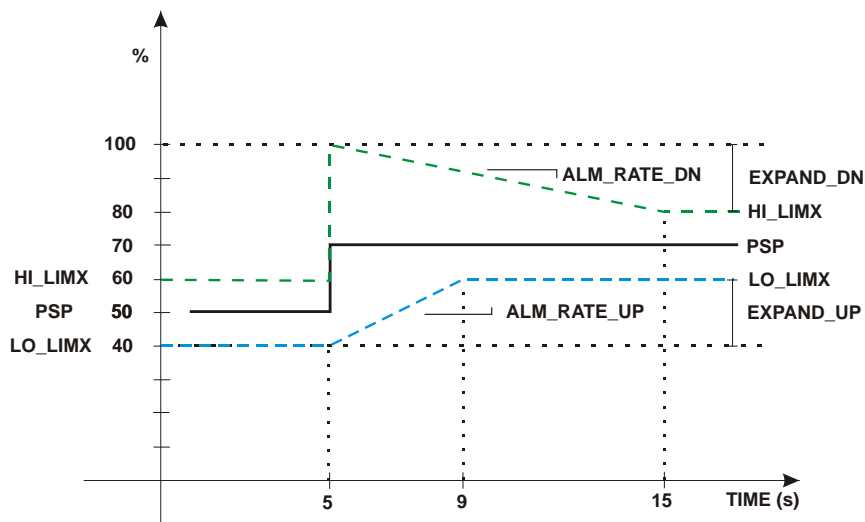
The OUT\_ALM parameter can be used for control purposes, for example, as an interlock signal, besides the basic function of alarm monitoring.

### Example of Dynamic Alarm

There is an alarm example of LEVEL 1 type. It considers the values below:

- OUT\_ALM\_SUM = Level 1
- HI\_GAIN = 1
- HI\_BIAS = 10
- LO\_GAIN = 1
- LO\_BIAS = 10
- ALM\_RATE\_DN = 2/seg
- ALM\_RATE\_UP = 5/seg

Initially, the PSP input is 50%, so the high alarm HI\_LIMX is 60% and the low alarm LO\_LIMX is 40%. In any time, the PSP input receives a step equals to 20, going to 70%. In this moment, the high alarm increases up to 100% and decreases in the ALM\_RATE\_DN rate. The low alarm decreases down to 40% and increases in the ALM\_RATE\_UP rate.



#### Simple alarm calculation: static alarm limits, no expansion and no delay to detection

The alarm limits will be static (HI\_HI\_LIM, HI\_LIM, LO\_LIM and LO\_LO\_LIM are the effective operating alarm limits) if the corresponding gain or bias is  $\pm \text{INF}$ , or the input PSP is left unconnected with status Bad – O/S.

The alarm limit expansion will be disabled by setting ALM\_RATE\_DN and ALM\_RATE\_UP to zero.

The detection of an alarm will be without delay setting IGNORE\_TIME to zero.

#### Supported Modes

O/S, MAN and AUTO.

#### Status Handling

The block will not filter an IN value with a bad status or uncertain status (and "Use Uncertain" option in STATUS\_OPTS is not set), but will instead, filter to the last usable value of PV and assign it the unusable status from IN. When the status of IN returns to a usable value (good or uncertain [and "Use Uncertain" option in STATUS\_OPTS is set]), the value of PV will again be filtered toward the value of IN with the status of IN.

The status of OUT is set to the status of PV (and IN) when in auto mode.

If the worst quality of the stati of PV and PSP is bad, or uncertain (and "Use Uncertain" option in STATUS\_OPTS is not set) the alarm test will not be performed and the status of PRE\_OUT\_ALM will be set to bad (non-specific). Otherwise, the alarm test will be performed and the quality of the status of PRE\_OUT\_ALM will be set to the worst quality of the stati of PV and PSP (good or uncertain). While the alarm condition is not being evaluated due to unusable stati, existing alarms will not be cleared and new alarms will not be generated. Prior alarm conditions may still be acknowledged.

In Auto mode, the status of OUT\_ALM will be set to the status of PRE\_OUT\_ALM. In Man mode, the limits status of OUT\_ALM is set to double-limited.

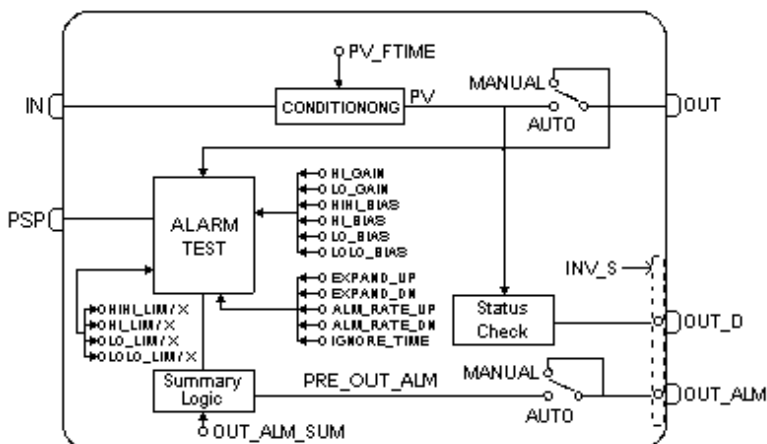
#### BLOCK\_ERR

The BLOCK\_ERR of the Analog Alarm block will reflect the following causes:

- Out of Service – When the block is in O/S mode.



## Schematic



## Parameters

| Idx | Parameter      | Data Type (length) | Valid Range/Options | Default Value | Units | Store / Mode | Description   |
|-----|----------------|--------------------|---------------------|---------------|-------|--------------|---|
| 1   | ST_REV         | Unsigned16         |                     | 0             | None  | S/RO         |   |
| 2   | TAG_DESC       | OctString(32)      |                     | Spaces        | Na    | S            |   |
| 3   | STRATEGY       | Unsigned16         |                     | 0             | None  | S            |   |
| 4   | ALERT_KEY      | Unsigned8          | 1 to 255            | 0             | None  | S            |   |
| 5   | MODE_BLK       | DS-69              |                     | O/S           | Na    | S            | See Mode Parameter  |
| 6   | BLOCK_ERR      | Bitstring(2)       |                     |               | E     | D / RO       |   |
| 7   | PV             | DS-65              |                     |               | PV    | D / RO       | Process analog value. This is the IN value after pass over the PV filter.   |
| 8   | OUT            | DS-65              | OUT_SCALE +/- 10%   |               | OUT   | N / Man      | The output value result of the block calculation.   |
| 9   | OUT_RANGE      | DS-68              |                     | 0-100%        | OUT   | S / Man      | The high and low scale values to the OUT parameter.   |
| 10  | GRANT_DENY     | DS-70              |                     | 0             | na    | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.                                  |
| 11  | STATUS_OPTIONS | Bitstring(2)       | See Block Options   | 0             | Na    | S / O/S      | See Block Options   |
| 12  | PV_FTIME       | Float              | Non-Negative        | 0             | Sec   | S            | Time constant of a single exponential filter for the PV, in seconds.  |
| 13  | IN             | DS-65              |                     |               | PV    | D            | The primary input value of the block, or PV value.  |
| 14  | PSP            | DS-65              |                     |               | PV    | D            | This is the process setpoint, which may be used for alarm limit determination.  |
| 15  | HI_GAIN        | Float              |                     | 1.1           |       | S            | This gain multiplies PSP before addition of biases for HI_LIM and HI_HI_LIM.  |
| 16  | LO_GAIN        | Float              |                     | 0.9           | Na    | S            | This gain multiplies PSP before subtraction of biases for LO_LIM and LO_LO_LIM.   |
| 17  | HI_HI_BIAS     | Float              | Positive            | 1.0           | Out   | S            | This bias is added to PSP*HI_GAIN to determine HI_HI_LIM.   |
| 18  | HI_BIAS        | Float              | Positive            | 0.0           | Out   | S            | This bias is added to PSP*HI_GAIN to determine HI_LIM.  |
| 19  | LO_BIAS        | Float              | Positive            | 0.0           | Out   | S            | This bias is subtracted from PSP*LO_GAIN to determine LO_LIM.   |
| 20  | LO_LO_BIAS     | Float              | Positive            | 1.0           | Out   | S            | This bias is subtracted from PSP*LO_GAIN to determine LO_LO_LIM.  |
| 21  | PRE_OUT_ALM    | DS-66              |                     |               | E     | D/RO         | This parameter is the alarm summary variable of the analog alarm block. If the block is in Man mode, it will have the same value and status of OUT_ALM parameter. |

| Idx | Parameter   | Data Type (length) | Valid Range/ Options   | Default Value | Units   | Store / Mode | Description  |
|-----|-------------|--------------------|--|---------------|---------|--------------|--|
| 22  | OUT_ALM     | DS-66              |  |               | E       | D            | This parameter is the alarm summary variable of the analog alarm block when in <i>Auto</i> mode and is the value specified by the operator/engineer in <i>Man</i> mode. It is a discrete parameter that indicates if the block is in alarm condition (value 1) or if not (Value 0).  |
| 23  | OUT_ALM_SUM | Unsigned8          | 0:NONE<br>1:LO_LO<br>2:LO<br>3:LOWs<br>4:HI<br>6:LEVEL1<br>8:HI_HI<br>9:LEVEL2<br>12:HIGHS<br>15:ANY | 0             | E       | S/Man        | Specifies the alarms conditions which must be <i>true</i> in order for OUT_ALM to be set to <i>true</i> : ANY, LOWs, HIGHs, LEVEL1, LEVEL2, LO_LO, LO, HI, or HI_HI.   |
| 24  | ALM_RATE_UP | Float              | Positive   | 0.0           | OUT/sec | S            | Ramp rate at which downward alarm expansion due to step PSP changes is decayed in the upward direction. Expressed in engineering units per second. Expansion feature is disabled when ALM_RATE_UP = 0 or INF.  |
| 25  | ALM_RATE_DN | Float              | Positive   | 0.0           | OUT/sec | S            | Ramp rate at which upward alarm expansion due to step PSP changes is decayed in the downward direction. Expressed in engineering units per second. Expansion feature is disabled when ALM_RATE_DN = 0 or INF.  |
| 26  | EXPAND_UP   | Float              |  |               | OUT     | D/RO         | Amount, in engineering units, that base HI and HI_HI limits are expanded after a setpoint change. Dynamically calculated by block. Initially expanded by the amount of a setpoint change and decayed at the rate of ALM_RATE_UP. (Positive)  |
| 27  | EXPAND_DN   | Float              |  |               | OUT     | D/RO         | Amount, in engineering units, that base LO and LO_LO limits are expanded after a setpoint change. Dynamically calculated by block. Initially expanded by the amount of a setpoint change and decayed at the rate of ALM_RATE_DN. (Positive)  |
| 28  | IGNORE_TIME | Float              | Positive   | 0.0           | Sec     | S            | The time, in seconds, to ignore the existence of a new alarm condition. There is no delay on clearing the existence of the alarm on return to normal. If the alarm does not persist for IGNORE_TIME seconds, it will not be reported. Does not apply to self-clearing (transient) type alarms.   |
| 29  | UPDATE_EVT  | DS-73              |  |               | Na      | D            | This alert is generated by any change to the static data.  |
| 30  | BLOCK_ALM   | DS-72              |  |               | Na      | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 31  | ALARM_SUM   | DS-74              | See Block Options  |               | Na      | S            | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 32  | ACK_OPTION  | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable  | 0             | Na      | S            | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 33  | ALARM_HYS   | Float              | 0 to 50 %  | 0.5%          | %       | S            | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis.  |
| 34  | HI_HI_PRI   | Unsigned8          | 0 to 15  | 0             |         | S            | Priority of the high high alarm.   |
| 35  | HI_HI_LIM   | Float              | OUT_SCALE, +INF  | +INF          | OUT     | S            | The setting for high high alarm in engineering units.  |

| Idx | Parameter  | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description   |
|-----|------------|--------------------|----------------------|---------------|-------|--------------|---|
| 36  | HI_HI_LIMX | Float              | OUT_SCALE, +INF      | +INF          | OUT   | D/RO         | The setting for high high alarm in engineering units.       |
| 37  | HI_PRI     | Unsigned8          | 0 to 15              | 0             |       | S            | Priority of the high alarm.                                 |
| 38  | HI_LIM     | Float              | OUT_SCALE, +INF      | +INF          | OUT   | S            | The setting for high alarm in engineering units.            |
| 39  | HI_LIMX    | Float              | OUT_SCALE, +INF      | +INF          | OUT   | D/RO         | The setting for high alarm in engineering units.            |
| 40  | LO_PRI     | Unsigned8          | 0 to 15              | 0             |       | S            | Priority of the low alarm.                                  |
| 41  | LO_LIM     | Float              | OUT_SCALE, -INF      | -INF          | OUT   | S            | The setting for low alarm in engineering units.             |
| 42  | LO_LIMX    | Float              | OUT_SCALE, -INF      | -INF          | OUT   | D/RO         | The setting for low alarm in engineering units.             |
| 43  | LO_LO_PRI  | Unsigned8          | 0 to 15              | 0             |       | S            | Priority of the low alarm.                                  |
| 44  | LO_LO_LIM  | Float              | OUT_SCALE, -INF      | -INF          | OUT   | S            | The setting for low alarm in engineering units.             |
| 45  | LO_LO_LIMX | Float              | OUT_SCALE, -INF      | -INF          | OUT   | D/RO         | The setting for low alarm in engineering units.             |
| 46  | HI_HI_ALM  | DS-71              |                      |               | OUT   | D            | The status for high alarm and its associated time stamp.    |
| 47  | HI_ALM     | DS-71              |                      |               | OUT   | D            | The status for high alarm and its associated time stamp.    |
| 48  | LO_ALM     | DS-71              |                      |               | OUT   | D            | The status for low alarm and its associated time stamp.     |
| 49  | LO_LO_ALM  | DS-71              |                      |               | OUT   | D            | The status for low low alarm and its associated time stamp. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The required mode for writing is the Actual mode for the OUT parameter, regardless the Target mode.

## EAALM – Enhanced Analog Alarm

### Description

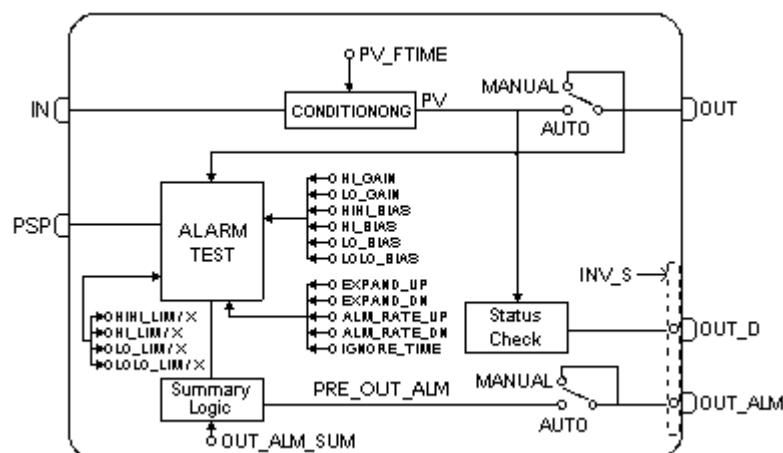
The EAALM block has all parameters of the AALM block. Additionally, it provides an output to indicate when the input has no usable status, and also it has the option to invert the outputs OUT\_D and OUT\_ALM.

### Additional Characteristics

The EAALM block has also the following characteristics:

- The OUT\_D indicating that IN is not usable. When IN.Status is Bad or Uncertain and the STATUS\_OPTS does not have the “Use Uncertain as Good” option selected, the OUT\_D value will be 1. Otherwise, it will be 0;
- Optionally, OUT\_D and OUT\_ALM can be inverted when the corresponding bits in the INVERT\_OPTS parameter are set.

### Schematic



### Parameters

It has all parameters of the AALM block plus these additional parameters:

| Idx | Parameter   | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description  |
|-----|-------------|--------------------|----------------------|---------------|-------|--------------|--|
| 50  | OUT_D       | DS_66              |                      |               |       | D/RO         | True: Indicate if the IN input is unusable with bad or uncertain status. |
| 51  | INVERT_OPTS | Bitstring(2)       |                      |               |       | S/O/S        | Parameter for inversion of the block discrete outputs.                   |

## ISEL - Input Selector

### Description

The signal selector block provides selection of up to four inputs and generates an output based on the configured action. This block would normally receive its input from an AI. In addition to signal selection the block can also perform max, min, mid, avg and first good selection. The SELECTED output shows the input that is being used by the algorithm.

The purpose of this block is providing the selection of the control signal only for the direct path. It is not developed to support input signals from a controller, so it does not support back calculations.

### Input Processing

If DISABLE\_n is true, the corresponding input is not used.

The inputs with bad or uncertain status are ignored (if the "Use uncertain as good" option of the STATUS\_OPTS parameter is selected).

MIN\_GOOD specifies the minimum number of good inputs for the selection process. If there are no good inputs, or the number of good inputs is lower than the specified in MIN\_GOOD, so OUT status will be bad and the SELECTED value will be zero.

### Selection Process

OP\_SELECT input selects an input for the output. This option has priority over the SELECT\_TYPE calculation. If the OP\_SELECT is higher than zero, OUT will follow the selected input in OP\_SELECT, ignoring the SELECT\_TYPE selection.

If SELECT\_TYPE = Maximum, OUT receives the higher value of all good inputs connected. SELECTED will show the selected input.

If SELECT\_TYPE = Minimum, OUT receives the lower value of all good inputs connected. SELECTED will show the selected input.

If SELECT\_TYPE = Middle, OUT receives the middle value of all good inputs connected. If there are 3 or 4 values, the upper and lower values will be discarded. If there are at least 2 values, the middle is done with them. SELECTED will be zero when the middle is done between the inputs, or it will show the selected input.

If SELECT\_TYPE = First Good, OUT will be the first good input encountered based on ascending evaluation of the inputs, from IN\_1 to IN\_4. SELECTED will show the selected input.

If SELECT\_TYPE = Average, OUT will be the average of all good inputs connected. SELECTED will show the number of the inputs used to calculate the average.

### BLOCK\_ERR

The BLOCK\_ERR of the ISEL block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the SELECT\_TYPE parameter has an invalid value;
- Out of Service – it occurs when the block is in O/S mode.

### Supported Modes

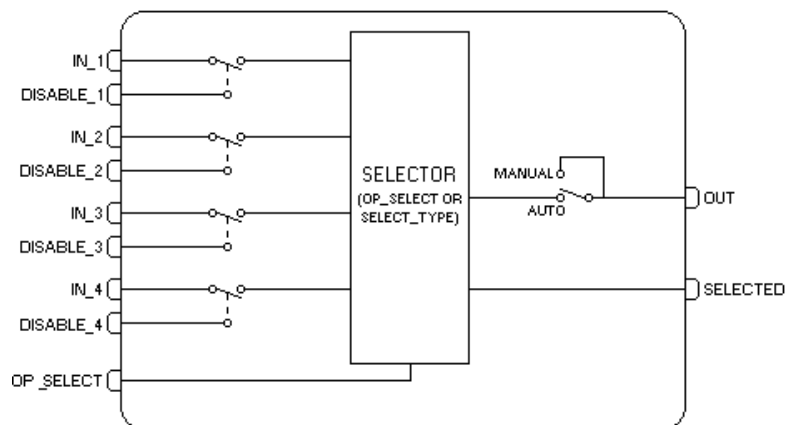
O/S, MAN and AUTO.

### Status Handling

If there are no usable inputs or the number of inputs is less than MIN\_GOOD, so the output status will be Bad Non Specific.

The status of OUT will be a copy of the selected input, but if the output is an average of inputs the status will be Good Non-cascade – Non-specific.

## Schematic



## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description   |
|-----|-------------|--------------------|----------------------|---------------|-------|--------------|---|
| 1   | ST_REV      | Unsigned16         |                      | 0             | None  | S/RO         |   |
| 2   | TAG_DESC    | OctString(32)      |                      | Spaces        | Na    | S            |   |
| 3   | STRATEGY    | Unsigned16         |                      | 0             | None  | S            |   |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255             | 0             | None  | S            |   |
| 5   | MODE_BLK    | DS-69              |                      | O/S           | Na    | S            | See Mode Parameter  |
| 6   | BLOCK_ERR   | Bitstring(2)       |                      |               | E     | D/RO         |   |
| 7   | OUT         | DS-65              | XD_SCALE             |               | OUT   | D / Man      | The primary analog value calculated as a result of executing the function.  |
| 8   | OUT_UNITS   | Unsigned16         |                      | 0             | E     | S            | The engineering units of the output for display.  |
| 9   | GRANT_DENY  | DS-70              |                      | 0             | Na    | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.    |
| 10  | STATUS_OPTS | Bitstring(2)       | See Block Options    | 0             | Na    | S / O/S      | See Block Options   |
| 11  | IN_1        | DS-65              |                      |               |       | D            | Numbered input parameter 1.   |
| 12  | IN_2        | DS-65              |                      |               |       | D            | Numbered input parameter 2.   |
| 13  | IN_3        | DS-65              |                      |               |       | D            | Numbered input parameter 3.   |
| 14  | IN_4        | DS-65              |                      |               |       | D            | Numbered input parameter 4.   |
| 15  | DISABLE_1   | DS-66              |                      |               |       | D            | Parameter to switch off the input from being used. If this parameter is true then don't use this input IN_1 determining the output. |
| 16  | DISABLE_2   | DS-66              |                      |               |       | D            | Parameter to switch off the input from being used. If this parameter is true then don't use this input IN_2 determining the output. |
| 17  | DISABLE_3   | DS-66              |                      |               |       | D            | Parameter to switch off the input from being used. If this parameter is true then don't use this input IN_3 determining the output. |
| 18  | DISABLE_4   | DS-66              |                      |               |       | D            | Parameter to switch off the input from being used. If this parameter is true then don't use this input IN_4 determining the output. |

| Idx | Parameter   | Data Type<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store /<br>Mode | Description   |
|-----|-------------|-----------------------|--|------------------|-------|-----------------|---|
| 19  | SELECT_TYPE | Unsigned8             | 1=First good<br>2=Minimum<br>3=Maximum<br>4=Middle<br>5=Average                              | 0                | E     | S               | <p>selector action</p> <p>max = select the max from all the connected and good inputs</p> <p>min = select the min from all the connected and good inputs</p> <p>mid = select the mid value from all the connected and good inputs, if less than MIN_GOOD inputs are connected then an error code is generated. An error code is also generated if less than MIN_GOOD values have good status. Not intended for use with 2 or 4 inputs.</p> <p>First Good = determine the first good input encountered based on ascending evaluation of the inputs, see text for further discussion</p> <p>avg = compute the average for all the connected and good inputs, if less than two inputs are connected then set the output equal to the input and generate an error code. An error code is also generated if less than MIN_GOOD inputs have good status</p> |
| 20  | MIN_GOOD    | Unsigned8             | 0 through 4  | 0                |       | S               | 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    | DS-66                 | None,<br>1= 1<br>2= 2<br>3= 3<br>4= 4  |                  |       | D / RO          | Number to identify which input is selected.   |
| 22  | OP_SELECT   | DS-66                 | 0 = Normal Operation<br>1= Selects IN1<br>2= Selects IN2<br>3= Selects IN3<br>4= Selects IN4 |                  | None  | D               | An operator adjustable parameter to force a given input to be used. Selecting 0 will indicate normal operation while choosing 1 to 4 will indicate the input to be used.  |
| 23  | UPDATE_EVT  | DS-73                 |  |                  | na    | D               | This alert is generated by any change to the static data.   |
| 24  | BLOCK_ALM   | DS-72                 |  |                  | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.  |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static;

**Gray Background Line: Default Parameters in Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

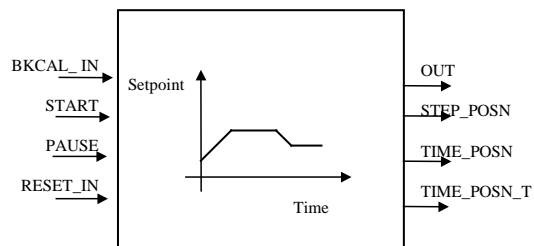
The default value of SELECT\_TYPE is First good.

The required mode for writing is the Actual mode for the OUT parameter, regardless the Target mode

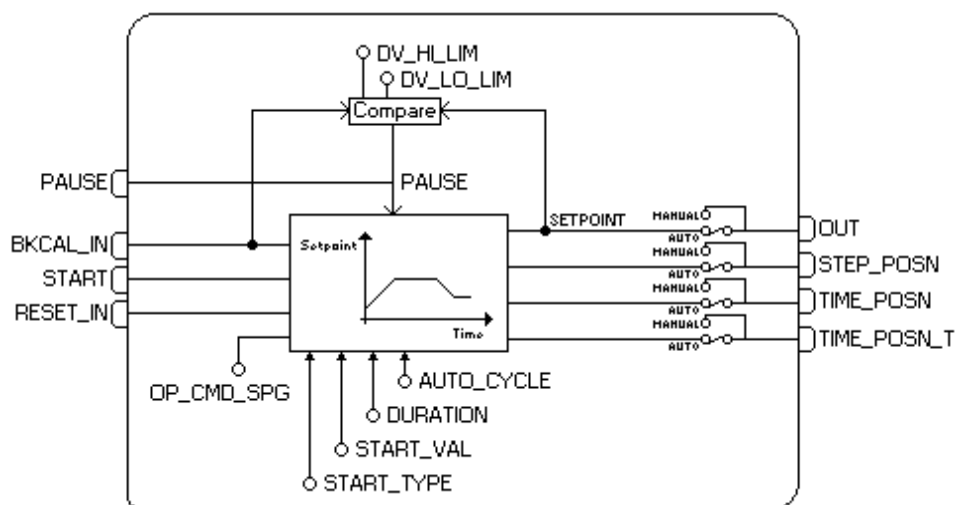
## SPG - Setpoint Ramp Generator

### Description

The Setpoint generator block is normally used to generate a Setpoint to a PID block in applications like temperature control, batch reactors, etc. In those applications, the Setpoint shall follow a certain profile in function of the time.



### Schematic



1) The profile is determined by up to ten segments or steps. Each segment is defined by a starting value [START\_VAL] and a time duration [DURATION]. The starting value of the next segment determines if the previous segment ramps up, down or remains constant. The curve is given by t

START\_VAL (Starting value) - Eleven floating point values defining the initial value of each segment, in engineering units.

TIME\_DURATION (Time duration) - Ten floating point values defining the duration, in seconds, of each segment. A zero value defines the last segment.

2) The two arrays define the Setpoint value (y-axis) in function of the time (t-axis). Between two given points, the Setpoint is calculated by interpolation. As each segment is defined by [START\_VAL]<sub>i</sub>, [TIME\_DURATION]<sub>i</sub> and [START\_VAL]<sub>i+1</sub>, a profile with "n" segments will need **n+1** starting values and **n** time durations. As example, the two following arrays define the profile shown on Fig. 1:

|               | 1  | 2  | 3   | 4   | 5   | 6  |
|---------------|----|----|-----|-----|-----|----|
| START_VAL     | 25 | 50 | 50  | 100 | 100 | 25 |
| TIME_DURATION | 60 | 60 | 120 | 60  | 60  | 0  |



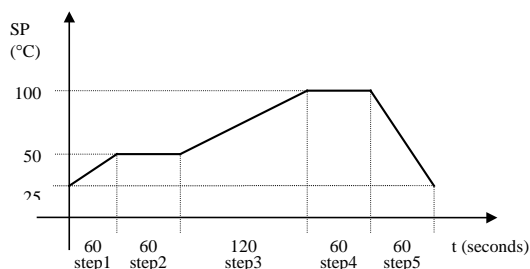


Fig.1 - Setpoint profile

3) The position in the time axis (t-axis) is controlled by an internal timer. This timer is started by a transition from false to true at START input, or by the OP\_CMD\_SPG="Start" parameter. Once started, it runs while the mode is Auto up to reach the sum of the durations calculated by TIME\_DURATION parameter.

4) The timer resets (that is, it is positioned in the starting point of the curve), if the OP\_CMD\_SPG="Reset" or the RESET\_IN input is 1. After resetting, a new Start is waited in order to reinitialize the timer again. When the RESET\_IN input is used and while its value is 1, the block is kept in Reset. So, the timer will be available to start only after the RESET\_IN parameter changes to 0.

5) The timer may be interrupted at any moment changing the Auto mode to any mode. When the mode returns to Auto mode, the timer will reinitialize from the point that is stopped. And also, it will be interrupted if START\_TYPE is "Cascade" (See item 17) and the cascade initialization is in progress, where the Actual mode is IMan.

6) The timer may be interrupted if PAUSE changes from false to true. It will restart when PAUSE is set to false and none condition interrupts the timer.

7) The timer is also interrupted by a PAUSE caused by the deviation between BCKAL\_IN and the generated Setpoint ( $\text{Deviation} = \text{BCKAL\_IN} - \text{PRE\_OUT}$ ). If the deviation exceeds DV\_HI\_LIM or DV\_LO\_LIM, an alarm is indicated in DV\_HI\_ALM or DV\_LO\_ALM, respectively. Both alarms stop the timer and return to normal operation when the deviation is within the prescribed limits.

8) The Setpoint is in the "y" axis, while the time is in the "t" axis. The Setpoint value is available at output OUT. It is also available in PRE\_OUT even when the block is in Man mode.

9) Three outputs inform the current point of the profile:

STEP\_POSN - Informs the current segment or step.

TIME\_POSN - Informs the time elapsed since the beginning of the current step.

TIME\_POSN\_T - Informs the time elapsed since the beginning of the profile.

10) With the block in Manual, the operator can write on the outputs STEP\_POSN, TIME\_POSN and TIME\_POSN\_T in order to select a particular point of the profile. The outputs are not independent. If TIME\_POSN\_T changes, it causes changes in TIME\_POSN and STEP\_POSN. If STEP\_POSN changes, it causes changes in TIME\_POSN\_T and STEP\_POSN is reset to zero. If TIME\_POSN changes, it causes changes in TIME\_POSN\_T but it can not be moved to out of the segment. When the block is switched back to auto, the profile will start from that point.

11) With the block in manual, the operator can also modify OUT. The value can be different of PRE\_OUT and there are no bumps when the mode is changed to Auto. In this situation, OUT starts from the actual value and reaches the PRE\_OUT\_D value according the value configured in BAL\_TIME.

12) Another operation that can be done with the block in Manual, is to advance or return the time through the following operator commands (OP\_CMD\_SPG):

ADVANCE – it puts the timer in the beginning of the next step.

REPEAT – it puts the timer in the beginning of the current step.

13) The outputs can only be modified with the block is in Manual mode.

14) If the AUTO\_CYCLE parameter is “Auto Cycle”, the timer will return to zero (Reset) and will restart (START) the curve automatically.

15) The operation status is given by the parameter SPG\_STATE

READY - When the block is at the beginning of the curve, waiting for the starting signal (START). The block is in this state after a Reset.

ACTIVE - When the timer is “on”. The block is in this state after a Start or when it exits from Pause.

PAUSE - When the timer is stopped (see the reason in the PAUSE\_CAUSE parameter). The timer will be active just after the PAUSE condition stops.

AT\_END - When the timer reaches the last point of the curve. To start the curve is necessary a Reset and a new Start.

16) The parameter PAUSE\_CAUSE enumerates the cause of the PAUSE state:

| PAUSE_CAUSE | Cause   |
|-------------|---|
| Mode        | The timer is stopped due to the Actual mode is not Auto (see item 5)    |
| Logic       | The timer is stopped due to the PAUSE input is active. (see item 6)     |
| Alarm       | The timer is stopped due to the deviation alarm is active. (see item 7) |

If there are more than one Pause condition, it will be indicates in the PAUSE\_CAUSE parameter. After stopping the condition that caused the pause in the timer, the timer starts from the point that was before the pause condition.

17) It is possible to start the curve from the BKCAL\_IN value when it is connected. When the timer starts, its initial value depends on the selection configured in the START\_TYPE parameter:

1 – Cascade - The curve starts as specified by START\_VAL and TIME\_DURATION. For this situation, BKCAL\_IN is connected to BKCAL\_OUT from the control block. The cascade initialization is used. When the BKCAL\_IN value is out of the curve, a bumpless starting may be used through the BAL\_TIME parameter.

2 – Use Duration – The timer starts from zero. START\_VAL for the first segment is equals to BKCAL\_IN, temporarily, when it is connected. If the BKCAL\_IN is not connected, the initial value will be START\_VAL [1].

3 – Use Rate – The curve starts from the BKCAL\_IN value (when it is connected) and uses the rate specified by the first values of START\_VAL and the first value of TIME\_DURATION. If the BKCAL\_IN is not connected, the initial value will be START\_VAL [1].

#### Procedure to Activate the Algorithm of the SPG Block

- 1) Reset the block. The reset of the block can occur in two ways: through a pulse in the input RESET\_IN (value change from 0 to 1 and after to 0) or OP\_CMD\_SPG=REST\_IN. In this case the algorithm is stopped and it is ready to start the curve, then the parameter SPG\_STATE=READY.
- 2) Start the algorithm. The initialization of the algorithm occurs through a pulse in the input START. At this moment the algorithm executes the curve, then SPG\_STATE=ACTIVE.
- 3) The algorithm can be interrupted momentarily at any moment of execution of the curve (Pause) through the input PAUSE=1. In this case the SPG\_STATE=PAUSE and the parameter PAUSE\_CAUSE indicate the stop condition.
- 4) When the end of the algorithm is detected, the SPG\_STATE=AT\_END.
- 5) To restart the algorithm, it is necessary to return to the step 1.

#### BLOCK\_ERR

The BLOCK\_ERR of the SPG block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the START\_TYPE parameter has an invalid value;
- Out of Service – it occurs when the block is in O/S mode.

#### Supported Modes

O/S, IMAN, MAN and AUTO.

#### Status Handling

If the status of any used input becomes bad or uncertain and the option “Use uncertain as Good” of the STATUS\_OPTS parameter is not set, the, the block Actual mode will be forced to Manual. The option “Target to Man if Bad IN” of the STATUS\_OPTS parameter can also be used.

The OUT status will be “Good Cascade” when START\_TYPE is “Cascade”, or it will be “Good Non Cascade” for the other situations.

## Parameters

| Idx | Parameter     | Data Type (length) | Valid Range/Options   | Default Value | Units  | Store / Mode | Description  |
|-----|---------------|--------------------|---|---------------|--------|--------------|--|
| 1   | ST_REV        | Unsigned16         |   | 0             | None   | S/RO         |  |
| 2   | TAG_DESC      | OctString(32)      |   | Spaces        | Na     | S            |  |
| 3   | STRATEGY      | Unsigned16         |   | 0             | None   | S            |  |
| 4   | ALERT_KEY     | Unsigned8          | 1 to 255  | 0             | None   | S            |  |
| 5   | MODE_BLK      | DS-69              |   | O/S           | Na     | S            | See Mode Parameter   |
| 6   | BLOCK_ERR     | Bitstring(2)       |   |               | E      | D / RO       |  |
| 7   | OUT           | DS-65              | OUT_SCALE +/- 10%   |               | OUT    | N / Man      | The analog value calculated as a result of executing the function.   |
| 8   | OUT_RANGE     | DS-68              |   | 0-100%        | OUT    | S / Man      | The high and low scale values to the OUT parameter only for display.   |
| 9   | GRANT_DENY    | DS-70              |   | 0             | na     | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.   |
| 10  | STATUS_OPTS   | Bitstring(2)       | Bit 2: Use Unc as Good<br>Bit 5: Target to Man if bad IN  | 0             | Na     | S / O/S      | See Block Options  |
| 11  | START_VAL     | 11 Floats          |   |               |        |              | An array of up to eleven points defines the starting point of each segment of the Setpoint profile, in engineering units.  |
| 12  | TIME_DURATION | 10 Floats          |   | 0's           | Sec    | S            | An array of up to ten points defines the duration of each segment of the Setpoint profile, in seconds.   |
| 13  | TIME_UNITS    | Unsigned8          | 1=seconds<br>2=minutes<br>3=hours<br>4=days<br>5=[day-[hr:[min[:sec]]]]   | 0             |        | E            | Display Time Units for TIME_POSN and TIME_POSN_T only for display.   |
| 14  | BKCAL_IN      | DS-65              |   |               | OUT    | N            | The value and status from a lower block's BKCAL_OUT that is used to initialize the control loop.   |
| 15  | START         | DS-66              |   |               | On/Off | D            | The timer is started when a transition from false to true occurs, and the timer is in the beginning of the curve (SPG_STATE=Ready)   |
| 16  | START_TYPE    | Unsigned8          | 1=Cascade<br>2=Use Duration<br>3=Use Rate   | 0             | E      | S            | This parameter selects the initialization type:<br>Cascade – it starts from the curve values.<br>Use Duration – It starts from the BKCAL_IN value with the duration of the first segment.<br>Use Rate – it starts from the BKCAL_IN value and the rate between the 2 first START_VALs and the duration of the first segment. |
| 17  | PAUSE         | DS-66              |   |               |        | D            | Stops the timer when set to <i>true</i> . It restarts when the time is set back to <i>false</i> .  |
| 18  | PAUSE_CAUSE   | Unsigned8          | 0=Not paused<br>1=Mode Pause<br>2=Logic Pause<br>4=Alarm Pause<br>3=Mode & Logic<br>5=Mode & Alarm<br>6=Logic & Alarm<br>7=Mode & Logic & Alarm |               |        | E            | This parameter enumerates the PAUSE causes (Two conditions can occur simultaneously):<br><br>Mode – The timer stopped due to changes in the Auto mode.<br><br>Logic – The timer stopped due to the PAUSE parameter is active.<br><br>Alarm – The timer stopped due to the deviation alarm.                                   |
| 19  | AUTO_CYCLE    | Unsigned8          | 1:AutoCycle   | 0             | E      | S            | When set the option "AutoCycle", the block automatically resets the time to the beginning of the first step and restarts the timer.  |
| 20  | STEP_POSN     | DS-66              | 0=none<br>1=step1<br>2=step 2<br>n=step n   | 0             | E      | D / Man      | Determine the current step or segment of the profile in auto mode. Direct the timer to the step specified by the operator when in manual mode.   |

| Idx | Parameter   | Data Type (length) | Valid Range/Options  | Default Value | Units | Store / Mode | Description  |
|-----|-------------|--------------------|--|---------------|-------|--------------|--|
| 21  | TIME_POSN   | DS-65              |  |               | Sec   | D / Man      | Determine the time elapsed since the beginning of the step in auto mode. The operator can set the time since the beginning of the current step when operating in manual.   |
| 22  | TIME_POSN_T | DS-65              |  |               | Sec   | N / Man      | Determine the time elapsed since the beginning of the curve in auto mode. The operator can set the time since the beginning of the curve when operating in manual.   |
| 23  | OP_CMD_SPG  | Unsigned8          | 0=No Action<br>1=RESET<br>2=START<br>3=ADVANCE<br>4=REPEAT | 0             | E     | D            | Enable the positioning in the curve. The options are:<br>RESET – it clears the timer. It puts the timer in the beginning of the curve.<br>START – It starts the timer.<br>ADVANCE – it puts the timer in the beginning of the next segment (only when in Man mode).<br>REPEAT – it puts the timer in the beginning of the actual segment (only when in Man mode)<br>After writing in one of the options, the parameter returns to No Action. |
| 24  | SPG_STATE   | Unsigned8          | 0=UNDEFINED<br>1=READY<br>2=ACTIVE<br>3=PAUSE<br>4=DONE    |               | E     | N/RO         | It indicates the actual state of the block operation. The available options are:<br>READY – when the timer is in the beginning of the curve waiting a signal to start.<br>ACTIVE – when the timer is on.<br>PAUSE – when the timer is stopped (see the cause in the PAUSE_CAUSE parameter)<br>DONE – When the timer reaches the last point of the curve.   |
| 25  | PRE_OUT     | DS-65              |  |               |       | D            | Displays what would be the OUT value and status if the mode was Auto or lower.   |
| 26  | RESET_IN    | DS-66              | 0:Off<br>1:Reset   |               | E     | D            | When the RESET_IN value is 1, it clears the timer and puts it in the beginning of the curve. The block will exist from the Reset only when the value returns to zero.  |
| 27  | BAL_TIME    | Float              | Positive   | 0             | sec   | S            | It specifies the time for OUT reaches the PRE_OUT when the block changes from Manual mode (or IMan – in the cascade initialization) to Auto mode.  |
| 28  | OUTAGE_LIM  | Float              | Positive   | 0             | Sec   | S            | The maximum tolerated duration for power failure.<br>This feature is not supported.  |
| 29  | UPDATE_EVT  | DS-73              |  |               | Na    | D            | This alert is generated by any change to the static data.  |
| 30  | BLOCK_ALM   | DS-72              |  |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.       |
| 31  | ALARM_SUM   | DS-74              | See Block Options  |               | Na    | S            | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 32  | ACK_OPTION  | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable                  | 0             | Na    | S            | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 33  | ALARM_HYS   | Float              | 0 to 50 %  | 0.5%          | %     | S            | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis.  |
| 34  | DV_HI_PRI   | Unsigned8          | 0 to 15  |               |       | S            | Priority of the high deviation alarm.  |
| 35  | DV_HI_LIM   | Float              | +(OUT_SCALE)<br>or +(INF)                                  | +INF          | OUT   | S            | The setting for high deviation alarm in engineering units.   |
| 36  | DV_LO_PRI   | Unsigned8          | 0 to 15  |               |       | S            | Priority of the low deviation alarm.   |

| Idx | Parameter | DataType<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|----------------------|---------------------------|------------------|-------|-----------------|--|
| 37  | DV_LO_LIM | Float                | -(OUT_SCALE)<br>or -(INF) | -INF             | OUT   | S               | The setting for low deviation alarm in engineering units.          |
| 38  | DV_HI_ALM | DS-71                |                           |                  | OUT   | D               | The status for high deviation alarm and its associated time stamp. |
| 39  | DV_LO_ALM | DS-71                |                           |                  | OUT   | D               | The status for low deviation alarm and its associated time stamp.  |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The default value of START\_TYPE is “Use Curve”.

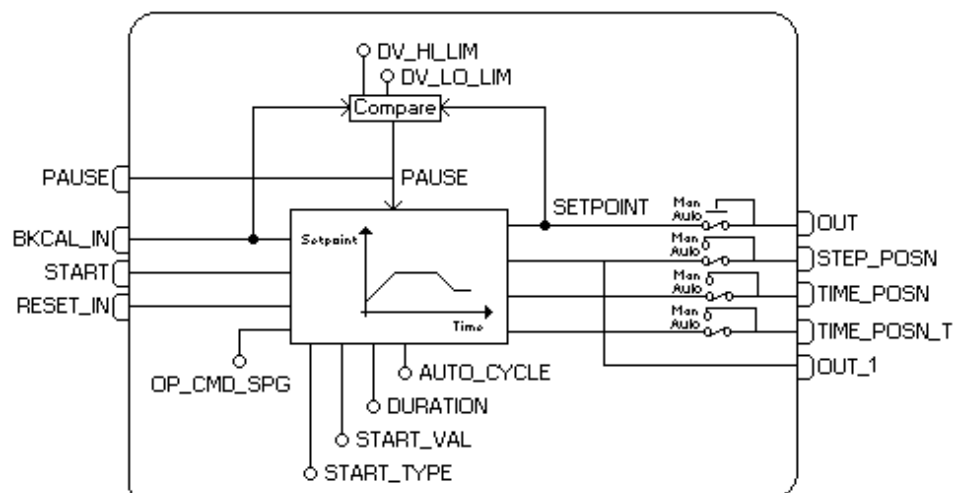
The required mode for writing is the actual mode, regardless the target mode: OUT, TIME\_POSN, TIME\_POSN\_T and STEP\_POSN

## ESPG – Enhanced Setpoint Ramp Generator

### Description

The Enhanced Setpoint Ramp Generator has all the features of the SPG block, and it has also an additional output parameter as it is possible to see in the schematic below. The OUT\_1 output indicates the current step or segment of profile like the SPG parameter STEP\_POSN. But the OUT\_1 format is float.

### Schematic



### Parameters

It has all parameters of the SPG block added of the

| Idx | Parameter | Data Type<br>(length) | Valid Range/<br>Options                                  | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|-----------------------|--|------------------|-------|-----------------|--|
| 40  | OUT_1     | DS-65                 | 0=None<br>1= Step1<br>2= Step2<br>3= Step 3<br>4= Step 4 | 0                | E     | D/RO            | This parameter identifies the current step or segment of profile. Similar to the STEP_POSN parameter but the format in this case is float. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

## TIME – Timer and Logic

### Description

The Timer and Logic function block provides logic combination and timing functions including the following:

- Combine multiple inputs as OR, AND, vote, or EXACTLY count.
- Measure the duration of the combined discrete input signal
- Accumulate, until reset, the duration of the combined input signal
- Count changes of the combined discrete input signal
- Set a discrete output if the duration of the combined input signal exceeds a limit
- Extend, Delay, Pulse, or Debounce the combined input as an output
- Provide outputs indicating amount of time expired and amount of time remaining
- Selectively invert any connected discrete input or output
- Reset timer

### Inputs Combination

Up to four inputs may be combined logically (AND, OR), voted (any 2 or more true, any 3 or more true), or counted (exactly 1 true, exactly 2 true, exactly 3, odd count, or even count). The combined input value is specified by the combination type (COMB\_TYPE) enumeration. Choices are indicated in the table below.

Connected inputs may have the values of true (1) or false (0). Values are different of 0 and 1 will be represented as false.

| COMB_TYPE | PV_D value   |
|-----------|--|
| OR        | true if one or more used inputs are true                             |
| ANY2      | true if two or more used inputs are true                             |
| ANY3      | true if three or more used inputs are true                           |
| AND       | true if all used inputs are true                                     |
| EXACTLY1  | true if exactly 1 used input is true                                 |
| EXACTLY2  | true if exactly 2 used inputs are true                               |
| EXACTLY3  | true if exactly 3 used inputs are true                               |
| EVEN      | true if exactly 0, 2 or 4 used inputs are true<br>(OR non exclusive) |
| ODD       | true if exactly 1 or 3 used inputs are true<br>(OR exclusive)        |

### Timer Processing

The timer processing type is specified by TIMER\_TYPE. It may operate to produce a measurement, delay, extension, pulse (non-re-triggerable or re-triggerable) or debounce, of the combined input signal.

TIMER\_SP is the specification for the time duration of delay, extension, pulse, debounce filter, or comparison limit.

OUT\_D output will be the PRE\_OUT\_D value when the block is in Auto mode. PRE\_OUT\_D is configured according to TIMER\_TYPE.

OUT\_EXP indicates the time elapsed and OUT\_REM indicates the remaining time when TIME\_TYPE is equal to the measurement, comparison, delay, extension, debounce, or pulse. See TIMER\_TYPE for details.

QUIES\_OPT allows the user to select the behavior for OUT\_EXP and OUT\_REM when the timer is quiescent-- that is, the timer is stopped waiting to start. The following table enumerates the definition of quiescent state for each option of TIMER\_TYPE:

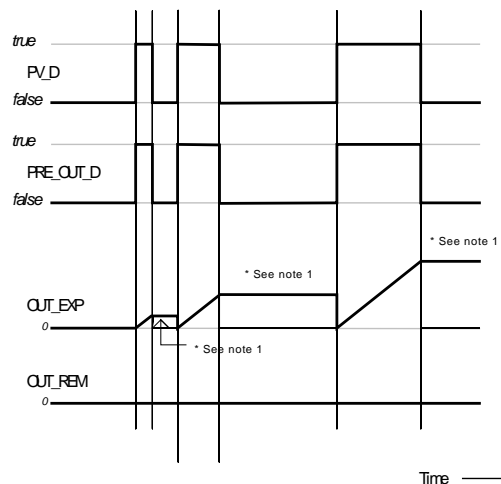
| Definition of quiescent state start and end as a function of TIMER_TYPE |   |   |
|---|---|---|
| TIMER_TYPE  | Quiescence state <b>starts</b><br>when combined input (PV_D): | Quiescence state <b>ends</b><br>when combined input (PV_D): |
| MEASURE   | returns to false  | changes from false-to-true                                  |
| ACCUM   | [QUIES_OPT does not apply]                                    | [QUIES_OPT does not apply]                                  |
| COMPARE   | returns to false  | changes from false-to-true                                  |
| DELAY   | returns to false  | changes from false-to-true                                  |
| EXTEND  | returns to true   | changes from true-to-false                                  |
| DEBOUNCE  | has changed <u>and</u> timer has expired                      | changes   |
| PULSE   | has returned to false <u>and</u> timer has expired            | changes from false-to-true                                  |
| RT_PULSE  | has returned to false <u>and</u> timer has expired            | changes from false-to-true                                  |

The CLEAR enumeration of QUIES\_OPT will cause both OUT\_EXP and OUT\_REM to be set to zero during quiescence. The LAST enumeration of QUIES\_OPT will cause both OUT\_EXP and OUT\_REM to be held to their values when the block becomes quiescent. That is, the time expired and time remaining will remain available until the quiescence ends with the start of the next activation. Note that a false-to-true transition on RESET\_IN will also reset OUT\_EXP and OUT\_REM.

N\_START is a count of the number of starts (false-to-true) transitions of the combined input, PV\_D, since the last false-to-true change seen on RESET\_IN.

TIMER\_TYPE may be one of the following, operating on the combined input signal:

- MEASURE - Indicate the duration of the most recent true signal
  - ACCUM - Accumulate the durations of a true signal
  - COMPARE - Compare a true signal duration to specified duration
  - DELAY - Delay a false-to-true transition, eliminating it if short
  - EXTEND - Extend a true-to-false transition, eliminating it if short
  - DEBOUNCE - Delay any transition, eliminating it if short
  - PULSE - Generate a true pulse on a false-to-true transition, non-retriggerable
  - RT\_PULSE - Generate a true pulse on a false-to-true transition, retriggerable
- If TIMER\_TYPE is **MEASURE**, PRE\_OUT\_D will be the same as the combined input, PV\_D. OUT\_EXP indicates the length of time, in seconds, that the combined signal is true. OUT\_REM is set to 0.

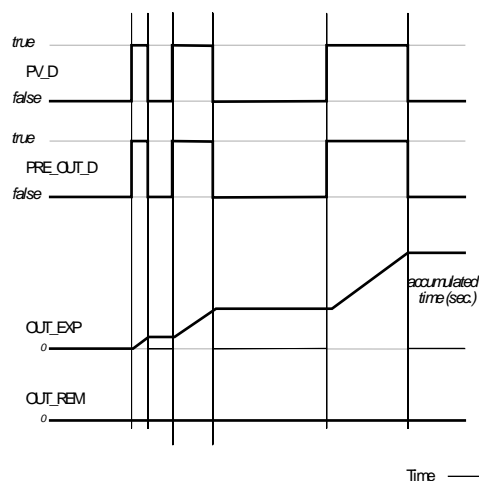


\* Note 1: Returns to zero if QUIES\_OPT = CLEAR

**Timer Example when TIMER\_TYPE = MEASURE**

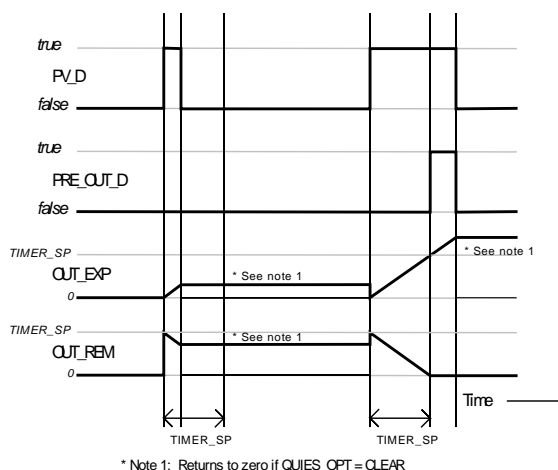


- If **TIMER\_TYPE** is **ACCUM**, **PRE\_OUT\_D** will be the same as the combined input, **PV\_D**. **OUT\_EXP** indicates the accumulated length of time, in seconds, that the combined signal has been true. Unlike **TIMER\_TYPE = MEAS**, it will not be automatically reset by the time of the next occurrence of a false-to-true change of **PV\_D**. Instead, it will continue to accumulate "on" time or "run" time until reset to 0 by a false-to-true change on **RESET\_IN**. **OUT\_REM** is unused (set to 0.0) for this timer type.



**Timer Example when *TIMER\_TYPE* = ACCUM**

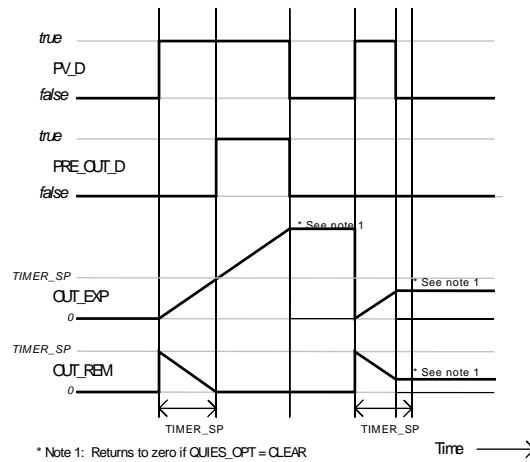
- If **TIMER\_TYPE** is **COMPARE**, the block will measure the time since a false-to-true change on the combined input, **PV\_D**. The current duration will be indicated by **OUT\_EXP**. **OUT\_REM** will indicate the time remaining between the current expired duration, **OUT\_EXP**, and current limit, **TIMER\_SP**. If **OUT\_EXP** does not exceed **TIMER\_SP**, **PRE\_OUT\_D** will be set to false. If **OUT\_EXP** equals or exceeds **TIMER\_SP**, **PRE\_OUT\_D** will be set to true and **OUT\_REM** will be set to zero. When the combined input returns to false, either with or without exceeding the limits specified by **TIMER\_SP**, **OUT\_D** will be set to false. [Note that this type of behavior is the same as **TIMER\_TYPE = DELAY**. The difference is merely in the application perspective]



\* Note 1: Returns to zero if **QUIES\_OPT** = **CLEAR**

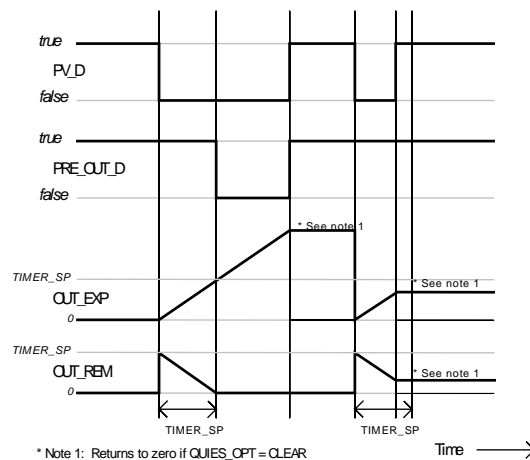
**Timer Example when *TIMER\_TYPE* = COMPARE**

- If **TIMER\_TYPE** is **DELAY**, a false-to-true change on the combined input, **PV\_D**, will be delayed at the output, **PRE\_OUT\_D**, until the amount of time specified by **TIMER\_SP** has been expired. If the combined input returns to false before the time expires, the output will remain as false, concealing the input transitions. If the **PRE\_OUT\_D** output has been set to true because the time has expired, a true-to-false transition in the combined input will be presented to **PRE\_OUT\_D** immediately. [Note that this type of behavior is the same as **TIMER\_TYPE = COMPARE**. The difference is merely in the application perspective.]



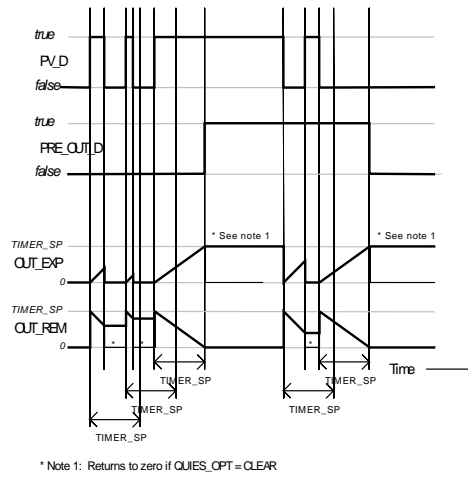
**Timer Example when *TIMER\_TYPE* = *DELAY***

- If **TIMER\_TYPE** is **EXTEND**, a true-to-false change on the combined input, **PV\_D**, will be delayed at the output, **PRE\_OUT\_D**, until the amount of time specified by **TIMER\_SP** has been expired. If the combined input returns to true before the time expires, the output will remain as true, concealing the input transitions. If the **PRE\_OUT\_D** output has been set to false because the time has expired, a false-to-true transition in the combined input will be presented to **PRE\_OUT\_D** immediately.



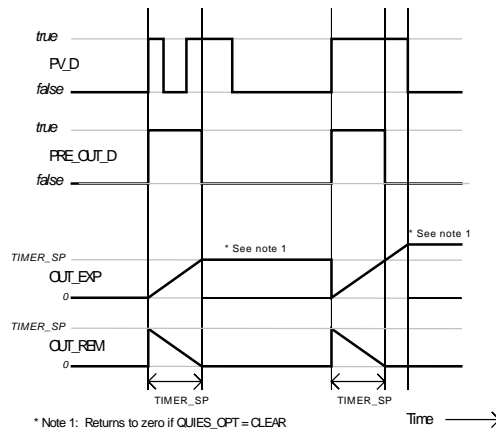
**Timer Example when *TIMER\_TYPE* = *EXTEND***

- If **TIMER\_TYPE** is **DEBOUNCE**, and if **PRE\_OUT\_D** is false, a false-to-true change on the combined input, **PV\_D**, will be delayed at the output, **PRE\_OUT\_D**, until the amount of time specified by **TIMER\_SP** has been expired. If the combined input returns to false before the time expires, the output will remain as false, concealing the input transitions. If **PRE\_OUT\_D** is true, a true-to-false change on the combined input, **PV\_D**, will be delayed at the output, **PRE\_OUT\_D**, until the amount of time specified by **TIMER\_SP** has been expired. If the combined input returns to true before the time expires, the output will remain as true, concealing the input transitions. This both delays true initiations and extends true terminations, acting as a filter for intermittent state changes.



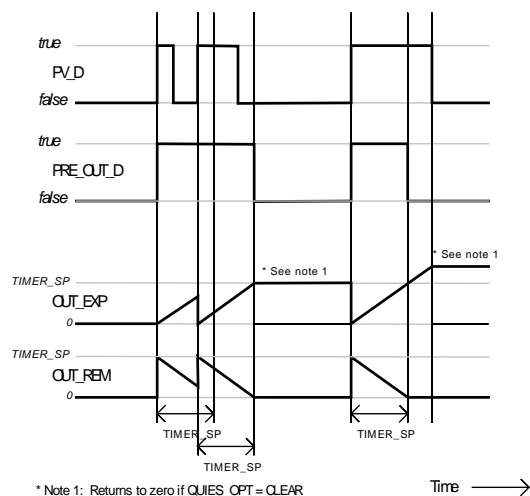
#### Timer Example when **TIMER\_TYPE = DEBOUNCE**

- If **TIMER\_TYPE** is **PULSE**, a false-to-true change on the combined input, **PV\_D**, will initiate a true pulse at **PRE\_OUT\_D** whose duration is determined by the **TIMER\_SP** value. At the end of the time duration, the output, will return to false. Further false-to-true transitions of the combined input while **PRE\_OUT\_D** is true will be ignored.



#### Timer Example when **TIMER\_TYPE = PULSE**

- If **TIMER\_TYPE** is **RT\_PULSE**, (Re-Triggerable pulse type) a false-to-true change on the combined input, **PV\_D**, will initiate a true pulse at **PRE\_OUT\_D** whose duration is determined by the **TIMER\_SP** value. At the end of that time duration **PRE\_OUT\_D** will return to false. If the combined input returns to false and presents a subsequent false-to-true transition while the timer is timing, the timer shall be reinitialized and **PRE\_OUT\_D** shall continue to be true.



#### Timer Example when $TIMER\_TYPE = RT\_PULSE$

RESET\_IN is a discrete input which, on a false-to-true transition, resets the timer. OUT\_EXP output is set to 0, and then the timer follows processing described in "Initialization Treatment" for the values PRE\_OUT\_D and OUT\_REM.

INVERT\_OPTS allows invert the discrete input or output values. Each input or output can be set individually.

In Man mode, the block allows change the OUT\_D outputs, the calculations is kept and PRE\_OUT\_D is calculated. In Auto mode, PRE\_OUT\_D is passed to the OUT\_D.

If the "Use Uncertain" option is selected in the STATUS\_OPTS, inputs with Uncertain status will be represented as Good (usable). If this option is not set, inputs with Uncertain status will be interpreted as Bad (Not usable).

When any one of the inputs receives a non usable status, the output will receive the non usable status and then the timer will stop the measurement. When the outputs return to usable status, the timer returns to the measurement and the OUT\_EXP and OUT\_REM status will be set to Uncertain while they are in quiescent state or a Reset occurs.

#### Initialization Treatment

The following table summarizes the values of PRE\_OUT\_D, OUT\_EXP and OUT\_REM after the initial execution, as a function of  $TIMER\_TYPE$  and the initial value of the combined input, PV\_D:

| TIMER_TYPE | PV_D  | PRE_OUT_D | OUT_EXP    | OUT_REM    | Timer Status |
|------------|-------|-----------|------------|------------|--------------|
| MEASURE    | False | False     | 0.0        | 0.0        | Inactive     |
| MEASURE    | True  | True      | 0.0        | 0.0        | Inactive     |
| ACCUM      | False | False     | 0.0        | 0.0        | Inactive     |
| ACCUM      | True  | True      | 0.0        | 0.0        | Inactive     |
| COMPARE    | False | False     | TIMER_SP † | 0.0        | Inactive     |
| COMPARE    | True  | False     | 0.0        | TIMER_SP † | Active       |
| DELAY      | False | False     | TIMER_SP † | 0.0        | Inactive     |
| DELAY      | True  | False     | 0.0        | TIMER_SP † | Active       |
| EXTEND     | False | True      | 0.0        | TIMER_SP † | Active       |
| EXTEND     | True  | True      | TIMER_SP † | 0.0        | Inactive     |
| DEBOUNCE   | False | False     | TIMER_SP † | 0.0        | Inactive     |
| DEBOUNCE   | True  | True      | TIMER_SP † | 0.0        | Inactive     |
| PULSE      | False | False     | 0.0        | 0.0        | Inactive     |
| PULSE      | True  | False     | TIMER_SP † | 0.0        | Inactive     |
| RT_PULSE   | False | False     | 0.0        | 0.0        | Inactive     |
| RT_PULSE   | True  | False     | TIMER_SP † | 0.0        | Inactive     |

† Initialize to TIMER\_SP value if QUIES\_OPT = LAST, initialize to 0.0 if QUIES\_OPT = CLEAR.

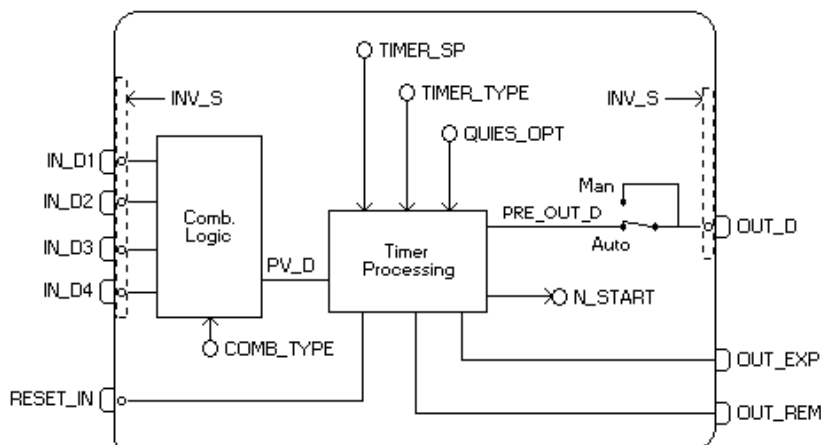
**BLOCK\_ERR**

The BLOCK\_ERR of the TIME block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the TIME\_UNITS or QUIES\_OPT parameters have an invalid value;
- Out of Service – it occurs when the block is in O/S mode.

**Modes Supported**

O/S, MAN and AUTO.

**Schematic****Parameters**

| Idx | Parameter   | Data Type (length) | Valid Range/Options | Default Value | Units | Store / Mode | Description  |
|-----|-------------|--------------------|---------------------|---------------|-------|--------------|--|
| 1   | ST_REV      | Unsigned16         |                     | 0             | None  | S/RO         |  |
| 2   | TAG_DESC    | OctString(32)      |                     | Spaces        | Na    | S            |  |
| 3   | STRATEGY    | Unsigned16         |                     | 0             | None  | S            |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255            | 0             | None  | S            |  |
| 5   | MODE_BLK    | DS-69              |                     | O/S           | Na    | S            | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)       |                     |               | E     | D/RO         | This is the timer duration used by the timer block for delay, extension, debouncing, and pulse time-processing.                  |
| 7   | PV_D        | DS-66              |                     |               |       | RO           | Either the primary discrete value for use in executing the function, or a process value associated with it.                      |
| 8   | OUT_D       | DS-66              |                     |               |       | D            | The primary discrete value calculated as a result of executing the function.   |
| 9   | TIMER_SP    | Float              | Positive            | 0             | Sec   | S            | Time used by the TIME block to set the delay, extend, debouncing and processing the pulse time.                                  |
| 10  | PV_STATE    | Unsigned16         |                     | 0             |       | S            | Index to the text describing the states of a discrete PV.  |
| 11  | OUT_STATE   | Unsigned16         |                     | 0             |       | S            | Index to the text describing the states of a discrete output.  |
| 12  | GRANT_DENY  | DS-70              |                     | 0             | Na    | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 13  | INVERT_OPTS | Bitstring(2)       | See Block Options   | 0             | Na    | S / O/S      | See Block Options  |
| 14  | STATUS_OPTS | Bitstring(2)       | See Block Options   | 0             | Na    | S / O/S      | See Block Options  |
| 15  | IN_D1       | DS-66              |                     |               |       | D            | Numbered discrete input parameter 1.   |
| 16  | IN_D2       | DS-66              |                     |               |       | D            | Numbered discrete input parameter 2.   |
| 17  | IN_D3       | DS-66              |                     |               |       | D            | Numbered discrete input parameter 3.   |
| 18  | IN_D4       | DS-66              |                     |               |       | D            | Numbered discrete input parameter 4.   |

| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|------------|-----------------------|---|------------------|-------|-----------------|--|
| 19  | COMB_TYPE  | Unsigned8             | 0=AND<br>1=OR<br>2=ANY2<br>3=ANY3<br>21=EXACTLY1<br>22=EXACTLY2<br>23=EXACTLY3<br>40=EVEN<br>41=ODD | 1                | E     | S/Man           | Determines how the inputs IN_D[i] are combined.  |
| 20  | TIMER_TYPE | Unsigned8             | 0=MEASURE<br>1=ACCUM<br>2=COMPARE<br>3=DELAY<br>4=EXTEND<br>5=DEBOUNCE<br>6=PULSE<br>7=RT_PULSE     | 0                | E     | S/Man           | Type of processing applied to PV_D to determine the PRE_OUT_D.   |
| 21  | PRE_OUT_D  | DS-66                 |   |                  |       | RO              | This parameter is the combined and time-processed output of the timer block.   |
| 22  | N_START    | Unsigned16            |   |                  | None  | D/RO            | Count of <i>false-to-true</i> transitions of the combined input, PV_D. Reset by <i>false-to-true</i> transition of RESET_IN.   |
| 23  | OUT_EXP    | DS-65                 |   |                  | Sec   | N / RO          | This is the time expired. Stops when TIMER_SP is reached. Reset to zero (1) by RESET_IN, (2) at start of next timer event if QUIES_OPT = LAST, or (3) when block becomes quiescent if QUIES_OPT = CLEAR.   |
| 24  | OUT_REM    | DS-65                 |   |                  | Sec   | N / RO          | This is the time remaining if the timer is active. Stops when event ceases (block becomes quiescent). Reset to 0.0 if QUIES_OPT = CLEAR, and the timer is inactive.  |
| 25  | RESET_IN   | DS-66                 | 0=Off<br>1=Reset  |                  |       |                 | Resets the timer.  |
| 26  | QUIES_OPT  | Unsigned8             | 1=CLEAR<br>2=LAST   | 0                | E     | S / O/S         | Behavior option for OUT_EXP and OUT_REM during quiescence. CLEAR resets them to zero. LAST causes last values to be held.  |
| 27  | TIME_UNITS | Unsigned8             | 1=seconds<br>2=minutes<br>3=hours<br>4=days<br>5=[day-<br>[hr:[min[:sec]]]]                         | 0                | E     | S               | This parameter has fixed unit: seconds. TIME_UNITS is not used.  |
| 28  | UPDATE_EVT | DS-73                 |   |                  | na    | D               | This alert is generated by any change to the static data.  |
| 29  | BLOCK_ALM  | DS-72                 |   |                  | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

If DIAG.BEHAVIOR parameter is “Adapted”:  
The default value of TIME\_UNITS is “Seconds”.  
The default value of QUIES\_OPT is “CLEAR”.

## LLAG - Lead Lag

### Description

The LLAG block provides dynamic compensation of the IN parameter. The block can function as a lead, lag, or both. The user would configure the LEAD\_TIME and LAG\_TIME parameters to obtain the desired input/output relationship.

The LAG\_TIME parameter specifies the time constant for the block. Based on a step change to the input this is the time to reach 63.2% of the final value.

The LEAD\_TIME parameter specifies the gain or impulse applied to the input parameter.

The FOLLOW parameter is used to start internal parameters. When FOLLOW is true, it forces OUT tracks IN.

The generalized form of the equation describing the action is as follows:

$$G(s) = \frac{(T_1s + 1)}{(T_2s + 1)}$$

Where:

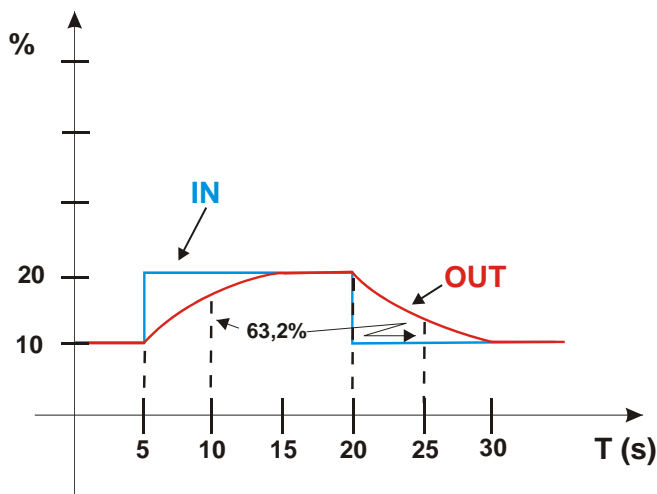
$T_1$  = Lead time Constant

$T_2$  = Lag Time Constant

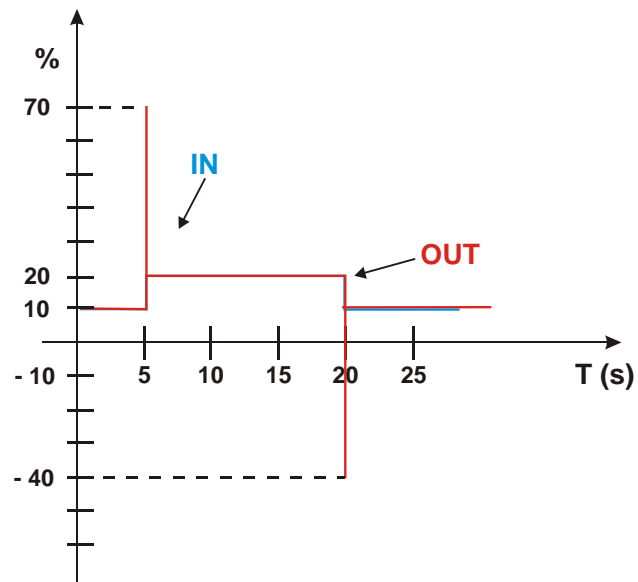
### Examples for Applications

Initially, it considers an input signal  $IN = 10$ . This input receives a positive step change equals to 10% in  $t = 5s$ . In  $t = 20s$ , the input receives a negative step change equals to 10%. The LLAG action can be observed for the following cases:

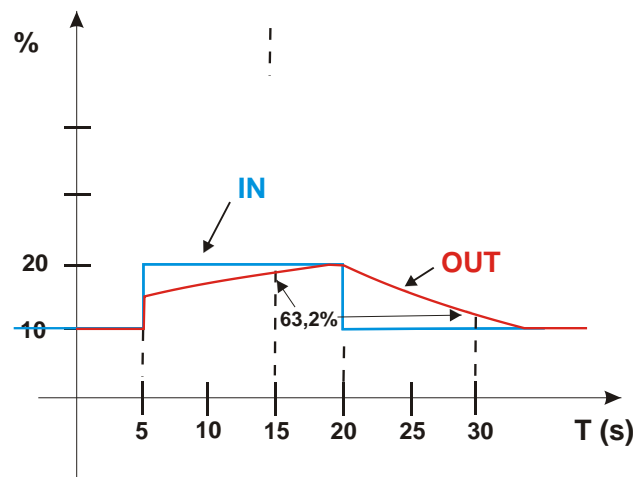
- 1) LEAD\_TIME = 0 and LAG\_TIME = 5



2) LEAD\_TIME = 5 and LAG\_TIME = 0



3) LEAD\_TIME = 5 and LAG\_TIME = 10



### Man to Auto Transition

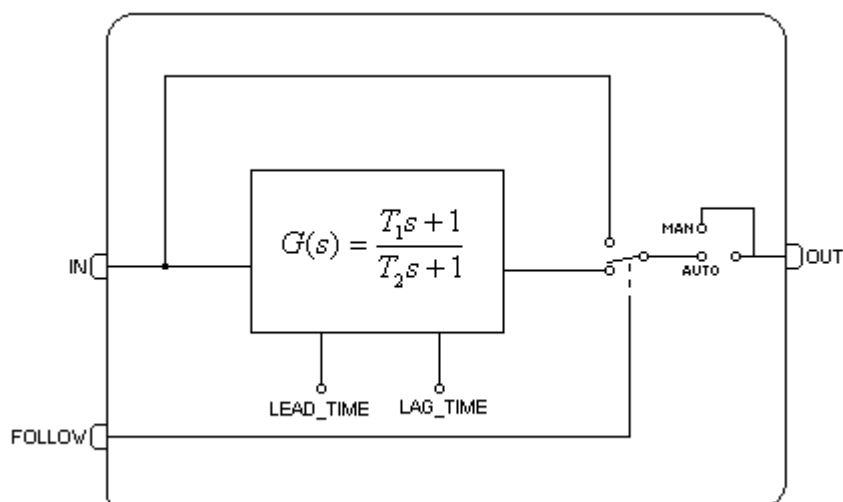
When the block is in Man mode, it is possible writing in OUT. During the transition from Manual mode to Auto mode, if the BAL\_TIME parameter is not configured, that is, it is zero; the transition from the written value to the calculated value will be done with bump, such as a pulse (Positive or negative). Through the BAL\_TIME configuration, given in seconds, a bumpless transition can be performed from the written value to the calculated value

### Supported Modes

O/S, MAN and AUTO.



## Schematic



## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/ Options   | Default Value | Units | Store / Mode | Description  |
|-----|-------------|--------------------|------------------------|---------------|-------|--------------|--|
| 1   | ST_REV      | Unsigned16         |                        | 0             | None  | S/RO         |  |
| 2   | TAG_DESC    | OctString(32)      |                        | Spaces        | Na    | S            |  |
| 3   | STRATEGY    | Unsigned16         |                        | 0             | None  | S            |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255               | 0             | None  | S            |  |
| 5   | MODE_BLK    | DS-69              |                        | O/S           | Na    | S            | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)       |                        |               | E     | D/RO         |  |
| 7   | OUT         | DS-65              |                        |               | OUT   | D            | The primary analog value calculated as a result of executing the function.   |
| 8   | OUT_RANGE   | DS-68              |                        | 0             | E     | S            | Defines the display scale for the output.  |
| 9   | GRANT_DENY  | DS-70              |                        | 0             | Na    | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.       |
| 10  | STATUS_OPTS | Bitstring(2)       | Bit 2: Use Unc as Good | 0             | Na    | S / O/S      | See Block Options  |
| 11  | IN          | DS-65              |                        |               |       | D            | The primary input value of the block, required for blocks that filter the input to get the PV.   |
| 12  | FOLLOW      | DS-66              | True=follow            |               | Na    | D            | Tracking input, when true causes the output to track the input.  |
| 13  | LAG_TIME    | Float              |                        |               | Sec   | D            | Specifies the lag time constant for the block. Based on a step change to the input this is the time to reach 63.2% of the final value. |
| 14  | LEAD_TIME   | Float              |                        |               | Sec   | D            | Specifies the lead time constant applied to the input parameter.   |
| 15  | BAL_TIME    | Float              | Positive               | 0             | Sec   | S            | This specifies the time for the internal working value of bias or ratio to return to the operator set bias or ratio, in seconds.       |
| 16  | OUTAGE_LIM  | Float              | Positive               | 0             | Sec   | S            | The maximum tolerated duration for power failure. This feature is not supported.   |
| 17  | UPDATE_EVT  | DS-73              |                        |               | na    | D            | This alert is generated by any change to the static data.  |

| Idx | Parameter | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 18  | BLOCK_ALM | DS-72                 |                         |                  | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The required mode for writing is the actual mode, regardless the target mode: OUT.

## OSDL - Output Signal Selector and Dynamic Limiter

### Description

The output signal selector and dynamic limiter block (OSDL) provides two different algorithms types:

As Output Selector the cascade input may be routed for one of two outputs based on the value of the OP\_SELECT input parameter. The output not selected may have two ways: keeping the last value when not selected, or receive a internal value.

As Dynamic Limiter the cascade input is transferred to both output, but it is limited by the secondary inputs multiplied by a gain, plus a bias. The Dynamic LIMITER is extremely useful in one of its most important applications: combustion control with double cross limits.

The OSDL\_TYPE parameter determines the algorithm used by the OSDL block. In order to change the OSDL\_TYPE the block must be in Out of Service mode.

### Output Signal Selector

The SP value may be controlled by an operator (Auto mode) or through a cascade control (Cas mode). In the cascade control the SP is supplied by another function block through the CAS\_IN parameter. The inputs IN and IN\_1 are not applied to this algorithm. It means that the block ignores the status and values of IN and IN\_1 when the OSDL\_TYPE is Output Selector.

The OP\_SELECT is a discrete input parameter that selects one of two outputs to receive SP parameter. When the OP\_SELECT is zero, the OUT parameter receives the SP parameter. Otherwise the OUT\_1 parameter receives the SP parameter.

When the OP\_SELECT status is not usable, the block changes to Auto, but the algorithm goes on working with the unusable value.

Most of the times, the SP is transferred to the selected output whatever is the status. Therefore, an unusable value and status in the CAS\_IN will be reflected to the selected output. Only the status of upper cascade initialization will not be copied to the selected output.

### Handling the non-selected output

There are two ways to handle the non-selected output, if the "Keep last value if not selected" bit in OSDL\_OPTS parameter is true, the non-selected output will keep the last value. Otherwise, the outputs OUT and OUT\_1 will receive the value contained in the NOT\_SEL\_VAL and NOT\_SEL\_VAL\_1, respectively.

The non-selected output receives the uncertain status indicating to the downstream block that it is not selected any more. The configuration of the STATUS\_OPTS in the downstream block will define how to deal with it.

### IFS status propagation

If the "IFS only for selected output" bit in OSDL\_OPTS parameter is true, the fault state status will be propagated only to selected output. Otherwise the status GoodCascade -IFS is propagated to both outputs, that is the default configuration.

### Downstream block is not in cascade

If the downstream block of the selected output is not in cascade mode, the OSDL block goes to lman mode. And the status of BKCAL\_OUT will be GoodCascade – Not Invited, it will force the upstream block to lman mode too.

If the downstream block of the non-selected output is not in cascade mode, the OSDL block will ignore it.

### Dynamic Limiter

As a dynamic limiter algorithm, the outputs are the value of the "CAS\_IN" parameter limited by the following values:

OUT:

High limit = HI\_GAIN\_1 \* IN\_1 + HI\_BIAS\_1

Low limit = LO\_GAIN\_1 \* IN\_1 - LO\_BIAS\_1

OUT\_1:

High limit = HI\_GAIN \* IN + HI\_BIAS

Low limit = LO\_GAIN \* IN - LO\_BIAS

After the limitation, the parameters GAIN and GAIN\_1 are applied as gain for the outputs OUT and OUT\_1, respectively.

The normal mode of operation for the OSDL block is Cas, as well the two downstream blocks. If one downstream block is not in cascade mode, indicated by not invited status (NI) on its BKCAL\_OUT, the OSDL block still continues in cascade mode. Only if both downstream blocks are not in cascade, then the OSDL changes to Iman mode and its BKCAL\_OUT output changes to NI.

If the OSDL block is in Iman mode, when the cascade is initialized with a IR by a downstream block, the respective output (OUT or OUT\_1) sends a IA for the downstream block and the BKCAL\_OUT of the OSDL block receive the value of the respective BKCAL\_IN. The OSDL block remains in IMAN mode until the downstream cascade is initialized. Then OSDL block goes to Auto mode and it sends an IR to upstream block to initialize the cascade.

After a downstream cascade initialization, the corresponding output must ramp from the last BKCAL\_IN to the calculated value in BAL\_TIME seconds.

The required actions as a dynamic limiter algorithm are summarized in the following table:

| Mode<br>Target/Actual | BKCAL_IN | BKCAL_IN1 | BKCAL_OUT | ACTION                              |
|-----------------------|----------|-----------|-----------|-------------------------------------|
| Cas/Iman              | NI or IR | NI or IR  | NI        |                                     |
| Cas/Cas               | NI or IR | OK        | OK        | BKCAL_OUT receive the CAS_IN value  |
| Cas/Cas               | OK       | NI or IR  | OK        | BKCAL_OUT receive the CAS_IN value. |
| Cas/Cas               | OK       | OK        | OK        | BKCAL_OUT receive the CAS_IN value. |

**Legend:** NI-not invited; IR-initialization request; OK-working in cascade

Optionally, when the block is working as “Dynamic Limiter, it can choose if the return value for the upper block through the BKCAL\_OUT output will be SP, Out or OUT\_1, as showed below:

- BKCAL\_OUT = SP (default)
- BKCAL\_OUT = OUT (when OSDL\_OPTS = “Use OUT for BKCAL\_OUT”).
- BKCAL\_OUT = OUT\_1 (When OSDL\_OPTS = “Use OUT\_1 for BKCAL\_OUT\_1”).

#### BLOCK\_ERR

The BLOCK\_ERR of the OSDL block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when the OUT\_TYPE parameter has an invalid value.
- Out of Service – When the block is in O/S mode.

#### Supported Modes

O/S, IMAN, AUTO and CAS.

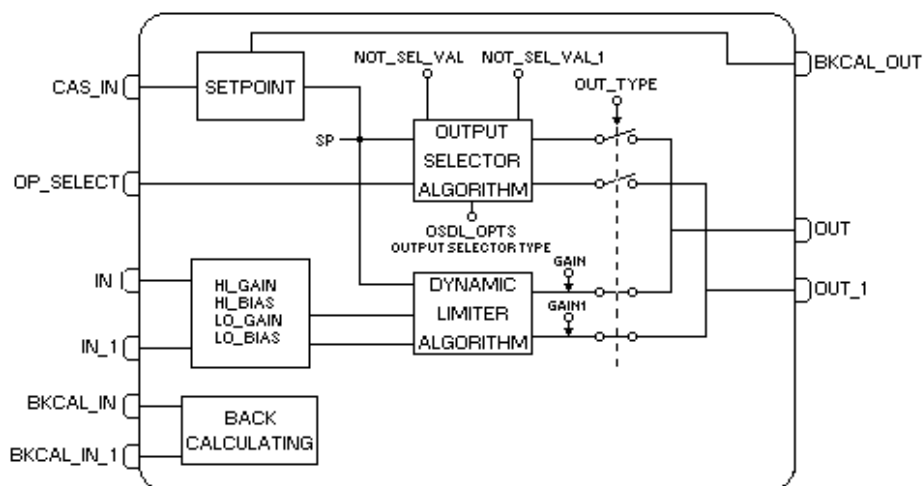
#### Status Handling

Standard, plus the following:

When one or both of the IN's input are Bad, special limiting action must be taken. If the “IFS if Bad IN\_x “ or “IFS if Bad CAS\_IN “ bit is true in the OSDL\_OPTS parameter and the respective input is Bad, both output status go to “good IFS”. If the bit is not true, the block goes to AUTO mode.

Sub-Status values received in CAS\_IN will be passed onto both outputs, except for those used in the cascade initialization. An IFS will go to both the selected and the non-selected output.

## Schematic



## Parameters

| Idx | Parameter  | Data Type (length) | Valid Range/Options   | Default Value | Units | Store/Mode | Description  |
|-----|------------|--------------------|---|---------------|-------|------------|--|
| 1   | ST_REV     | Unsigned16         |   | 0             | None  | S/RO       |  |
| 2   | TAG_DESC   | OctString(32)      |   | Spaces        | Na    | S          |  |
| 3   | STRATEGY   | Unsigned16         |   | 0             | None  | S          |  |
| 4   | ALERT_KEY  | Unsigned8          | 1 to 255  | 0             | None  | S          |  |
| 5   | MODE_BLK   | DS-69              |   | O/S           | Na    | S          | See Mode Parameter   |
| 6   | BLOCK_ERR  | Bitstring(2)       |   |               | E     | D/RO       |  |
| 7   | CAS_IN     | DS-65              |   |               |       | D          | This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link. |
| 8   | SP         | DS-65              |   |               |       | N / Auto   | The analog set point.  |
| 9   | IN         | DS-65              |   |               |       | D          | The primary input value of the block.  |
| 10  | IN_1       | DS-65              |   |               |       | D          | Numbered input parameter 1.  |
| 11  | OP_SELECT  | DS-66              | 0 thru 4  |               | None  | D          | An operator adjustable parameter to force a given input to be used.  |
| 12  | OUT        | DS-65              |   |               |       | D / RO     | The primary analog value calculated as a result of executing the function.   |
| 13  | OUT_1      | DS-65              |   |               |       | D / RO     | Numbered output parameter 1.   |
| 14  | GRANT_DENY | DS-70              |   | 0             | Na    | S / O/S    | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block. |
| 15  | OUT_TYPE   | Unsigned8          | 0 : Invalid Value<br>1 : Output Selector<br>2 : Dynamic Limiter | 0             | E     | S / Man    | This parameter specifies the algorithm type that will be calculated.   |
| 16  | OSDL_OPTS  | Bitstring(2)       |   | 0             | Na    | S / O/S    | Option bit string for handling the block processing.   |
| 17  | HI_GAIN    | Float              |   | 1.1           | None  | S          | It is used to calculate the high limit for OUT_1. This gain multiplies IN before adding HI_BIAS.                                 |
| 18  | HI_BIAS    | Float              | Positive  | 0             |       | S          | It is used to calculate the high limit for OUT_1. This bias is added to IN after multiplying by HI_GAIN.                         |

| Idx | Parameter     | Data Type (length) | Valid Range/ Options | Default Value | Units | Store/ Mode | Description  |
|-----|---------------|--------------------|----------------------|---------------|-------|-------------|--|
| 19  | LO_GAIN       | Float              |                      | 0.9           | None  | S           | It is used to calculate the low limit for OUT_1. This gain multiplies IN before subtracting LO_BIAS.   |
| 20  | LO_BIAS       | Float              | Positive             | 0             |       | S           | It is used to calculate the low limit for OUT_1. This bias is subtracted of IN after multiplying by LO_GAIN.   |
| 21  | HI_GAIN_1     | Float              |                      | 1.1           | None  | S           | It is used to calculate the high limit for OUT. This gain multiplies IN_1 before adding HI_BIAS_1.   |
| 22  | HI_BIAS_1     | Float              | Positive             | 0             |       | S           | It is used to calculate the high limit for OUT. This bias is added to IN_1 after multiplying by HI_GAIN_1.   |
| 23  | LO_GAIN_1     | Float              |                      | 0.9           | None  | S           | It is used to calculate the low limit for OUT. This gain multiplies IN_1 before subtracting LO_BIAS_1.   |
| 24  | LO_BIAS_1     | Float              | Positive             | 0             |       | S           | It is used to calculate the low limit for OUT. This bias is subtracted of IN_1 after multiplying by LO_GAIN_1.   |
| 25  | GAIN          | Float              |                      | 1             | None  | S           | Gain applied to OUT after limiting.  |
| 26  | GAIN_1        | Float              |                      | 1             | None  | S           | Gain applied to OUT_1 after limiting.  |
| 27  | BKCAL_IN      | DS-65              |                      |               |       | N           | The value and status from a lower block's BKCAL_OUT that is used to prevent reset windup and to initialize the control loop.   |
| 28  | BKCAL_IN_1    | DS-65              |                      |               |       | N           | The back calculated input required to initialize a lower cascade 1.  |
| 29  | BKCAL_OUT     | DS-66              |                      |               |       | D           | The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control.  |
| 30  | BAL_TIME      | Float              |                      | 0             | Sec   | S           | This specifies the time for the internal working value of bias or ratio to return to the operator set bias or ratio, in seconds.   |
| 27  | NOT_SEL_VAL   | Float              |                      | 0             |       | S           | Contained parameter that will set the respective output when OUT is not selected by the OP_SELECT.   |
| 28  | NOT_SEL_VAL_1 | Float              |                      | 0             |       | S           | Contained parameter that will set the respective output when OUT_1 is not selected by the OP_SELECT.   |
| 33  | UPDATE_EVT    | DS-73              |                      |               | na    | D           | This alert is generated by any change to the static data.  |
| 34  | BLOCK_ALM     | DS-72              |                      |               | na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

If DIAG.BEHAVIOR parameter is “Adapted”:

The default value of OUT\_TYPE is “Dynamic limiter”.

The required mode for writing is the actual mode, regardless the target mode: SP.

## DENS - Density

### Overview

This function block has an algorithm to calculate density in different kinds of engineering units, as Plato degree, Brix, TC and INPM.

### Description

The algorithm to calculate the density is based on the pressure in two points of the tank in a known difference height. The calculation is done using the average from the sensor pressure samples (the number of the samples is determined by the NUM\_SAMPLES parameter). Then, the previous density is calculated using the following formula:

$$D = \frac{\text{Conv\_factor} * (\overline{P\_1} - \overline{P\_2})}{\text{HEIGHT} * \text{GRAVITY}} \left[ \frac{\text{g}}{\text{cm}^3} \right]$$

Where in the formula above the average pressure is given by:

$$\overline{P\_i} = \frac{\sum_{j=1}^{\text{NUM\_SAMPLES}} \text{IN\_i}}{\text{NUM\_SAMPLES}} [\text{PRESSURE\_UNITS}]$$

*Conv\_Factor* is a factor to transform the formula coefficients in the same units.

*HEIGHT* and *GRAVITY* ≠ 0

The temperature is compensated in the density calculation. The DENS\_OUT parameter is the compensated density in g/cm<sup>3</sup>. The OUT parameter is the compensated density in a different engineering unit for chosen by the EU\_SEL parameter.

The density block provides alarm condition and the discrete alarm output to be used any block. If the density exceeds HI\_LIM or LO\_LIM an alarm is indicated in HI\_ALM or LO\_ALM and the output OUT\_D will be set to true. If one or both of the limit parameters are set +/- INF this indicate that this alarm is disabled.

The RESET\_IN is a discrete input, which on a false-to-true transition resets the density block, including the previous density, the round buffer and all outputs of the block. The status will be the same as the last cycle.

### BLOCK\_ERR

The BLOCK\_ERR of the DENS block will reflect the following causes:

- Out of Service – it occurs when the block is in O/S mode.

### Supported Modes

O/S, MAN and AUTO.

### Mode Handling

Manual mode disconnects the outputs from the algorithm and permits manual substitution of the OUT, DENS\_OUT, OUT\_D values for test or other purposes. Although the OUT\_D parameter is also disconnected from the alarm, the alarm and the limits (OUT\_HI\_LIM and OUT\_LO\_LIM) continue to check the output.

### Status Handling

The primary input status (IN\_1 and IN\_2 parameters) are propagated to the outputs.

If the status of any primary input becomes bad or uncertain and their respective option "use uncertain" of STATUS\_OPTS is not set, the actual mode block will be forced to manual and the algorithm stops the calculation.

If the secondary input (IN\_3 parameter) is unusable, the algorithm uses the last usable value and the output status will be Uncertain.

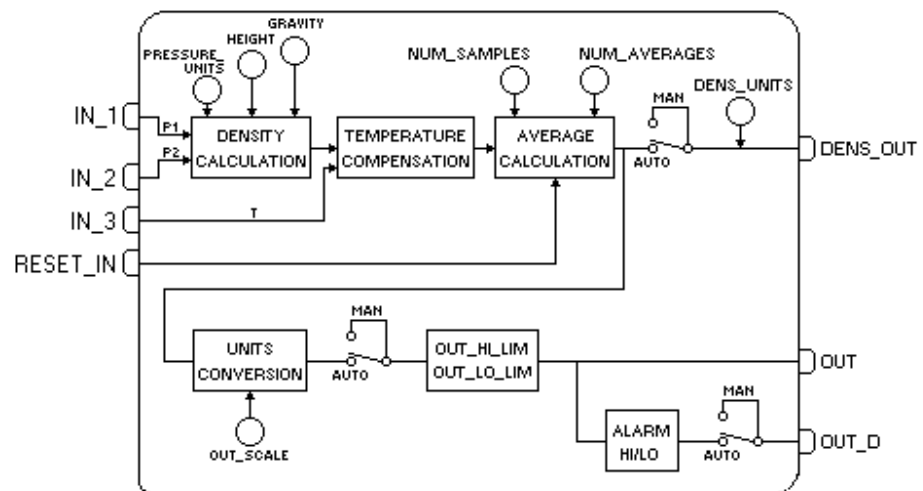
A bad status in RESET\_IN input does not stop the algorithm.

If target mode is Man then the output status is Good.

### Valid Pressure Units

| Index | Unit    |
|-------|---------|
| 1130  | PA      |
| 1133  | KPA     |
| 1132  | MPA     |
| 1137  | BAR     |
| 1138  | MBAR    |
| 1139  | TORR    |
| 1140  | ATM     |
| 1141  | PSI     |
| 1144  | GCM2    |
| 1145  | KGCM2   |
| 1148  | INH20   |
| 1147  | INH204C |
| 1151  | MMH20   |
| 1150  | MMH204C |
| 1154  | FTH20   |
| 1156  | INHG    |
| 1158  | MMHG    |

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/Options | Default Value | Units | Store/Mode | Description        |
|-----|-----------|--------------------|---------------------|---------------|-------|------------|--------------------|
| 4   | ST_REV    | Unsigned16         |                     | 0             | None  | S/RO       |                    |
| 5   | TAG_DESC  | OctString(32)      |                     | Spaces        | Na    | S          |                    |
| 6   | STRATEGY  | Unsigned16         |                     | 0             | None  | S          |                    |
| 4   | ALERT_KEY | Unsigned8          | 1 to 255            | 0             | None  | S          |                    |
| 5   | MODE_BLK  | DS-69              |                     | O/S           | Na    | S          | See Mode Parameter |



| Idx | Parameter      | Data Type<br>(length) | Valid Range/<br>Options                        | Default<br>Value          | Units             | Store/<br>Mode | Description  |
|-----|----------------|-----------------------|--|---------------------------|-------------------|----------------|--|
| 6   | BLOCK_ERR      | Bitstring(2)          |  |                           | E                 | D / RO         |  |
| 7   | OUT            | DS-65                 | OUT_SCALE<br>+/- 10%                           |                           | OUT               | D / Man        | The analog value calculated as a result of executing the function.   |
| 8   | OUT_SCALE      | DS-68                 |  | 0-100%                    | OUT               | S / Man        | The high and low scale values to the OUT parameter.  |
| 9   | EU_SEL         | Unsigned8             | 0: Plato degree<br>1: Brix<br>2: TC<br>3: INPM | 0                         | E                 | S / Man        | Selection of engineering unit for density.   |
| 10  | GRANT_DENY     | DS-70                 |  | 0                         | na                | D              |  |
| 11  | STATUS_OPTS    | Bitstring(2)          | See Block Options                              | 0                         | Na                | S / O/S        | See Block Options  |
| 12  | IN_1           | DS-65                 |  |                           |                   | D              | The first input (pressure P1) must have the same engineering units of IN_2.  |
| 13  | IN_2           | DS-65                 |  |                           |                   | D              | The second input (pressure P2) must have the same engineering units of IN_1.   |
| 14  | IN_3           | DS-65                 |  |                           |                   | D              | The third input. (Temperature T in Degrees Celsius)  |
| 15  | PRESSURE_UNITS | Unsigned16            | See valid pressure units                       | 1144 (g/cm <sup>2</sup> ) | E                 | S              | This is the pressure engineering unit of IN_1 and IN_2.  |
| 16  | HEIGHT         | Float                 | Positive                                       | 1000.0                    | mm                | S              | Distance between the two pressure transmitters. The engineering units must be compatible to the inputs IN_1 and IN_2. If it is mmH2O, the EU of HEIGHT is mm.  |
| 17  | GRAVITY        | Float                 | Positive                                       | 9.80665                   | m/s <sup>2</sup>  | S              | The gravity acceleration used in the density calculation, the EU of GRAVITY is m/s <sup>2</sup> .  |
| 18  | NUM_SAMPLES    | Unsigned16            | 1-1000   | 10                        | Na                | S              | Number of samples.   |
| 19  | NUM_AVERAGES   | Unsigned16            | 1-30   | 10                        | Na                | S              | Number of averages in the round buffer.  |
| 20  | DENS_OUT       | DS-65                 |  |                           | g/cm <sup>3</sup> | D / Man        | The density compensated by the temperature.  |
| 21  | DENS_UNITS     | Unsigned16            |  | 1100                      | g/cm <sup>3</sup> | S / RO         | This is the engineering unit of DENS_OUT that is fixed in g/cm <sup>3</sup> .  |
| 22  | OUT_D          | DS-66                 |  |                           |                   | D              | This is a discrete output parameter to indicate alarm state.   |
| 23  | RESET_IN       | DS-66                 |  |                           |                   | D              | When it is true, it resets the average calculation and it clears the round buffer of averages.   |
| 24  | OUT_HI_LIM     | Float                 |  | 100                       | OUT               | S              | High limit for OUT.  |
| 25  | OUT_LO_LIM     | Float                 |  | 0                         | OUT               | S              | Low limit for OUT.   |
| 26  | UPDATE_EVT     | DS-73                 |  |                           | Na                | D              | This alert is generated by any change to the static data.  |
| 27  | BLOCK_ALM      | DS-72                 |  |                           | Na                | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 28  | ALARM_SUM      | DS-74                 |  |                           | Na                | S              | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 29  | ACK_OPTION     | Bitstring(2)          | 0: Auto ACK Disable<br>1: Auto ACK Enable      | 0                         | Na                | S              | Selection of whether alarms associated with the block will be automatically acknowledged   |

| Idx | Parameter | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store/<br>Mode | Description   |
|-----|-----------|-----------------------|-------------------------|------------------|-------|----------------|---|
| 30  | ALARM_HYS | Float                 | 0 to 50 %               | 0.5%             | %     | S              | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis. |
| 31  | HI_PRI    | Unsigned8             | 0 to 15                 |                  |       | S              | Priority of the high alarm.   |
| 32  | HI_LIM    | Float                 | OUT_SCALE, +INF         | +INF             | OUT   | S              | The setting for high alarm in engineering units.  |
| 33  | LO_PRI    | Unsigned8             | 0 to 15                 |                  |       | S              | Priority of the low alarm.  |
| 34  | LO_LIM    | Float                 | OUT_SCALE, -INF         | -INF             | OUT   | S              | The setting for low alarm in engineering units.   |
| 35  | HI_ALM    | DS-71                 |                         |                  | OUT   | D              | The status for high alarm and its associated time stamp.  |
| 36  | LO_ALM    | DS-71                 |                         |                  | OUT   | D              | The status for low alarm and its associated time stamp.   |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is “Adapted”:**

The required mode for writing is the actual mode, regardless the target mode : OUT, DENS\_OUT,

## CTRW – Constant and Contained RW

### Overview

The Constant function block generates constant values to use in input parameters of other blocks. It can also read/write in contained parameters of other blocks into the same device.

### Description

The Constant function block has two functions:

- As Constant:

It generates constant values to use in input parameters of other blocks. This block allows up to 6 discrete constant and 6 analogical constant.

- As Contained RW:

The input writes in contained parameters of any other block into the same device.

The output reads data of the contained parameters of any other block into the same device.

#### ➤ Generating Constant Values for other Blocks

This block can generate up to six discrete outputs and 6 analogic outputs, where:

- The OUT\_1 to OUT\_4 and OUT\_D1 to OUT\_D4 outputs become available in the output the constant values of CT\_VALxx and CT\_STATUS\_xx.
- The OUT\_5, OUT\_6, OUT\_D5 and OUT\_D6 outputs have the feature Reading of Contained Parameter of other blocks and are associated with the LOC\_OUT\_xx parameter. If these parameters are not configured, the output will show the CT\_VAL\_xx and CT\_STATUS\_xx constant values.

If the Mode is Man, it is allowed the manual substitution of all output values. In the Auto mode, the output values are the respective constant values.

#### ➤ Reading or Writing from/to Contained Variables of other blocks

The writing and/or reading in the contained parameters of other blocks is associated to the block execution in the Auto mode.

##### ○ Input Treatment – Writing in Contained Parameters

During the block execution in the Auto mode, the block writes the value of the IN\_xx parameter in a parameter<sup>1</sup> of any other block in the same device.

The value will be written in the parameter configured in the following situations:

- While the value had a usable value in the IN\_xx input (i.e., IN\_xx Status was Good or Uncertain and the option "Use Uncertain as Good" in STATUS\_OPTS was set).
- The DISABLE\_IN\_xx input is usable and with FALSE value, or when the IN\_xx input was with an usable value.
- For the IN\_Dx inputs will just be done the writing when the actual value was different of the previous. This avoids cyclical writing in static parameters which causes a cyclical increment of the ST\_REV and event generation by the UPD\_EVT.
- For the IN\_x inputs will just be done the writing when the actual value was upper or lower than DEAD\_BAND\_x. In this range would not have writing in the parameter. When the DEAD\_BAND\_X parameter is equal zero, it means continued writing.

If the writing does not accomplish in the desired block, the input with failure will be indicated in the BAD\_STATUS parameter.

##### ○ Output Treatment – Reading of the Contained Parameters

During the block execution in the Auto mode, the block reads the parameter<sup>1</sup> value of any other block in the same device and become available in the OUT\_xx.VALUE parameter (where xx is only for the OUT\_5, OUT\_6, OUT\_D5 and OUT\_D6 outputs). The Output status will be GoodNonCascade in this case.

If the block does not read the parameter, the correspondent output with failure will be indicated in the BAD\_STATUS parameter, and also OUT\_xx.STATUS = Bad No Comm.

When BLOCK\_TAG\_OUT\_xx parameter is not configured (BLOCK\_TAG = Spaces), then the OUT\_xx output becomes available the corresponding constant value and status (CT\_VAL\_xx and CT\_STATUS\_xx).

○ **Configuration of Contained Parameter to be Read/Write**

To address the contained parameter, the respective input or output will have a LOC\_xxx parameter composed of the following structure (DS-262):

| Element Name       | Data Type         | Size | Description  |
|--------------------|-------------------|------|--|
| BLOCK_TAG_xxx      | VisibleString(32) | 32   | Block Tag for monitoring (this tag is case sensitive).   |
| INDEX_RELATIVE_xxx | Unsigned16        | 2    | Relative Index of parameter.   |
| SUB_INDEX_xxx      | Unsigned8         | 1    | Parameter subindex started by 1. When the parameter is a structure (DS_yyy), it indicates the number of the structure element.<br>When it was a Bitstring parameter, it indicates the byte of the parameter to be considered.<br>In simple parameters, this sub index is not considered. |

In the table below, xxx means the selected input or output (IN1,IN2,IN\_D1,etc). For example, to address IN\_1, the configuration parameters are BLOCK\_TAG\_IN1, INDEX\_RELATIVE\_IN1, SUB\_INDEX\_IN1.

The Input/Output is considered **NOT CONFIGURED** when **BLOCK\_TAG\_xxx** is **blank** or **INDEX\_RELATIVE\_xxx** and SUB\_INDEX\_xxx are equals to **zero**.

When there was a configuration error in the contained parameter, the correspondent bit of the input/output in CONFIG\_STATUS will be set and the BLOCK\_ERR will indicate "Configuration Error".

<sup>1</sup>The reading or writing is not supported in all parameter types. The table below summarizes the operations supported by the block:

| Parameter Input/Output | Data Type  | Direction             |                       | Data Type of/ Parameter of Other Supported Block   |
|------------------------|------------|-----------------------|-----------------------|--|
|                        |            | From                  | To                    |  |
| IN_x                   | Float      | In_x                  | Other Block Parameter | Boolean *<br>Float<br>Integer8 *<br>Integer16 *<br>Integer32<br>Unsigned8 *<br>Unsigned16 *<br>Unsigned32<br>Bitstring * |
| In_Dx                  | Unsigned8  | In_Dx                 | Other Block Parameter | Boolean<br>Float<br>Integer8<br>Integer16<br>Integer32<br>Unsigned8<br>Unsigned16<br>Unsigned32<br>Bitstring *           |
| Out_x                  | Float      | Other Block Parameter | Out_x                 | Boolean<br>Float<br>Integer8<br>Integer16<br>Integer32<br>Unsigned8<br>Unsigned16<br>Unsigned32<br>Bitstring             |
| Out_Dx                 | Unsigned8* | Other Block Parameter | Out_Dx                | Boolean<br>Float<br>Integer8<br>Integer16<br>Integer32<br>Unsigned8<br>Unsigned16<br>Unsigned32<br>Bitstring             |

**Note 1:** The data type assigned with \* means the block will cut the values out

of range of the correspondent type:

- Unsigned 8 / Bitstring – 0 to 255
- Integer 8 – (-127) a (+127)
- Boolean – 0 e 1
- Unsigned16 – 0 to 65535
- Integer16 – (-32767) to (+32767)

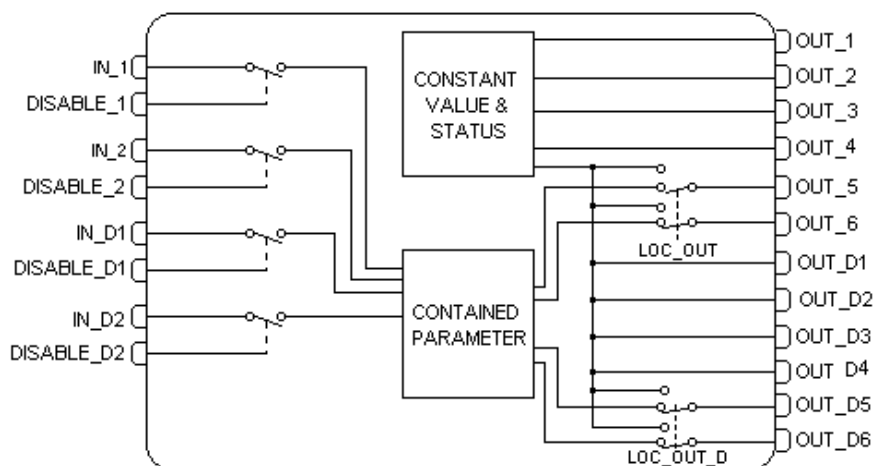
**Note 2:** For the BitString type, the Subindex identifies the correspondent Byte.

**Note 3:** The reading or writing in contained parameters are not supported in the same block Constant.

### Supported modes

O/S, MAN, AUTO

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description   |
|-----|-----------|--------------------|----------------------|---------------|-------|--------------|---|
| 1   | ST_REV    | Unsigned16         |                      | 0             | None  | S/RO         |   |
| 2   | TAG_DESC  | OctString(32)      |                      | Spaces        | Na    | S            |   |
| 3   | STRATEGY  | Unsigned16         |                      | 0             | None  | S            |   |
| 4   | ALERT_KEY | Unsigned8          | 1 to 255             | 0             | None  | S            |   |
| 5   | MODE_BLK  | DS-69              |                      | O/S           | Na    | S            | See Mode Parameter  |
| 6   | BLOCK_ERR | Bitstring(2)       |                      |               | E     | D/RO         |   |
| 7   | OUT_1     | DS-65              |                      |               |       | N / Man      | This output is determined by the constant CT_VAL_1. It can not be used to read contained parameters.                                    |
| 8   | OUT_2     | DS-65              |                      |               |       | D / Man      | This output is determined by the constant CT_VAL_2. It can not be used to read contained parameters.                                    |
| 9   | OUT_3     | DS-65              |                      |               |       | D / Man      | This output is determined by the constant CT_VAL_3. It can not be used to read contained parameters.                                    |
| 10  | OUT_4     | DS-65              |                      |               |       | D / Man      | This output is determined by the constant CT_VAL_4. It can not be used to read contained parameters.                                    |
| 11  | OUT_5     | DS-65              |                      |               |       | D / Man      | This output can have the value of a constant CT_VAL_5 or the value of a contained parameter from another block depends on BLK_TAG_OUT5. |

| Idx | Parameter    | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|--------------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 12  | OUT_6        | DS-65                 |                         |                  |       | D / Man         | This output can have the value of a constant CT_VAL_6 or the value of a contained parameter from another block depends on BLK_TAG_OUT6.  |
| 13  | OUT_D1       | DS-66                 |                         |                  |       | N / Man         | This output is determined by the constant CT_VAL_D1. It can not be used to read contained parameters.  |
| 14  | OUT_D2       | DS-66                 |                         |                  |       | D / Man         | This output is determined by the constant CT_VAL_D2. It can not be used to read contained parameters.  |
| 15  | CT_VAL_1     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_1.   |
| 16  | CT_VAL_2     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_2.   |
| 17  | CT_VAL_3     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_3.   |
| 18  | CT_VAL_4     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_4.   |
| 19  | CT_VAL_5     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_5.   |
| 20  | CT_VAL_6     | Float                 |                         | 0                |       | S               | Analog constant value transferred to the output OUT_6.   |
| 21  | CT_VAL_D1    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D1.  |
| 22  | CT_VAL_D2    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D2.  |
| 23  | UPDATE_EVT   | DS-73                 |                         |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 24  | BLOCK_ALM    | DS-72                 |                         |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 25  | CT_VAL_D3    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D3.  |
| 26  | CT_VAL_D4    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D4.  |
| 27  | CT_VAL_D5    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D5.  |
| 28  | CT_VAL_D6    | Unsigned8             |                         | 0                |       | S               | Discrete constant value transferred to the output OUT_D6.  |
| 29  | CT_STATUS_1  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_1.   |
| 30  | CT_STATUS_2  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_2.   |
| 31  | CT_STATUS_3  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_3.   |
| 32  | CT_STATUS_4  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_4.   |
| 33  | CT_STATUS_5  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_5.   |
| 34  | CT_STATUS_6  | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_6.   |
| 35  | CT_STATUS_D1 | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_D1.  |
| 36  | CT_STATUS_D2 | Unsigned8             |                         | GNC              |       | S               | Constant Status transferred to the output OUT_D2.  |

| Idx | Parameter            | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description   |
|-----|----------------------|--------------------|----------------------|---------------|-------|--------------|---|
| 37  | CT_STATUS_D3         | Unsigned8          |                      | GNC           |       | S            | Constant Status transferred to the output OUT_D3.   |
| 38  | CT_STATUS_D4         | Unsigned8          |                      | GNC           |       | S            | Constant Status transferred to the output OUT_D4.   |
| 39  | CT_STATUS_D5         | Unsigned8          |                      | GNC           |       | S            | Constant Status transferred to the output OUT_D5.   |
| 40  | CT_STATUS_D6         | Unsigned8          |                      | GNC           |       | S            | Constant Status transferred to the output OUT_D6.   |
| 41  | OUT_D3               | DS-66              |                      |               |       | D/Man        | This output is determined by the constant CT_VAL_D3. It can not be used to read contained parameters.                             |
| 42  | OUT_D4               | DS-66              |                      |               |       | D/Man        | This output is determined by the constant CT_VAL_D4. It can not be used to read contained parameters.                             |
| 43  | OUT_D5               | DS-66              |                      |               |       | D/Man        | This output can have the value of a CT_VAL_D5 or the value of a contained parameter from another block depends on BLK_TAG_OUT_D5. |
| 44  | OUT_D6               | DS-66              |                      |               |       | D/Man        | This output can have the value of a CT_VAL_D6 or the value of a contained parameter from another block depends on BLK_TAG_OUT_D6. |
| 45  | IN_1                 | DS-65              |                      |               |       | D            | Analog input that writes to a contained parameter configured in BLK_TAG_IN1.  |
| 46  | DISABLE_1            | DS-66              |                      |               |       | D            | Disable the IN_1 writing in the correspondent contained parameter.  |
| 47  | BLK_TAG_IN1          | Visible(32)        |                      | Blank         |       | S / OOS      | Block tag that will be written by IN_1.   |
| 48  | INDEX_RELATIVE_IN1   | Unsigned16         |                      | 0             |       | S / OOS      | Relative index of the block parameter that will be written by IN_1.   |
| 49  | SUB_INDEX_IN1        | Unsigned8          |                      | 0             |       | S / OOS      | Subindex of the block parameter that will be written by IN_1.   |
| 50  | DEAD_BAND_1          | Float              |                      | 1.0           | IN_1  | S            | Dead band for the IN_1 input where the variation into this range would not cause writing in the block parameter.                  |
| 51  | IN_2                 | DS-66              |                      |               |       | D            | Analog input that writes to a contained parameter configured in BLK_TAG_IN2.  |
| 52  | DISABLE_2            | DS-66              |                      |               |       | D            | Disable the IN_2 writing in the correspondent contained parameter.  |
| 53  | BLK_TAG_IN2          | Visible(32)        |                      | Blank         |       | S / OOS      | Block tag that will be written by IN_2.   |
| 54  | INDEX_RELATIVE_IN2   | Unsigned16         |                      | 0             |       | S / OOS      | Relative index of the block parameter that will be written by IN_2.   |
| 55  | SUB_INDEX_IN2        | Unsigned8          |                      | 0             |       | S / OOS      | Subindex of the block parameter that will be written by IN_2.   |
| 56  | DEAD_BAND_2          | Float              |                      | 1.0           | IN_2  | S            | Dead band for the IN_2 input where the variation into this range would not cause writing in the block parameter.                  |
| 57  | IN_D_1               | DS-66              |                      |               |       | D            | Discrete input that writes to a contained parameter configured in BLK_TAG_IN_D1.  |
| 58  | DISABLE_D1           | DS-66              |                      |               |       | D            | Disable the IN_D1 writing in the correspondent contained parameter  |
| 59  | BLK_TAG_IN_D1        | Visible(32)        |                      | Blank         |       | S / OOS      | Block tag that will be written by IN_D_1.   |
| 60  | INDEX_RELATIVE_IN_D1 | Uns16              |                      | 0             |       | S / OOS      | Relative index of the block parameter that will be written by IN_D_1.   |
| 61  | SUB_INDEX_IN_D_1     | Uns8               |                      | 0             |       | S / OOS      | Subindex of the block parameter that will be written by IN_D_1.   |
| 62  | IN_D_2               | DS-66              |                      |               |       | D            | Discrete input that writes to a contained parameter configured in BLK_TAG_IN_D2.  |
| 63  | DISABLE_D2           | DS-66              |                      |               |       | D            | Disable the IN_D2 writing in the correspondent contained parameter  |
| 64  | BLK_TAG_IN_D2        | Visible(32)        |                      | Blank         |       | S / OOS      | Block tag that will be written by IN_D_2.   |
| 65  | INDEX_RELATIVE_IN_D2 | Uns16              |                      | 0             |       | S / OOS      | Relative index of the block parameter that will be written by IN_D_2.   |

| Idx | Parameter                 | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description   |
|-----|---------------------------|-----------------------|-------------------------|------------------|-------|-----------------|---|
| 66  | SUB_INDEX_IN_D<br>2       | Uns8                  |                         | 0                |       | S / OOS         | Subindex of the block parameter that will be written by IN_D_2.   |
| 67  | BLK_TAG_OUT5              | Visible(32)           |                         | Blank            |       | S / OOS         | Block tag that will be written by OUT_5.  |
| 68  | INDEX_RELATIVE<br>_OUT5   | Uns16                 |                         | 0                |       | S / OOS         | Relative index of the block parameter that will be written by OUT_5.  |
| 69  | SUB_INDEX_OUT<br>5        | Uns8                  |                         | 0                |       | S / OOS         | Subindex of the block parameter that will be written OUT_5.   |
| 70  | BLK_TAG_OUT6              | Visible(32)           |                         | Blank            |       | S / OOS         | Block tag that will be written by OUT_6.  |
| 71  | INDEX_RELATIVE<br>_OUT6   | Uns16                 |                         | 0                |       | S / OOS         | Relative index of the block parameter that will be written by OUT_6.  |
| 72  | SUB_INDEX_OUT<br>6        | Uns8                  |                         | 0                |       | S / OOS         | Subindex of the block parameter that will be written OUT_6.   |
| 73  | BLK_TAG_OUT_D<br>5        | Visible(32)           |                         | Blank            |       | S / OOS         | Block tag that will be written by OUT_D5.   |
| 74  | INDEX_RELATIVE<br>_OUT_D5 | Uns16                 |                         | 0                |       | S / OOS         | Relative index of the block parameter that will be written by OUT_D5.   |
| 75  | SUB_INDEX_OUT<br>_D5      | Uns8                  |                         | 0                |       | S / OOS         | Subindex of the block parameter that will be written OUT_D6.  |
| 76  | BLK_TAG_OUT_D<br>6        | Visible(32)           |                         | Blank            |       | S / OOS         | Block tag that will be written by OUT_D6.   |
| 77  | INDEX_RELATIVE<br>_OUT_D6 | Uns16                 |                         | 0                |       | S / OOS         | Relative index of the block parameter that will be written by OUT_D6.   |
| 78  | SUB_INDEX_OUT<br>_D6      | Uns8                  |                         | 0                |       | S / OOS         | Subindex of the block parameter that will be written OUT_D6.  |
| 79  | BAD_STATUS                | Bitstring(2)          |                         |                  |       | D/RO            | Writing/reading status of the parameter. The set bit indicates the algorithm does not get read/write the specified "Block.Parameter". |
| 80  | CONFIG_STATUS             | Bitstring(2)          |                         |                  |       | D/RO            | Indicates which input or output is with configuration error.  |
| 81  | STATUS_OPTS               | Bitstring(2)          |                         |                  |       | S/ OOS          | See the parameter description in "Function Block Options".  |

**Legend:** *E* – Enumerated parameter; *Na* – Dimensionless parameter; *RO* – Read only; *D* – dynamic; *N* – non-volatile; *S* - static

**Gray Background Line: Default Parameters of Syscon**

**If DIAG.BEHAVIOR parameter is "Adapted":**

The required mode for writing is the actual mode, regardless the target mode: OUT\_1, OUT\_2, OUT\_3, OUT\_4, OUT\_5, OUT\_6, OUT\_D1 and OUT\_D2.

**BAD\_STATUS and CONFIG\_STATUS Bitstring**

| Bit | Description |
|-----|-------------|
| 0   | IN_1        |
| 1   | IN_2        |
| 2   | IN_D1       |
| 3   | IN_D2       |
| 4   | OUT_5       |
| 5   | OUT_6       |
| 6   | OUT_D5      |
| 7   | OUT_D6      |



## FFET - Flip-Flop and Edge Trigger

### Overview

It can be configured to work as:

- SR flip-flop
- RS flip-flop
- D-latch
- Rising edge trigger
- Falling edge trigger
- Bi-directional edge trigger

### Description

The following tables summarize the block behavior:

| RESET_IN | SET_IN | OUT_D (SR flip-flop) | OUT_D (RS flip-flop) |
|----------|--------|----------------------|----------------------|
| L        | L      | $Q_{n-1}$            | $Q_{n-1}$            |
| H        | L      | L                    | L                    |
| L        | H      | H                    | H                    |
| H        | H      | H                    | L                    |

| RESET_IN | SET_IN | IN_D1          | IN_D2 | OUT_D (D-latch) |
|----------|--------|----------------|-------|-----------------|
| L        | H      | X              | X     | H               |
| H        | L      | X              | X     | L               |
| H        | H      | X              | X     | H               |
| L        | L      | Rising         | L     | L               |
| L        | L      | Rising         | H     | H               |
| L        | L      | H,L or Falling | X     | $OUT\_D_{n-1}$  |

| RESET_IN | SET_IN | IN_D1         | OUT_D (Rising Edge) | OUT_D (Falling Edge) | OUT_D (Bi-directional) |
|----------|--------|---------------|---------------------|----------------------|------------------------|
| L        | L      | Rising        | H                   | L                    | H                      |
| L        | L      | Falling       | L                   | H                    | H                      |
| L        | L      | No transition | L                   | L                    | L                      |
| X        | H      | X             | H                   | H                    | H                      |
| H        | L      | X             | L                   | L                    | L                      |

### BLOCK\_ERR

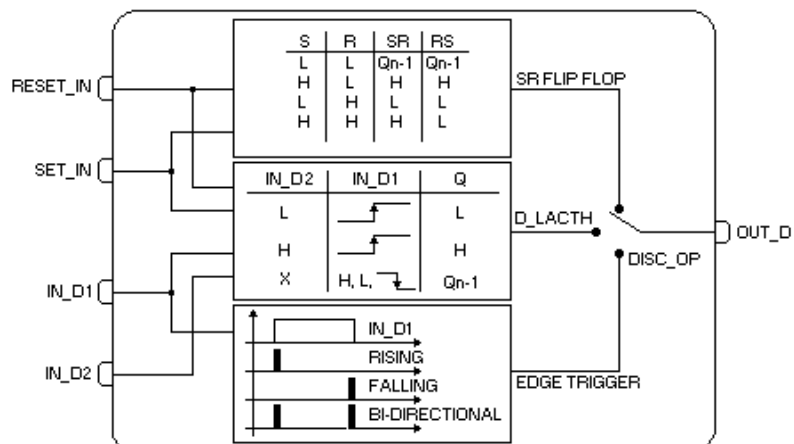
The BLOCK\_ERR of the FFET block will reflect the following causes:

- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S, MAN and AUTO.

### Schematic



## Parameters

| Idx | Parameter   | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-------------|-----------------------|---|------------------|-------|-----------------|--|
| 1   | ST_REV      | Unsigned16            |   | 0                | None  | S/RO            |  |
| 2   | TAG_DESC    | OctString(32)         |   | Spaces           | Na    | S               |  |
| 3   | STRATEGY    | Unsigned16            |   | 0                | None  | S               |  |
| 4   | ALERT_KEY   | Unsigned8             | 1 to 255  | 0                | None  | S               |  |
| 5   | MODE_BLK    | DS-69                 |   | O/S              | Na    | S               | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)          |   |                  | E     | D / RO          |  |
| 7   | DISC_OP     | Unsigned8             | 0 : SR flip-flop<br>1 : RS flip-flop<br>2 : D-latch<br>3 : rising edge<br>4 : falling edge<br>5 : bi-directional edge |                  | E     | S / Man         | Selection of discrete operation.   |
| 8   | STATUS_OPTS | Bitstring(2)          |   |                  |       | S / O/S         |  |
| 9   | IN_D1       | DS-66                 |   |                  |       | D               | Numbered discrete input 1.   |
| 10  | IN_D2       | DS-66                 |   |                  |       | D               | Numbered discrete input 2.   |
| 11  | SET_IN      | DS-66                 | 0 : Off<br>1 : set  |                  |       | D               | The set input.   |
| 12  | RESET_IN    | DS-66                 | 0 : Off<br>1 : reset  |                  |       | D               | The reset input.   |
| 13  | OUT_D       | DS-66                 |   |                  |       | N / Man         | The output of flip-flop.   |
| 14  | UPDATE_EVT  | DS-73                 |   |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 15  | BLOCK_ALM   | DS-72                 |   |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

If DIAG.BEHAVIOR parameter is “Adapted”:

The required mode for writing is the actual mode, regardless the target mode : OUT\_D.

## AEQU – Advanced Equations

### Overview

This block was specially designed to support specific calculations.

### Description

As this block has a general purpose, some parameters may not be used by a selected equation. It follows a description of each equation type, as well the meaning of used parameters:

| Parameter                    | Description   | Inputs  | Configurable Parameters | Outputs  |
|------------------------------|---|---|-------------------------|--|
| <b>Ln x</b>                  | It calculates the natural logarithm   | <b>IN_1:</b> input to the function, x   | None                    | <b>OUT:</b> result of natural logarithm  |
| <b>Log x</b>                 | It calculates base 10 logarithm   | <b>IN_1:</b> input to the function, x   | None                    | <b>OUT:</b> result of base 10 logarithm  |
| <b>Exp x</b>                 | It calculates e to power of x   | <b>IN_1:</b> input to the function, x   | None                    | <b>OUT:</b> result of e to power of x  |
| <b>Dew point temperature</b> | It calculates the dew point temperature, water vapor saturation pressure (psia) and water vapor pressure (pw) | <b>IN_1:</b> dry bulb temperature (F)<br><b>IN_2:</b> relative humidity (percent) | None                    | <b>OUT:</b> dew point temperature<br><b>OUT_1:</b> water vapor saturation pressure (psia)<br><b>OUT_2:</b> water vapor pressure (pw)<br><b>Especial:</b> Reserved option |

### BLOCK\_ERR

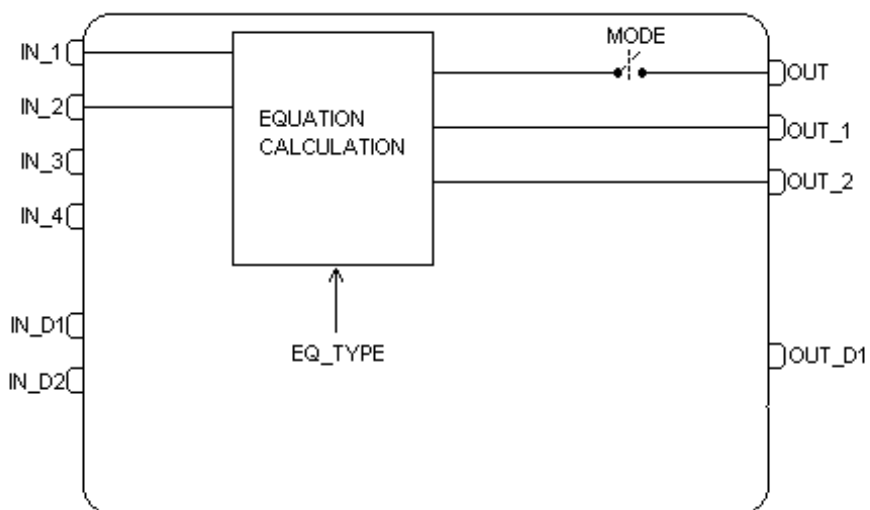
The BLOCK\_ERR of the AEQU block will reflect the following causes:

- Out of Service – When the block is in O/S mode.
- Block Configuration Error – When an abnormal result occurs (+/- INF, NaN).

### Supported modes

O/S, MAN, AUTO.

### Schematic



## Parameters

| Idx | Parameter  | Data Type<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|------------|-----------------------|---|------------------|-------|-----------------|--|
| 1   | ST_REV     | Unsigned16            |   | 0                | None  | S/RO            |  |
| 2   | TAG_DESC   | OctString(32)         |   | Spaces           | Na    | S               |  |
| 3   | STRATEGY   | Unsigned16            |   | 0                | None  | S               |  |
| 4   | ALERT_KEY  | Unsigned8             | 1 to 255  | 0                | None  | S               |  |
| 5   | MODE_BLK   | DS-69                 |   | O/S              | Na    | S               | See Mode Parameter   |
| 6   | BLOCK_ERR  | Bitstring(2)          |   |                  | E     | D/RO            |  |
| 7   | EQ_TYPE    | Unsigned8             | 0 : Ln x<br>1 : Log x<br>2 : Exp x<br>3 : Dew Point<br>Temperature<br>255 : Special | 0                | E     | S / Man         | Equation type  |
| 8   | IN_1       | DS-65                 |   |                  |       | D               | Numbered input 1.  |
| 9   | IN_2       | DS-65                 |   |                  |       | D               | Numbered input 2.  |
| 10  | IN_3       | DS-65                 |   |                  |       | D               | Numbered input 3.  |
| 11  | IN_4       | DS-65                 |   |                  |       | D               | Numbered input 4.  |
| 12  | IN_D1      | DS-66                 |   |                  |       | D               | Numbered discrete input 1.   |
| 13  | IN_D2      | DS-66                 |   |                  |       | D               | Numbered discrete input 2.   |
| 14  | OUT        | DS-65                 |   |                  |       | D / Man         | Primary output.  |
| 15  | OUT_D1     | DS-66                 |   |                  |       | D / Man         | Numbered discrete output 1.  |
| 16  | OUT_1      | DS-65                 |   |                  |       | D / RO          | Numbered output 1.   |
| 17  | OUT_2      | DS-65                 |   |                  |       | D / RO          | Numbered output 2.   |
| 18  | CT_VAL_1   | Float                 |   | 0                |       | S               | Constant value 1.  |
| 19  | CT_VAL_2   | Float                 |   | 0                |       | S               | Constant value 2.  |
| 20  | CT_VAL_3   | Float                 |   | 0                |       | S               | Constant value 3.  |
| 21  | CT_VAL_4   | Float                 |   | 0                |       | S               | Constant value 4.  |
| 22  | CT_VAL_5   | Float                 |   | 0                |       | S               | Constant value 5.  |
| 23  | CT_VAL_6   | Float                 |   | 0                |       | S               | Constant value 6.  |
| 24  | CT_VAL_D1  | Unsigned8             |   | 0                |       | S               | Integer constant value 1.  |
| 25  | CT_VAL_D2  | Unsigned8             |   | 0                |       | S               | Integer constant value 2.  |
| 26  | OUT_HI_LIM | Float                 |   | 100              | OUT   | S               | High limit for OUT.  |
| 27  | OUT_LO_LIM | Float                 |   | 0                | OUT   | S               | Low limit for OUT.   |
| 28  | BLOCK_ALM  | DS-72                 |   |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

## PRED – Smith Predictor

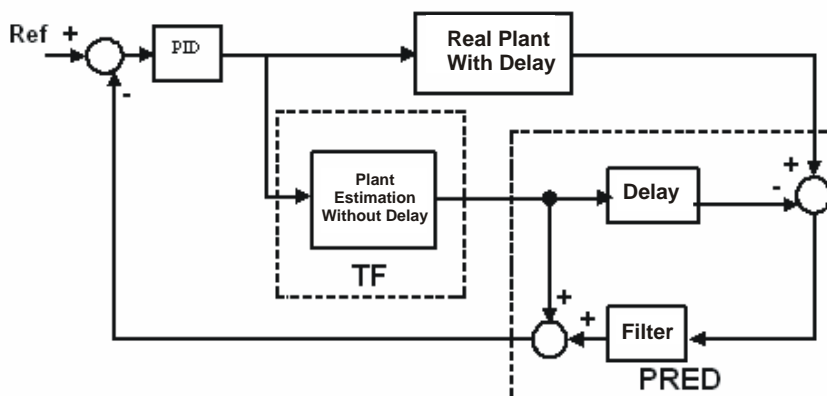
### Description

The delay is the main part of several industrial processes. For systems with low delay, the PID controls the process through setting the parameters properly. The derivative term must be set correctly because it represents the linear prediction.

When the delay is upper than two times of the time constant of the process, PID has limited action, and in many situations it is not possible to control the process in order to obtain responses more quickly when comparing closed loops and opened loops. In these situations, another control type or with delay compensation should be selected.

The PRED block, with TF block (Transfer Function) allow to the user develops a control strategy using PID blocks in order to control systems with considerable delay times.

The Predictor block is an implementation of the scheme below:



**Scheme using Smith Predictor**

This block has three inputs and one output:

- **In\_1** – Input for the Real system with delay.
- **In\_2** – Input for the Plant Estimation without delay.
- **Delay Time** – Input providing the delay value, in seconds. This value can not be higher than the *Sample time*.
- **Out** – Block configurable output.

The block output can be configured by three different ways, according to the PRED\_SELECTOR parameter:

- **Bypass** – Output will have the same value of the IN\_1 input.
- **Delay** – Block will have only the delay function, if the signal was received by IN\_2 input.
- **Smith Predictor** – Block will have the Smith Predictor function.

Optionally, a filter can be used (as showed in the picture above). The filter provides robustness to the systems with errors in high frequency between the profile and the real process. The filter has the following configuration:

$$G(s) = \frac{1}{Ts + 1}$$

In the equation above, T coefficient is configured by user through the FILTER parameter.

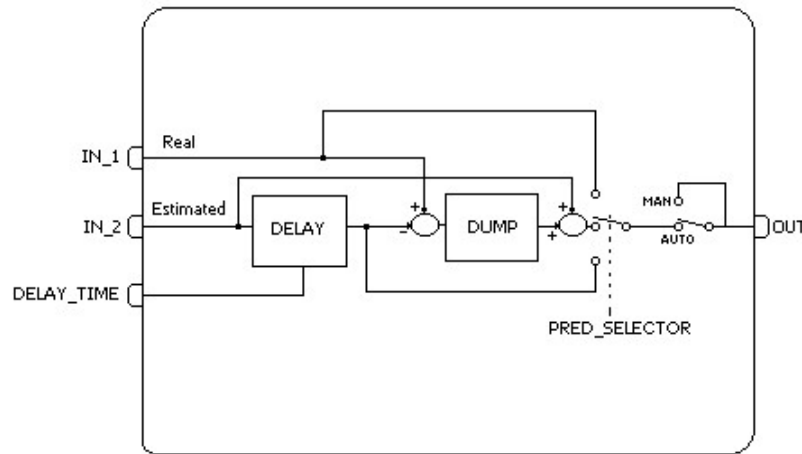
### OS → Auto Transition

When a block transition from OS to Auto occurs and the block is configured in Smith Predictor, this block will need a time (the same of the DELAY\_TIME) to store data, while this the block will work in the Bypass option.

### Supported Modes

O/S, MAN and AUTO.

### Schematic



### Parameters

| Idx | Parameter        | Data type (length) | Valid Range / Options                          | Default Value | Unids | Store /Mode | Description   |
|-----|------------------|--------------------|--|---------------|-------|-------------|---|
| 1   | ST_REV           | Unsigned16         |  | 0             | None  | S           |   |
| 2   | TAG_DESC         | OctString(32)      |  | Spaces        | Na    | S           |   |
| 3   | STRATEGY         | Unsigned16         |  | 0             | None  | S           |   |
| 4   | ALERT_KEY        | Unsigned8          | 1 to 255                                       | 0             | None  | S           |   |
| 5   | MODE_BLK         | DS-69              | O/S,MAN,AUTO                                   | O/S           | Na    | S           | See Mode parameter.   |
| 6   | BLOCK_ERR        | Bitstring(2)       |  |               | E     | D           |   |
| 7   | IN_1             | DS-65              |  |               |       | D           | Input for the Real system with delay.   |
| 8   | IN_2             | DS-65              |  |               |       | D           | Input for the Plant Estimation without delay.   |
| 9   | DELAY_TIME       | DS-65              |  |               | Sec   | D           | Input providing the delay value, in seconds.  |
| 10  | OUT              | DS-65              |  |               |       | D / Man     | Block configurable output.  |
| 11  | PRED_SELECTOR    | Unsigned8          | 0 = Bypass<br>1 = Delay<br>2 = Smith Predictor | 0             | E     | S           | Output selector:<br><b>Bypass</b> – Output will have the same value of the IN_1 input;<br><b>Delay</b> – The output will be the signal of the IN_2 input with delay.<br><b>Smith Predictor</b> – The output will be according to the Smith Predictor algorithm. |
| 12  | FILTER           | float              | Positive                                       | 0             | Sec   | S           | Parameter to configure the block filter.  |
| 13  | PRED_SAMPLE_TIME | float              | Positive                                       | 1             | Sec   | S           | Sample time to collect data.  |
| 14  | STATUS_OPTS      | Bitstring(2)       | Bit 2 – Uncertain as Good                      | 0             | Na    | S / OOS     | See Block Option.   |
| 15  | UPDATE_EVT       | DS-73              |  |               | Na    | D           | This alert is generated by any change to the static data.   |
| 16  | BLOCK_ALM        | DS-72              |  |               | Na    | D           |   |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

Gray Background Line: Default Parameters of Syscon

## TF – Transfer Function

### Description

The Transfer Function block (TF) must represent systems up to 2nd order, through the configuration of the coefficients A, B, C, D, E and F of the transfer function as:

$$G(s) = \frac{As^2 + Bs + C}{Ds^2 + Es + F}$$

This block has one output and one input.

### System Representation

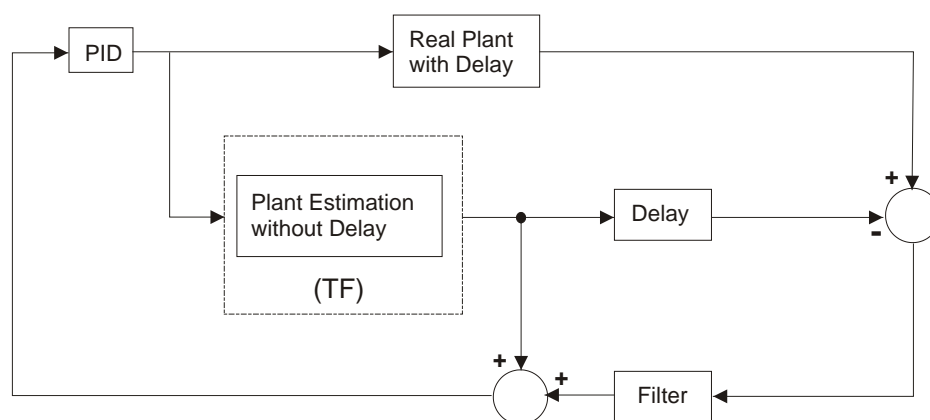
A system to be accomplished should have the denominator degree higher than the numerator degree, thus when D parameter was equal zero, A parameter must be zero too.

The parameters D and E can not be zero.

### TF Block Application

For systems with delay is difficult to control the system using only the PID controller. It is necessary to make a scheme that supplies to the PID the value of the process variable without delay. A scheme used is the *Predictor Smith* and its configuration is showed below:

To represent the plant estimation without delay, it is necessary to use the TF block.



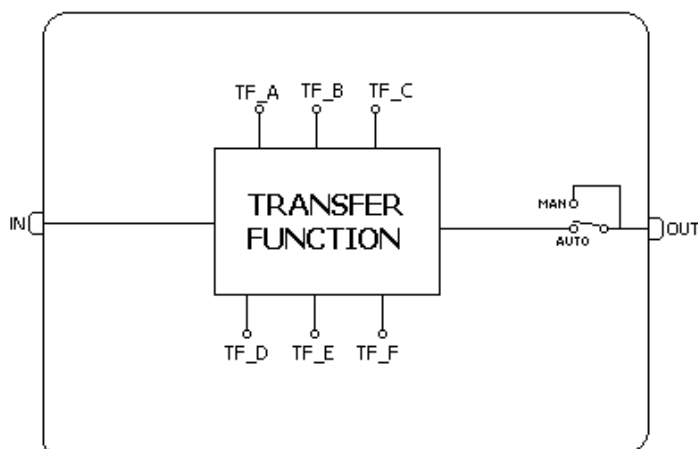
**Scheme using Smith Predictor**

Another application which the TF block can be used is using the Constant block to make multimodel systems. One example of this application type is the conic tank, where it is possible to determine its operation points and for each point is possible to configure different transfer functions.

### Supported Modes

O/S, MAN and AUTO.

## Schematic



## Parameters

| Idx | Parameter   | Data type (length) | Valid Range/Options       | Default Value | Units | Store/Mode | Description   |
|-----|-------------|--------------------|---------------------------|---------------|-------|------------|---|
| 1   | ST_REV      | Unsigned16         |                           | 0             | None  | S          |   |
| 2   | TAG_DESC    | OctString(32)      |                           | Spaces        | Na    | S          |   |
| 3   | STRATEGY    | Unsigned16         |                           | 0             | None  | S          |   |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255                  | 0             | None  | S          |   |
| 5   | MODE_BLK    | DS-69              | O/S,MAN,AUTO              | O/S           | Na    | S          | See Mode parameter.   |
| 6   | BLOCK_ERR   | Bitstring(2)       |                           |               | E     | D          |   |
| 7   | IN          | DS-65              |                           |               |       | D          | System analog input.  |
| 8   | OUT         | DS-65              |                           |               |       | D / Man    | System analog output.   |
| 9   | TF_A        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 10  | TF_B        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 11  | TF_C        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 12  | TF_D        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 13  | TF_E        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 14  | TF_F        | Float              |                           | 0.0           |       | S          | Transfer function coefficient.  |
| 15  | TF_BAL_TIME | Float              | Positive                  | 0.0           | Sec   | S          | Specifies the time from the last output value returning to the calculate algorithm value when the block is changing from Manual to Automatic Mode, providing bumpless start up. |
| 16  | STATUS_OPTS | Bitstring(2)       | Bit 2 : Uncertain as Good | 0             | Na    | S / OOS    | See Block options.  |
| 17  | UPDATE_EVT  | DS-73              |                           |               | Na    | D          | This alert is generated by any change to the static data.   |
| 18  | BLOCK_ALM   | DS-72              |                           |               | Na    | D          |   |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S - static

**Gray Background Line: Default Parameters of Syscon**



## LCF – Liquid Correction Factors

### Description

This block calculates the correction factors (CTL, CPL and BSW in operation temperature) for liquid measurement.

### Product Configuration

Using the PRODUCT\_TYPE parameter is possible to select the product type to be used in the calculation. There is also the selection of the density type in the DENSITY\_TYPE parameter, the application of the correction for the glass density meter (HYDROMETER\_CORRECTION) and the coefficient of Thermal Expansion in base temperature or equilibrium absolute pressure.

### CCF Calculation

If the TEMPERATURE\_METER and DENSITY\_METER inputs are connected, the CTL\_OIL factor will be calculated. And if the PRESSURE\_METER input is also connected, the CPL\_OIL factor will be also calculated.

If the CALL\_BSW parameter is configured to accomplish the BSW calculation, thus:

$$CCF = CTL * CPL * MF * (1 - X_{w,m}) * SF$$

### Block Inputs

| Input             | Link Necessity | Description  |
|-------------------|----------------|--|
| TEMPERATURE_METER | mandatory      | Liquid flow temperature<br>If the system has an online density meter, the temperature which the density measurement is being accomplished must be lower than the variation of the acceptable limits related to the flow temperature in the flow meter. |
| PRESSURE_METER    | Optional       | Flow manometric pressure<br>If this input is not connected, CPL = 1.   |
| DENSITY_METER     | mandatory      | Density of the measured product (emulsão), which can be in the flow conditions or in base conditions, it depends on the DENSITY_TYPE configuration   |

### Block Outputs

This block provides four outputs described below. In the applications which the CPL factor is not calculated and the PRESSURE\_METER input is not connected, thus the CPL\_OIL output will indicate 1.

| Output    | Description  | Value in the exception condition (*) |
|-----------|--|--------------------------------------|
| BASE_DENS | Density in base condition. It is defined by the LKD block. | DENSITY_METER                        |
| CTL_OIL   | Temperature Correction Factor                              | 1.0000                               |
| CPL_OIL   | Pressure Correction Factor                                 | 1.0000                               |
| CCF_OUT   | Combined Correction Factor                                 | 1.0000                               |

(\*) Situation which is impossible to accomplish the calculation. It can be because of the input status or it is out of range specified by the correspondent standard.

| TEMPERATURE_METER and DENSITY_METER | PRESSURE_METER | CALC_BSW                  | CCF   |
|-------------------------------------|----------------|---------------------------|---|
| No connected                        | -              | -                         | 1   |
| Connected                           | No connected   | None                      | CTL * MF                                      |
| Connected                           | Connected      | None                      | CTL * CPL * MF                                |
| Connected                           | No connected   | Dual range / Lab analysis | MF * (1 - X <sub>w,m</sub> ) * CTL * SF       |
| Connected                           | Connected      | Dual range / Lab analysis | MF * (1 - X <sub>w,m</sub> ) * CTL * CPL * SF |

The CCF\_OUT output results of the three factors multiplication (CTL, CPL and MF), if any calculation of these factors is impossible, thus the value in the exception conditions must be used.

#### Temperature Correction Factor for Liquid Hydrocarbon (CTL\_OIL)

The density used in the CTL calculation depends on the configuration, as indicated below:

| Density used  | Configuration  | Comments  |
|---------------|--|---|
| DENSITY_METER | There are two conditions: <ul style="list-style-type: none"> <li>CALC_BSW set to Dual range and LO_SW is 100%.</li> <li>CALC_BSW set to None.</li> </ul> | The dry oil density in operation or base conditions (configured in DENSITY_TYPE) indicated in DENSITY_METER input is used in the CTL_OIL calculation.             |
| LAB_DENS_OIL  | CALC_BSW set to Dual range and LO_SW is different of 100%.   | The DENSITY_METER input is the emulsion density (not compliant for the CTL_OIL calculation CTL_OIL), which is used for the DWS calculation in operation condition |
|               | CALC_BSW set to Lab analysis.  | The DENSITY_METER input is not used because the dry oil density must be obligatorily supplied for the BSW calculation.  |

For the measurement of crude oil, general products, MTBE and lubricant oil is used API-11.1 standard. For the measurement of light liquid hydrocarbon are used GPA-TP25 and GPA-TP15 standards.

#### Compressibility Factor – F

The Compressibility Factor for the measured liquid is calculated using the base density and flow temperature. If it is impossible to calculate the compressibility factor, the F parameter will be zero.

The CPL factor is calculated using compressibility factor, flow manometric pressure and equilibrium pressure.

If the selected product is water, the compressibility factor will be zero and the CPL factor is 1.

The standards used for the compressibility factor calculation are API-11.2.1 and API-11.2.1.M for crude oil, general products, MTBE and lubricant oil. For the measurement of light hydrocarbon liquid are used API-11.2.2 and API-11.2.2.M standards.

#### Meter factor – MF

If the meter is submitted to the proving, thus the meter factor value obtained must be written in the MF parameter. Otherwise, the default value of the MF parameter must be kept, that is, 1.

#### BSW Calculation– Dual range

If the CALC\_BSW parameter is set to “Dual range”, the FLOWING\_SW\_OUT parameter is the SW\_METER input, if this input was lower than LO\_SW. Otherwise, the BSW will be calculated using the result of laboratory analysis and emulsion density in flow condition.

If LO\_SW = 0.0, thus BSW will be always calculated

If LO\_SW = 100.0, thus the FLOW\_SW\_IN input will be always used.

**Block Inputs**

The inputs used are indicated below:

| Input             | Link Necessity | Description   |
|-------------------|----------------|---|
| TEMPERATURE_METER | mandatory      | Liquid flow temperature   |
| DENSITY_METER     | mandatory      | Density of the measured product which is must be in flow condition obligatorily for the Dual range option.  |
| SW_METER          | Optional       | Input with the value of the BSW measured online for values lower than the configured values in LO_SW parameter, up to the value of BSW will be used.<br>If this input is not connected, it will be ignored, thus the BSW will be always calculated. |

**Output Blocks**

| Outputs        | Description   |
|----------------|---|
| FLOWING_SW_OUT | If the value of the SW_METER input is lower than the LO_SW parameter, this output will follow the FLOW_SW_IN input. Otherwise, it will be calculated the value. |
| CTL_WATER      | Temperature correction factor for water.  |
| BASE_SW        | BSW calculated for base temperature condition   |

**BSW Calculation– LAB\_DENS\_OIL and LAB\_DENS\_WATER**

The BSW is calculated using the dry oil and water densities in laboratory analysis condition, the density in flow condition and the flow temperature. All factors are measured online.

These calculations assume constant features (base density) of the oil and water made.

**FLOWING\_SW\_OUT Calculation:**

- It calculates :  $DENS_{\text{óleo},T} = f(DENS_{\text{óleo},T_{\text{lab}}}, T)$

Where :

$DENS_{\text{óleo},T}$  : dry oil density in flow temperature

$DENS_{\text{óleo},T_{\text{lab}}} = \text{LAB\_DENS\_OIL}$  : dry oil density in laboratory analysis temperature

T : flow temperature

- It calculates :  $DENS_{\text{água},T} = f(DENS_{\text{água},15/60}, T)$

Where :

$DENS_{\text{água},T}$  : water density in flow temperature

$DENS_{\text{água},T_{\text{lab}}} = \text{LAB\_DENS\_WATER}$ : water temperature in laboratory analysis temperature

T : flow temperature

- It calculates  $BSW_T$  (FLOWING\_SW\_OUT parameter).

Where :

$BSW_T$  : BSW in flow temperature

$$BSW_T = \frac{DENS_{\text{emulsão},T} - DENS_{\text{óleo},T}}{DENS_{\text{água},T} - DENS_{\text{óleo},T}}$$

**BASE\_SW Calculation:**

- $CTL_A = f(DENS_{\text{água},15/60}, T_b)$  is the CTL\_WATER parameter, which converts the volume in flow temperature to base temperature.
- $CTL_O = f(DENS_{\text{óleo},T_{\text{lab}}}, T_{\text{lab}}, T_b)$ , which converts the volume in laboratory analysis temperature to flow temperature

- It calculates  $BSW_{Tb}$  (BASE\_SW parameter).

$$BSW_{Tb} = \frac{BSW_T * CTL_A}{BSW_T * CTL_A + (1 - BSW_T) * CTL_o}$$

### BSW Calculation – Lab analysis

If the CALC\_BSW parameter is set to “Lab analysis”, the value of the FLOWING\_SW\_OUT output will be calculated using only the laboratory analysis results. This assumes the stability/regularity of the oil base density and BSW (changes occurred from the temperature changes and the difference of the water thermal expansion and oil thermal expansion coefficients).

This equation is calculated as indicated in the API-201 standard. – Allocation measurement, located in the B Appendix

### Block Inputs

The input is :

| Input             | Link Necessity | Description              |
|-------------------|----------------|--------------------------|
| TEMPERATURE_METER | mandatory      | Liquid flow temperature. |

### Block Output

| Outputs        | Description   |
|----------------|---|
| FLOWING_SW_OUT | Value calculated for the BSW parameter in flow condition. |
| CTL_WATER      | Temperature correction factor for water.                  |
| BASE_SW        | BSW calculated for base temperature condition             |

### BSW – LAB\_DENS\_OIL, LAB\_DENS\_WATER and XWS Calculations

The BSW is calculated using the laboratory analysis results: dry oil density, water density and BSW in laboratory analysis temperature. These calculations assume constant features (base density) for oil and water made.

### FLOWING\_SW\_OUT Calculation:

The equation below must be calculated:

$$X_{w,m} = \frac{X_{w,lab} * (CTL_{w,lab} / CTL_{w,m})}{X_{w,lab} * (CTL_{w,lab} / CTL_{w,m}) + (1 - X_{w,lab}) * (CTL_{o,lab} / (CTL_{o,m} * SF))}$$

Where :

$X_{w,m}$  : BSW in flow condition

$X_{w,lab}$  : BSW in laboratory analysis condition

$CTL_{w,lab}$  : Temperature correction factor for water, from the temperature of the laboratory analysis to a 15 °Celsius temperature (LD\_UNITS= Kg/m<sup>3</sup>); or to a 60 Fahrenheit temperature (LD\_UNITS=API/SG).

$CTL_{w,m}$  : Temperature correction factor for water, from the flow temperature to a 15 °Celsius temperature (LD\_UNITS= Kg/m<sup>3</sup>), or to a 60 Fahrenheit temperature (LD\_UNITS=API/SG).

$CTL_{o,lab}$  : Temperature correction factor for oil, from the temperature of laboratory analysis to a 15 °Celsius temperature (LD\_UNITS= Kg/m<sup>3</sup>), or to a 60 Fahrenheit temperature (LD\_UNITS=API/SG).

$CTL_{o,m}$  : Temperature correction factor for oil, from the flow temperature to a 15 °Celsius temperature (LD\_UNITS= Kg/m<sup>3</sup>), or to a 60 Fahrenheit temperature (LD\_UNITS=API/SG).

SF : oil shrinkage temperature

### Diagnosing and Correcting Problems

1. BLOCK\_ERR. Block configuration: The Temperature or Density Inputs are not connected.
2. BLOCK\_ERR. Out of Service: LCF block can continue in Out of service mode, although the target mode is Auto, because the Resource block is in O/S.

### Special Indications for BATCH\_STATUS

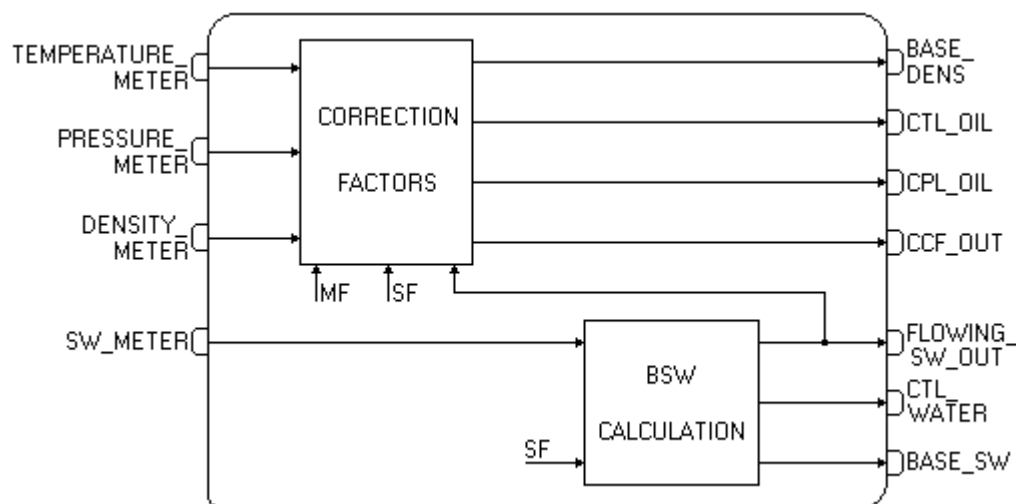
“Abnormal Conditions” – Problems in the BSW calculation.

The input values for SW are out of range 0-100 % (caso CALC\_BSW = “Dual Range”).

### Supported Modes

O/S and AUTO.

## Schematic



## Parameters

| Idx    | Type/View | Parameter         | Data Type     | Valid Range / Options           | Default Value | Units    | Store./Mode | Description  |
|--------|-----------|-------------------|---------------|---------------------------------|---------------|----------|-------------|--|
| 1      | 1,2,3,4   | ST_REV            | Unsigned16    |                                 | 0             | None     | S / RO      |  |
| 2      |           | TAG_DESC          | OctString(32) |                                 | Spaces        | Na       | S           | If this parameter is configured with a string other than blank spaces, then this parameter will replace the block tag in the QTR report. |
| 3 (A2) | 4         | STRATEGY          | Unsigned16    | 0 to 4                          | 0             | None     | S           | This parameter identifies the number of the measured flow.   |
| 4      | 4         | ALERT_KEY         | Unsigned8     | 1 to 255                        | 0             | None     | S           |  |
| 5      | 1,3       | MODE_BLK          | DS-69         |                                 | O/S           | Na       | S           | See Mode parameter.  |
| 6      | 1,3       | BLOCK_ERR         | Bitstring(2)  |                                 |               | E        | D / RO      |  |
| 7      | I,1,3     | TEMPERATURE_METER | DS-65         |                                 |               | T_UNITS  | N / RO      | Temperature used to calculate the correction factor for the thermal expansion of a liquid.   |
| 8      | I,1,3     | PRESSURE_METER    | DS-65         |                                 |               | P_UNITS  | N / RO      | Manometric pressure used to calculate the correction factor for the liquid compressibility.  |
| 9      | I,1,3     | DENSITY_METER     | DS-65         |                                 |               | LD_UNITS | N / RO      | Density used to calculate the factors CPLm and CTLm.   |
| 10     | I,1,3     | SW_METER          | DS-65         |                                 |               | %        | N / RO      | Percentage of sand and water mixed in the oil.   |
| 11     | O,1,3     | BASE_DENS         | DS-65         |                                 |               | LD_UNITS | N / RO      | Base density of the dry oil. This parameter is calculated.   |
| 12     | O,1,3     | CTL_OIL           | DS-65         |                                 |               |          | N / RO      | Temperature Corrention Factor.   |
| 13     | O,1,3     | CPL_OIL           | DS-65         |                                 |               |          | N / RO      | Pressure Correction Factor.  |
| 14     | O,1,3     | CCF_OUT           | DS-65         |                                 |               |          | N / RO      | Combined Correction Factor.  |
| 15     | O,1,3     | FLOWING_SW_OUT    | DS-65         |                                 |               | %        | N / RO      | Percentage calculated of sand and water mixed in the oil.  |
| 16     | O,1,3     | CTL_WATER         | DS-65         |                                 |               |          | N / RO      | Temperature Correction Factor.   |
| 17     | O,1,3     | BASE_SW           | DS-65         |                                 |               | %        | N / RO      | Percentage of sand and water mixed in the oil calculated in base conditions.   |
| 18     | 4         | BASE_PRESSURE     | Float         | 101.325 kPa<br>or<br>14.696 psi | 101.325 kPa   | P_UNITS  | S           | Base pressure for the fluid according to the selected unit in the P_UNITS parameter.   |

| Idx | Type/<br>View | Parameter             | Data Type  | Valid Range /<br>Options   | Default<br>Value  | Units   | Store/<br>Mode | Description   |
|-----|---------------|-----------------------|------------|--|-------------------|---------|----------------|---|
| 19  | 4             | BASE_TEMPERATURE      | Float      | 15.0 °C<br>or<br>20.0 °C<br>or<br>60.0 °F  | 15.0 °C           | T_UNITS | S              | Base temperature for the fluid according to the selected unit in the T_UNITS parameter.   |
| 20  | 4             | T_UNITS               | Unsigned16 | 1000=Kelvin<br>1001=Celsius<br>1002=Fahrenheit<br>1003=Rankine   | Celsius           | E       | S              | Engineering Unit for temperature.   |
| 21  | 4             | P_UNITS               | Unsigned16 | 1130=Pa<br>1132=Mpa<br>1133=kPa<br>1137=bar<br>1138=mbar<br>1139=torr<br>1140=atm<br>1141=psi<br>1144=g/cm <sup>2</sup><br>1145=kgf/cm <sup>2</sup><br>1147=inH2O 4°C<br>1148=inH2O 68 °F<br>1150=mmH2O 4°C<br>1151= mmH2O 68 °F<br>1154=ftH2O 68 °F | KPa               | E       | S              | Engineering Unit for static pressure  |
| 22  | 4             | LD_UNITS              | Unsigned16 | 1097= Kg/m <sup>3</sup><br>1113=API<br>1599 = relative density/SG  | Kg/m <sup>3</sup> | E       | S              | Engineering Unit for liquid density. The selection of this unit indicates which table uses for the correction factor calculations (CTL and CPL).  |
| 23  | 4             | PRODUCT_TYPE          | Unsigned8  | 0=Crude oil(Table suffix A)<br>1=Generalized products (Table suffix B)<br>2=MTBE (Table suffix C)<br>3=Lubricating oil (Table suffix D)<br>4=Water<br>5=Light hydrocarbon (NGL&LPG)  | 0                 | E       | S              | Product type.   |
| 24  | 4             | DENSITY_TYPE          | Unsigned8  | 1=Density at base<br>2=Measured density  | 1                 | E       | S              | Density type.   |
| 25  | 4             | HYDROMETER_CORRECTION | Unsigned8  | 0=No correction<br>1=Correction is done  | 0                 | E       | S              | Hydrometer correction.  |
| 26  | 4             | COEF_OF_THERMAL_EXP   | Float      | >= 0.0   | 0.0               |         | S              | If the selected product is MTBE, the thermal expansion coefficient is in base temperature.<br>If the selected product is Light hydrocarbon, the absolute equilibrium pressure is at 100 °F. |
| 27  | 2             | MF                    | Float      | 0.8 to 1.2   | 1.0               | Na      | S              | MF used for the combined correction factor (CCF).   |

| Idx | Type/<br>View | Parameter      | Data Type    | Valid Range /<br>Options  | Default<br>Value | Units     | Store/<br>Mode | Description   |
|-----|---------------|----------------|--------------|---|------------------|-----------|----------------|---|
| 28  | 4             | CALC_BSW       | Unsigned8    | 0=None<br>1=Dual range<br>2=Lab analysis                                  | 0                | Na        | S              | It selects one of the ways of calculating the BSW.  |
| 29  | 2             | LO_SW          | Float        | 0.0 to 100.0<br>0.0 = Always<br>calculated<br>100.0 = Never<br>calculated | 0.0              | %         | S              | Lower limit to calculate the BSW, if the "Dual range" option is selected in CALC_BSW.   |
| 30  | 2             | LAB_TEMP       | Float        |   | 15               | T_UNITS   | S              | Temperature which the laboratory analysis is accomplished to obtain the XWS.  |
| 31  | 2             | LAB_DENS_WATER | Float        | >= 0.0  | 1000             | LD_UNITS  | S              | Water density in laboratory analysis condition (LAB_TEMP).  |
| 32  | 2             | LAB_DENS_OIL   | Float        | >= 0.0  | 900              | LD_UNITS  | S              | Oil density in laboratory analysis condition (LAB_TEMP).  |
| 33  | 2             | LAB_SW         | Float        | 0 to 100  | 0                | %         | S              | BSW value obtained in laboratory analysis condition (LAB_TEMP).   |
| 34  | 2             | SF             | Float        | 1=disabled<br>0< SF <= 1  | 1                | Na        | S              | Shrinkage factor obtained in laboratory analysis.   |
| 35  | 3             | F              | Float        |   |                  | 1/P_UNITS | N / RO         | <b>Compressibility factor</b>   |
| 36  | 3             | STATUS_CURRENT | Bitstring[2] | See Block<br>Options  | 0                | Na        | N/ RO          | Current status. Similar to BATCH_STATUS.  |
| 37  |               | PE_TF          | Float        |   |                  | P_UNITS   | N / RO         | Equilibrium pressure in flow conditions.  |
| 38  |               | UPDATE_EVT     | DS-73        |   |                  | Na        | D              | This alert is generated by any change to the static data.   |
| 39  |               | BLOCK_ALM      | DS-72        |   |                  | Na        | 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 indicated in the subcode field. The first active alert will set the Active status in the Status attribute. When the Unreported status is removed by the alert reporting task, another block alert can be reported without clearing the Active status, if the subcode has been changed. |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static; I – Input Parameter; O – Output Parameter

**Gray Background Line: Default Parameters in Syscon**

## FMTH – Flexible Mathematical Block

### Description

This block provides mathematical expression execution with inputs, outputs and auxiliary variables generated by the user, and also including conditional expressions.

The FMTH block has the following characteristics:

- It allows execute several mathematical expressions “customized” by user with input and output values, and also using auxiliary variables in these expressions.
- Friendly edition of the mathematical expressions, similar to the Microsoft Excel;
- It allows the usage of the following operations described in the table below:

| Operation                | Representation | Variable Type   |
|--------------------------|----------------|-----------------|
| AND                      | and(Num;Num)   | Real            |
| Arc Sine                 | asin(Num)      | [-1;1]          |
| Arco Cosine              | acos(Num)      | [-1;1]          |
| Making around to down    | floor(Num)     | Real            |
| Making around to up      | ceil(Num)      | Real            |
| Making around to nearest | round(Num)     | Real            |
| Status Bad               | bad(Status)    | Status Variable |
| Cosine                   | cos(Num)       | Radian          |
| Separation               | /              | Real            |
| Exponential              | exp(Num)       | Real            |
| Status Good              | good(Status)   | Status Variable |
| Logarithm                | log(Num;Base)  | Real            |
| Module                   | abs(Num)       | Real            |
| Multiplication           | *              | Real            |
| Negative                 | (-Num)         | Real            |
| OR                       | or(Num;Num)    | Real            |
| Potence                  | ^              | Real            |
| Sine                     | sin(Num)       | Radian          |
| Addition                 | +              | Real            |
| Subtraction              | -              | Real            |
| Tangent                  | tan(Num)       | Radian          |
| Uncertain or Bad Status  | uob(Status)    | Status Variable |
| Uncertain or Good Status | uog(Status)    | Status Variable |

- It provides constant variables for using in the expressions:

| Constant Variable | Description   |
|-------------------|---|
| MC                | Time for block execution  |
| PI                | 3,14159   |
| E                 | 2,71828   |
| GOOD              | “Good Non Cascade Non Specific Not Limited” Status (Value =128) |
| BAD               | “Bad Non Specific Not Limited” Status (Value= 0)                |
| UNC               | “Uncertain Non Specific Not Limited” Status (Value = 64)        |

- It allows make conditional expressions (if(expression;affirmative possibility;negative possibility));
- It supports up to mathematical expressions;
- It has identifier of the following error types for the equations inserted by user:
  - Incorrect variable name
  - Incorrect usage of mathematical expressions, for example dividing by zero.



This block becomes available to the user:

- 10 analog inputs;
- 4 discrete inputs;
- 2 analog outputs;
- 2 discrete outputs;
- 20 auxiliary analog variables;
- 10 auxiliary discrete variables.

The FMTH block has also a MC parameter, which represents the Macro Cycle. The user can correlate two or more fields to make a mathematical expression.

### Mathematical Operations

Some of the mathematical and conditional operations supported by the FMTH block are described below

#### 1 - AND:

- TRUE: If all arguments are TRUE
- FALSE: if all, one or more arguments are FALSE

Syntax – *and(condition; condition)* or *and(condition; condition; condition)*  
Condition is a number or a conditional expression.

Example:  
*and(1,0)* is False

#### 2 - ARC SINE:

It calculates the arc sine of the value. This value should be between -1 and 1.

Syntax – *asin(num)*  
Num can be a number or an expression

Example:  
*asin(0.5)* = 0.523599

#### 3 - ARC COSINE:

It calculates the arc cosine of the value. This value should be between -1 and 1.

Syntax – *acos(num)*  
Num can be a number or an expression.

Example:  
*acos(0.5)* = 1.047198

#### 4 – MAKING AROUND TO DOWN

It makes around the number for the nearest low integer number. The value can be in REAL format.

Syntax – *floor(num)*  
Num is the number that will be making around

Example:  
*floor(2,343)* is equal to 2

#### 5 – MAKING AROUND TO UP:

It makes around the number for the nearest upper integer number. The value can be in REAL format.

Syntax – *ceil(num)*  
Num is the number that will be making around

Example:  
*ceil(2,343)* is equal to 3

#### 6 – MAKING AROUND TO THE NEAREST:

It makes around to the nearest integer number. Arredonda um número para o inteiro mais próximo. The value can be in REAL format.

Syntax – *round(num)*  
Num is the number that will be making around

Example:  
*round(2,343)* é igual a 2

**7 - COSINE:**

It calculates the cosine of the angle given in radian.

Syntax – *cos(num)*

Num is the angle that will be obtained the cosine.

Example:

$\cos(0)$  is equal to 1

**8 - EXPONENTIAL:**

It calculates  $e$  in the potence of num. The constant  $e$  is equal to 2.71828182845904, in the base of natural logarithm.

syntax – *exp(num)*

Num is the exponent applied to the base  $e$ .

Example:

$\exp(2)$  is equal to  $e^2$ , or 7.389056

**9 - LOGARITHM:**

It calculates the logarithm of the number for a specified base.

Syntax – *log(num; base )*

Num is the positive real number that the logarithm will be obtained.

Base is the base of the logarithm.

Example:

$\log(8; 2)$  is equal to 3

**10 - MODULE:**

It calculates the absolute value of the number. The absolute value of the number is the number without signal (positive or negative).

Syntax - *abs(num)*

Num is the real number that the absolute value will be obtained

Example:

$\text{abs}(-2)$  is equal to 2

**11 - NEGATIVE:**

The result is the number with the signal changed.

Syntax - *(-num)*

Num is the real number that the signal will be changed

Example:

$((-1)+2) = 1$

Obs: all negative operations must be accomplished using parentheses. If there are no parentheses, errors can occur in the logic expression. See the example below:

$(-1+2)$  will result  $-3$  , not 1.

**12 - OR:**

- TRUE if any argument is TRUE
- FALSE if all arguments are FALSE

Syntax - *or(condition; condition) or or(condition; condition; condition)*

Condition is the real number of the conditional expression

Example:

$\text{or}(1; 0)$  is equal to True

**13 - POTENCE:**

It provides the result of the number in the potence required.

Syntax – *num^potence*

Num is the base number. It can be any real number.

Potence is the exponent for that the base is elevated.

Example:

$3^2$  - is equal to 9

**14 - SINE:**

It calculates the sine of an angle given in radian.

Syntax –  $\sin(num)$

Num is the angle in radian for that the sine will be obtained.

Example:

$\sin(0)$  is equal to 0

**15 - TANGENT:**

It calculates the tangent for an angle given in radian.

Syntax –  $\tan(num)$

Num is the angle given in radian for that the tangent will be obtained

Example:

$\tan(0)$  is equal to 0

**Representation of the Variables in the Mathematical Expressions**

The representation of the output, input and auxiliary variables is showed in the table below

| Variable                    | Representation |
|-----------------------------|----------------|
| Analog Input                | AIX            |
| Discrete Input              | DIX            |
| Analog Input Status         | AIXS           |
| Discrete Input Status       | DIXS           |
| Analog Output               | AOX            |
| Discrete Output             | DOX            |
| Analog Output Status        | AOXS           |
| Discrete Output Status      | DOXS           |
| Auxiliary Analog Variable   | AAX            |
| Auxiliary Discrete Variable | DAX            |

X means the number of the variable, for example:

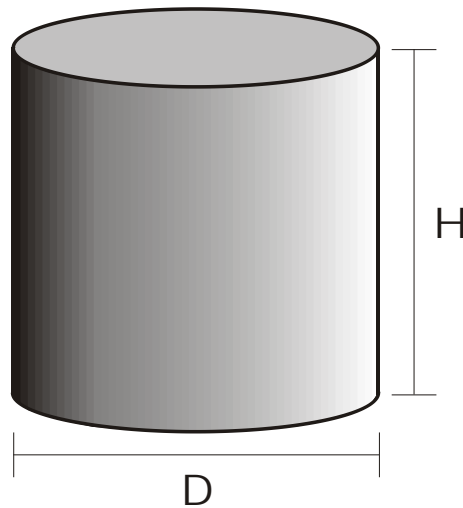
- AI1 – It represents the analog input variable 1
- DA7 – It represents the auxiliary discrete variable 7

**NOTE for SYSCON**

Before inserting the equations in the `EXPR_X` variables, it is necessary to click in the expression line and where the equation is edited, put the mouse cursor in the end of the line and then push the Backspace key (keyboard) from the ending to the beginning of the line. This avoids that residues damage the edition of the equations.

**Example of the Calculation for the Volume of the Tube**

Below there is a tube that is necessary to calculate the volume. It has the base diameter equals to 10m and its height is 30m



D = 10  
H = 30

Through the FAUX auxiliary variables, it is possible storage the values that will be used in the equations, and also the intermediate calculations performed.

```
<FAUX[1]>.. 3.1416 // it corresponds to the constant π
<FAUX[2]> 10      // it corresponds to the D diameter
<FAUX[3]> D/2     // R Ray (It will be calculated in the EXPR_1 expression)
<FAUX[4]> π*R2   // Base Area (It will be calculated in the EXPR_2 expression)
<FAUX[5]> 30      // it corresponds to the H height
```

the calculations are performed in the expressions showed below

Base Area:  $A = \pi * R^2$

```
<EXPR_1> AA3=AA2/2
<EXPR_2> AA4=AA1*AA3^2
```

Volume:  $V = A * H$

```
<EXPR_3> AO1=AA4*AA5
```

Using the values for this tube, its volume will be 2356.2 m<sup>3</sup>.

### Using Conditional

This block allows using conditional in the expressions. It shows the result of a condition, specified by user, if the algorithm evaluates as true; and another value if it evaluates as false. This condition can be a value or an expression that can be evaluated as true or false.

It supports if (conditional) in sequence.

Syntax – *if(condition;value\_if\_true;value\_if\_false)*  
 - Value\_if\_true is the result when the condition is true.  
 - Value\_if\_false is the result when the condition is false

Example1:

if(2=1+1;3;4) the expression 2 = 1+1 is true, so the result will be 3 (the corresponding value when Value\_if\_true).

Example 2:

Algorithm: Consider the following example:

```

if (AA1 > 0) AND (AA1 < 500)
  if (AA2 >= (AA1 * 10))
    AO1 = AA2
  else
    AO1 = AA1 * 20
else if (AA1 > 500) AND (AA1 < 1000)
  AO1 = AA1
else
  AO1 = AA3

```

FMATH expression:

AO1 = if (and(AA1 > 0;AA1<500) ; if (AA2 >= (AA1\*10); AA2; AA1 \* 20) ; if (and (AA1 > 500; AA1 < 1000);AA1;AA3) )

## Status Handle

For the non usable inputs (bad or uncertain status), the block keeps the calculation by using the last usable value. The output status will be the worst input status by default.

The output status in the expressions can be determined by using the input status optionally

There are several status functions (good(x), bad(x), uob(x), uog(x) in order to make easy the input status handle in the expressions. Numeric values or Constant variables, such as GOOD or BAD, can also be compared (for example, AIxS > GOOD when it is necessary to know if the quality status is good). To determine the output status only the numeric value will be used (AOxS = 128 to write GOOD in the output status).

The next example has two input values for the same variable (with measurement redundancy, for example) which it used the first good input. If the first input fails, the algorithm keeps the calculation using the other input. A way to implement the algorithm should be:

```

If (STATUS_ENTRADA_1 = GOOD)
  SAIDA_1 = ENTRADA_1 * AA1
else
  if (STATUS_ENTRADA_2 = GOOD)
    SAIDA_1 = ENTRADA_2 * AA1
  else
    SAIDA_1 = 100
if (STATUS_ENTRADA_1 = GOOD) OR (STATUS_ENTRADA_2 = GOOD)
  STATUS_SAIDA_1 = GOOD
else
  STATUS_SAIDA_1 = BAD

```

The block expressions should be:

<EXPR\_1> AO1 = if (good(AI1S);AI1\*AA1;if(goodAI2S);AI2\*AA1;100))

<EXPR\_2> AO1S = if (or(good(AI1S);good(AI2S));128;0)

## Correlating the Mathematical Expressions

Each parameter in the mathematical expression has a limit of 100 characters. This avoids that long mathematical expressions be inserted in only one expression field. It is possible to insert long expressions correlating two or more expression fields. This correlation is accomplished through inserting the # character in the ending of the expression that is being correlating with the next expression field. This operator allows correlate more than two expression fields, just inserting the character # in the ending of all correlated fields except in the last one.

There is an example of correlated expression below:

```

<Expr_1> AO1 = 2+2*sin(ln_1/R) + ... + #
<Expr_2> log(10;2) + ... + 2 #
<Expr_3> ^2 - if(BAD(AI1S);12;14)
<Expr_4> AA2 = 2*3^2*(-2)+23

```

In the example above, the *Expr\_1*, *Expr\_2* and *Expr\_3* fields are correlated resulting in the mathematical expression:

$$AO1 = 2+2*\sin(\ln_1/R) + \dots + \log(10;2) + \dots + 2^2 - \text{if}(\text{BAD}(\text{AI1S});12;14)$$

### Using Macrocycle for calculations

The macrocycle should also be used in the expressions when it is necessary to use the time as base. The MC variable has the block execution value.

Example: It is required to totalize the output when IN\_1 has its value less than 1000.

The FMTH expressions are:

```
<EXPR_1>AA1=AI1*MC
<EXPR_2>AA2=AA2+AA1
<EXPR_3>AA2=if(AA2 >=1000;0;AA2)
<EXPR_4>AO1=AA2
```

### Error Handle

The FMTH block uses the *Error\_Code*, *Error\_Line* and *Error\_Col* parameters to show the type and place that the errors occurred. The *Error\_Line* parameter shows the line that the error occurred, and the *Error\_Col* parameter shows the place (in tokens) that the error occurred. Token means any expression element (AA1, AO2, +, =, (, ), are token examples). The error types are identified in the *Error\_Code* parameter, and are indicated in the table below:

| Code | Error                            |
|------|----------------------------------|
| 0    | No Errors                        |
| 1    | Invalid Variable Name            |
| 2    | Invalid Function Name            |
| 3    | Division by Zero                 |
| 4    | Incorrect use of Logarithm       |
| 5    | Square Root of a Negative Number |
| 6    | Inexistent Arc                   |
| 7    | Unidentified Error               |

For example, consider the following expression:

```
<EXPR_10> AO1 = (AA1 * 10) * 2 + AI2 ^ 2 * sen (AA3)
```

There is an error in the expression above: the function name “sin” is incorrect. The block shows the following message:

```
FMTH_ERROR_LINE = 10    // Error was found in the Expression 10.
FMTH_ERROR_COL   = 15    // Error in the fifteenth token erro (from left to right)
FMTH_ERROR_CODE  = 2     // Error is Invalid functio name
```

The syntax errors (Incorrect Variable or Function name and incorrect use of the mathematical operators, for example, 2 + + 2) are identified when the mathematical expression is done. So, these error types never appear in the block execution.

There are some execution errors that can occur during the execution time, for example, dividing the number by zero, logarithm of a negative number or square root of a negative number. When a execution error occurs, the block takes the following actions:

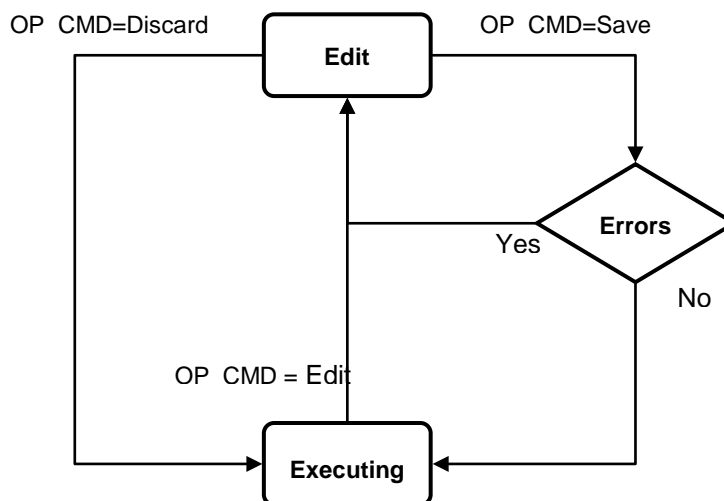
- The Analog Output Values (OUT\_1 and OUT\_2) and the Discrete Output 1 (OUT\_D1) will repeat the last value calculated without errors and their status will be GOOD;
- The OUT\_D2 discrete output will be 1, indicating that the execution error occurs, with GOOD status;
- The ERROR\_CODE, ERROR\_LINE and ERROR\_COL parameters will show, respectively, error code, line and column that the error occurs.

## Edit and Execution States

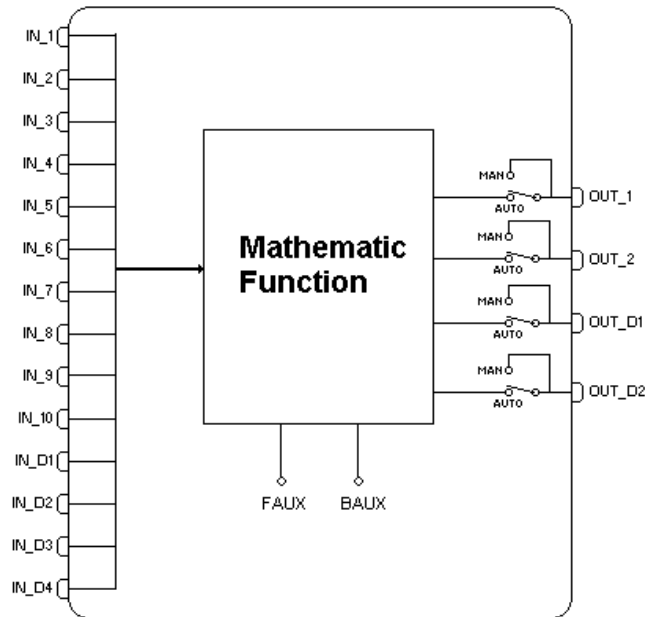
The block allows to edit mathematical expressions in online mode. In order to do this, the block has two states:

- Edit State:** In this state the user can edit the expressions while the previous expressions are executed in each cycle. When inserting an expression, an error check ("Invalid Variable Name", "Invalid Function Name") is done and the *Error\_Code* parameter shows the error type and *Error\_Line* and *Error\_Code* parameters show the place for the identified error. The possible errors generated during the "Edit State" will not be indicated. To back to the execution mode, the user has two options that should be selected through the *OP\_CMD\_FMTH* parameter:
  - **Save** – This command can be used when the user wants to execute the edited expressions. Therefore, before starting the executing of the edited expressions, a syntax error check is done, and if there is an error, its type and place are indicated in the *Error\_Code*, *Error\_Line* and *Error\_Code* parameters, and the block continues in the edit state. If there are no errors, the block replaces the expressions that are being executed by the next edited expressions and goes to the executing state identified by the *OP\_CMD\_FMTH* (*Executing*) parameter.
  - **Discard** – This command can be used when the user does not want to execute the edited expressions, so the expressions that are being executed will be kept. When using this command, the edited expression will be lost and the expressions that are being executed will be showed again.
- Executing State** – In this mode, it is impossible to edit the mathematical expressions, only the visualization of the mathematical expressions are allowed

The next state diagram shows all possible transitions:



## Schematic



## Parameters

| Idx | Parameter       | Data Type     | Valid Range / Options   | Default Value | Units | Store/ Mode | Description   |
|-----|-----------------|---------------|---|---------------|-------|-------------|---|
| 1   | ST_REV          | Unsigned16    |   | 0             | None  | S           |   |
| 2   | TAG_DESC        | OctString(32) |   | Spaces        | Na    | S           |   |
| 3   | STRATEGY        | Unsigned16    |   | 0             | None  | S           |   |
| 4   | ALERT_KEY       | Unsigned8     | 1 to 255  | 0             | None  | S           |   |
| 5   | MODE_BLK        | DS-69         |   | O/S           | Na    | S           | See Mode Parameter  |
| 6   | BLOCK_ERR       | Bitstring(2)  |   |               | E     | D           |   |
| 7   | FMTH_ERROR_LINE | Unsigned8     |   | 0             |       | D           | Line in which occurred an error                                     |
| 8   | FMTH_ERROR_COL  | Unsigned8     |   | 0             |       | D           | Position in the line that occurred an error                         |
| 9   | FMTH_ERROR_CODE | Unsigned8     | 0 = No Errors<br>1 = Invalid Variable Name<br>2 = Invalid Function Name<br>3 = Division by Zero<br>4 = Incorrect use of Logarithm<br>5 = Square Root of a Negative Number<br>6 = Inexistent Arc<br>7 = Unidentified Error | 0             |       | D           | Code representing an error in the mathematical expressions inserted |
| 10  | OUT_1           | DS-65         |   |               |       | N / Man     | Numbered analog output 1.   |
| 11  | OUT_2           | DS-65         |   |               |       | D / Man     | Numbered analog output 2.   |
| 12  | OUT_D1          | DS-66         |   |               |       | D / Man     | Numbered discrete output 1.   |
| 13  | OUT_D2          | DS-66         |   |               |       | D / Man     | Numbered discrete output 2.   |



| Idx | Parameter   | Data Type           | Valid Range / Options                                | Default Value | Units | Store/ Mode | Description  |
|-----|-------------|---------------------|--|---------------|-------|-------------|--|
| 14  | OP_CMD_FMTH | Unsigned8           | 0 = Edit<br>1 = Save<br>2 = Discard<br>3 = Executing |               |       | D / Man     | Parameter responsible for changing the state of the block                                  |
| 15  | EXPR_1      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 1   |
| 16  | EXPR_2      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 2   |
| 17  | EXPR_3      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 3   |
| 18  | EXPR_4      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 4   |
| 19  | EXPR_5      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 5   |
| 20  | EXPR_6      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 6   |
| 21  | EXPR_7      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 7   |
| 22  | EXPR_8      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 8   |
| 23  | EXPR_9      | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 9   |
| 24  | EXPR_10     | VisibleString (100) |  | Blank         |       | S/RW        | Expression input 10  |
| 25  | IN_1        | DS-65               |  |               |       | D           | Analog input 1.  |
| 26  | IN_2        | DS-65               |  |               |       | D           | Analog input 2.  |
| 27  | IN_3        | DS-65               |  |               |       | D           | Analog input 3.  |
| 28  | IN_4        | DS-65               |  |               |       | D           | Analog input 4.  |
| 29  | IN_5        | DS-65               |  |               |       | D           | Analog input 5.  |
| 30  | IN_6        | DS-65               |  |               |       | D           | Analog input 6.  |
| 31  | IN_7        | DS-65               |  |               |       | D           | Analog input 7.  |
| 32  | IN_8        | DS-65               |  |               |       | D           | Analog input 8.  |
| 33  | IN_9        | DS-65               |  |               |       | D           | Analog input 9.  |
| 34  | IN_10       | DS-65               |  |               |       | D           | Analog input 10.   |
| 35  | IN_D1       | DS-66               |  |               |       | D           | Discrete input 1.  |
| 36  | IN_D2       | DS-66               |  |               |       | D           | Discrete input 2.  |
| 37  | IN_D3       | DS-66               |  |               |       | D           | Discrete input 3.  |
| 38  | IN_D4       | DS-66               |  |               |       | D           | Discrete input 4.  |
| 39  | FAUX        | Float[20]           |  | 0             |       | D/RW        | Auxiliary analog variable to use to build the mathematical expressions in any EXPR field   |
| 40  | BAUX        | Unsigned8[10]       |  | 0             |       | D/RW        | Auxiliary discrete variable to use to build the mathematical expressions in any EXPR field |
| 41  | STATUS_OPTS | Bitstring(2)        | See Block Options                                    | 0             | Na    | S/0/S       | See Block Options  |
| 42  | UPDATE_EVT  | DS-73               |  |               | Na    | D           | This alert is generated by any change to the static data.                                  |

| Idx | Parameter | Data Type | Valid Range / Options | Default Value | Units | Store/ Mode | Description   |
|-----|-----------|-----------|-----------------------|---------------|-------|-------------|---|
| 43  | BLOCK_ALM | DS-72     |                       |               | Na    | 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 indicated in the subcode field. The first active alert will set the Active status in the Status attribute. When the Unreported status is removed by the alert reporting task, another block alert can be reported without clearing the Active status, if the subcode has been changed. |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static;

**Gray Background Line: Default Parameters in Syscon**

## ADT – Analog Data Transfer

### Overview

The Analog Data Transfer block obtains the analog values and status from the inputs and makes available for the outputs. Optionally the block can be configured to generate in its outputs constant values to be used in the inputs of other blocks.

### Description

The ADT block has two functions:

- **Data Transfer (default feature):** Transfer the values of the analog parameters of input IN\_x for the analog outputs OUT\_x. This block has 16 analog inputs and outputs.
- **Constant:** Generate in its analog outputs OUT\_x constant values stored in the internal parameters CT\_VAL\_x. The parameter CT\_OPTS enables this block feature.

### BLOCK\_ERR

The BLOCK\_ERR of the ADT block will reflect the following causes:

- Out of Service – occurs when the block is in O/S mode.

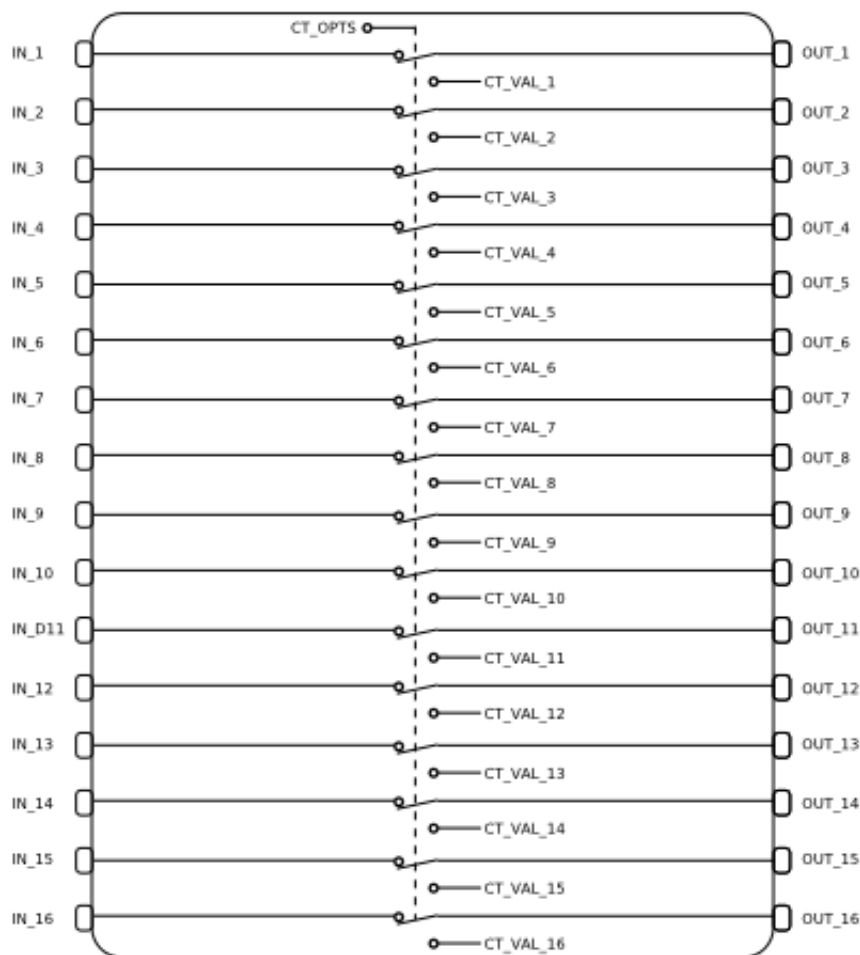
### Supported Modes

O/S, MAN and AUTO

### Modes

The manual mode disconnects the algorithm outputs and it allows manual substitution of the OUT\_x values.

### Schematic



## Parameters

| Idx | Parameter | Data Type  | Valid Range | Default Value                    | Units | Store / Mode | Description                                   |
|-----|-----------|------------|-------------|----------------------------------|-------|--------------|---|
| 1   | ST_REV    | Uns6       | 0 ... 65535 | 0                                | None  | S/RO         | Universal parameter                           |
| 2   | TAG_DESC  | VisStr(32) |             | Space                            | Na    | S/OOS        | Universal parameter                           |
| 3   | STRATEGY  | Uns16      | 0 ... 65535 | 0                                | None  | S/OOS        | Universal parameter                           |
| 4   | ALERT_KEY | Uns8       | 0 to 255    | 0                                | None  | S/OOS        | Universal parameter                           |
| 5   | MODE_BLK  | DS-69      |             | target:<br>OOS<br>actual:<br>OOS | Na    | S            | Universal parameter                           |
| 6   | BLOCK_ERR | Bitstr(2)  |             | 0, 0                             | E     | D/RO         | Universal parameter                           |
| 7   | IN_1      | DS-65      |             |                                  |       | D            | Analogic Input IN_1                           |
| 8   | IN_2      | DS-65      |             |                                  |       | D            | Analogic Input IN_2                           |
| 9   | IN_3      | DS-65      |             |                                  |       | D            | Analogic Input IN_3                           |
| 10  | IN_4      | DS-65      |             |                                  |       | D            | Analogic Input IN_4                           |
| 11  | IN_5      | DS-65      |             |                                  |       | D            | Analogic Input IN_5                           |
| 12  | IN_6      | DS-65      |             |                                  |       | D            | Analogic Input IN_6                           |
| 13  | IN_7      | DS-65      |             |                                  |       | D            | Analogic Input IN_7                           |
| 14  | IN_8      | DS-65      |             |                                  |       | D            | Analogic Input IN_8                           |
| 15  | IN_9      | DS-65      |             |                                  |       | D            | Analogic Input IN_9                           |
| 16  | IN_10     | DS-65      |             |                                  |       | D            | Analogic Input IN_10                          |
| 17  | IN_11     | DS-65      |             |                                  |       | D            | Analogic Input IN_11                          |
| 18  | IN_12     | DS-65      |             |                                  |       | D            | Analogic Input IN_12                          |
| 19  | IN_13     | DS-65      |             |                                  |       | D            | Analogic Input IN_13                          |
| 20  | IN_14     | DS-65      |             |                                  |       | D            | Analogic Input IN_14                          |
| 21  | IN_15     | DS-65      |             |                                  |       | D            | Analogic Input IN_15                          |
| 22  | IN_16     | DS-65      |             |                                  |       | D            | Analogic Input IN_16                          |
| 23  | OUT_1     | DS-65      |             |                                  |       | D            | Analogic Output OUT_1                         |
| 24  | OUT_2     | DS-65      |             |                                  |       | D            | Analogic Output OUT_2                         |
| 25  | OUT_3     | DS-65      |             |                                  |       | D            | Analogic Output OUT_3                         |
| 26  | OUT_4     | DS-65      |             |                                  |       | D            | Analogic Output OUT_4                         |
| 27  | OUT_5     | DS-65      |             |                                  |       | D            | Analogic Output OUT_5                         |
| 28  | OUT_6     | DS-65      |             |                                  |       | D            | Analogic Output OUT_6                         |
| 29  | OUT_7     | DS-65      |             |                                  |       | D            | Analogic Output OUT_7                         |
| 30  | OUT_8     | DS-65      |             |                                  |       | D            | Analogic Output OUT_8                         |
| 31  | OUT_9     | DS-65      |             |                                  |       | D            | Analogic Output OUT_9                         |
| 32  | OUT_10    | DS-65      |             |                                  |       | D            | Analogic Output OUT_10                        |
| 33  | OUT_11    | DS-65      |             |                                  |       | D            | Analogic Output OUT_11                        |
| 34  | OUT_12    | DS-65      |             |                                  |       | D            | Analogic Output OUT_12                        |
| 35  | OUT_13    | DS-65      |             |                                  |       | D            | Analogic Output OUT_13                        |
| 36  | OUT_14    | DS-65      |             |                                  |       | D            | Analogic Output OUT_14                        |
| 37  | OUT_15    | DS-65      |             |                                  |       | D            | Analogic Output OUT_15                        |
| 38  | OUT_16    | DS-65      |             |                                  |       | D            | Analogic Output OUT_16                        |
| 39  | CT_VAL_1  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_1  |
| 40  | CT_VAL_2  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_2  |
| 41  | CT_VAL_3  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_3  |
| 42  | CT_VAL_4  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_4  |
| 43  | CT_VAL_5  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_5  |
| 44  | CT_VAL_6  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_6  |
| 45  | CT_VAL_7  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_7  |
| 46  | CT_VAL_8  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_8  |
| 47  | CT_VAL_9  | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_9  |
| 48  | CT_VAL_10 | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_10 |
| 49  | CT_VAL_11 | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_11 |
| 50  | CT_VAL_12 | Float      |             | 0                                |       | S            | Constant analogic value transferred to OUT_12 |

| Idx | Parameter  | Data Type | Valid Range       | Default Value | Units | Store / Mode | Description  |
|-----|------------|-----------|-------------------|---------------|-------|--------------|--|
| 51  | CT_VAL_13  | Float     |                   | 0             |       | S            | Constant analogic value transferred to OUT_13  |
| 52  | CT_VAL_14  | Float     |                   | 0             |       | S            | Constant analogic value transferred to OUT_14  |
| 53  | CT_VAL_15  | Float     |                   | 0             |       | S            | Constant analogic value transferred to OUT_15  |
| 54  | CT_VAL_16  | Float     |                   | 0             |       | S            | Constant analogic value transferred to OUT_16  |
| 55  | CT_OPTS    | Bitstr(2) | See Block Options | 0             |       | S            | When the bit "CT_VAL_1 to OUT_1" is enabled the output OUT_1 receives the value of CT_VAL_1. And like this consecutively for each CT_VAL_x corresponding to each OUT_x, where x varies of 1 up to 16.  |
| 56  | UPDATE_EVT | DS-73     |                   |               | Na    | D            | This alert is generated by any change to the static data.  |
| 57  | BLOCK_ALM  | DS-72     |                   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static

## DDT – Discrete Data Transfer

### Overview

The Discrete Data Transfer block obtains the discrete values from the inputs and makes available for the outputs. Optionally the block can be used to make available the constants for the outputs.

### Description

The DDT block has two functions:

- **Data Transfer (default feature):** Transfer the values of the discrete parameters of input IN\_Dx for the discrete outputs OUT\_Dx. This block has 16 discrete inputs and outputs.
- **Constant:** Generate in its discrete outputs OUT\_Dx constant values stored in the internal parameters CT\_VAL\_Dx. To operate in this mode, the bit of the parameter CT\_OPTS corresponding to the output which will be used as constant should be enabled. For example, for the OUT\_D1 output to receive the constant value of CT\_VAL\_D1 the bit “CT\_VAL\_1 to OUT\_1” from the CT\_OPTS parameter should be enabled.

### BLOCK\_ERR

The BLOCK\_ERR of the DDT block will reflect the following causes:

- Out of Service – occurs when the block is in O/S mode.

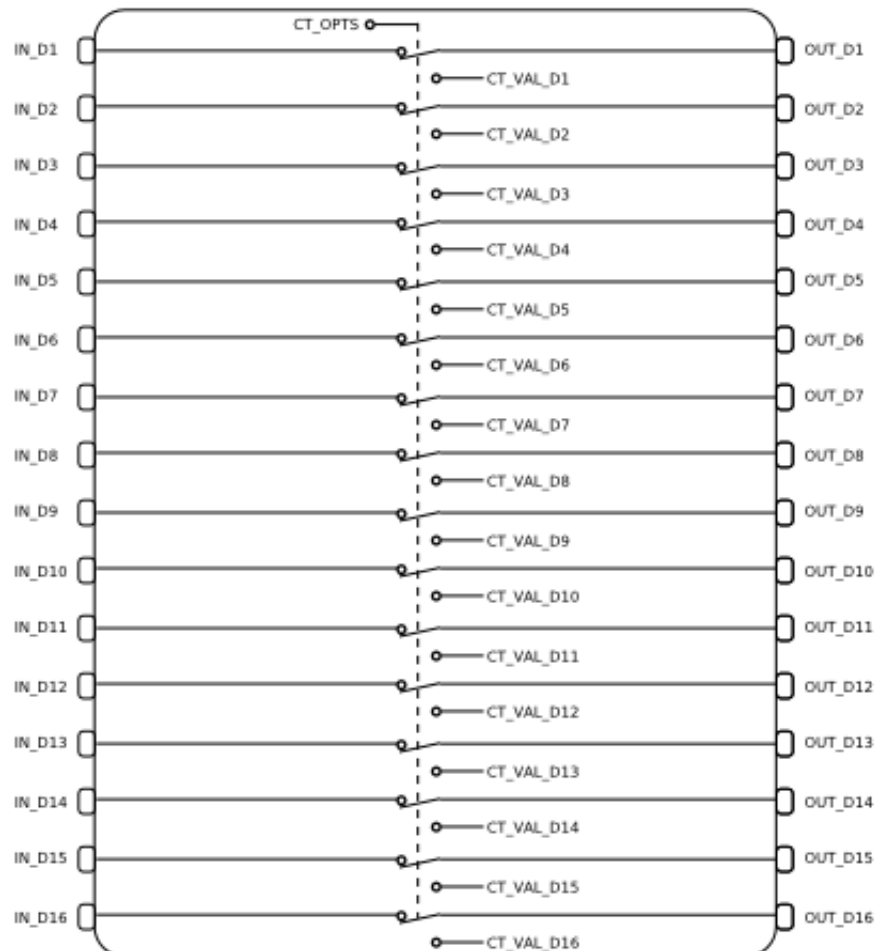
### Supported Modes

O/S, MAN and AUTO

### Modes

The manual mode disconnects the algorithm outputs and it allows manual substitution of the OUT\_Dx values.

### Schematic



## Parameters

| Idx | Parameter  | Data Type  | Valid Range | Default Value              | Units | Store / Mode | Description                                    |
|-----|------------|------------|-------------|----------------------------|-------|--------------|--|
| 1   | ST_REV     | Uns6       | 0 ... 65535 | 0                          | None  | S/RO         | Universal parameter                            |
| 2   | TAG_DESC   | VisStr(32) |             | Space                      | Na    | S/OOS        | Universal parameter                            |
| 3   | STRATEGY   | Uns16      | 0 ... 65535 | 0                          | None  | S/OOS        | Universal parameter                            |
| 4   | ALERT_KEY  | Uns8       | 0 to 255    | 0                          | None  | S/OOS        | Universal parameter                            |
| 5   | MODE_BLK   | DS-69      |             | target: OOS<br>actual: OOS | Na    | S            | Universal parameter                            |
| 6   | BLOCK_ERR  | Bitstr(2)  |             | 0, 0                       | E     | D/RO         | Universal parameter                            |
| 7   | IN_D1      | DS-66      |             |                            |       | D            | Discrete Input IN_D1                           |
| 8   | IN_D2      | DS-66      |             |                            |       | D            | Discrete Input IN_D2                           |
| 9   | IN_D3      | DS-66      |             |                            |       | D            | Discrete Input IN_D3                           |
| 10  | IN_D4      | DS-66      |             |                            |       | D            | Discrete Input IN_D4                           |
| 11  | IN_D5      | DS-66      |             |                            |       | D            | Discrete Input IN_D5                           |
| 12  | IN_D6      | DS-66      |             |                            |       | D            | Discrete Input IN_D6                           |
| 13  | IN_D7      | DS-66      |             |                            |       | D            | Discrete Input IN_D7                           |
| 14  | IN_D8      | DS-66      |             |                            |       | D            | Discrete Input IN_D8                           |
| 15  | IN_D9      | DS-66      |             |                            |       | D            | Discrete Input IN_D9                           |
| 16  | IN_D10     | DS-66      |             |                            |       | D            | Discrete Input IN_D10                          |
| 17  | IN_D11     | DS-66      |             |                            |       | D            | Discrete Input IN_D11                          |
| 18  | IN_D12     | DS-66      |             |                            |       | D            | Discrete Input IN_D12                          |
| 19  | IN_D13     | DS-66      |             |                            |       | D            | Discrete Input IN_D13                          |
| 20  | IN_D14     | DS-66      |             |                            |       | D            | Discrete Input IN_D14                          |
| 21  | IN_D15     | DS-66      |             |                            |       | D            | Discrete Input IN_D15                          |
| 22  | IN_D16     | DS-66      |             |                            |       | D            | Discrete Input IN_D16                          |
| 23  | OUT_D1     | DS-66      |             |                            |       | D            | Discrete Output OUT_D1                         |
| 24  | OUT_D2     | DS-66      |             |                            |       | D            | Discrete Output OUT_D2                         |
| 25  | OUT_D3     | DS-66      |             |                            |       | D            | Discrete Output OUT_D3                         |
| 26  | OUT_D4     | DS-66      |             |                            |       | D            | Discrete Output OUT_D4                         |
| 27  | OUT_D5     | DS-66      |             |                            |       | D            | Discrete Output OUT_D5                         |
| 28  | OUT_D6     | DS-66      |             |                            |       | D            | Discrete Output OUT_D6                         |
| 29  | OUT_D7     | DS-66      |             |                            |       | D            | Discrete Output OUT_D7                         |
| 30  | OUT_D8     | DS-66      |             |                            |       | D            | Discrete Output OUT_D8                         |
| 31  | OUT_D9     | DS-66      |             |                            |       | D            | Discrete Output OUT_D9                         |
| 32  | OUT_D10    | DS-66      |             |                            |       | D            | Discrete Output OUT_D10                        |
| 33  | OUT_D11    | DS-66      |             |                            |       | D            | Discrete Output OUT_D11                        |
| 34  | OUT_D12    | DS-66      |             |                            |       | D            | Discrete Output OUT_D12                        |
| 35  | OUT_D13    | DS-66      |             |                            |       | D            | Discrete Output OUT_D13                        |
| 36  | OUT_D14    | DS-66      |             |                            |       | D            | Discrete Output OUT_D14                        |
| 37  | OUT_D15    | DS-66      |             |                            |       | D            | Discrete Output OUT_D15                        |
| 38  | OUT_D16    | DS-66      |             |                            |       | D            | Discrete Output OUT_D16                        |
| 39  | CT_VAL_D1  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D1  |
| 40  | CT_VAL_D2  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D2  |
| 41  | CT_VAL_D3  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D3  |
| 42  | CT_VAL_D4  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D4  |
| 43  | CT_VAL_D5  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D5  |
| 44  | CT_VAL_D6  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D6  |
| 45  | CT_VAL_D7  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D7  |
| 46  | CT_VAL_D8  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D8  |
| 47  | CT_VAL_D9  | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D9  |
| 48  | CT_VAL_D10 | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D10 |
| 49  | CT_VAL_D11 | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D11 |
| 50  | CT_VAL_D12 | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D12 |
| 51  | CT_VAL_D13 | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred to OUT_D13 |
| 52  | CT_VAL_D14 | Unsig8     |             | 0                          |       | S            | Constant Discrete value transferred            |

| Idx | Parameter  | Data Type | Valid Range       | Default Value | Units | Store / Mode | Description  |
|-----|------------|-----------|-------------------|---------------|-------|--------------|--|
|     |            |           |                   |               |       |              | to OUT_D14   |
| 53  | CT_VAL_D15 | Unsig8    |                   | 0             |       | S            | Constant Discrete value transferred to OUT_D15   |
| 54  | CT_VAL_D16 | Unsig8    |                   | 0             |       | S            | Constant Discrete value transferred to OUT_D16   |
| 55  | CT_OPTS    | Bitstr(2) | See Block Options | 0             |       | S            | When the bit “CT_VAL_1 to OUT_1” is enabled the output OUT_D1 receives the value of CT_VAL_D1. And like this consecutively for each CT_VAL_Dx corresponding to each OUT_Dx, where x varies of 1 up to 16.  |
| 56  | UPDATE_EVT | DS-73     |                   |               | Na    | D            | This alert is generated by any change to the static data.  |
| 57  | BLOCK_ALM  | DS-72     |                   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static



## Flexible Function Block 1131

### FFB\_1131 – Flexible Function Block

#### Description

O FFB\_1131 has as main purpose to interconnect the ladder logic (usually used for discrete control strategies) and continuous control system, configured through function blocks.

The input/output parameter definitions are performed as following the user applications.

FFB\_1131 has 12 basic parameters. Besides that, it can have up to 242 additional parameters, combining DI, DO, AI and AO parameters (OUT\_D, IN\_D, OUT, IN), which the parameters names are also defined by user.

In Auto mode, this block updates the block inputs for the Ladder and sends the Ladder values for the block outputs. In Man mode, the block stops the block update and the user is able to change the output values.

The input/output status are mapped in boolean parameters into the Ladder. For the inputs, any status with Good quality (Good Non Cascade or Good Cascade) will be 0 (zero) in the corresponding Ladder status. Status qualities different of Good (Bad or Uncertain) will be 1 (one) in the Ladder. For the outputs, the value 0 (zero) in the Ladder will correspond to Good Non Cascade status and the value 1 (one) will correspond to Bad Non Specific status. The table below summarizes the status:

| FFB Parameter | FF Status [datatype byte]                       | Ladder Status [Datatype boolean] |
|---------------|---|----------------------------------|
| Input         | Good Non Cascade or Good Cascade ( $\geq 128$ ) | 0                                |
|               | Bad or Uncertain ( $< 128$ )                    | 1                                |
| Output        | Good Non Cascade Non Specific Not Limited (128) | 0                                |
|               | Bad Non Specific Not Limited (0)                | 1                                |

This block is available up to the following versions of controllers Firmware and Device Revision:

| Controller | Firmware | Device Revision Base |
|------------|----------|----------------------|
| DF62       | V2_x_x   | 03                   |
| DF63       |          | 02                   |
| DF75       |          | 02                   |
| DF79       | V1_x_x   | 02                   |
| DF89       |          | 01                   |
| DF73       | V2_x_x   | 04                   |
| DF95       |          | 01                   |
| DF97       |          | 01                   |
| DF81       | V1_x_x   | 01                   |

Where, in the Firmware column, x\_x indicates the releases of firmware version. The Device Revision Base column indicates the Device Description (DD base) of the equipment.

#### ATTENTION

Changes of device description in controllers are made using the Syscon Exchange procedure, where the links of the FFB will be lost.

#### BLOCK\_ERR

BLOCK\_ERR will be “Out of Service” – when the block is in O/S mode.

#### Supported Modes

O/S, MAN and AUTO.

## Parameters

| Idx | Parameter         | Data Type     | Valid Range / Options | Default Value | Units | Store./ Mode | Description   |
|-----|-------------------|---------------|-----------------------|---------------|-------|--------------|---|
| 1   | ST_REV            | Unsigned16    |                       | 0             | None  | S            |   |
| 2   | TAG_DESC          | OctString(32) |                       | Spaces        | Na    | S            |   |
| 3   | STRATEGY          | Unsigned16    |                       | 0             | None  | S            |   |
| 4   | ALERT_KEY         | Unsigned8     | 1 to 255              | 0             | None  | S            |   |
| 5   | MODE_BLK          | DS-69         | O/S, Man, Auto        | O/S           | Na    | S            | See Mode parameter.   |
| 6   | BLOCK_ERR         | Bitstring(2)  |                       |               | E     | D / RO       |   |
| 7   | ALGORITHM_SEL     | Unsigned32    |                       | 0             | None  | S            | Not used.   |
| 8   | CONTENTS_REV      | Unsigned32    |                       |               | None  | S            | Not used.   |
| 9   | FILE_REV          | Unsigned32    |                       |               | None  | S            | Not used.   |
| 10  | FILE_LOCATOR      | Unsigned32    |                       |               | None  | S            | Not used.   |
| 11  | UPDATE_EVT        | DS-73         |                       |               | na    | D            | This alert is generated by any change to the static data.   |
| 12  | BLOCK_ALM         | DS-72         |                       |               | na    | 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 indicated in the subcode field. The first active alert will set the Active status in the Status attribute. When the Unreported status is removed by the alert reporting task, another block alert can be reported without clearing the Active status, if the subcode has been changed. |
| 13  | OUT_D/IN_D/OUT/IN | DS-65/DS-66   |                       |               |       | D            | The input/output parameters as well as type, name and quantity are determined by user.  |
| ... | ....              | ...           | ...                   | ...           | ...   | ...          |   |
| 254 | OUT_D/IN_D/OUT/IN | DS-65/DS-66   |                       |               |       | D            |   |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static;

**Gray Background Line: Default Parameters in Syscon**

## FFB2\_1131 – Flexible Function Block 2

### Description

O FFB2\_1131 has as main purpose to interconnect the ladder logic (usually used for discrete control strategies) and continuous control system, configured through function blocks.

The parameters names and input/output parameter definitions are performed as following the user applications.

O FFB2\_1131 has 12 basic parameters. Besides that, it can have up to 242 additional parameters, combining DI, DO, AI and AO (OUT\_D, IN\_D, OUT, IN) input/output parameters.

For additional parameters, the block supports four data structure, DS-65, DS-66, DS-158\* and DS-174\*. These structures are limited in input/output parameters: up to 242 parameters for DS-65 and DS-66 structures, and up to 16 parameters for DS-158 and DS-174 structures.

The following Table details information about the quantity of input/output parameters for each FFB2\_1131 data structure:

| Data Structures | Quantity of Default Parameters | Maximum of Input/Output Parameters supported by Structure | Maximum of Input/Output Parameters supported by FFB |
|-----------------|--------------------------------|---|---|
| IN – DS65       | 32                             | 242 parameters  | 242 parameters                                      |
| OUT – DS65      | 32                             |   |   |
| IN – DS66       | 32                             | 242 parameters  |   |
| OUT – DS66      | 32                             |   |   |
| IN – DS158 *    | 4                              | 16 parameters   |   |
| OUT – DS158 *   | 4                              |   |   |
| IN – DS174 *    | 4                              | 16 parameters   |   |
| OUT – DS174 *   | 4                              |   |   |

\* Structures used to optimize the communication.

For example, it is possible a FFB block with a 60 IN\_DS65 + 60 IN\_DS66 + 60 OUT\_DS65 + 62 OUT\_DS66 combination, totaling 242 flexible parameters, or then 16\_IN\_DS158 + 16 OUT\_DS174 + 100 IN\_DS66 + 110 OUT\_DS66 totaling 242 parameters, or other combination within the limits mentioned in the Table above.

In Auto mode, this block updates the block inputs for the Ladder and sends the Ladder values for the block outputs. In Man mode, the block stops the block update and the user is able to change the output values.

The input/output statuses are mapped in boolean parameters into the Ladder. For the inputs, any status with Good quality (Good Non Cascade or Good Cascade) will be 0 (zero) in the corresponding Ladder status. Status qualities different of Good (Bad or Uncertain) will be 1 (one) in the Ladder. For the outputs, the value 0 (zero) in the Ladder will correspond to Good Non Cascade status and the value 1 (one) will correspond to Bad Non Specific status. The table below summarizes the status:

| FFB Parameter | FF Status [datatype byte]                       | Ladder Status [Datatype boolean] |
|---------------|---|----------------------------------|
| Input         | Good Non Cascade or Good Cascade ( $\geq 128$ ) | 0                                |
|               | Bad or Uncertain ( $< 128$ )                    | 1                                |
| Output        | Good Non Cascade Non Specific Not Limited (128) | 0                                |
|               | Bad Non Specific Not Limited (0)                | 1                                |

The above rules apply to inputs and outputs of DS-65, DS-66 and DS-158 types. For inputs and outputs of DS-174 type the above rules apply to the individual status of each of the 16 variables. The Overall Status is the link communication status of the input, and the block execution mode of the output (Auto-GoodNC:Non Specific:Constant, Man-initial value GoodNC: Non Specific:Constant, O/S-Bad:Out of Service), but in both cases there is no mapping in the ladder variable.

This block is available in the following versions of controllers Firmware and Device Revision:

| Controller | Firmware | Device Revision Base |
|------------|----------|----------------------|
| DF62       | V3_x_x   | 04                   |
| DF63       |          | 03                   |
| DF75       |          | 03                   |
| DF79       | V2_x_x   | 03                   |
| DF89       |          | 02                   |
| DF73       | V3_x_x   | 05                   |
| DF95       |          | 02                   |
| DF97       |          | 02                   |
| DF81       | V2_x_x   | 02                   |

Where, in the Firmware column, x\_x indicates the releases of firmware version. The Device Revision Base column indicates the Device Description (DD base) of the equipment.

#### ATTENTION

Changes of Device Description in controllers are made using the Syscon Exchange procedure, where the links of the FFB will be lost.

#### BLOCK\_ERR

BLOCK\_ERR will be “Out of Service” – when the block is in O/S mode.

#### Supported Modes

O/S, MAN and AUTO

#### Parameters

| Idx | Parameter         | Data Type                    | Valid Range / Options | Default Value | Units | Store/ Mode | Description   |
|-----|-------------------|------------------------------|-----------------------|---------------|-------|-------------|---|
| 1   | ST_REV            | Unsigned16                   |                       | 0             | None  | S           |   |
| 2   | TAG_DESC          | OctString(32)                |                       | Spaces        | Na    | S           |   |
| 3   | STRATEGY          | Unsigned16                   |                       | 0             | None  | S           |   |
| 4   | ALERT_KEY         | Unsigned8                    | 1 to 255              | 0             | None  | S           |   |
| 5   | MODE_BLK          | DS-69                        | O/S, Man, Auto        | O/S           | Na    | S           | See Mode parameter.   |
| 6   | BLOCK_ERR         | Bitstring(2)                 |                       |               | E     | D / RO      |   |
| 7   | ALGORITHM_SEL     | Unsigned32                   |                       | 0             | None  | S           | Not used.   |
| 8   | CONTENTS_REV      | Unsigned32                   |                       |               | None  | S           | Not used.   |
| 9   | FILE_REV          | Unsigned32                   |                       |               | None  | S           | Not used.   |
| 10  | FILE_LOCATOR      | Unsigned32                   |                       |               | None  | S           | Not used.   |
| 11  | UPDATE_EVT        | DS-73                        |                       |               | na    | D           | This alert is generated by any change to the static data.   |
| 12  | BLOCK_ALM         | DS-72                        |                       |               | na    | 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 indicated in the subcode field. The first active alert will set the Active status in the Status attribute. When the Unreported status is removed by the alert reporting task, another block alert can be reported without clearing the Active status, if the subcode has been changed. |
| 13  | OUT_D/IN_D/OUT/IN | DS-65/DS-66<br>DS-158/DS-174 |                       |               |       | D           | The input/output parameters as well as type, name and quantity are determined by user.  |
| ... | ....              | ...                          | ...                   | ...           | ...   | ...         |   |
| 254 | OUT_D/IN_D/OUT/IN | DS-65/DS-66<br>DS-158/DS-174 |                       |               |       | D           |   |

**Legend:** E – Enumerated Parameter; NA – Dimensionless Parameter; RO – Read Only; D – Dynamic; N – Non volatile; S – Static; Gray Background Line: Default Parameters in Syscon

## Modbus Function Blocks

### MBCF – Modbus Configuration

#### Overview

This block allows configuration of several communication parameters of the Modbus protocol.

#### Description

This block allows setting parameters of the communication between DFI302 and Modbus slave devices through Ethernet and serial (EIA-232). User defines rate of transference of data of the serial ports, parity, timeout and number of retransmissions.

#### NOTE

Every time a Modbus parameter is changed it is necessary to set the ON\_APPLY parameter of the MBCF block to "Apply". Otherwise these alterations will not be effective.

User must set ONLY one MBCF block for each device.

#### Modbus Addresses

User must attribute a Modbus address to the DFI302. However this address cannot be the same of any other device in the Modbus network in case it is connected in the serial and Ethernet mean. In this case user must set the parameter DEVICE\_ADDRESS. The default value of this parameter is 247.

In applications where the DFI302 acts as master TCP/IP user will have also to inform the IP address of the devices in the parameter SLAVE\_ADRESSES.

#### Parameters MASTER\_SLAVE and MEDIA

These parameters set the DFI302 behavior and media where the communication is done. In the MASTER\_SLAVE parameter is defined if the DFI302 will work as a slave or master Modbus device. MEDIA may be serial or TCP/IP. It is necessary that DEVICE\_ADDRESS is unique within the Modbus network.

#### Rate of transference of the serial ports

It is possible to select the baud rate of data in the serial ports. They may be set through the parameter BAUD\_RATE. It allows the selection among the following baud rates:

- 0:100 bps
- 1:300 bps
- 2:600 bps
- 3:1200 bps
- 4:2400 bps
- 5:4800 bps
- 6:9600 bps (default)
- 7:19200 bps
- 8:38400 bps
- 9:57600 bps
- 10:115200 bps

#### Parity

Parameter PARITY defines the type o parity to the serial ports.

- 0: No parity
- 1: Even Parity
- 2: Odd parity

#### Timeout, number of retransmissions

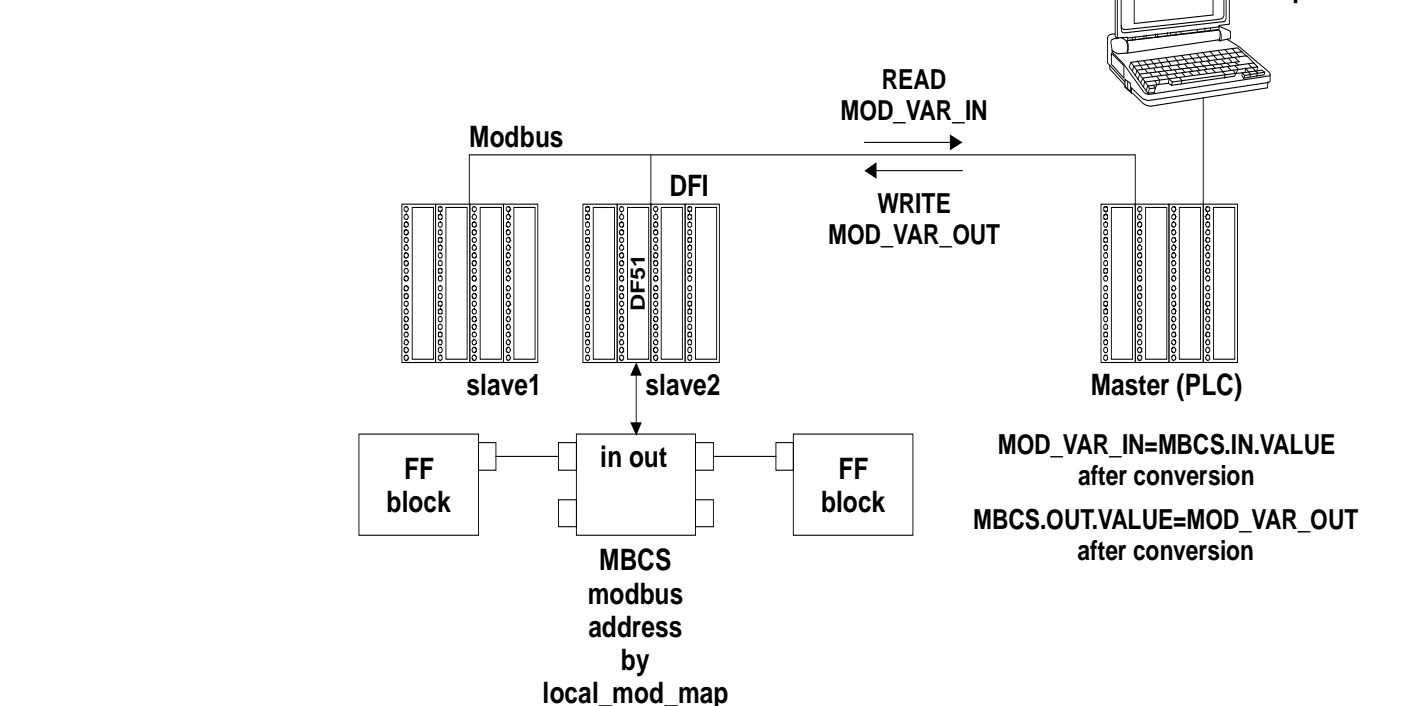
Timeout is the time waited for answer from a slave after a message having been sent to the serial port or Ethernet. The default value is 1000 ms, this parameter is directly connected with the parameter NUMBER\_RETRANSMISSIONS.

Number of retransmissions is the number of times the DFI302 will retry to establish communication with the slave device after not getting a reply. The time waited for this answer is set by the TIME\_OUT parameter. The number of retransmissions is chosen through the NUMBER OF RETRANSMISSIONS parameter. User may select a value in the range 0 to 255 to this parameter. The default value is 1.

## Parameters

| Idx | Parameter               | Data Type (length) | Valid Range/ Options  | Default Value | Units | Store/ Mode | Description  |
|-----|-------------------------|--------------------|---|---------------|-------|-------------|--|
| 1   | ST_REV                  | Unsigned16         |   | 0             | None  | S/RO        |  |
| 2   | TAG_DESC                | OctString(32)      |   | Spaces        | Na    | S           |  |
| 3   | STRATEGY                | Unsigned16         |   | 0             | None  | S           |  |
| 4   | ALERT_KEY               | Unsigned8          | 1 to 255  | 0             | None  | S           |  |
| 5   | MODE_BLK                | DS-69              |   | O/S           | Na    | S           | See Mode Parameter   |
| 6   | BLOCK_ERR               | Bitstring(2)       |   |               | E     | D / RO      |  |
| 7   | MEDIA                   | Unsigned8          | 0:Serial, 1:TCP/IP  | Serial        | E     | S           | Define the type of Modbus channel.   |
| 8   | MASTER_SLAVE            | Unsigned8          | 0:Master, 1:Slave   | Slave         | E     | S           | Define if DFI is master or slave.  |
| 9   | DEVICE_ADDRESS          | Unsigned8          | 1-247   | 1             | E     | S           | Define the DFI Modbus address (only for DFI slave).  |
| 10  | BAUD_RATE               | Unsigned8          | 0:110, 1:300, 2:600, 3:1200, 4:2400, 5:4800, 6:9600, 7:19200, 8:38400, 9:57600, 10:115200 | 19200         | E     | S           | Define the baud rate (only for media serial).  |
| 11  | STOP_BITS               | Unsigned8          | 0:1, 1:2  | 1             | E     | S           | Define the number of stop bits (only for media serial).  |
| 12  | PARITY                  | Unsigned8          | 0:None, 1:Even, 2:Odd.  | Even          | E     | S           | Define the parity (only for media serial).   |
| 13  | TIMEOUT                 | Unsigned16         | 200-65535   | 2000          | ms    | S           | Time value to wait a response from a slave (for DFI master), or time to wait the OUTs are updated (for DFI slave).   |
| 14  | NUMBER_RETRANS MISSIONS | Unsigned8          | 0-255   | 1             |       | S           | Number of retransmission if DFI doesn't receive response from slave.   |
| 15  | SLAVE_ADDRESSES         | DS-263             |   |               |       | S           | IP number and Modbus addresses of slaves ( only for DFI master in TCP/IP media);   |
| 16  | RESTART_MODBUS          | Boolean            |   | FALSE         |       | S           | Not used.  |
| 17  | TIME_TO_RESTART         | Unsigned16         | 100-65535 (Master)<br>0-65535 (Slave)   | 500           | ms    | S           | When the device is working as master, it is the scan cycle of Modbus requests. When the device is working as slave, it is the delay for responses in Modbus TCP, aimed at restricting too short cycle by the master. |
| 18  | RTS_CTS                 | Boolean            |   | FALSE         |       | S           | Enable or not handshaking.   |
| 19  | ON_APPLY                | Unsigned8          | 0:None, 1: Apply  | None          | E     | S           | <b>Apply the changes made in the Modbus blocks.</b>  |
| 20  | CHECK_COMM_STANDBY      | Unsigned8          | 0 ~ 255   | 0             | NA    | S / RW      | It configures for the Standby if it will test the Modbus communication with the slave devices.<br><br>0: Disable test.<br>1 – 255: Enable test defining the time interval between each test (s).                     |

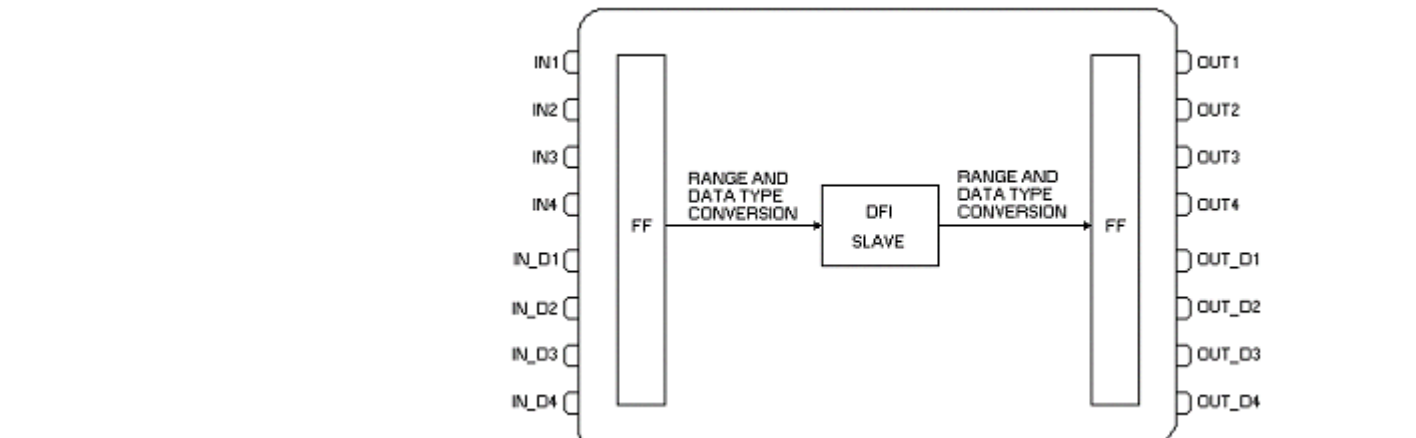
**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
Gray Background Line: Default Parameters of Syscon



MBCS block generates a communication strategy between a Modbus master and a FOUNDATION fieldbus slave. In the present case, the slave is the Smar's linking device DFI302 that has slave behavior for the Modbus network. It allows Modbus variables to be associated with fieldbus variables and data between these two "worlds" to be exchanged through DFI302.

**NOTE**

Every time a Modbus parameter is changed it is necessary to set the ON\_APPLY parameter of the MBCF block to "Apply". Otherwise these alterations will not be effective.



### Inputs and Outputs

This block has 4 digital inputs, 4 analog inputs, 4 digital outputs and 4 analog outputs that may be connected to other fieldbus function blocks or to the Modbus world.

- IN1, IN2, IN3 and IN4 are analog inputs.
- IN\_D1, IN\_D2, IN\_D3 and IN\_D4 are digital inputs.
- OUT1, OUT2, OUT3 and OUT4 are analog outputs.
- OUT\_D1, OUT\_D2, OUT\_D3 and OUT\_D4 are digital outputs.

Digital outputs and Digital inputs are of the DS-66 data type. Thus they contain both a Status and a value (both Unsigned 8). The analog outputs and inputs are of the DS-65 data type, containing status and value as well. Type of values is Float. For more details (see on Chapter 1, "Data Type and Data Structure Definition")

### Parameter LOCAL\_MOD\_MAP

This parameter defines the address range of the Modbus addresses attributed to the input and output fieldbus variables of the MBCS block. In order to set this parameter properly user needs first to check the tables below:

| LOCAL_MOD_MAP (MBCS) |  |                             |
|----------------------|--|-----------------------------|
| PARAMETER            | LOCAL_MOD_MAP = x<br>OFFSET = 40 * x<br>x = 0 ~ 15 | e.g.<br>LOCAL_MOD_MAP<br>=1 |
| IN1-Value            | 40001+ OFFSET<br>40002+ OFFSET                     | 40041<br>40042              |
| IN2-Value            | 40003+ OFFSET<br>40004+ OFFSET                     | 40043<br>40044              |
| IN3-Value            | 40005+ OFFSET<br>40006+ OFFSET                     | 40045<br>40046              |
| IN4-Value            | 40007+ OFFSET<br>40008+ OFFSET                     | 40047<br>40048              |
| OUT1-Value           | 40009+ OFFSET<br>40010+ OFFSET                     | 40049<br>40050              |
| OUT2-Value           | 40011+ OFFSET<br>40012+ OFFSET                     | 40051<br>40052              |
| OUT3-Value           | 40013+ OFFSET<br>40014+ OFFSET                     | 40053<br>40054              |
| OUT4-Value           | 40015+ OFFSET<br>40016+ OFFSET                     | 40055<br>40056              |
| IN1-Status           | 40017+ OFFSET                                      | 40057                       |
| IN2-Status           | 40018+ OFFSET                                      | 40058                       |
| IN3-Status           | 40019+ OFFSET                                      | 40059                       |
| IN4-Status           | 40020+ OFFSET                                      | 40060                       |
| OUT1-Status          | 40021+ OFFSET                                      | 40061                       |
| OUT2-Status          | 40022+ OFFSET                                      | 40062                       |
| OUT3-Status          | 40023+ OFFSET                                      | 40063                       |
| OUT4-Status          | 40024+ OFFSET                                      | 40064                       |
| IN_D1-Status         | 40025+ OFFSET                                      | 40065                       |
| IN_D2-Status         | 40026+ OFFSET                                      | 40066                       |
| IN_D3-Status         | 40027+ OFFSET                                      | 40067                       |
| IN_D4-Status         | 40028+ OFFSET                                      | 40068                       |
| OUT_D1-Status        | 40029+ OFFSET                                      | 40069                       |
| OUT_D2-Status        | 40030+ OFFSET                                      | 40070                       |
| OUT_D3-Status        | 40031+ OFFSET                                      | 40071                       |
| OUT_D4-Status        | 40032+ OFFSET                                      | 40072                       |
| IN_D1-Value          | 1+ OFFSET  | 41                          |
| IN_D2-Value          | 2+ OFFSET  | 42                          |
| IN_D2-Value          | 3+ OFFSET  | 43                          |



| LOCAL_MOD_MAP (MBCS) |           |    |
|----------------------|-----------|----|
| IN_D2-Value          | 4+ OFFSET | 44 |
| OUT_D1-Value         | 5+ OFFSET | 45 |
| OUT_D2-Value         | 6+ OFFSET | 46 |
| OUT_D3-Value         | 7+ OFFSET | 47 |
| OUT_D4-Value         | 8+ OFFSET | 48 |

The second column of the table above shows the Modbus address attributed to each input and output of the MBCS block, according to the value set for LOCAL\_MODE\_MAP. The address has the following equation:

$$\begin{aligned}\text{LOCAL\_MOD\_MAP} &= X \\ \text{OFFSET} &= 40 * X\end{aligned}$$

Where, X is the value of the LOCAL\_MOD\_MAP parameter that is specified by user for each MBCS block, whereas the value must be from 0 to 15.

For example, if LOCAL\_MOD\_MAP equal to 0, it will result in the Modbus addresses range shown in the second column with OFFSET equal to zero. For LOCAL\_MOD\_MAP equal to 1, it will result in OFFSET=40 and it corresponds to the Modbus addresses shown in the third column from the table above. Like this, different values of LOCAL\_MOD\_MAP will result in a different Modbus addresses range.

From the table above, the analog values (IN\_1, ..., IN\_4, OUT\_1, ..., OUT\_4) are float of 4 bytes and have two Modbus registers. However, to configure this address in the Modbus master, it is necessary only to write the first address. For example, the address of the OUT\_1 variable of the MBCS block with LOCAL\_MOD\_MAP equal to 1 is 40049. The IN\_Dn and OUT\_Dn discrete values have only one Modbus register (for example the Modbus address of IN\_D1 for LOCAL\_MOD\_MAP equal to 1 is 41). The status values also use only one register.

This block allows Modbus Scale Conversion. To do the conversion procedure, see the item "Modbus Scale Conversion" in the Section 1.

### Output Status

The output status can be set in two ways: by Modbus master or by status chosen by user. In the first case the Modbus master sends the value and status (i.e., two different Modbus variables). In the second case the master sends only the value.

When the output status is updated by master, the STATUS\_OUT\_Dn or SCALE\_CONV\_OUTn.Output\_Status parameter is equal to "set by master", that is the default option of the block. In this case, the Modbus master sends the status periodically, writing a fieldbus status in the status parameter (OUTn.Status). For example, to write the "Good Non Cascade; Non Specific" status whose value is 128 (see section status composition) in the OUT\_D1 parameter of the MBCS block with LOCAL\_MOD\_MAP equal to 1, it is necessary the master to write 128 in the 40069 Modbus address. For this case, the master also should write in the 45 Modbus address corresponding to the parameter value (OUT\_D1.Value).

If the user configures a value different of "Set by master" for the STATUS\_OUT\_Dn or SCALE\_CONV\_OUTn.Output\_Status parameters, then this status will be the status of the parameter for each master update. I.e., for the example above, if the user configures the STATUS\_OUT\_D1 parameter equal to "Good Non Cascade; Non Specific", and whenever the master to write a new value for the OUT\_D1.VALUE parameter (45 Modbus address), the status will be the status corresponding to the STATUS\_OUT\_D1 parameter.

For the two ways, if the outputs are not updated by Modbus Master during the time interval set in TIMEOUT (TIMEOUT parameter in the MBCF block), a "bad status" will be generated (Bad:NoComm\_withusablevalue).

To choose the best option of status observes the following sceneries:

- If for the logic is necessary a treatment more complete of fieldbus status, as logic of cascade treatment, failure type, etc. In these cases, the status set by master should be used.
- For simple logics of status verification only for Modbus communication, i.e., only to know if the data are being updated by master, then the status chosen by user should be used.

**BLOCK\_ERR**

The BLOCK\_ERR of the MBCS block will reflect the following causes:

- Other: it occurs when the conversion from Y to DATA\_TYPE\_IN results in a value out of range of this data type.
- Out of Service: it occurs when the block is in O/S mode.

**Parameters**

| Idx | Parameter       | DataType (length) | Valid Range/ Options   | Default Value | Units | Store / Mode | Description   |
|-----|-----------------|-------------------|--|---------------|-------|--------------|---|
| 1   | ST_REV          | Unsigned16        |  | 0             | None  | S/RO         |   |
| 2   | TAG_DESC        | OctString(32)     |  | Spaces        | Na    | S            |   |
| 3   | STRATEGY        | Unsigned16        |  | 0             | None  | S            |   |
| 4   | ALERT_KEY       | Unsigned8         | 1 to 255   | 0             | None  | S            |   |
| 5   | MODE_BLK        | DS-69             |  | O/S           | Na    | S            | See Mode Parameter.   |
| 6   | BLOCK_ERR       | Bitstring(2)      |  |               | E     | D / RO       |   |
| 7   | LOCAL_MOD_MAP   | Unsigned8         | 0 to 15  | 0             |       | S / O/S      | Define the Modbus addresses.  |
| 8   | IN1             | DS-65             |  |               |       | N            | Analog input 1  |
| 9   | SCALE_CONV_IN1  | DS-256            |  |               |       | S / O/S      | Information to generate constants A and B in equation $Y=A*X+B$ .   |
| 10  | IN2             | DS-65             |  |               |       | N            | Analog input 2  |
| 11  | SCALE_CONV_IN2  | DS-256            |  |               |       | S / O/S      | Information to generate constants A and B in equation $Y=A*X+B$ .   |
| 12  | IN3             | DS-65             |  |               |       | N            | Analog input 3  |
| 13  | SCALE_CONV_IN3  | DS-256            |  |               |       | S / O/S      | Information to generate constants A and B in equation $Y=A*X+B$ .   |
| 14  | IN4             | DS-65             |  |               |       | N            | Analog input 4  |
| 15  | SCALE_CONV_IN4  | DS-256            |  |               |       | S / O/S      | Information to generate constants A and B in equation $Y=A*X+B$ .   |
| 16  | IN_D1           | DS-66             |  |               |       | N            | Discrete input 1  |
| 17  | IN_D2           | DS-66             |  |               |       | N            | Discrete input 2  |
| 18  | IN_D3           | DS-66             |  |               |       | N            | Discrete input 3  |
| 19  | IN_D4           | DS-66             |  |               |       | N            | Discrete input 4  |
| 20  | OUT1            | DS-65             |  |               |       | N / Man      | Analog output 1   |
| 21  | SCALE_CONV_OUT1 | DS-257            | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |               |       | S / O/S      | Information to generate constants A and B in equation $Y=A*X+B$ plus output status.<br><br>Define the status for the OUT1 parameter. If the option is “Set by Master”, the status will be sent by master. If the option is different of “Set by Master”, the status of the parameter will be the SCALE_CONV_OUT1 when the master is sending the value (OUT1.Value). |
| 22  | OUT2            | DS-65             |  |               |       | N / Man      | Analog output 2   |

| Idx | Parameter       | Data Type<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store /<br>Mode | Description   |
|-----|-----------------|-----------------------|--|------------------|-------|-----------------|---|
| 23  | SCALE_CONV_OUT2 | DS-257                | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Information to generate constants A and B in equation $Y=A*X+B$ plus output status.<br>Define the status for the OUT2 parameter. If the option is “Set by Master”, the status will be sent by master. If the option is different of “Set by Master”, the status of the parameter will be the SCALE_CONV_OUT2 when the master is sending the value (OUT2.Value). |
| 24  | OUT3            | DS-65                 |  |                  |       | N / Man         | Analog output 3   |
| 25  | SCALE_CONV_OUT3 | DS-257                | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Information to generate constants A and B in equation $Y=A*X+B$ plus output status.<br>Define the status for the OUT3 parameter. If the option is “Set by Master”, the status will be sent by master. If the option is different of “Set by Master”, the status of the parameter will be the SCALE_CONV_OUT3 when the master is sending the value (OUT3.Value). |
| 26  | OUT4            | DS-65                 |  |                  |       | N / Man         | Analog output 4   |
| 27  | SCALE_CONV_OUT4 | DS-257                | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Information to generate constants A and B in equation $Y=A*X+B$ plus output status.<br>Define the status for the OUT4 parameter. If the option is “Set by Master”, the status will be sent by master. If the option is different of “Set by Master”, the status of the parameter will be the SCALE_CONV_OUT4 when the master is sending the value (OUT4.Value). |
| 28  | OUT_D1          | DS-66                 |  |                  |       | N / Man         | Discrete output 1   |

| Idx | Parameter     | DataType<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|---------------|----------------------|--|------------------|-------|-----------------|--|
| 29  | STATUS_OUT_D1 | Unsigned8            | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Define the status for the OUT_D1 parameter. If the option is "Set by Master", the status will be sent by master. If the option is different of "Set by Master", the status of the parameter will be the STATUS_OUT_D1 when the master is sending the value (OUT_D1.Value). |
| 30  | OUT_D2        | DS-66                |  |                  |       | N / Man         | Discrete output 2  |
| 31  | STATUS_OUT_D2 | Unsigned8            | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Define the status for the OUT_D2 parameter. If the option is "Set by Master", the status will be sent by master. If the option is different of "Set by Master", the status of the parameter will be the STATUS_OUT_D2 when the master is sending the value (OUT_D2.Value). |
| 32  | OUT_D3        | DS-66                |  |                  |       | N / Man         | Discrete output 3  |
| 33  | STATUS_OUT_D3 | Unsigned8            | 0 – Bad: Non Specific<br>.<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>.<br>192 – Good Cascade: Non Specific<br>.<br>.<br>252 – Set by Master |                  |       | S / O/S         | Define the status for the OUT_D3 parameter. If the option is "Set by Master", the status will be sent by master. If the option is different of "Set by Master", the status of the parameter will be the STATUS_OUT_D3 when the master is sending the value (OUT_D3.Value). |
| 34  | OUT_D4        | DS-66                |  |                  |       | N / Man         | Discrete output 4  |

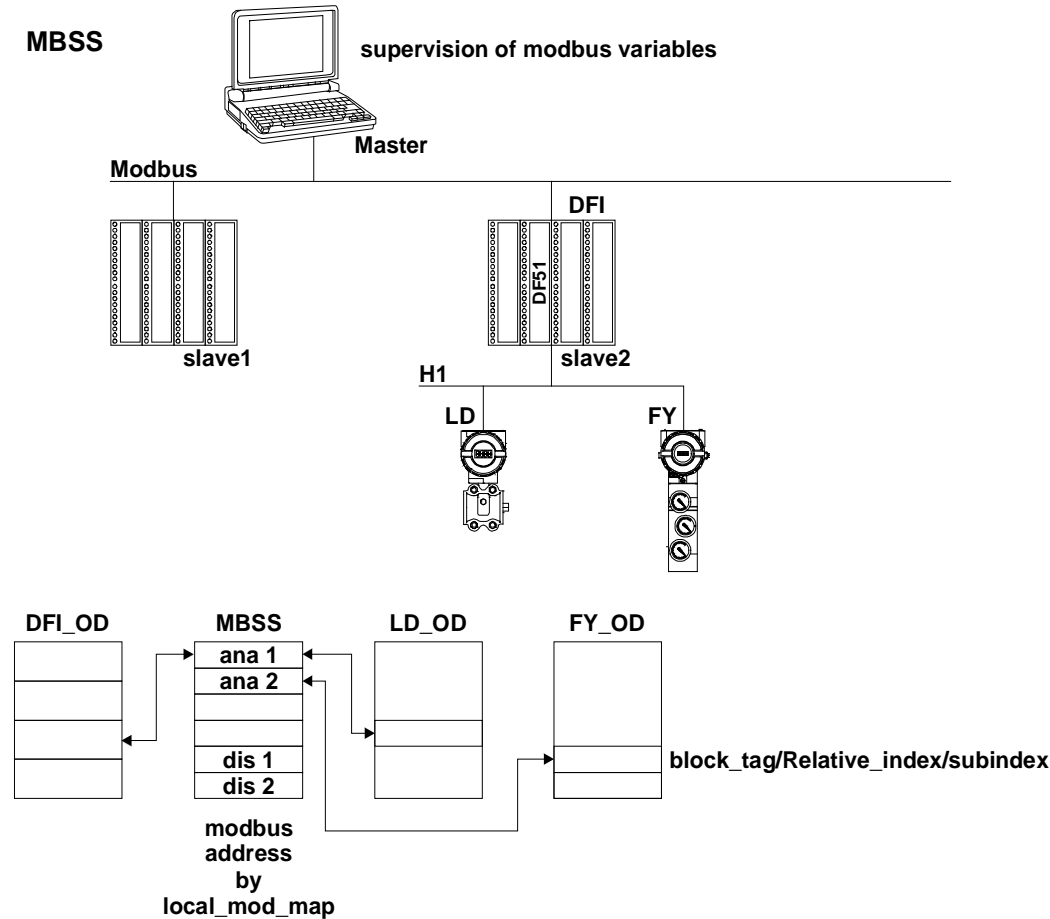
| Idx | Parameter     | DataType<br>(length) | Valid Range/<br>Options   | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|---------------|----------------------|---|------------------|-------|-----------------|--|
| 35  | STATUS_OUT_D4 | Unsigned8            | 0 – Bad: Non Specific<br>.<br>128 – Good Non Cascade: Non Specific<br>.<br>192 – Good Cascade: Non Specific<br>.<br>252 – Set by Master |                  |       | S / O/S         | Define the status for the OUT_D4 parameter. If the option is "Set by Master", the status will be sent by master. If the option is different of "Set by Master", the status of the parameter will be the STATUS_OUT_D4 when the master is sending the value (OUT_D4.Value).   |
| 36  | UPDATE_EVT    | DS-73                |   |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 37  | BLOCK_ALM     | DS-72                |   |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static

**Gray Background Line: Default Parameters of Syscon**

## MBSS – Modbus Supervision Slave

### Overview



### Description

MBSS block generates a communication strategy between a Modbus master and a FOUNDATION fieldbus slave. In the present case, the slave is the Smar's linking device DFI302 that has slave behavior for the Modbus network. The MBSS block allows that fieldbus variables are monitored. Unlike the MBCS block, the MBSS does not have inputs or outputs that may be connected. In another words, links to other function blocks cannot be made. It will allow only the Modbus master to monitor specific variables set. For example, suppose there is a PID function block in a FIELDBUS control strategy and it is required to visualize this value in the Modbus master. With the MBSS this value may be monitored.

### NOTE

Every time a Modbus parameter is changed it is necessary to set the ON\_APPLY parameter of the MBCF block to "Apply". Otherwise these alterations will not be effective.

### I\_IDn, F\_IDn, B\_IDn parameters

I\_IDn are integer variables, F\_IDn are float variables and D\_IDn refers to Boolean variables. These parameters are of the DS-262 data type, which has 3 elements, and their descriptions are in the Chapter 1 "Data Type and Structure Definitions".

### LOCAL\_MOD\_MAP parameter

This parameter will attribute Modbus address to the variables you need to monitor. See the following table:

| LOCAL_MOD_MAP (MBSS) |  |                             |
|----------------------|--|-----------------------------|
| PARAMETER            | LOCAL_MOD_MAP = x<br>OFFSET = 40 * x<br>x = 0 ~ 15 | e.g.<br>LOCAL_MOD_MAP<br>=1 |
| FVALUE1              | 42601+ OFFSET<br>42602+ OFFSET                     | 42641<br>42642              |
| FVALUE2              | 42603+ OFFSET<br>42604+ OFFSET                     | 42643<br>42644              |
| FVALUE3              | 42605+ OFFSET<br>42606+ OFFSET                     | 42645<br>42646              |
| FVALUE4              | 42607+ OFFSET<br>42608+ OFFSET                     | 42647<br>42648              |
| FVALUE5              | 42609+ OFFSET<br>42610+ OFFSET                     | 42649<br>42650              |
| FVALUE6              | 42611+ OFFSET<br>42612+ OFFSET                     | 42651<br>42652              |
| FVALUE7              | 42613+ OFFSET<br>42614+ OFFSET                     | 42653<br>42654              |
| FVALUE8              | 42615+ OFFSET<br>42616+ OFFSET                     | 42655<br>42656              |
| IVALUE1              | 42617+ OFFSET<br>42618+ OFFSET                     | 42657<br>42658              |
| IVALUE2              | 42619+ OFFSET<br>42620+ OFFSET                     | 42659<br>42660              |
| IVALUE3              | 42621+ OFFSET<br>42622+ OFFSET                     | 42661<br>42662              |
| IVALUE4              | 42623+ OFFSET<br>42624+ OFFSET                     | 42663<br>42664              |
| BVALUE1              | 2601+ OFFSET                                       | 2641                        |
| BVALUE2              | 2602+ OFFSET                                       | 2642                        |
| BVALUE3              | 2603+ OFFSET                                       | 2643                        |
| BVALUE4              | 2604+ OFFSET                                       | 2644                        |
| BAD_STATUS           | 42625+OFFSET                                       | 42665                       |

LOCAL\_MOD\_MAP= X  
OFFSET = 40\*X

Once values for LOCAL\_MOD\_MAP are set, MODBUS ADDRESSES are given to the variables you wish to monitor. So, each integer, float or Boolean variable will have a Modbus address associated.

For example, suppose LOCAL\_MOD\_MAP = 1 and a float value will be monitored. Picking the F\_ID1 and setting its parameters, we have:

F\_ID1.Tag = Tag of the float parameter necessary to monitor

F\_ID1.Index= Index of the first column of the parameter necessary to monitor.

F\_ID1.subindex = The sub index is used for parameters that have a structure. In this case it is necessary to indicate which element of the structure is being referred.

See the table above The Modbus addresses given to this parameter (remember, float values use two Modbus registers) are 42641 and 42642.

#### **BAD\_STATUS Parameter**

This parameter indicates if the Fieldbus communication is OK or no. If the correspondent bit is in logic level 1 this means there was an error during writing/reading of the respective parameter. The table below presents the values for these status values.

#### **Relation between the bits in BAD\_STATUS and Modbus addresses**

| BIT | PARAMETER |
|-----|-----------|
| 0   | FVALUE1   |
| 1   | FVALUE2   |

| BIT | PARAMETER |
|-----|-----------|
| 2   | FVALUE3   |
| 3   | FVALUE4   |
| 4   | FVALUE5   |
| 5   | FVALUE6   |
| 6   | FVALUE7   |
| 7   | FVALUE8   |
| 8   | IVALUE1   |
| 9   | IVALUE2   |
| 10  | IVALUE3   |
| 11  | IVALUE4   |
| 12  | BVALUE1   |
| 13  | BVALUE2   |
| 14  | BVALUE3   |
| 15  | BVALUE4   |

**BLOCK\_ERR**

The BLOCK\_ERR of the MBSS block will reflect the following causes:

- Block Configuration Error: If it is requested a tag with a data type different from permitted or invalid or not found block tag;
- Out of Service: it occurs when the block is in O/S mode.

**Remarks**

BVALUEx parameters can address FF block parameters of the following data types: Boolean, integer8 and unsigned8. Those data types are automatically converted to bit (0 or 1) and vice versa for Modbus supervision and also converted to Boolean parameter (BVALUEx).

IVALUEx parameters can address FF block parameters of the following data types: Integer8, Integer16, Integer32, Unsigned8, Unsigned16 and Unsigned32.

Each analog parameter (IVALUEx) is mapping as two analog registers in Modbus, i.e., four bytes. Thus, when addressing a FF block parameter with one or two bytes, such parameter will be promoted to Unsigned32 or Integer32.

If Relative Index = 5 (MODE\_BLK) e Sub Index = 0, it is performed a writing in Sub Index 1 and a reading in Sub Index 2.

**Data Type and Supported Structures by MBSS**

The Modbus supervision blocks (MBSS) used in the controllers configured as slave, have some restrictions about data types and structures that they support when supervising the block parameters tags. In such case, the next table shows the data types and structures which can be monitored by the MBSS block.

| DATA TYPE * | STRUCTURE TYPES |
|-------------|-----------------|
| Boolean     | DS-65           |
| Float       | DS-66           |
| Unsigned 8  | DS-68           |
| Unsigned 16 | DS-69           |
| Unsigned 32 | DS-71           |
| Integer8    | DS-72           |
| Integer16   | DS-74           |
| Integer32   | DS-159 (DC302)  |
|             | DS-160 (DC302)  |

\*For the DF51 controller the data types Swapped Float and Swapped Integer can be obtained by setting the RTS\_CTS parameter to TRUE value in the MBSS block.

For further information about the blocks parameters and their data and structures types, as mentioned in the previous table, please see in this manual the Data Type field of each table presented.



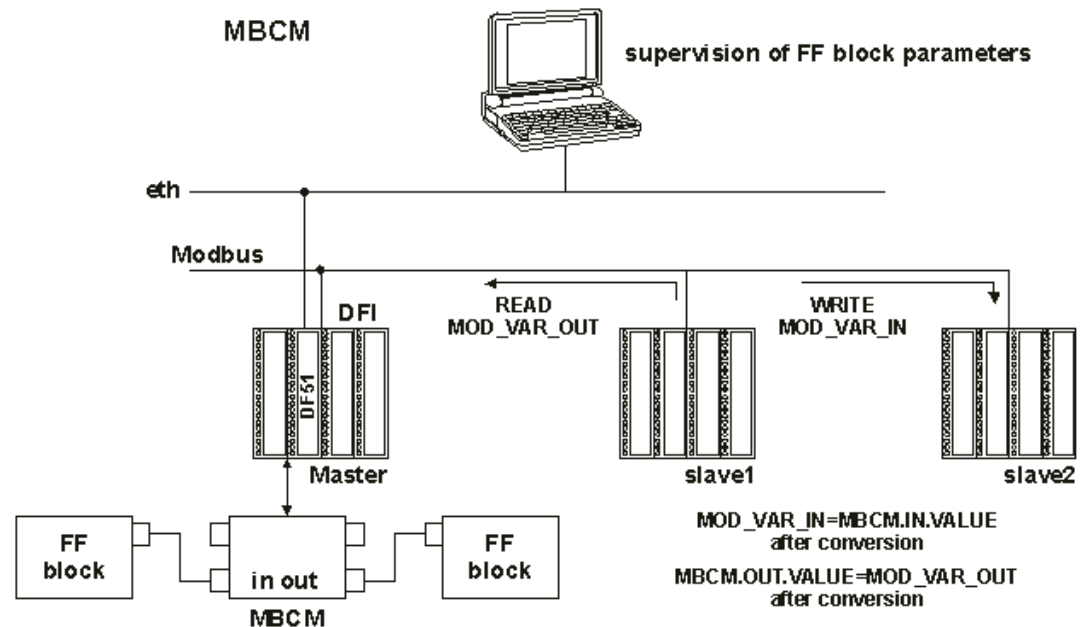
## Parameters

| Idx | Parameter     | Data Type<br>(length) | Valid<br>Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|---------------|-----------------------|----------------------------|------------------|-------|-----------------|--|
| 1   | ST_REV        | Unsigned16            |                            | 0                | None  | S/RO            |  |
| 2   | TAG_DESC      | OctString(32)         |                            | Spaces           | Na    | S               |  |
| 3   | STRATEGY      | Unsigned16            |                            | 0                | None  | S               |  |
| 4   | ALERT_KEY     | Unsigned8             | 1 to 255                   | 0                | None  | S               |  |
| 5   | MODE_BLK      | DS-69                 |                            | O/S              | Na    | S               | See Mode Parameter   |
| 6   | BLOCK_ERR     | Bitstring(2)          |                            |                  | E     | D / RO          |  |
| 7   | LOCAL_MOD_MAP | Unsigned8             | 0 to 15                    | 0                |       | S / O/S         | Define the Modbus addresses.   |
| 8   | F_ID1         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 9   | FVALUE1       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 10  | F_ID2         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 11  | FVALUE2       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 12  | F_ID3         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 13  | FVALUE3       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 14  | F_ID4         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 15  | FVALUE4       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 16  | F_ID5         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 17  | FVALUE5       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 18  | F_ID6         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 19  | FVALUE6       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 20  | F_ID7         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 21  | FVALUE7       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 22  | F_ID8         | DS-262                |                            |                  |       | S / O/S         | Information to locate float parameter.   |
| 23  | FVALUE8       | Float                 |                            | 0                |       | N               | Value from requested float parameter.  |
| 24  | I_ID1         | DS-262                |                            |                  |       | S / O/S         | Information to locate integer parameter.   |
| 25  | IVALUE1       | Integer32             |                            | 0                |       | N               | Value from requested integer parameter.  |
| 26  | I_ID2         | DS-262                |                            |                  |       | S / O/S         | Information to locate integer parameter.   |
| 27  | IVALUE2       | Integer32             |                            | 0                |       | N               | Value from requested integer parameter.  |
| 28  | I_ID3         | DS-262                |                            |                  |       | S / O/S         | Information to locate integer parameter.   |
| 29  | IVALUE3       | Integer32             |                            | 0                |       | N               | Value from requested integer parameter.  |
| 30  | I_ID4         | DS-262                |                            |                  |       | S / O/S         | Information to locate integer parameter.   |
| 31  | IVALUE4       | Integer32             |                            | 0                |       | N               | Value from requested integer parameter.  |
| 32  | B_ID1         | DS-262                |                            |                  |       | S / O/S         | Information to locate boolean parameter.   |
| 33  | BVALUE1       | Boolean               |                            | TRUE             |       | N               | Value from requested boolean parameter.  |
| 34  | B_ID2         | DS-262                |                            |                  |       | S / O/S         | Information to locate boolean parameter.   |
| 35  | BVALUE2       | Boolean               |                            | TRUE             |       | N               | Value from requested boolean parameter.  |
| 36  | B_ID3         | DS-262                |                            |                  |       | S / O/S         | Information to locate boolean parameter.   |
| 37  | BVALUE3       | Boolean               |                            | TRUE             |       | N               | Value from requested boolean parameter.  |
| 38  | B_ID4         | DS-262                |                            |                  |       | S / O/S         | Information to locate boolean parameter.   |
| 39  | BVALUE4       | Boolean               |                            | TRUE             |       | N               | Value from requested boolean parameter.  |
| 40  | UPDATE_EVT    | DS-73                 |                            |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 41  | BLOCK_ALM     | DS-72                 |                            |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 42  | BAD_STATUS    | BitString (2)         |                            |                  |       | D/RO            | This parameter indicates if the status of correspondent variable is bad or not.  |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
 Gray Background Line: Default Parameters of Syscon

## MBCM – Modbus Control Master

### Overview



### Description

This block allows control of communication in a strategy where the DFI302 is a Modbus master and the slaves may exchange data between them and with the DFI302. With this block it is not only possible to read Modbus variables, but also writing variables in the Modbus world, exchange data and interact with the FOUNDATION fieldbus control strategy.

### NOTE

Every time a Modbus parameter is changed it is necessary to set the ON\_APPLY parameter of the MBCM block to "Apply". Otherwise these alterations will not be effective.

### LOCAL\_MOD\_MAP parameter

All MBCM blocks added to the strategy must have different values for LOCAL\_MOD\_MAP. Otherwise the block will not work properly.

### Inputs and Outputs

This block has 4 digital inputs and outputs and 4 analog inputs and outputs. These inputs and outputs may be connected to other fieldbus function blocks in order to be connected to Modbus I/O modules or registers

- INn: Analog input. DS-65 Data type. Value and Status. (See on Chapter 1, "Data Type and Data Structure Definition"). In this parameter user will visualize the value of the parameter set for this input and its status.
- IN\_Dn: Digital input. DS-66 Data type. Value and Status. (See on Chapter 1, "Data Type and Data Structure Definition"). In this parameter user will visualize the value of the parameter set for this input and its status
- OUTn: Analog output. DS-65 Data type. Value and Status. (See on Chapter 1, "Data Type and Data Structure Definition"). In this parameter user will visualize the value of the parameter set for this output and its status.
- OUT\_Dn: Digital output. DS-66 Data type. Value and Status. (see on Chapter 1, "Data Type and Data Structure Definition"). In this parameter user will visualize the value of the parameter set for this output and its status

### SCALE\_LOC\_INn and SCALE\_LOC\_OUTn

These parameters are of the DS-259 data type. They both convert the value to Engineering Units and address the variable in the Modbus network. The INn and OUTn inputs and outputs have SCALE\_LOC\_INn and SCALE\_LOC\_OUTn parameters associated. It is necessary to set these parameters so the monitoring and data exchanges are properly made.

Each one of these parameters consists of the following elements:

- √ From Eu 100 %
- √ From Eu 0 %
- √ To Eu 100 %
- √ To Eu 0 %
- √ Data Type
- √ Slave Address
- √ Modbus Address of Value
- √ Modbus Address of Status

This block allows Modbus Scale Conversion, to do the conversion procedure see the item “Modbus Scale Conversion” in the Chapter 1 for more details.

The Modbus status is related with the Modbus value. Thus, when the “MODBUS\_ADDRESS\_OF\_STATUS” parameter was configured, it is necessary to configure also “MODBUS\_ADDRESS\_OF\_VALUE”.

The treatment of inputs and outputs are described in the table below:

| Input/Output               | Status Configured<br>(MODBUS_ADDRESS_OF_STATUS ≠ 0)  | Status Not Configured<br>(MODBUS_ADDRESS_OF_STATUS = 0)   |
|----------------------------|--|---|
| Inputs<br>(IN_n , IN_Dn)   | The block sends to the Modbus slave device the status corresponding of its input.(The status has the FF standard format)                                     | No status information is sent to the slave device.  |
| Outputs<br>(OUT_n, OUT_Dn) | The block reads from the slave device the corresponding status. (The block make the interpretation that the Modbus variable is the same format of FF Status) | <ul style="list-style-type: none"> <li>- The block updates the status to “Good Non Cascade” when the communication with the Modbus slave device is ok.</li> <li>- The block update the status to “Bad No Communication with last value” when the communication with the Modbus slave device is not ok.</li> </ul> |

Float values use two Modbus registers, but it is necessary only to inform the first one.

#### Setting the inputs and outputs of the MBCM block

To read a Modbus variable, connect it to an output of the MBCM function block. To write in a Modbus register connect it to an input of the MBCM block.

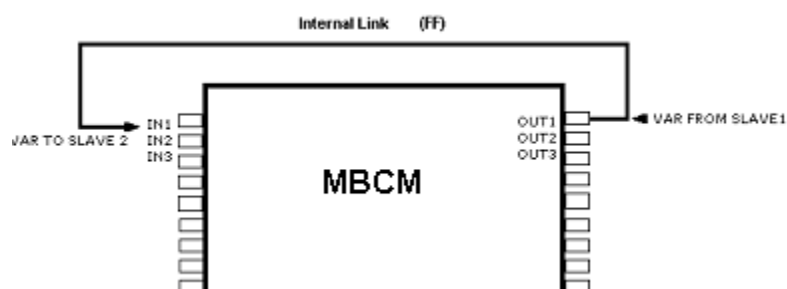
Generally Modbus addresses are:

The standard of the Modbus protocol specifies the division of the address range to the variables.

- 0001 to 9999 => Digital Outputs
- 10001 to 19999 => Digital Inputs.
- 30001 to 39999 => Analog Inputs.
- 40001 to 49999 => Analog Outputs

Once the variables required to be mapped are defined and referenced in the MBCM block it is now possible to set the strategy.

It is possible to connect the variables to other fieldbus function blocks (Connect the output or input of the block to blocks in the strategy), to write in Modbus registers (Connect the Input of the MBCM block to a Modbus register). Exchanging data between two slaves (set the input of the MBCM block with the slave address and specific Modbus address where the value will be written and set the output of the MBCM block with the slave address and Modbus address of the variable where the value will be read). This last application is showed below:



### BAD\_STATUS Parameter

This parameter indicates if the communication between slaves was established properly. If the correspondent bit is in logic level 1 this means there was an error during writing/reading of the respective parameter. The table below presents the values for these status values.

### Relation between the bits in BAD\_STATUS and Modbus addresses

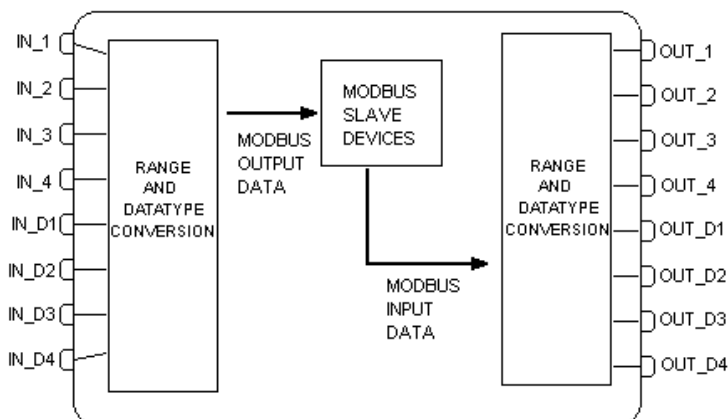
| BIT | PARAMETER |
|-----|-----------|
| 0   | IN1       |
| 1   | IN2       |
| 2   | IN3       |
| 3   | IN4       |
| 4   | IN_D1     |
| 5   | IN_D2     |
| 6   | IN_D3     |
| 7   | IN_D4     |
| 8   | OUT1      |
| 9   | OUT2      |
| 10  | OUT3      |
| 11  | OUT4      |
| 12  | OUT_D1    |
| 13  | OUT_D2    |
| 14  | OUT_D3    |
| 15  | OUT_D4    |

### Remarks

Each bit corresponds to an OR between the value and status, indicating if communication with slave is good or bad.

- If it is only used the value, the status is considered zero.
- If it is only used the status, the value is considered zero.

### Schematic



**BLOCK\_ERR**

The BLOCK\_ERR of the MBCM block will reflect the following causes:

- Other: it occurs when the conversion from Y to DATA\_TYPE\_IN results in a value out of range of this data type.
- Out of Service: it occurs when the block is in O/S mode.

**Parameters**

| Idx | Parameter      | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description   |
|-----|----------------|-----------------------|-------------------------|------------------|-------|-----------------|---|
| 1   | ST_REV         | Unsigned16            |                         | 0                | None  | S/RO            |   |
| 2   | TAG_DESC       | OctString(32)         |                         | Spaces           | Na    | S               |   |
| 3   | STRATEGY       | Unsigned16            |                         | 0                | None  | S               |   |
| 4   | ALERT_KEY      | Unsigned8             | 1 to 255                | 0                | None  | S               |   |
| 5   | MODE_BLK       | DS-69                 |                         | O/S              | Na    | S               | See Mode Parameter  |
| 6   | BLOCK_ERR      | Bitstring(2)          |                         |                  | E     | D / RO          |   |
| 7   | LOCAL_MOD_MAP  | Unsigned8             | 0 to 15                 | 0                |       | S / O/S         | Define the Modbus addresses.  |
| 8   | BAD_STATUS     | Bitstring(2)          |                         | 0                | E     | D / RO          | Indicate if communication from slave is good or not (each bit corresponds to a Modbus variable).      |
| 9   | IN1            | DS-65                 |                         |                  |       | N               | Analog input 1  |
| 10  | SCALE_LOC_IN1  | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 11  | IN2            | DS-65                 |                         |                  |       | N               | Analog input 2  |
| 12  | SCALE_LOC_IN2  | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 13  | IN3            | DS-65                 |                         |                  |       | N               | Analog input 3  |
| 14  | SCALE_LOC_IN3  | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 15  | IN4            | DS-65                 |                         |                  |       | N               | Analog input 4  |
| 16  | SCALE_LOC_IN4  | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 17  | IN_D1          | DS-66                 |                         |                  |       | N               | Discrete input 1  |
| 18  | LOCATOR_IN_D1  | DS-261                |                         |                  |       | S / O/S         | Addresses in a slave device.  |
| 19  | IN_D2          | DS-66                 |                         |                  |       | N               | Discrete input 2  |
| 20  | LOCATOR_IN_D2  | DS-261                |                         |                  |       | S / O/S         | Addresses in a slave device.  |
| 21  | IN_D3          | DS-66                 |                         |                  |       | N               | Discrete input 3  |
| 22  | LOCATOR_IN_D3  | DS-261                |                         |                  |       | S / O/S         | Addresses in a slave device.  |
| 23  | IN_D4          | DS-66                 |                         |                  |       | N               | Discrete input 4  |
| 24  | LOCATOR_IN_D4  | DS-261                |                         |                  |       | S / O/S         | Addresses in a slave device.  |
| 25  | OUT1           | DS-65                 |                         |                  |       | N / Man         | Analog output 1   |
| 26  | SCALE_LOC_OUT1 | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 27  | OUT2           | DS-65                 |                         |                  |       | N / Man         | Analog output 2   |
| 28  | SCALE_LOC_OUT2 | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |
| 29  | OUT3           | DS-65                 |                         |                  |       | N / Man         | Analog output 3   |
| 30  | SCALE_LOC_OUT3 | DS-259                |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device. |

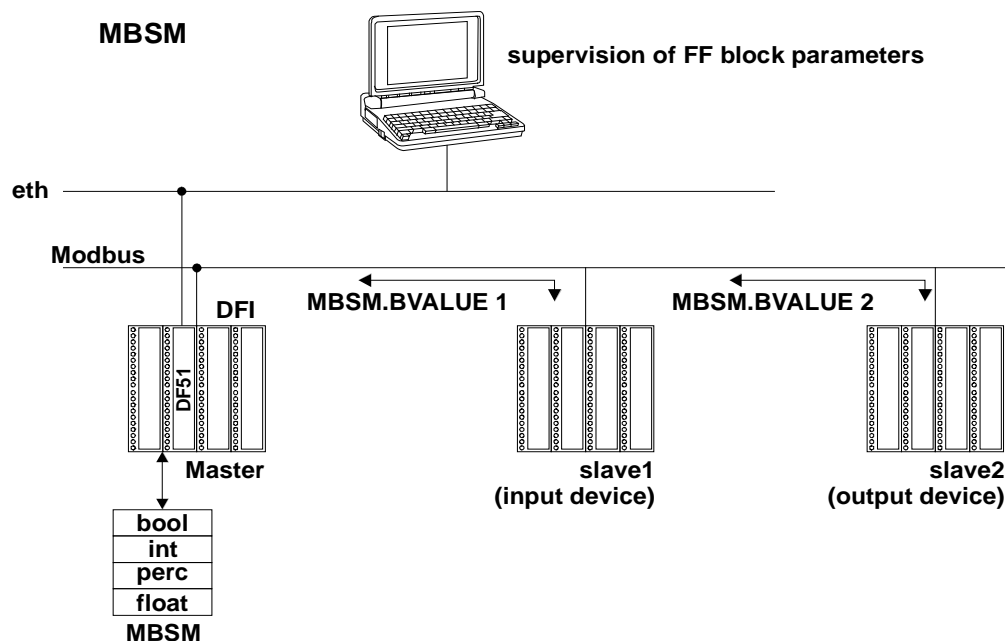
| Idx | Parameter      | DataType<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|----------------|----------------------|-------------------------|------------------|-------|-----------------|--|
| 31  | OUT4           | DS-65                |                         |                  |       | N / Man         | Analog output 4  |
| 32  | SCALE_LOC_OUT4 | DS-259               |                         |                  |       | S / M           | Information to generate constants A and B in equation $Y=A*X+B$ plus the addresses in a slave device.  |
| 33  | OUT_D1         | DS-66                |                         |                  |       | N / Man         | Discrete output 1  |
| 34  | LOCATOR_OUT_D1 | DS-261               |                         |                  |       | S / O/S         | Addresses in a slave device.   |
| 35  | OUT_D2         | DS-66                |                         |                  |       | N / Man         | Discrete output 2  |
| 36  | LOCATOR_OUT_D2 | DS-261               |                         |                  |       | S / O/S         | Addresses in a slave device.   |
| 37  | OUT_D3         | DS-66                |                         |                  |       | N / Man         | Discrete output 3  |
| 38  | LOCATOR_OUT_D3 | DS-261               |                         |                  |       | S / O/S         | Addresses in a slave device.   |
| 39  | OUT_D4         | DS-66                |                         |                  |       | N / Man         | Discrete output 4  |
| 40  | LOCATOR_OUT_D4 | DS-261               |                         |                  |       | S / O/S         | Addresses in a slave device.   |
| 41  | UPDATE_EVT     | DS-73                |                         |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 42  | BLOCK_ALM      | DS-72                |                         |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static

Gray Background Line: Default Parameters of Syscon

## MBSM – Modbus Supervision Master

### Overview



### Description

This block enables the DFI302 to monitor Modbus variables. The DFI302 is the master the slaves contain the Modbus variables desired to be read. Unlike the MBCM this block does not have inputs and outputs that may be connected.

#### NOTE

Every time a Modbus parameter is changed it is necessary to set the ON\_APPLY parameter of the MBCF block to "Apply". Otherwise these alterations will not be effective.

### LOCAL\_MOD\_MAP

All MBCM blocks you add to your strategy must have different values for LOCAL\_MOD\_MAP. Otherwise the block will not work properly.

### Parameters FVALUEn, PVALUEn, IVALUEn and BVALUEn

User may select these parameters according to his needs. If the variable required to be monitored is a float it is necessary to use a FVALUE parameter. If it is a percentage, the PVALUEn will work. IVALUE refers to Integer values and BVALUE refers to Boolean values.

To each of these parameters are associated parameters to address them in the Modbus network so that the MBSM block knows their location.

### Parameter FLOCATORn

It refers to the FVALUEn parameter.

This parameter is of the data type DS-260, so it is required to set two elements for this parameter (see on Chapter 1, "Data Type and Data Structure Definition"):

The FVALUEn parameters will display the values of the variables set in FLOCATORn. Float values use two Modbus registers, but it is necessary only to inform the first one.

### Modbus Addresses

- 0001 to 9999 => Digital Outputs
- 10001 to 19999 => Digital Inputs.
- 30001 to 39999 => Analog Inputs.
- 40001 to 49999 => Analog Outputs

### Parameter PLOCATORn

It refers to the PVALUEn parameter.

These parameters are of the DS-258 data type. Each of these parameters consists of the following elements:

- From Eu 100 %
- From Eu 0 %
- To Eu 100 %
- To Eu 0 %
- Data Type:
- Slave Address:
- Modbus Address of Value:

This block allows Modbus Scale Conversion, to do the conversion procedure see the item “Modbus Scale Conversion” in the Chapter 1 for more details.

### Parameter ILOCATORn

It refers to the IVALUEn parameter. (See on Chapter 1, “Data Type and Data Structure Definition”):

- Slave Address:
- Modbus Address OF Value:

The IVALUEn parameters will display the values of the variables set in ILOCATORn.

### Parameter BLOCATORn

It refers to the BVALUEn parameter. This parameter is of the data type DS-260, so you will have to set two elements for this parameter (see on Chapter 1, “Data Type and Data Structure Definition”):

- Slave Address:
- Modbus Address OF Value:

The BVALUEn parameters will display the values of the variables set in BLOCATORn.

### BAD\_STATUS Parameter

This parameter indicates if the communication between slaves was established properly. If the correspondent bit is in logic level 1 this means there was an error during writing/reading of the respective parameter. The table below presents the values for these status values.

#### Relation between the bits in BAD\_STATUS and Modbus addresses

| Bit | Mnemonic | Parameter |
|-----|----------|-----------|
| 0   | B1       | BVALUE1   |
| 1   | B2       | BVALUE2   |
| 2   | B3       | BVALUE3   |
| 3   | B4       | BVALUE4   |
| 4   | B5       | BVALUE5   |
| 5   | B6       | BVALUE6   |
| 6   | B7       | BVALUE7   |
| 7   | B8       | BVALUE8   |
| 8   | I1       | IVALUE1   |
| 9   | I2       | IVALUE2   |
| 10  | P1       | PVALUE1   |
| 11  | P2       | PVALUE2   |
| 12  | F1       | FVALUE1   |
| 13  | F2       | FVALUE2   |



## Parameters

| Idx | Parameter     | Data Type (length) | Valid Range/Options | Default Value | Units | Store / Mode | Description  |
|-----|---------------|--------------------|---------------------|---------------|-------|--------------|--|
| 1   | ST_REV        | Unsigned16         |                     | 0             | None  | S/RO         |  |
| 2   | TAG_DESC      | OctString(32)      |                     | Spaces        | Na    | S            |  |
| 3   | STRATEGY      | Unsigned16         |                     | 0             | None  | S            |  |
| 4   | ALERT_KEY     | Unsigned8          | 1 to 255            | 0             | None  | S            |  |
| 5   | MODE_BLK      | DS-69              |                     | O/S           | Na    | S            | See Mode Parameter   |
| 6   | BLOCK_ERR     | Bitstring(2)       |                     |               | E     | D / RO       |  |
| 7   | LOCAL_MOD_MAP | Unsigned8          | 0 to 15             | 0             |       | S / O/S      | Define the Modbus addresses.   |
| 8   | BAD_STATUS    | Bitstring(2)       |                     | 0             | E     | D / RO       | Indicate if communication from slave is good or not (each bit corresponds to a Modbus variable). |
| 9   | FLOCATOR1     | DS-260             |                     |               |       | S / O/S      | Information to locate float parameter  |
| 10  | FVALUE1       | Float              |                     | 0             |       | N            | Value from requested address.  |
| 11  | FLOCATOR2     | DS-260             |                     |               |       | S / O/S      | Information to locate float parameter  |
| 12  | FVALUE2       | Float              |                     | 0             |       | N            | Value from requested address.  |
| 13  | PLOCATOR1     | DS-258             |                     |               |       | S / O/S      | Information to locate percentage parameter   |
| 14  | PVALUE1       | Float              |                     | 0             |       | N            | Value from requested address.  |
| 15  | PLOCATOR2     | DS-258             |                     |               |       | S / O/S      | Information to locate percentage parameter   |
| 16  | PVALUE2       | Float              |                     | 0             |       | N            | Value from requested address.  |
| 17  | ILOCATOR1     | DS-260             |                     |               |       | S / O/S      | Information to locate integer parameter  |
| 18  | ILENGTH1      | Integer8           | 1,2,4               | 2             |       | S / O/S      | Data length.   |
| 19  | IVALUE1       | Integer32          |                     | 0             |       | N            | Value from requested address.  |
| 20  | ILOCATOR2     | DS-260             |                     |               |       | S / O/S      | Information to locate integer parameter  |
| 21  | ILENGTH2      | Integer8           | 1,2,4               | 2             |       | S / O/S      | Data length.   |
| 22  | IVALUE2       | Integer32          |                     | 0             |       | N            | Value from requested address.  |
| 23  | BLOCATOR1     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 24  | BVALUE1       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 25  | BLOCATOR2     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 26  | BVALUE2       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 27  | BLOCATOR3     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 28  | BVALUE3       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 29  | BLOCATOR4     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 30  | BVALUE4       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 31  | BLOCATOR5     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 32  | BVALUE5       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 33  | BLOCATOR6     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 34  | BVALUE6       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 35  | BLOCATOR7     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 36  | BVALUE7       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 37  | BLOCATOR8     | DS-260             |                     |               |       | S / O/S      | Information to locate boolean parameter  |
| 38  | BVALUE8       | Boolean            |                     | TRUE          |       | N            | Value from requested addresses.  |
| 39  | UPDATE_EVT    | DS-73              |                     |               | Na    | D            | This alert is generated by any change to the static data.  |

| Idx | Parameter | DataType<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-----------|----------------------|-------------------------|------------------|-------|-----------------|--|
| 40  | BLOCK_ALM | DS-72                |                         |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static**

**Gray Background Line: Default Parameters of Syscon**

## Output Function Blocks

### AO - Analog Output

#### Overview

The Analog Output Block is a function block used by devices that work as output elements in a control loop, like valves, actuators, positioners, etc. The AO block receives a signal from another function block and passes its results to an output transducer block through an internal channel reference.

#### NOTE

There is IO point use consistency between block and Ladder. The block allocates a point in an IO module if the block channel is configured and if there is a link in a specific point. For example, to use the first two points of an IO module with 8 discrete outputs in the rack 2 slot 0 with one MDO block, it will be necessary to configure the channel (in this case equal to 2009), and the MDO.IN\_D1 and MDO.IN\_D2 inputs must be linked in another block of the control strategy.

In this case, it will not be allowed an online change (after a download) of block channel if the Ladder is using the same point to be allocated for this block. In this case, the online change of channel can be made only if the rack, slot is not being used by the Ladder.

#### Description

The AO block is connected to the transducer block through the CHANNEL parameter that must match with the following parameter in the transducer block:

TERMINAL\_NUMBER parameter for the FI302

The CHANNEL parameter must be set to 1 (one) if the AO block is running in the FY302 or FP302, and no configuration is necessary in the transducer block to connect it to the AO block.

#### Treatment of Input Values

The SP value may be controlled automatically through a cascade or remote cascade control or manually by an operator. The PV\_SCALE and XD\_SCALE are used to do the scaling conversion of the SP.

#### Treatment of Output Values

The transducer scaling (XD\_SCALE) is used to convert percent of span to the number used by the transducer. This allows portions of the SP span to cause full span movement of the output.

$$\text{OUT} = \text{SP}\% * (\text{EU}_{100\%} - \text{EU}_{0\%}) + \text{EU}_{0\%} [\text{XD\_SCALE}]$$

The bit "Increase to Close" in IO\_OPTS allows the output to be inverted relative to the span of the input value. For example, if the SP is 100. (PV\_SCALE=0-100%; XD\_SCALE = 3-15Psi):

If the "Increase to Close" bit in IO\_OPTS is clear, SP converted to OUT\_SCALE will be 15 psi. Therefore the actuator type will be "air to open".

If the "Increase to Close" bit in IO\_OPTS is true, SP converted to OUT\_SCALE will be 3 psi. Therefore the actuator type will be "air to close".

#### Simulate

The SIMULATE parameter is used for the diagnostics and checkout purposes. When it is active, the transducer value and status will be overridden by the simulate value and status. The SIMULATE can be disabled either by software in the SIMULATE parameter or hardware through the jumper.

The SIMULATE structure is composed by the following attributes:

- Simulate Value and Status
- Transducer Value and Status
- Simulate Enable/Disable

The Transducer Value/Status attributes of SIMULATE parameter are always showing the value that the AO block receives from the corresponding transducer block.

There is a hardware jumper to disable the SIMULATE parameter. If this jumper is placed Off, then the simulation will be disabled. In this case, the user cannot change the ENABLE/DISABLE attribute. This jumper prevents simulation from accidentally being enabled during plant operations. When the jumper is placed ON, it will cause "Simulate Active" attribute in the BLOCK\_ERR of Resource block to be true.

If the following conditions exist, simulate will be activated:

- The simulate hardware jumper is not placed Off;
- The SIMULATE.ENABLE/DISABLE parameter is “Active”.

When simulation is active, the READBACK and PV parameters will be calculated based on the attribute Simulate Value/Status of the SIMULATE parameter. Otherwise it will be that one supplied by the transducer block in the Transducer Value/Status attribute of the SIMULATE parameter.

#### Readback parameter

If the hardware supports a readback value, such as valve position, then the value will be read by the transducer block and it will be provided to the corresponding AO block through the Transducer Value/Status attribute of the SIMULATE parameter. If not supported, the Transducer Value/Status attribute of the SIMULATE parameter is generated from AO.OUT by the transducer block.

The READBACK parameter has a copy of the Transducer Value/Status attribute of the SIMULATE parameter if the simulation is disabled, otherwise it is a copy of the Simulate Value/Status attribute of the SIMULATE parameter

The PV is the READBACK parameter converted to the PV\_SCALE, therefore the PV can be simulated through the SIMULATE parameter.

In addition, the block admits safe action as described early in the fault state processing.

The AO block supports the mode-shedding feature as described early in the mode parameter.

#### BLOCK\_ERR

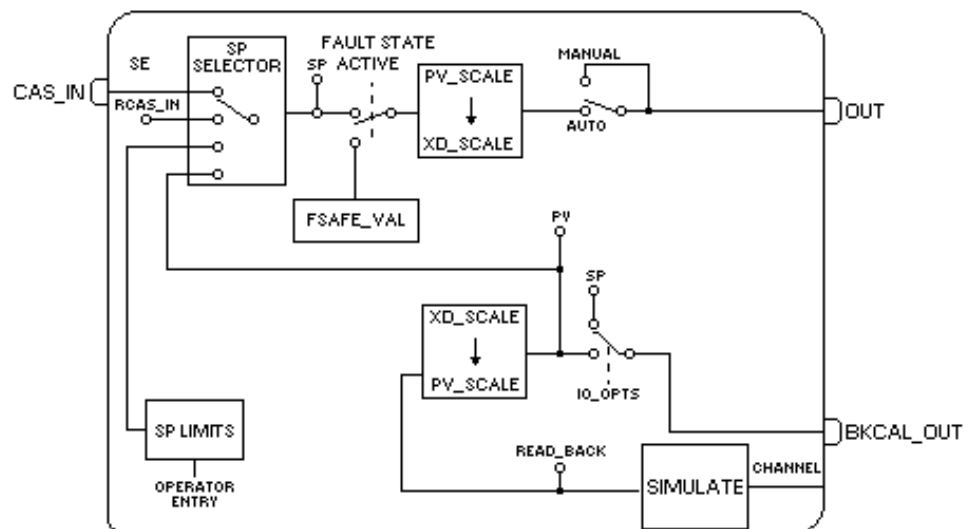
The BLOCK\_ERR of the AO block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when one or more of the following situations occur:
  - When the CHANNEL or SHED\_OPT parameters have an invalid value;
  - When the XD\_SCALE does not have a supported engineering unit and/or range for the respective the transducer block.
  - When the transducer block is in O/S mode.
  - When it is not compatible the CHANNEL parameter and HC configuration (DFI302).
- Simulate Active – When the Simulate is active.
- Local Override – When the block is in LO mode because the fault state is active.
- Output Failure – I/O module failure (DFI302)
- Out of Service – Occur when the block is in O/S mode.

#### Supported Modes

O/S, IMAN, LO, MAN, AUTO, CAS and RCAS.

#### Schematic



## Parameters

| Idx | Parameter   | Data Type (length) | Valid Range/ Options  | Default Value  | Units  | Store/ Mode | Description  |
|-----|-------------|--------------------|---|--|--------|-------------|--|
| 1   | ST_REV      | Unsigned16         |   | 0  | None   | S/RO        |  |
| 2   | TAG_DESC    | OctString(32)      |   | Spaces   | Na     | S           |  |
| 3   | STRATEGY    | Unsigned16         |   | 0  | None   | S           |  |
| 4   | ALERT_KEY   | Unsigned8          | 1 to 255  | 0  | None   | S           |  |
| 5   | MODE_BLK    | DS-69              |   | O/S  | Na     | S           | See Mode Parameter   |
| 6   | BLOCK_ERR   | Bitstring(2)       |   |  | E      | D/RO        |  |
| 7   | PV          | DS-65              |   |  | PV     | D / RO      | Process analog value.  |
| 8   | SP          | DS-65              | PV_SCALE +/- 10%  |  | PV     | N / Auto    | The analog set point. Can be set manually, automatically through the interface device or another field device.   |
| 9   | OUT         | DS-65              | XD_SCALE  |  | OUT    | N / Man     | The output value result to the transducer block.   |
| 10  | SIMULATE    | DS-82              | 1: Disable ;<br>2: Active<br>are the Enable/Disable options.          | Disable  |        | D           | Allows the readback value to be manually supplied when simulate is enabled. In this case, the simulate value and status will be the PV value.  |
| 11  | PV_SCALE    | DS-68              |   | 0-100%   | PV     | S / Man     | The high and low scale values to the SP parameter.   |
| 12  | XD_SCALE    | DS-68              | Depends on the device type. See the corresponding manual for details. | Depends on the Device type. See description for details. | XD     | S / Man     | The high and low scale values, to transducer for a specified channel.<br>The Default value for each Smar device is showed below:<br>DFI302 0 to 100 [%]  |
| 13  | GRANT_DENY  | DS-70              |   | 0  | Na     | D           |  |
| 14  | IO_OPTS     | Bitstring(2)       | See Block Options   | 0  | Na     | S / O/S     | See Block Options  |
| 15  | STATUS_OPTS | Bitstring(2)       | See Block Options   | 0  | Na     | S / O/S     | See Block Options  |
| 16  | READBACK    | DS-65              |   |  | XD     | D / RO      | Indicate the readback of the actual position of the transducer, in transducer units.   |
| 17  | CAS_IN      | DS-65              |   |  |        | D           | This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link.   |
| 18  | SP_RATE_DN  | Float              | Positive  | +INF   | PV/Sec | S           | Ramp rate at which upward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply in AUTO, CAS and RCAS modes.   |
| 19  | SP_RATE_UP  | Float              | Positive  | +INF   | PV/Sec | S           | Ramp rate at which downward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply in AUTO, CAS and RCAS modes.   |
| 20  | SP_HI_LIM   | Float              | PV_SCALE +/- 10%  | 100  | PV     | S           | The setpoint high limit is the highest setpoint operator entry that can be used for the block.   |
| 21  | SP_LO_LIM   | Float              | PV_SCALE +/- 10%  | 0  | PV     | S           | The setpoint low limit is the lowest setpoint operator entry that can be used for the block.   |
| 22  | CHANNEL     | Unsigned16         |   | 0  | None   | S / O/S     | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 23  | FSTATE_TIME | Float              | Positive  | 0  | Sec    | S           | The time in seconds to ignore the existence of a new fault state condition. If the fault state condition does not persist for FSTATE_TIME seconds and while this time does not elapse, the block will execute in the last actual mode. |
| 24  | FSTATE_VAL  | Float              | PV_SCALE +/- 10%  | 0  | PV     | S           | The preset analog SP value to use when fault occurs. This value will be used if the I/O option fault state to value is selected.   |

| Idx | Parameter  | Data Type (length) | Valid Range/ Options   | Default Value | Units | Store/ Mode | Description   |
|-----|------------|--------------------|--|---------------|-------|-------------|---|
| 25  | BKCAL_OUT  | DS-65              |  |               | PV    | D / RO      | The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to close the loop control.    |
| 26  | RCAS_IN    | DS-65              |  |               | PV    | D           | Target setpoint and status provided by a supervisory Host to an analog control or output block.   |
| 27  | SHED_OPT   | Unsigned8          | 1: NormalShed, NormalReturn<br>2: NormalShed, NoReturn<br>3: ShedToAuto, NormalReturn<br>4: ShedToAuto, NoReturn<br>5: ShedToMan, NormalReturn<br>6: ShedToMan, NoReturn<br>7: ShedToRetained Target, NormalReturn<br>8: ShedToRetained Target, NoReturn | 0             |       | S           | Defines action to be taken on remote control device timeout.  |
| 28  | RCAS_OUT   | DS-65              |  |               | PV    | D / RO      | Block setpoint and status after ramping – provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change. |
| 29  | UPDATE_EVT | DS-73              |  |               | Na    | D           |   |
| 30  | BLOCK_ALM  | DS-72              |  |               | Na    | D           |   |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
Gray Background Line: Default Parameters of Syscon

If DIAG.BEHAVIOR parameter is “Adapted”:

The default value of CHANNEL is the lowest available number.

The default value of SHED\_OPT is NormalShed/NormalReturn.

The required mode for writing is the actual mode, regardless the target mode: SP and OUT

## DO - Discrete Output

### Overview

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

#### NOTE

There is IO point use consistency between block and Ladder. The block allocates a point in an IO module if the block channel is configured and if there is a link in a specific point. For example, to use the first two points of an IO module with 8 discrete outputs in the rack 2 slot 0 with one MDO block, it will be necessary to configure the channel (in this case equal to 2009), and the MDO.IN\_D1 and MDO.IN\_D2 inputs must be linked in another block of the control strategy.

In this case, it will not be allowed an online change (after a download) of block channel if the Ladder is using the same point to be allocated for this block. In this case, the online change of channel can be made only if the rack, slot is not being used by the Ladder.

### Description

The Invert I/O option can be used to do a Boolean NOT function between the SP\_D and the hardware.

The SP\_D supports the full cascade sub-function. Cas mode must be used to transfer the output of another block to the SP\_D of the DO.

There are additional I/O options which will cause the SP\_D value to track the PV\_D value when the block is in an actual mode of LO or Man.

If the hardware supports a readback value, it is used for READBACK\_D, which, after accounting for the Invert I/O option, acts as the PV\_D for this block. If not supported, READBACK\_D is generated from OUT\_D. The OUT\_D and READBACK\_D parameters both use XD\_STATE. The PV\_D and SP\_D use PV\_STATE.

### BLOCK\_ERR

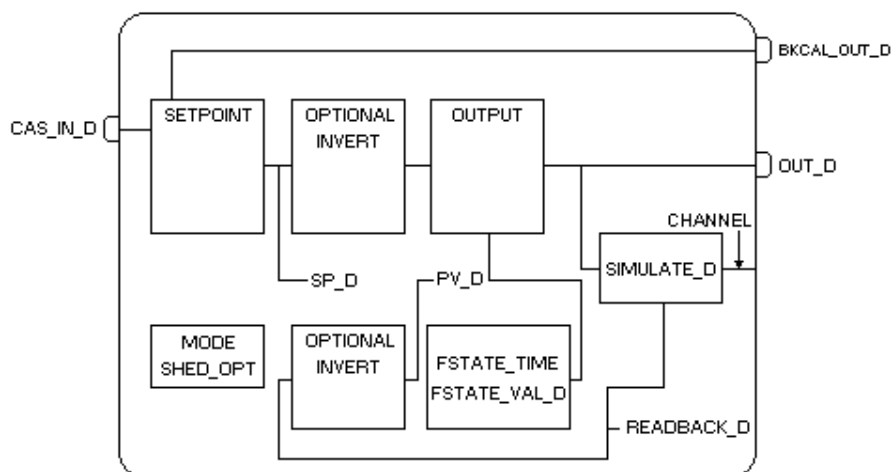
The BLOCK\_ERR of the DO block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when one or more of the following situations occur:
  - When the CHANNEL or SHED\_OPT parameters have an invalid value;
  - When it is not compatible the CHANNEL parameter and HC configuration (DFI302).
- Simulate Active – When the Simulate is active.
- Local Override – When the block is in LO mode because the fault state is active.
- Output Failure – I/O module failure (DFI302)
- Out of Service – Occur when the block is in O/S mode.

### Supported Modes

O/S, LO, Iman, Man, Auto, Cas, and RCas. The Man mode can be used to force the output, in a PLC sense. It may be that Man mode is not permitted, but it must be supported so that Man mode may be entered when leaving O/S. The IMan mode is used to indicate that there is no path to the final element.

### Schematic



## Parameters

| Idx | Parameter    | Data Type (length) | Valid Range/ Options   | Default Value | Units | Store / Mode | Description  |
|-----|--------------|--------------------|--|---------------|-------|--------------|--|
| 1   | ST_REV       | Unsigned16         |  | 0             | None  | S/RO         |  |
| 2   | TAG_DESC     | OctString(32)      |  | Spaces        | Na    | S            |  |
| 3   | STRATEGY     | Unsigned16         |  | 0             | None  | S            |  |
| 4   | ALERT_KEY    | Unsigned8          | 1 to 255   | 0             | None  | S            |  |
| 5   | MODE_BLK     | DS-69              |  | O/S           | Na    | S            | See Mode Parameter   |
| 6   | BLOCK_ERR    | Bitstring(2)       |  |               | E     | D/RO         |  |
| 7   | PV_D         | DS-66              |  |               | PV    | D / RO       | Either the primary discrete value for use in executing the function, or a process value associated with it.<br><br>May also be calculated from the READBACK_D value of a DO block.   |
| 8   | SP_D         | DS-66              | PV_STATE   |               | PV    | N / Auto     | The discrete setpoint of this block.   |
| 9   | OUT_D        | DS-66              | OUT_STATE  |               | OUT   | N / Man      | The primary discrete value calculated as a result of executing the function.   |
| 10  | SIMULATE_D   | DS-83              | 1: Disable ;<br>2: Active<br>are the Enable/Disable options. | Disable       |       | D            | Allows the transducer discrete input or output to the block to be manually supplied when simulate is enabled. When simulation is disabled, the simulate value and status track the actual value and status.                            |
| 11  | PV_STATE     | Unsigned16         |  | 0             | PV    | S            | Index to the text describing the states of a discrete PV.  |
| 12  | XD_STATE     | Unsigned16         |  | 0             | XD    | S            | Index to the text describing the states of a discrete for the value obtained from the transducer.  |
| 13  | GRANT_DENY   | DS-70              |  | 0             | Na    | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.   |
| 14  | IO_OPTS      | Bitstring(2)       | See Block Options  | 0             | Na    | S / O/S      | See Block Options  |
| 15  | STATUS_OPTS  | Bitstring(2)       | See Block Options  | 0             | Na    | S / O/S      | See Block Options  |
| 16  | READBACK_D   | DS-66              |  |               | XD    | D / RO       | This indicates the readback of the actual discrete valve or other actuator position, in the transducer state.  |
| 17  | CAS_IN_D     | DS-66              |  |               | PV    | D / RW       | This parameter is the remote setpoint value of a discrete block, which must come from another Fieldbus block, or a DCS block through a defined link.   |
| 18  | CHANNEL      | Unsigned16         |  | 0             | None  | S / O/S      | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 19  | FSTATE_TIME  | Float              | Positive   | 0             | Sec   | S            | The time in seconds to ignore the existence of a new fault state condition. If the fault state condition does not persist for FSTATE_TIME seconds and while this time does not elapse, the block will execute in the last actual mode. |
| 20  | FSTATE_VAL_D | Unsigned8          |  | 0             | PV    | S            | The preset discrete SP_D value to use when fault occurs. This value will be used if the I/O option Fault State to value is selected.   |



| Idx | Parameter   | Data Type<br>(length) | Valid Range/<br>Options  | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|-------------|-----------------------|--|------------------|-------|-----------------|--|
| 21  | BKCAL_OUT_D | DS-66                 |  |                  | PV    | D / RO          | The output value and status provided to an upstream discrete block. This information is used to provide bumpless transfer to closed loop control.  |
| 22  | RCAS_IN_D   | DS-66                 |  |                  | PV    | D               | Target setpoint and status provided by a supervisory Host to a discrete control or output block.   |
| 23  | SHED_OPT    | Unsigned8             | 1: NormalShed, NormalReturn<br>2: NormalShed, NoReturn<br>3: ShedToAuto, NormalReturn<br>4: ShedToAuto, NoReturn<br>5: ShedToMan, NormalReturn<br>6: ShedToMan, NoReturn<br>7: ShedToRetainedTarget, NormalReturn<br>8: ShedToRetainedTarget, NoReturn | 0                |       | S               | Defines action to be taken on remote control device timeout.   |
| 24  | RCAS_OUT_D  | DS-66                 |  |                  | PV    | D / RO          | Block setpoint and status provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change.  |
| 25  | UPDATE_EVT  | DS-73                 |  |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 26  | BLOCK_ALM   | DS-72                 |  |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
 Gray Background Line: Default Parameters of Syscon

## MAO - Multiple Analog Output

### Description

The MAO block makes available to the I/O subsystem its eight input parameters IN\_1 through IN\_8.

For DFI working with Analog Output, the 4-20mA or 1-5V range must be worked. In this case, the input values are in percent from 0 to 100%. If it is necessary to work in another configuration, the AO block must be used.

This function block has the same fault state characteristics as the AO block. It includes option to hold the last value or go to a preset value when fault state active, individual preset values for each point, besides a delay time to go into the fault state.

The actual mode will be LO only due to the resource block, otherwise bad status in input parameter and configuration of MO\_STATUS\_OPTS will not affect the mode calculation. However the functionality of fault state will be done only for that input parameter.

### NOTE

There is IO point use consistency between block and Ladder. The block allocates a point in an IO module if the block channel is configured and if there is a link in a specific point. For example, to use the first two points of an IO module with 8 discrete outputs in the rack 2 slot 0 with one MDO block, it will be necessary to configure the channel (in this case equal to 2009), and the MDO.IN\_D1 and MDO.IN\_D2 inputs must be linked in another block of the control strategy.

In this case, it will not be allowed an online change (after a download) of block channel if the Ladder is using the same point to be allocated for this block. In this case, the online change of channel can be made only if the rack, slot is not being used by the Ladder.

### BLOCK\_ERR

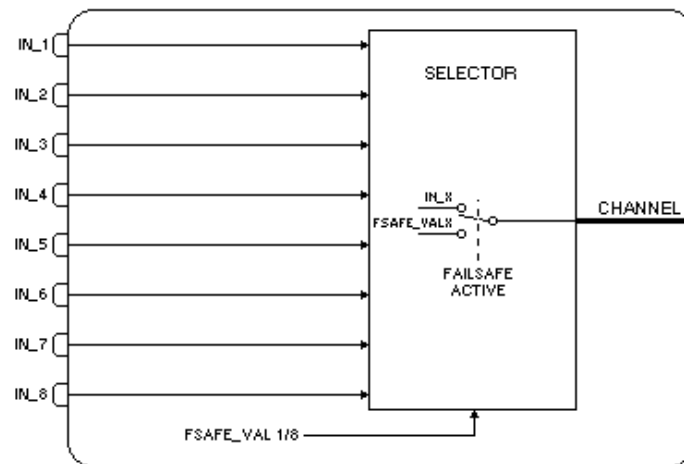
The BLOCK\_ERR of the MAO block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when CHANNEL has an invalid value;
- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S, LO and AUTO.

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/ Options | Default Value | Units | Store / Mode | Description        |
|-----|-----------|--------------------|----------------------|---------------|-------|--------------|--------------------|
| 1   | ST_REV    | Unsigned16         |                      | 0             | None  | S/RO         |                    |
| 2   | TAG_DESC  | OctString(32)      |                      | Spaces        | Na    | S            |                    |
| 3   | STRATEGY  | Unsigned16         |                      | 0             | None  | S            |                    |
| 4   | ALERT_KEY | Unsigned8          | 1 to 255             | 0             | None  | S            |                    |
| 5   | MODE_BLK  | DS-69              |                      | O/S           | Na    | S            | See Mode Parameter |
| 6   | BLOCK_ERR | Bitstring(2)       |                      |               | E     | D/RO         |                    |

| Idx | Parameter     | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store /<br>Mode | Description  |
|-----|---------------|-----------------------|-------------------------|------------------|-------|-----------------|--|
| 7   | CHANNEL       | Unsigned16            |                         | 0                | None  | S / O/S         | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 8   | IN_1          | DS-65                 |                         |                  | %     | D               | Numbered input 1.  |
| 9   | IN_2          | DS-65                 |                         |                  | %     | D               | Numbered input 2.  |
| 10  | IN_3          | DS-65                 |                         |                  | %     | D               | Numbered input 3.  |
| 11  | IN_4          | DS-65                 |                         |                  | %     | D               | Numbered input 4.  |
| 12  | IN_5          | DS-65                 |                         |                  | %     | D               | Numbered input 5.  |
| 13  | IN_6          | DS-65                 |                         |                  | %     | D               | Numbered input 6.  |
| 14  | IN_7          | DS-65                 |                         |                  | %     | D               | Numbered input 7.  |
| 15  | IN_8          | DS-65                 |                         |                  | %     | D               | Numbered input 8.  |
| 16  | MO_OPTS       | Bitstring(2)          | See Block Options       | 0                | Na    | S / O/S         | See Block Options  |
| 17  | FSTATE_TIME   | Float                 | Positive                | 0                | Sec   | S               | The time in seconds to ignore the existence of a new fault state condition. If the fault state condition does not persist for FSTATE_TIME seconds and while this time does not elapse, the block will execute in the last actual mode.   |
| 18  | FSTATE_VAL1   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_1. Ignored if the "Fault state to value 1" in the MO_OPTS parameter is false.   |
| 19  | FSTATE_VAL2   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_2. Ignored if the "Fault state to value 2" in the MO_OPTS parameter is false.   |
| 20  | FSTATE_VAL3   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_3. Ignored if the "Fault state to value 3" in the MO_OPTS parameter is false.   |
| 21  | FSTATE_VAL4   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_4. Ignored if the "Fault state to value 4" in the MO_OPTS parameter is false.   |
| 22  | FSTATE_VAL5   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_5. Ignored if the "Fault state to value 5" in the MO_OPTS parameter is false.   |
| 23  | FSTATE_VAL6   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_6. Ignored if the "Fault state to value 6" in the MO_OPTS parameter is false.   |
| 24  | FSTATE_VAL7   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_7. Ignored if the "Fault state to value 7" in the MO_OPTS parameter is false.   |
| 25  | FSTATE_VAL8   | Float                 |                         | 0                |       | S               | The preset analog value to use when failure occurs in IN_8. Ignored if the "Fault state to value 8" in the MO_OPTS parameter is false.   |
| 16  | FSTATE_STATUS | Unsigned8             |                         |                  | None  | D / RO          | It shows which points are in fault state active.   |
| 27  | UPDATE_EVT    | DS-73                 |                         |                  | Na    | D               | This alert is generated by any change to the static data.  |
| 28  | BLOCK_ALM     | DS-72                 |                         |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
Gray Background Line: Default Parameters of Syscon

## MDO - Multiple Discrete Output

### Description

The MDO block makes available to the I/O subsystem its eight input parameters IN\_D1 through IN\_D8.

This function block has the same fault state characteristics as the DO block. It includes option to hold the last value or go to a preset value when fault state active, individual preset values for each point, besides a delay time to go into the fault state.

The actual mode will be LO only due to the resource block, otherwise bad status in input parameter and configuration of MO\_STATUS\_OPTS will not affect the mode calculation. However the functionality of fault state will be done only for that input parameter.

The parameter FSTATE\_STATE shows which points are in fault state active.

### NOTE

There is IO point use consistency between block and Ladder. The block allocates a point in an IO module if the block channel is configured and if there is a link in a specific point. For example, to use the first two points of an IO module with 8 discrete outputs in the rack 2 slot 0 with one MDO block, it will be necessary to configure the channel (in this case equal to 2009), and the MDO.IN\_D1 and MDO.IN\_D2 inputs must be linked in another block of the control strategy.

In this case, it will not be allowed an online change (after a download) of block channel if the Ladder is using the same point to be allocated for this block. In this case, the online change of channel can be made only if the rack, slot is not being used by the Ladder.

### BLOCK\_ERR

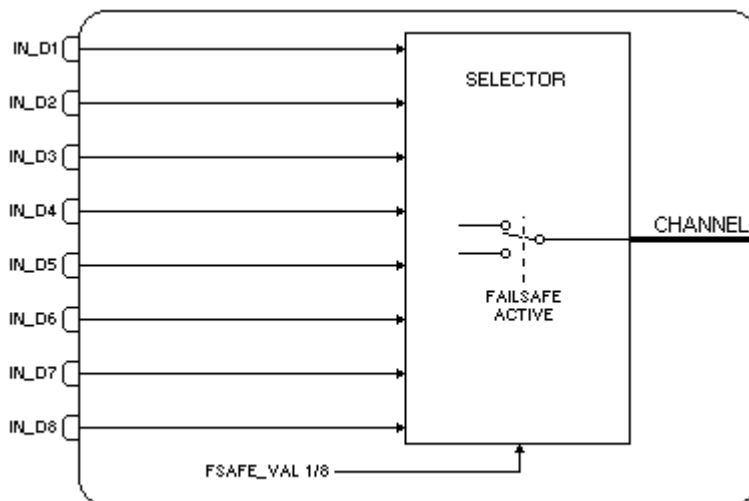
The BLOCK\_ERR of the MDO block will reflect the following causes:

- Block Configuration Error – the configuration error occurs when CHANNEL has an invalid value (FB700);
- Out of Service – When the block is in O/S mode.

### Supported Modes

O/S, LO and AUTO.

### Schematic



### Parameters

| Idx | Parameter | Data Type (length) | Valid Range/ Options | Default Value | Units | Store/ Mode | Description |
|-----|-----------|--------------------|----------------------|---------------|-------|-------------|-------------|
| 1   | ST_REV    | Unsigned16         |                      | 0             | None  | S/RO        |             |
| 2   | TAG_DESC  | OctString(32)      |                      | Spaces        | Na    | S           |             |
| 3   | STRATEGY  | Unsigned16         |                      | 0             | None  | S           |             |
| 4   | ALERT_KEY | Unsigned8          | 1 to 255             | 0             | None  | S           |             |

| Idx | Parameter     | Data Type<br>(length) | Valid Range/<br>Options | Default<br>Value | Units | Store/<br>Mode | Description  |
|-----|---------------|-----------------------|-------------------------|------------------|-------|----------------|--|
| 5   | MODE_BLK      | DS-69                 |                         | O/S              | Na    | S              | See Mode Parameter   |
| 6   | BLOCK_ERR     | Bitstring(2)          |                         |                  | E     | D/RO           |  |
| 7   | CHANNEL       | Unsigned16            |                         | 0                | None  | S / O/S        | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".   |
| 8   | IN_D1         | DS-66                 |                         |                  |       | D              | Numbered discrete input 1.   |
| 9   | IN_D2         | DS-66                 |                         |                  |       | D              | Numbered discrete input 2.   |
| 10  | IN_D3         | DS-66                 |                         |                  |       | D              | Numbered discrete input 3.   |
| 11  | IN_D4         | DS-66                 |                         |                  |       | D              | Numbered discrete input 4.   |
| 12  | IN_D5         | DS-66                 |                         |                  |       | D              | Numbered discrete input 5.   |
| 13  | IN_D6         | DS-66                 |                         |                  |       | D              | Numbered discrete input 6.   |
| 14  | IN_D7         | DS-66                 |                         |                  |       | D              | Numbered discrete input 7.   |
| 15  | IN_D8         | DS-66                 |                         |                  |       | D              | Numbered discrete input 8.   |
| 16  | MO_OPTS       | Bitstring(2)          | See Block Options       | 0                | Na    | S / O/S        | See Block Options  |
| 17  | FSTATE_TIME   | Float                 | Positive                | 0                | Sec   | S              | The time in seconds to ignore the existence of a new fault state condition. If the fault state condition does not persist for FSTATE_TIME seconds and while this time does not elapse, the block will execute in the last actual mode.   |
| 18  | FSTATE_VAL_D1 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D1. Ignored if the "Fault state to value 1" in the MO_OPTS parameter is false.  |
| 19  | FSTATE_VAL_D2 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D2. Ignored if the "Fault state to value 2" in the MO_OPTS parameter is false.  |
| 20  | FSTATE_VAL_D3 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D3. Ignored if the "Fault state to value 3" in the MO_OPTS parameter is false.  |
| 21  | FSTATE_VAL_D4 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D4. Ignored if the "Fault state to value 4" in the MO_OPTS parameter is false.  |
| 22  | FSTATE_VAL_D5 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D5. Ignored if the "Fault state to value 5" in the MO_OPTS parameter is false.  |
| 23  | FSTATE_VAL_D6 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D6. Ignored if the "Fault state to value 6" in the MO_OPTS parameter is false.  |
| 24  | FSTATE_VAL_D7 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D7. Ignored if the "Fault state to value 7" in the MO_OPTS parameter is false.  |
| 25  | FSTATE_VAL_D8 | Unsigned8             |                         | 0                |       | S              | The preset discrete value to use when failure occurs in IN_D8. Ignored if the "Fault state to value 8" in the MO_OPTS parameter is false.  |
| 26  | FSTATE_STATUS | Unsigned8             |                         |                  | None  | D / RO         | It shows which points are in fault state active.   |
| 27  | UPDATE_EVT    | DS-73                 |                         |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 28  | BLOCK_ALM     | DS-72                 |                         |                  | Na    | 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 subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static

Gray Background Line: Default Parameters of Syscon

## STEP – Step Output PID

### Overview

A Step Control Output block is used most commonly, when the final control element has an actuator driven by an electric motor. The final control element is positioned by rotating the motor clockwise or anticlockwise, which is accomplished by activating a discrete signal for each direction. A control valve, for example, needs a signal to open and another to close. If none of the signals is present, the valve stem would stay at the same position.

Fieldbus actuators and switchgears are the transducer blocks of this block.

### Description

As shown on Fig.1, electric actuators require a switchgear module to power the electric motor and reverse it as demanded by the control loop. Most of the electric actuators require an interlock circuit to prevent the motor to overheat, or even burn, when the actuator reaches one of the travel limits or something blocks the movement in any direction, increasing the torque beyond an established limit. These actuators are normally equipped with torque switches and limit switches to provide this kind of protection.

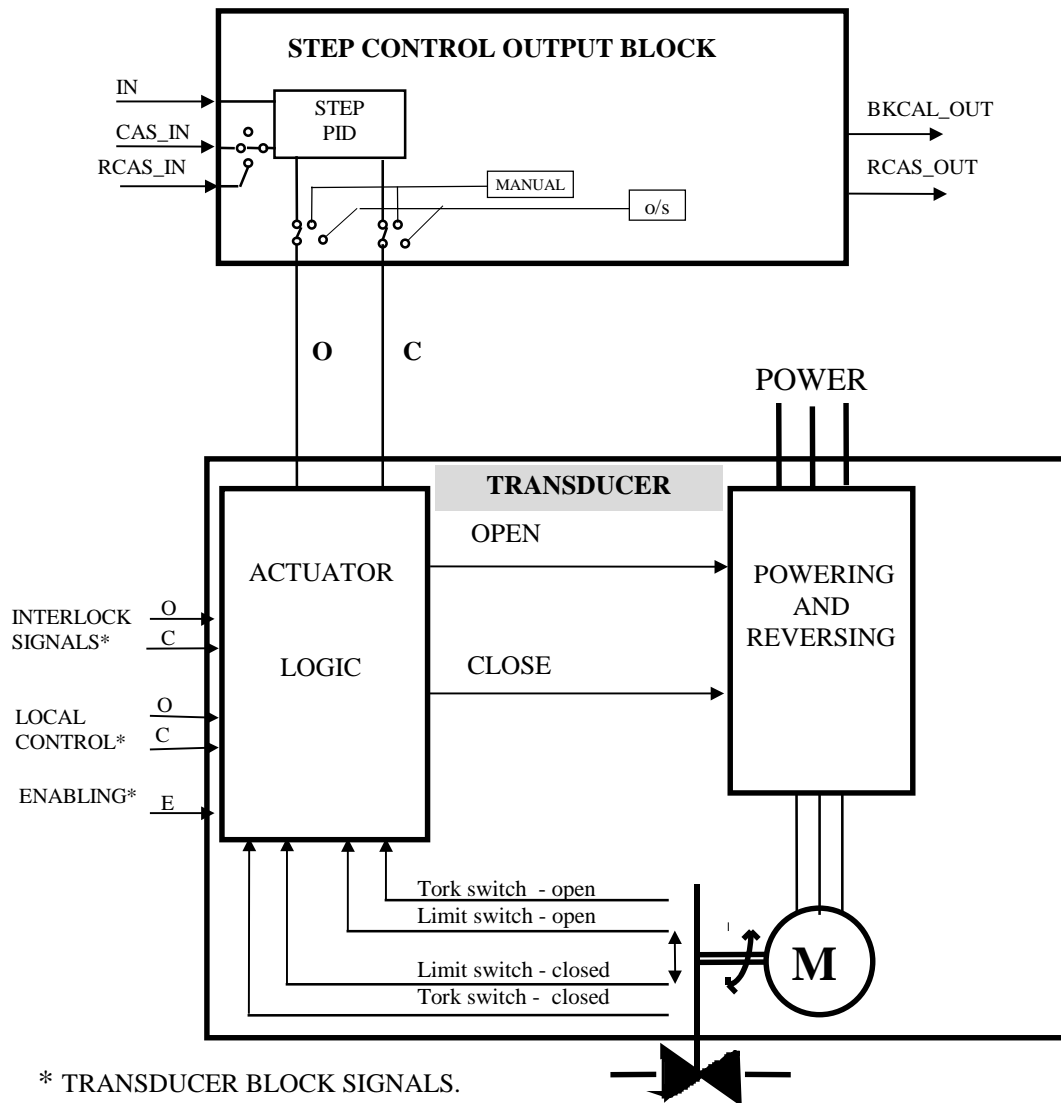
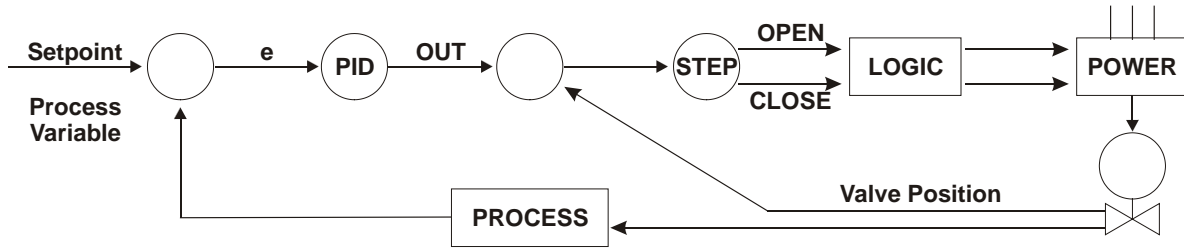


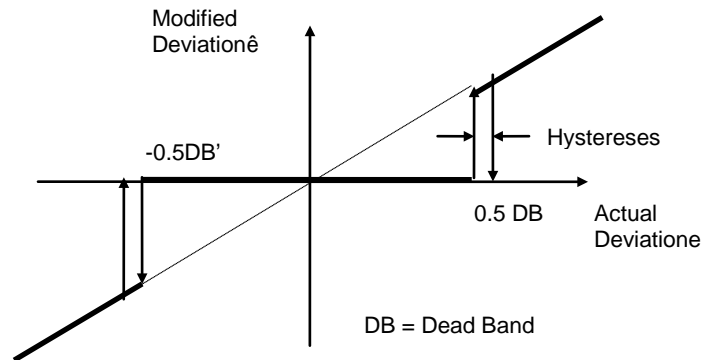
Fig.1- Electric Actuator

**Using a standard PID controller in cascade with a PI Step Controller.**

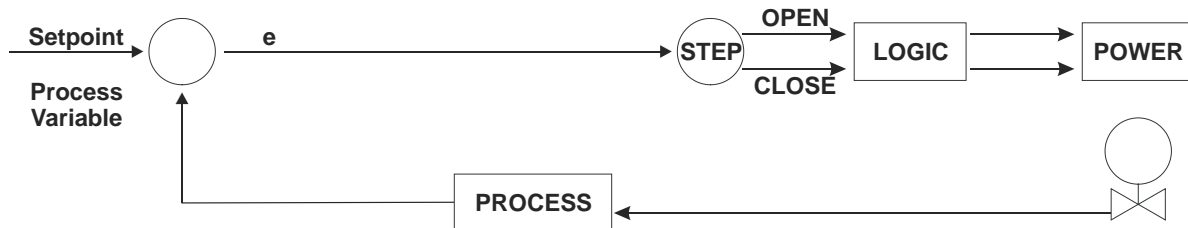
The slave process variable is the position of the final control element, as shown in Fig.2. Controlling the process variable regardless of the valve position measurement.



**Fig. 2 - Step controller working as a positioner**



**Fig. 3 - Modified deviation or gap deviation**



**Fig. 4- Step Controller**

The actuator has a Travel Time, that is the time it takes to drive the final control element from one end limit to another. For a control valve, for example, it is the time required to drive it from closed to completely open.

The proportional action will actuate the final control element in the required direction during a time proportional to:

$$t_p = [\text{GAIN}] * (\hat{e} / 100) * [\text{TRAVEL\_TIME}] \dots (\text{s})$$

If the proportional action is not enough to turn  $\hat{e} = 0$ , the Integral Action will move the final control element at a speed of:

$$V = [\text{GAIN}] * \hat{e} / [\text{RESET}] \dots (\% / \text{s})$$

where the reset is the Integral time constant in seconds.

As most of the actuators work with constant and fixed speed, they can not give a speed larger than:

$$\text{Maximum speed} = 100\% / [\text{TRAVEL TIME}] \dots (\% / \text{s})$$

while the smaller speeds required by the Integral action are obtained by giving pulses of a specified duration [PULSE\_DUR]. Each pulse will move the final control element a  $\Delta x$  % in the required direction.

$$\Delta x \% = [\text{PULSE\_DUR}] * 100\% / [\text{TRAVEL\_TIME}] \dots\dots\dots (\%)$$

The pulse frequency is given by:

$$f = V / \Delta x \% \dots\dots\dots (\text{pulses} / \text{s})$$

The Derivative or Rate action is given by:

$$t_D = [\text{GAIN}] * (d\hat{e}/dt) * [\text{RATE}]$$

Where Rate is the derivative time constant in seconds and  $d\hat{e}/dt$  can be calculated in several ways, including derivative gain, filtering, etc.

The PID Step Controller activates the OPEN or CLOSE signals according to the modified deviation,  $\hat{e}$ , the PID parameters and the other parameters in the following way:

The signal is activated during a time equivalent to:

$$= t_P + t_D$$

If the modified deviation is still different of zero, the Integral or Reset action will give pulses with a duration defined by [PULSE\_DUR], with a frequency calculated by "f."  $t$  and  $f$  are dynamically modified by  $\hat{e}$ .

In order to avoid the Reset wind-up, the actuation time in one direction must be integrated and limited.

If the actuation time in one direction is larger than the [TRAVEL\_TIME], there is no use in making the respective output signal to pulse, therefore it is recommendable to maintain it continuously activated.

The block provides a full PV and Deviation alarm support.

The meaning of possible values for OUT\_D are :

OUT\_D.value = 0 → Stop

OUT\_D.value = 1 → Close

OUT\_D.value = 2 → Open

As the STEP block requires two discrete outputs, when setting the CHANNEL parameter, indeed two outputs are allocated. The value of CHANNEL points to the CLOSE output, and the next channel points to the OPEN output.

### **Supported Modes**

Out-of-service, Manual, Auto, CAS and RCAS are supported.

In Out-of-service mode, the status of the output will be Out-of -service.

In Manual mode, the OUT\_D can be set by the operator. The block stops output calculation.

In Auto mode the block operates normally.

In CAS (cascade), the Setpoint is supplied by another function block through the CAS\_IN parameter.

In RCAS the block setpoint is set by a control application running on a computer, DCS or PLC.

### **Status Handling**

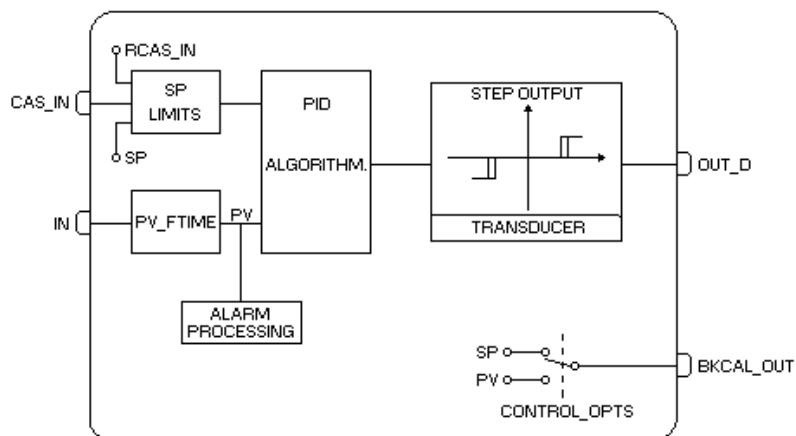
The status of OUT\_Di shall reflect the worst quality of the status of any connected input.

### **Initial Value Handling**

The initial value of OUT\_Di should be zero, that is, no action in both directions and the Integral action value should also go to zero.



## Schematic



## Parameters

| Idx | Parameter    | Data Type (length) | Valid Range/Options | Default Value | Units  | Store / Mode | Description  |
|-----|--------------|--------------------|---------------------|---------------|--------|--------------|--|
| 1   | ST_REV       | Unsigned16         |                     | 0             | None   | S/RO         |  |
| 2   | TAG_DESC     | OctString(32)      |                     | Spaces        | Na     | S            |  |
| 3   | STRATEGY     | Unsigned16         |                     | 0             | None   | S            |  |
| 4   | ALERT_KEY    | Unsigned8          | 1 to 255            | 0             | None   | S            |  |
| 5   | MODE_BLK     | DS-69              |                     | O/S           | Na     | S            | See Mode Parameter   |
| 6   | BLOCK_ERR    | Bitstring(2)       |                     |               | E      | D / RO       |  |
| 7   | PV           | DS-65              |                     |               | PV     | D / RO       | Process analog value. This is the IN value after pass over the PV filter.  |
| 8   | SP           | DS-65              | PV_SCALE +/- 10%    |               | PV     | N / Auto     | The analog set point. Can be set manually, automatically through the interface device or another field device.                                     |
| 9   | OUT_D        | DS-66              |                     |               |        | N / Man      | The output value result of the Step Output PID calculation.  |
| 10  | PV_SCALE     | DS-68              |                     | 0-100%        | PV     | S / Man      | The high and low scale values to the PV and SP parameter.  |
| 11  | XD_STATE     | Unsigned16         |                     | 0             | XD     | S            | Index to the text describing the states of a discrete for the value obtained from the transducer.  |
| 12  | GRANT_DENY   | DS-70              |                     | 0             | na     | D            | Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.                   |
| 13  | CONTROL_OPTS | Bitstring(2)       | See Block Options   | 0             | na     | S / O/S      | See Block Options  |
| 14  | STATUS_OPTS  | Bitstring(2)       | See Block Options   | 0             | Na     | S / O/S      | See Block Options  |
| 15  | IN           | DS-65              |                     |               | PV     | D            | The primary input value of the block, or PV value.   |
| 16  | PV_FTIME     | Float              | Non-Negative        | 0             | Sec    | S            | Time constant of a single exponential filter for the PV, in seconds.   |
| 17  | JOG_TIME     | Float              | Positive            | 0             | Sec    | S            | Duration of OUT_D in the active state when commanded by the operator to jog open or jog closed.  |
| 18  | CAS_IN       | DS-65              |                     |               |        | D            | This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link.                   |
| 19  | SP_RATE_DN   | Float              | Positive            | +INF          | PV/Sec | S            | Ramp rate at which upward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply only in AUTO mode.   |
| 20  | SP_RATE_UP   | Float              | Positive            | +INF          | PV/Sec | S            | Ramp rate at which downward setpoint changes in PV units per second. It is disable if is zero or +INF. Rate limiting will apply only in AUTO mode. |

| Idx | Parameter        | Data Type (length) | Valid Range/ Options   | Default Value | Units       | Store / Mode  | Description   |
|-----|------------------|--------------------|--|---------------|-------------|---------------|---|
| 21  | SP_HI_LIM        | Float              | PV_SCALE +/- 10%   | 100           | PV          | S             | The setpoint high limit is the highest setpoint operator entry that can be used for the block.  |
| 22  | SP_LO_LIM        | Float              | PV_SCALE +/- 10%   | 0             | PV          | S             | The setpoint low limit is the lowest setpoint operator entry that can be used for the block.  |
| 23  | <b>GAIN</b>      | <b>Float</b>       |  | <b>0</b>      | <b>None</b> | <b>S</b>      | <b>Proportional term of the PID. It is the Kp value.</b>  |
| 24  | <b>RESET</b>     | <b>Float</b>       | <b>Positive</b>  | <b>+INF</b>   | <b>sec</b>  | <b>S</b>      | <b>Integral term of the PID. It is the Tr value.</b>  |
| 25  | BAL_TIME         | Float              | Positive   | 0             | sec         | S             | This specifies the time for the internal working value of bias or ratio to return to the operator set bias or ratio, in seconds.<br><br>In the PID block, it may be 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.   |
| 26  | <b>RATE</b>      | <b>Float</b>       | <b>Positive</b>  | <b>0</b>      | <b>sec</b>  | <b>S</b>      | <b>Derivative term of the PID. It is the Td value.</b>  |
| 27  | IO_OPTS          | Bitstring(2)       | See Block Options  | 0             | na          | S / O/S       | See Block Options   |
| 28  | CHANNEL          | Unsigned16         |  | 0             | None        | S / O/S       | For more details about the configuration of this parameter, see Chapter 1 "CHANNEL Configuration".<br><br>In the DFI302, this parameter is selecting two discrete outputs. The first one is the CLOSE output and the next point in the same group will be the OPEN output.<br><br>The CHANNEL parameter will be addressing the CLOSE output, despite of it is allocating the OPEN output too. |
| 29  | FSTATE_TIME      | Float              | Positive   | 0             | Sec         | S             | The time in seconds from detection of fault of the output block remote setpoint to the output action of the block output if the condition still exists.   |
| 30  | FSTATE_VAL_D     | Unsigned8          |  | 0             |             | S             | The preset discrete SP_D value to use when fault occurs. This value will be used if the I/O option Fault State to value is selected.  |
| 31  | <b>BKCAL_OUT</b> | <b>DS-65</b>       |  |               | <b>PV</b>   | <b>D / RO</b> | <b>The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control.</b>  |
| 32  | RCAS_IN          | DS-65              |  |               | PV          | D             | Target setpoint and status provided by a supervisory Host to a analog control or output block.  |
| 33  | SHED_OPT         | Unsigned8          | 1: NormalShed, NormalReturn<br>2: NormalShed, NoReturn<br>3: ShedToAuto, NormalReturn<br>4: ShedToAuto, NoReturn<br>5: ShedToMan, NormalReturn<br>6: ShedToMan, NoReturn<br>7: ShedToRetainedTarget, NormalReturn<br>8: ShedToRetainedTarget, NoReturn | 0             |             | S             | Defines action to be taken on remote control device timeout.  |
| 34  | RCAS_OUT         | DS-65              |  |               | PV          | D / RO        | Block setpoint and status after ramping - provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change.   |
| 35  | TRAVEL_TIME      | Float              | Positive   | 60            | Sec         | S / Man       | The time required by the actuator to drive the final control element from one end position to another, in seconds.  |

| Idx | Parameter  | Data Type (length) | Valid Range/ Options                      | Default Value | Units | Store / Mode | Description  |
|-----|------------|--------------------|---|---------------|-------|--------------|--|
| 36  | PULSE_DUR  | Float              | Positive                                  | 1             | Sec   | S / Man      | It is the width, in seconds, of the pulses given due to the integral action.   |
| 37  | DEAD_BAND  | Float              | Non-negative                              | 0             | %     | S / Man      | It is the interval where changes in the Input will not change the output   |
| 38  | HYSTERESIS | Float              | Non-negative                              | 0             | %     | S / Man      | Difference between the switching points.   |
| 39  | UPDATE_EVT | DS-73              |   |               | Na    | D            | This alert is generated by any change to the static data.  |
| 40  | BLOCK_ALM  | DS-72              |   |               | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |
| 41  | ALARM_SUM  | DS-74              | See Block Options                         |               | Na    | S            | The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.  |
| 42  | ACK_OPTION | Bitstring(2)       | 0: Auto ACK Disable<br>1: Auto ACK Enable | 0             | Na    | S            | Selection of whether alarms associated with the block will be automatically acknowledged   |
| 43  | ALARM_HYS  | Float              | 0 to 50 %                                 | 0.5%          | %     | S            | Alarm hysteresis parameter. In order to clear the alarm the amount the PV must return within the alarm limit plus hysteresis.  |
| 44  | HI_HI_PRI  | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the high high alarm.   |
| 45  | HI_HI_LIM  | Float              | OUT_SCALE, +INF                           | +INF          | PV    | S            | The setting for high high alarm in engineering units.  |
| 46  | HI_PRI     | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the high alarm.  |
| 47  | HI_LIM     | Float              | OUT_SCALE, +INF                           | +INF          | PV    | S            | The setting for high alarm in engineering units.   |
| 48  | LO_PRI     | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the low alarm.   |
| 49  | LO_LIM     | Float              | OUT_SCALE, -INF                           | -INF          | PV    | S            | The setting for low alarm in engineering units.  |
| 50  | LO_LO_PRI  | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the low low alarm.   |
| 51  | LO_LO_LIM  | Float              | OUT_SCALE, -INF                           | -INF          | PV    | S            | The setting for low low alarm in engineering units.  |
| 52  | DV_HI_PRI  | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the deviation high alarm.  |
| 53  | DV_HI_LIM  | Float              | 0 to PV span, +INF                        | +INF          | PV    | S            | The setting for deviation high alarm in engineering units.   |
| 54  | DV_LO_PRI  | Unsigned8          | 0 to 15                                   | 0             |       | S            | Priority of the deviation low alarm.   |
| 55  | DV_LO_LIM  | Float              | -INF, -PV span to 0                       | -INF          | PV    | S            | The setting for deviation low alarm in engineering units.  |
| 56  | HI_HI_ALM  | DS-71              |   |               | PV    | D            | The status for high high alarm and its associated time stamp.  |
| 57  | HI_ALM     | DS-71              |   |               | PV    | D            | The status for high alarm and its associated time stamp.   |
| 58  | LO_ALM     | DS-71              |   |               | PV    | D            | The status for low alarm and its associated time stamp.  |
| 59  | LO_LO_ALM  | DS-71              |   |               | PV    | D            | The status for low low alarm and its associated time stamp.  |
| 60  | DV_HI_ALM  | DS-71              |   |               | PV    | D            | The status for deviation high alarm and its associated time stamp.   |
| 61  | DV_LO_ALM  | DS-71              |   |               | PV    | D            | The status for deviation low alarm and its associated time stamp.  |

**Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static**  
**Gray Background Line: Default Parameters of Syscon**

## ROMAO – ROM Analog Output

### Description

This block has all characteristics of the standard AO block in addition the features related to alarms and I/O mapping via CHANNEL\_TAG parameter.

### CHANNEL\_TAG Parameter

This parameter allows association with the physical points of analog output, which occurs when the value of this parameter coincides with a compatible VAR\_NAME in terms of type, ie, analog output. The output physical point has to be associated to just one CHANNEL\_TAGS of output function block.

The configuration of the CHANNEL\_TAG parameter indirectly implies in the configuration of the CHANNEL parameter, which depends on the match with a valid VAR\_NAME. So, the CHANNEL parameter becomes a read-only parameter to the user.

### BLOCK\_ERR

It indicates a problem in the output point, which can be: Configuration Error and Output Failure.

### Diagnostic and troubleshooting

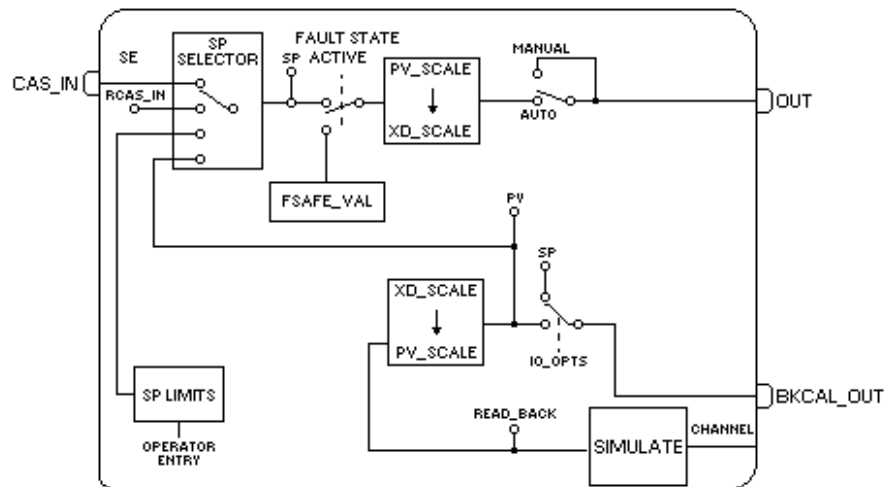
The diagnostics are based on the value of **BLOCK\_ERR** parameter, as follows:

- BLOCK\_ERR = Block configuration: if tag configured for CHANNEL\_TAG was not found in an output transducer block, for example **TB4AO**;
- BLOCK\_ERR = Out-of-service: It indicates that Actual Mode is O/S.

### Supported Modes

Auto, Man, O/S.

### Schematic



### Parameters

| Idx | Type/View | Parameter   | Data Type         | Valid Range | Initial Value | Units | Store/Mode | Description   |
|-----|-----------|-------------|-------------------|-------------|---------------|-------|------------|---|
| 31  | 4         | CHANNEL_TAG | VisibleString[32] |             | Blanks        |       | S          | Tag of analog output point in the transducer block to effect the association to the physical point. Blank tag means that the point is not configured. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
 Gray Background Line: Default Parameters of Syscon



| Idx | Type/<br>View | Parameter          | Data Type             | Valid<br>Range | Initial<br>Value | Units | Store/<br>Mode | Description  |
|-----|---------------|--------------------|-----------------------|----------------|------------------|-------|----------------|--|
| 1   | 1,2,3,4       | ST_REV             | Unsigned16            |                | 0                | None  | S / RO         |  |
| 2   |               | TAG_DESC           | OctString(32)         |                | Spaces           | Na    | S              |  |
| 3   | 4             | STRATEGY           | Unsigned16            |                | 255              | None  | S              |  |
| 4   | 4             | ALERT_KEY          | Unsigned8             |                | 1                | None  | S              |  |
| 5   | 1,3           | MODE_BLK           | DS-69                 |                | Auto             | Na    | S              |  |
| 6   | 1,3           | BLOCK_ERR          | Bitstring(2)          |                |                  | E     | D / RO         |  |
| 7   | 2,4           | FORCE_ENB_16       | Bit String[2]         |                | 0                |       | S              | Controls the updating of each of the 16 analog points when the block is in AUTO mode. A value of 1 will stop the updating of the output point.   |
| 8   |               | CHANNEL_TAG_16     | VisibleString[16][32] |                | Blanks           |       | S              | TAGs of the output points to be mapped in the transducer block.<br>Blank Tag means that the output point is not configured.<br>The MAP_MM_16 parameter indicates if any Tags specified here are not found in a transducer block.                               |
| 9   | 1,3           | SPM_16             | DS-174                |                |                  | %     | N / Auto       | Setpoint parameter with 16 float values and status.  |
| 10  | I,1,3         | CAS_INM_16         | DS-174                |                |                  | %     | N              | Cascade input parameter with 16 float values and status.   |
| 11  | O,1,3         | OUTM_16            | DS-174                |                |                  | %     | N / Man        | Output parameter with 16 float values and status.  |
| 12  | 1,3           | MAP_MM_16          | BitString[2]          |                | 0                | Na    | D / RO         | Indicates which of 16 points configured in CHANNEL_TAG_16 are not associated to an input point in the transducer block: (1) tag was not found in the transducer block, (2) type incompatible, (3) physical point is already associated to another CHANNEL_TAG. |
| 13  | 4             | MO_FSTATE_OPTS_16  | BitString[2]          |                | 0                | Na    | S              | Action to be taken on each of the 16 outputs, when there is fault state. (0: Freezes, 1: Goes to the fault state value configured in FSTATE_VAL_16)  |
| 14  | 4             | MO_POWERUP_OPTS_16 | BitString[2]          |                | 0                | Na    | S              | Bit in 1 indicates that the value of fault state configured in FSTATE_VAL_16 has to be used in power up.   |
| 15  | 4             | FSTATE_TIME        | Float                 | Positive       | 0                | Sec   | S              | Time in seconds between the detection of the fault state condition up to the fault state action.   |
| 16  | 4             | FSTATE_VAL_16      | Float[16]             |                | 0                | %     | S              | Value to be used in case of active fault state, when the corresponding bit in MO_FSTATE_OPTS is 1.   |
| 17  | 1,3           | FSTATE_ST_16       | BitString[2]          |                |                  |       | D / RO         | Shows which points are in active fault state that is indicated by bit in 1.  |

| Idx | Type/<br>View | Parameter     | Data Type | Valid<br>Range | Initial<br>Value | Units | Store/<br>Mode | Description  |
|-----|---------------|---------------|-----------|----------------|------------------|-------|----------------|--|
| 18  | O,3           | BKCAL_OUTM_16 | DS-174    |                |                  |       | D / RO         | Output parameter of back calculation with 16 float values and status.  |
| 19  |               | UPDATE_EVT    | DS-73     |                |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 20  |               | BLOCK_ALM     | DS-72     |                |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

**Legend:** E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static  
Gray Background Line: Default Parameters of Syscon

## MBO64 – 64 Multiple Binary Output

### Description

The cascade and fault state features are available in this block.

### CHANNEL\_TAG\_64A/B parameter

This parameter allows association with the physical points of binary output, which occurs when the value of this parameter coincides with a VAR\_NAME with type compatible, i.e., binary output.

A single physical output point can be associated to only one CHANNEL\_TAG in function block.

### MAP\_MM\_64 parameter

It indicates which of 64 points configured in CHANNEL\_TAG\_64A/B is not associated to binary output point in transducer block, which can be for the following reasons:

- (1) VAR\_NAME equals to CHANNEL\_TAG configured was not found,
- (2) Type is incompatible with the binary output.
- (3) Output point is already associated to some function block.

### FORCE\_ENB\_B64 parameter

When in Cas mode, this parameter enables/disables the point updating in OUTM\_B64, i.e., the transference from the CAS\_IN\_B64 input parameter to the OUTM\_B64 parameter. In the same way for the Auto mode in the transference from SP\_B64 point to OUTM\_B64. The physical point scanning and the transducer block updating continue to be performed and with the value of OUTM\_B64. The transducer block also will indicate this same value. So this parameter allows similar functionality to the Auto/Man mode individually for each point.

### Conditions to Fault State:

| Cause of Fault State       | Previous Actual Mode | Next Actual Mode | FSTATE_ST_64  | Description  |
|----------------------------|----------------------|------------------|---|--|
| CAS_IN_B64.Status=Bad      | Cas                  | LO               | All points configured, which are not in force mode, have the corresponding bit in 1, indicating that they are in fault state. | Force mode has priority over this cause of Fault State.    |
| CAS_IN_B64.Status=Good:IFS | Cas                  | LO               | All points configured, which are not in force mode, have the corresponding bit in 1, indicating that they are in fault state. | Force mode has priority over this cause of Fault State.    |
| RS.FAULT_STATE = Active    | Any                  | LO               | All points configured have the corresponding bit in 1, indicating that they are in fault state.                               | This cause of Fault State has priority over FORCE_ENB_B64. |

### Decision:

- FSTATE\_VAL\_B64 and FORCE\_VAL\_B64 (to be added) are values that will be used by SPM\_B64, when the point is in fault state and force mode, respectively.

### Conditions to Fault State:

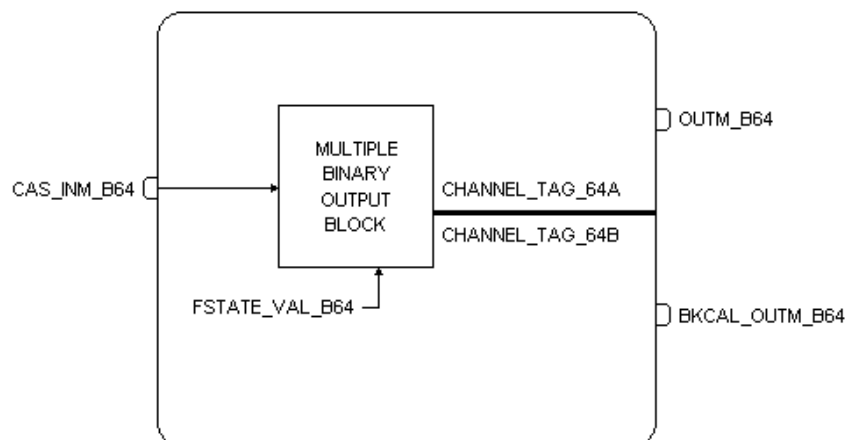
1. Target mode is Cas and CAS\_IN\_B64 status is bad
2. Status of CAS\_IN\_B64 is GoodCas:IFS
3. RS.FAULT\_STATE = Active

### Supported Modes

Cas, Auto, LO, O/S.



## Schematic



## Parameters

| Idx | Type/View | Parameter       | Data Type                 | Range Valid | Initial Value | Units | Store/Mode | Description  |
|-----|-----------|-----------------|---------------------------|-------------|---------------|-------|------------|--|
| 1   | 1,2,3,4   | ST_REV          | Unsigned16                |             | 0             | None  | S / RO     |  |
| 2   |           | TAG_DESC        | OctString(32)             |             | Spaces        | Na    | S          |  |
| 3   | 4         | STRATEGY        | Unsigned16                |             | 255           | None  | S          |  |
| 4   | 4         | ALERT_KEY       | Unsigned8                 |             | 1             | None  | S          |  |
| 5   | 1,3       | MODE_BLK        | DS-69                     |             | Auto          | Na    | S          |  |
| 6   | 1,3       | BLOCK_ERR       | Bitstring(2)              |             |               | E     | D / RO     |  |
| 7   | 2,4       | FORCE_ENB_B64   | Bit String[8]             |             | 0             |       | S          | It controls the updating of each of the 64 binary points when the block is in Auto. A value of 1 in bit stops updating the output point.   |
| 8   |           | CHANNEL_TAG_64A | VisibleString[32]<br>[32] |             | Blanks        |       | S          | Tags of the first group of 32 output points that must be mapped to points on the transducer block. Blank tag means that the output point is not configured. The MAP_MM_64 parameter indicates if some configured tag does not have corresponding in transducer block.  |
| 9   |           | CHANNEL_TAG_64B | VisibleString[32]<br>[32] |             | Blanks        |       | S          | Tags of the second group of 32 output points that must be mapped to points on the transducer block. Blank tag means that the output point is not configured. The MAP_MM_64 parameter indicates if some configured tag does not have corresponding in transducer block. |
| 10  | 1,3       | SPM_B64         | DS-158                    |             |               | Na    | N / Auto   | Setpoint parameter with 64 binary values and status.   |
| 11  | I,1,3     | CAS_INM_B64     | DS-158                    |             |               | Na    | N          | Cascade input parameter with 64 binary values and status.  |
| 12  | O,1,3     | OUTM_B64        | DS-158                    |             |               | none  | N / Man    | Output parameter with 64 binary values and status.   |

| Idx | Type/<br>View | Parameter          | Data Type    | Range<br>Valid | Initial<br>Value | Units | Store/<br>Mode | Description  |
|-----|---------------|--------------------|--------------|----------------|------------------|-------|----------------|--|
| 13  | 1,3           | MAP_MM_64          | BitString[8] |                | 0                | Na    | D / RO         | Indicates which of 64 points configured in CHANNEL_TAG_64A/B is not associated to an input point on transducer block: (1) tag was not found on transducer block, (2) incompatible type. (3) physical point is already associated to another CHANNEL_TAG.   |
| 14  | 4             | MO_FSTATE_OPTS_64  | BitString[8] |                | 0                | Na    | S              | Action to be taken on each of the 64 outputs, when there is fault state. (0: Freezes, 1: Goes to the fault state value configured in FSTATE_VAL_B64)   |
| 15  | 4             | MO_POWERUP_OPTS_64 | BitString[8] |                | 0                | Na    | S              | Bit in 1 indicates that the value of fault state configured in FSTATE_VAL_B64 has to be used in power up.  |
| 16  | 4             | FSTATE_TIME        | Float        | Positive       | 0                | Sec   | S              | Time in seconds between the detection of the fault state condition up to the fault state action.   |
| 17  | 4             | FSTATE_VAL_B64     | BitString[8] |                | 0                |       | S              | Value to be used in case of active fault state, when the corresponding bit in MO_FSTATE_OPTS_64 is 1.  |
| 18  | 1,3           | FSTATE_ST_64       | BitString[8] |                |                  |       | D / RO         | It shows which points are in fault state active, that is indicated by bit in 1.  |
| 19  | O,3           | BKCAL_OUTM_B64     | DS-158       |                |                  |       | D / RO         | Output parameter of back calculation with 64 binary values and status.   |
| 20  |               | UPDATE_EVT         | DS-73        |                |                  | Na    | D              | This alert is generated by any change to the static data.  |
| 21  |               | BLOCK_ALM          | DS-72        |                |                  | Na    | 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 attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed. |

Legend: E – Enumerated parameter; Na – Dimensionless parameter; RO – Read only; D – dynamic; N – non-volatile; S – Static

Gray Background Line: Default Parameters of Syscon

## Block Options

### Resource Block Options

#### HARD\_TYPES

Hardware types supported

| Bit | Meaning            |
|-----|--------------------|
| 0   | Scalar input (LSB) |
| 1   | Scalar output      |
| 2   | Discrete input     |
| 3   | Discrete output    |
| 4   |                    |
| 5   |                    |
| 6   |                    |
| 7   |                    |
| 8   |                    |
| 9   |                    |
| 10  |                    |
| 11  |                    |
| 12  |                    |
| 13  |                    |
| 14  |                    |
| 15  |                    |

#### CYCLE\_TYPE and CYCLE\_SEL

Types of cycle supported

| Bit | Meaning          |
|-----|------------------|
| 0   | Scheduled (LSB)  |
| 1   | Block Execution  |
| 2   | Manufac Specific |
| 3   |                  |
| 4   |                  |
| 5   |                  |
| 6   |                  |
| 7   |                  |
| 8   |                  |
| 9   |                  |
| 10  |                  |
| 11  |                  |
| 12  |                  |
| 13  |                  |
| 14  |                  |
| 15  |                  |

#### FEATURES and FEATURE\_SEL

Things that this resource supports

| Bit | Meaning                                   |
|-----|---|
| 0   | Unicode                                   |
| 1   | Reports                                   |
| 2   | Fault State                               |
| 3   | Soft Write lock                           |
| 4   | Hard Write lock                           |
| 5   | Output readback (*)                       |
| 6   | Direct Write                              |
| 7   | Change of bypass in an automatic mode (*) |
| 8   | MVC supported (*)                         |
| 9   |   |
| 10  |   |
| 11  |   |
| 12  |   |
| 13  |   |
| 14  |   |
| 15  |   |

#### Order of Resource Block Alerts

For ALARM\_SUM and ACK\_OPTION

| Bit | Meaning                  |
|-----|--------------------------|
| 0   | Writes have been enabled |
| 1   |                          |
| 2   |                          |
| 3   |                          |
| 4   |                          |
| 5   |                          |
| 6   |                          |
| 7   | Block alarm              |
| 8   |                          |
| 9   |                          |
| 10  |                          |
| 11  |                          |
| 12  |                          |
| 13  |                          |
| 14  |                          |
| 15  |                          |

(\*) This feature depends on the Hardware Type

**RES\_OPTS**

| Bit | Mnemonic                    | Meaning                             |
|-----|-----------------------------|-------------------------------------|
| 0   | Sched and Link Optimization | Enable function block optimization. |
| 1   | Reserved                    |                                     |
| 2   | Reserved                    |                                     |
| 3   | Reserved                    |                                     |
| 4   | Reserved                    |                                     |
| 5   | Reserved                    |                                     |
| 6   | Reserved                    |                                     |
| 7   | Reserved                    |                                     |
| 8   | Reserved                    |                                     |
| 9   | Reserved                    |                                     |
| 10  | Reserved                    |                                     |
| 11  | Reserved                    |                                     |
| 12  | Reserved                    |                                     |
| 13  | Reserved                    |                                     |
| 14  | Reserved                    |                                     |
| 15  | Reserved                    |                                     |

**RES\_STATUS**

| Bit | Mnemonic             | Meaning   |
|-----|----------------------|---|
| 0   | Schedule Overflow    | Indicates that the resource cannot attend the current macrocycle. |
| 1   | Download in Progress | Indicates the HSE download is in progress.                        |
| 2   | CPU Overload         | Indicates when CPU Overload is higher than 80%.                   |
| 3   | Reserved             |   |
| 4   | Reserved             |   |
| 5   | Reserved             |   |
| 6   | Reserved             |   |
| 7   | Reserved             |   |
| 8   | Reserved             |   |
| 9   | Reserved             |   |
| 10  | Reserved             |   |
| 11  | Reserved             |   |
| 12  | Reserved             |   |
| 13  | Reserved             |   |
| 14  | Reserved             |   |
| 15  | Reserved             |   |

**FEATURES and FEATURE\_SEL****Unicode strings**

This feature is not supported.

**Reports supported**

It is necessary to set this feature in order to enable alert reporting in the resource.

**Fault State supported**

If this feature is selected in FEATURE\_SEL, setting the SET\_FSTATE parameter will force all output function blocks (AO and DO) in the resource to go to fault state.

Individual output function block will go to Fault State due to a loss of communication to CAS\_IN or IFS status in CAS\_IN, regardless the selection of this feature.

**Soft Write lock supported**

It is necessary to select this feature to set the WRITE\_LOCK parameter.

**Hard Write lock supported**

This feature is not supported.

**Output readback supported**

Only the FY302 and FP302 support output readback and this feature is used regardless the selection in FEATURE\_SEL.

**Direct write to output hardware**

This feature is not supported.

**Change of BYPASS in an automatic mode**

If this feature is selected in FEATURE\_SEL, it is allowed to write in BYPASS in an automatic mode, otherwise only in Man or O/S modes.

**MVC supported**

The selection of this feature allows optimize communication performance by transferring a grouped data as a single variable list in either publisher/subscriber transactions for function block links, or report distribution to a host device.

## IO\_OPTS

| Bit | Meaning                            | AI | DI | AO | DO | STEP |
|-----|------------------------------------|----|----|----|----|------|
| 0   | Invert (LSB)                       |    | X  |    | X  |      |
| 1   | SP tracks PV if Man                |    |    | X  | X  |      |
| 2   | SP tracks PV if LO                 |    |    | X  | X  |      |
| 3   | SP tracks Rcas or Cas if LO or Man |    |    | X  | X  |      |
| 4   | Increase to close                  |    |    | X  |    |      |
| 5   | Faultstate Type                    |    |    | X  | X  | X    |
| 6   | Faultstate restart                 |    |    | X  | X  | X    |
| 7   | Target to Man                      |    |    | X  | X  | X    |
| 8   | PV for BKCal_Out                   |    |    | X  | X  |      |
| 9   | Low cutoff                         | X  |    |    |    |      |
| 10  | Low cutoff                         |    |    |    |    |      |
| 11  | Reserved                           |    |    |    |    |      |
| 12  | Reserved                           |    |    |    |    |      |
| 13  | Reserved                           |    |    |    |    |      |
| 14  | Reserved                           |    |    |    |    |      |
| 15  | Reserved                           |    |    |    |    |      |

### Invert

Indicate whether the discrete input value should be logically inverted before it is stored in the process variable.

### SP-PV Track in Man

Permit the setpoint to track the process variable when the target mode of the block is Man.

### SP-PV Track in LO

Permit the setpoint to track the process variable when the actual mode of the block is LO. IMan is not possible in an I/O block.

### SP tracks RCAS or CAS if LO or MAN

Permit the set point to track the Rcas or Cas parameter based on the retained target mode when the actual mode of the block is LO or Man.

Increase to close

Indicate whether the output value should be inverted before it is communicated to the I/O channel.

### Faultstate Type

The output action to take when failure occurs. (0: freeze, 1: go to preset value)

### Faultstate restart

Use the value of FSTATE\_VAL if the device is restarted, otherwise use the non-volatile value. This does not act like Fault State, just uses the value.

### Target to Man if FAULT STATE activated

Set the target mode to Man, thus losing the original target, if Fault State is activated. This latches an output block into the manual mode.

### Use PV for BKCAL\_OUT

The BKCAL\_OUT value is normally the working SP. This option changes it to the PV.

### Low cutoff

The AI low cutoff algorithm is enabled.

## CONTROL\_OPTS

| Bit | Meaning                           | PID | EPID | APID | CHAR | STEP |
|-----|-----------------------------------|-----|------|------|------|------|
| 0   | Bypass Enable (LSB)               | X   | X    | X    | X    |      |
| 1   | SP-PV Track in Man                | X   | X    | X    |      | X    |
| 2   | SP-PV Track in Rout               | X   | X    | X    |      |      |
| 3   | SP-PV Track in LO or Iman         | X   | X    | X    |      | X    |
| 4   | SP Track Retained Target          | X   | X    | X    |      | X    |
| 5   | Direct Acting                     | X   | X    | X    |      | X    |
| 6   | Balance Ramp                      |     |      |      |      |      |
| 7   | Track Enable                      | X   | X    | X    |      |      |
| 8   | Track in Manual                   | X   | X    | X    |      |      |
| 9   | PV for BKCAL_OUT                  | X   | X    | X    |      | X    |
| 10  | Bias may be adjusted              |     |      |      |      |      |
| 11  | Convert IN_1 to Out_Scale         |     |      |      |      |      |
| 12  | Restrict SP limits if Cas or Rcas | X   | X    | X    |      | X    |
| 13  | No OUT limits in Manual           | X   | X    | X    |      |      |
| 14  | Reserved                          |     |      |      |      |      |
| 15  | Reserved                          |     |      |      |      |      |

### Bypass Enable

This parameter, if true, allows BYPASS to be set. Some control algorithm applications cannot provide closed loop control if bypassed.

### SP-PV Track in Man

Permit the setpoint to track the process variable when the target mode of the block is Man.

### SP-PV Track in Rout

Permit the setpoint to track the process variable when the actual mode of the block is ROut.

### SP-PV Track in LO or IMan

Permit the setpoint to track the process variable when the actual mode of the block is LO or IMan.

### SP Track retained target

Permit the setpoint to track the Rcas or Cas parameter based on the retained target mode when the actual mode of the block is IMAN, LO, Man or ROut. When SP-PV Track options are enable, then SP track retained target will have precedence in the selection of the value to track when the actual mode is MAN, IMAN, ROUT and LO.

### Direct Acting

Define the relationship between a change in PV and corresponding change in output. When Direct is selected, an increase in PV results in an increase in the output.

### Track Enable

This enables the external tracking function. If true, the value in TRK\_VAL will replace the value of OUT if TRK\_IN\_D becomes true and the target mode is not Man.

### Track in Manual

This enables TRK\_VAL to replace the value of OUT when the target mode is Man and TRK\_IN\_D is true. The actual mode will then be LO.

### PV for BKCAL\_OUT

The BKCAL\_OUT and RCAS\_OUT values are normally the working SP. If this option is enabled, then the PV value will be used after the CASCADE is closed.

### Convert IN\_1 to Out\_Scale

This feature is not used.

**Restrict SP to limits if Cas or Rcas**

Normally the setpoint will not be restricted to the setpoint limits except when entered by a human interface device. However, if this option is selected, the setpoint will be restricted to the setpoint absolute limits in the Cas and Rcas modes.

**No OUT limits in Manual**

Do not apply OUT\_HI\_LIM or OUT\_LO\_LIM when target and actual mode are Man. Trust the operator to do the right thing.

**STATUS\_OPTS**

| Bit | Meaning                    | AI | DI | PUL | PID | EPID | APID | SPLT | AALM | ISEL | SPG | TIME | LLAG | DENS | FFET | AO | DO | STEP | INTG |
|-----|----------------------------|----|----|-----|-----|------|------|------|------|------|-----|------|------|------|------|----|----|------|------|
| 0   | IFS if BAD IN (LSB)        |    |    |     | X   | X    | X    |      |      |      |     |      |      |      |      |    |    |      |      |
| 1   | IFS if BAD CAS_IN          |    |    |     | X   | X    | X    | X    |      |      |     |      |      |      |      |    |    |      |      |
| 2   | Use Uncertain as Good      |    |    |     | X   | X    | X    | X    | X    | X    | X   | X    | X    | X    | X    |    |    | X    |      |
| 3   | Propagate Fail Forward     | X  | X  | X   |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 4   | Propagate Fail Backward    |    |    |     |     |      |      |      |      |      |     |      |      |      |      | X  | X  | X    |      |
| 5   | Target to Manual if BAD IN |    |    |     | X   | X    | X    |      |      |      |     |      |      |      |      |    |    | X    |      |
| 6   | Uncertain if Limited       | X  |    | X   |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 7   | BAD if Limited             | X  |    | X   |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 8   | Uncertain if Man mode      | X  | X  | X   |     |      |      |      |      |      |     |      |      |      |      |    |    |      | X    |
| 9   | No select if No AUTO       |    |    |     | X   | X    | X    |      |      |      |     |      |      |      |      |    |    | X    |      |
| 10  | No select if no Cas        |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 11  | Reserved                   |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 12  | Reserved                   |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 13  | Reserved                   |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 14  | Reserved                   |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |
| 15  | Reserved                   |    |    |     |     |      |      |      |      |      |     |      |      |      |      |    |    |      |      |

**IFS if BAD IN**

Set Initiate Fault State status in the OUT parameter if the status of the IN parameter is BAD.

**IFS if BAD CAS\_IN**

Set Initiate Fault State status in the OUT parameter if the status of the CAS\_IN parameter is BAD.

**Use Uncertain as Good**

If the status of the IN parameter is Uncertain, treat it as Good. Otherwise, treat it as BAD.

**Propagate Fail Forward**

If the status from the sensor is Bad, Device failure or Bad, Sensor failure, propagate it to OUT without generating an alarm. The use of these sub-status in OUT is determined by this option. Through this option, the user may determine whether alarming (sending of an alert) will be done by the block or propagated downstream for alarming.

**Propagate Fail Backward**

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

### Target to Manual if BAD IN

Set the target mode to Man if the status of the IN parameter is BAD. This latches a PID block into the Man state if the input ever goes bad.

### Uncertain if Limited

Set the output status of an input or calculation block to uncertain if the measured or calculated value is limited.

### BAD if Limited

Set the output status to Bad if the sensor is at a high or low limit.

### Uncertain if Man Mode

Set the output status of an input or calculation block to uncertain if the actual mode of the block is Man.

## ALARM\_SUM and ACK\_OPTION

Valid for all blocks, except for Resource Block.

| Bit | Description   | Meaning              | AI | PUL | DI | APID | PID/EPID | AALM | SPG | DENS | STEP |
|-----|---------------|----------------------|----|-----|----|------|----------|------|-----|------|------|
| 0   | Unack Alarm1  | Discrete alarm       |    |     | x  |      |          |      |     |      |      |
| 1   | Unack Alarm2  | High High alarm      | x  | x   |    | x    | x        | x    |     |      | x    |
| 2   | Unack Alarm3  | High alarm           | x  | x   |    | x    | x        | x    |     | x    | x    |
| 3   | Unack Alarm4  | Low Low alarm        | x  | x   |    | x    | x        | x    |     |      | x    |
| 4   | Unack Alarm5  | Low alarm            | x  | x   |    | x    | x        | x    |     | x    | x    |
| 5   | Unack Alarm6  | Deviation High alarm |    |     |    | x    | x        |      | x   |      | x    |
| 6   | Unack Alarm7  | Deviation Low alarm  |    |     |    | x    | x        |      | x   |      | x    |
| 7   | Unack Alarm8  | Block alarm          | x  | x   | x  | x    | x        | x    | x   | x    | x    |
| 8   | Unack Alarm9  | Not used             |    |     |    |      |          |      |     |      |      |
| 9   | Unack Alarm10 | Not used             |    |     |    |      |          |      |     |      |      |
| 10  | Unack Alarm11 | Not used             |    |     |    |      |          |      |     |      |      |
| 11  | Unack Alarm12 | Not used             |    |     |    |      |          |      |     |      |      |
| 12  | Unack Alarm13 | Not used             |    |     |    |      |          |      |     |      |      |
| 13  | Unack Alarm14 | Not used             |    |     |    |      |          |      |     |      |      |
| 14  | Unack Alarm15 | Not used             |    |     |    |      |          |      |     |      |      |
| 15  | Unack Alarm16 | Not used             |    |     |    |      |          |      |     |      |      |



**PID\_OPTS**

| Bit | Meaning                          | APID | EPID |
|-----|----------------------------------|------|------|
| 0   | IFS if Bad TRK_IN_D              | x    | x    |
| 1   | IFS if Bad TRK_VAL               | x    | x    |
| 2   | Man if Bad TRK_IN_D              | x    | x    |
| 3   | Man if Bad TRK_VAL               | x    | x    |
| 4   | Target to Manual if BAD TRK_IN_D | x    | x    |
| 5   | Target to Manual if BAD TRK_VAL  | x    | x    |
| 6   | Target to Man if Tracking Active | x    | x    |
| 7   | Target to Man if Power Up        | x    | x    |
| 8   | Reserved                         |      |      |
| 9   | Reserved                         |      |      |
| 10  | Reserved                         |      |      |
| 11  | Reserved                         |      |      |
| 12  | Reserved                         |      |      |
| 13  | Reserved                         |      |      |
| 14  | Reserved                         |      |      |
| 15  | Reserved                         |      |      |

**INTEG\_OPTS**

| Bit | Meaning            | INTG |
|-----|--------------------|------|
| 0   | Input 1 accumulate | x    |
| 1   | Input 2 accumulate | x    |
| 2   | Flow forward       | x    |
| 3   | Flow reverse       | x    |
| 4   | Use Uncertain      | x    |
| 5   | Use Bad            | x    |
| 6   | Carry              | x    |
| 7   | Reserved           |      |
| 8   | Reserved           |      |
| 9   | Reserved           |      |
| 10  | Reserved           |      |
| 11  | Reserved           |      |
| 12  | Reserved           |      |
| 13  | Reserved           |      |
| 14  | Reserved           |      |
| 15  | Reserved           |      |

**INVERT\_OPTS**

| Bit | Meaning        | TIMER | AALM | EAI |
|-----|----------------|-------|------|-----|
| 0   | Invert IN_D1   | x     |      |     |
| 1   | Invert IN_D2   | x     |      |     |
| 2   | Invert IN_D3   | x     |      |     |
| 3   | Invert IN_D4   | x     |      |     |
| 4   | Invert OUT_D   | x     | x    |     |
| 5   | Invert OUT_ALM |       | x    | x   |
| 6   | Reserved       |       |      |     |
| 7   | Reserved       |       |      |     |
| 8   | Reserved       |       |      |     |
| 9   | Reserved       |       |      |     |
| 10  | Reserved       |       |      |     |
| 11  | Reserved       |       |      |     |
| 12  | Reserved       |       |      |     |
| 13  | Reserved       |       |      |     |
| 14  | Reserved       |       |      |     |
| 15  | Reserved       |       |      |     |

**INPUT\_OPTS**

| Bit | Meaning             | ARTH |
|-----|---------------------|------|
| 0   | IN Use uncertain    | x    |
| 1   | IN_LO Use uncertain | x    |
| 2   | IN_1 Use uncertain  | x    |
| 3   | IN_1 Use bad        | x    |
| 4   | IN_2 Use uncertain  | x    |
| 5   | IN_2 Use bad        | x    |
| 6   | IN_3 Use uncertain  | x    |
| 7   | IN_3 Use bad        | x    |
| 8   | Reserved            |      |
| 9   | Reserved            |      |
| 10  | Reserved            |      |
| 11  | Reserved            |      |
| 12  | Reserved            |      |
| 13  | Reserved            |      |
| 14  | Reserved            |      |
| 15  | Reserved            |      |

**OSDL\_OPTS**

| Bit | Meaning                       | OSDL |
|-----|-------------------------------|------|
| 0   | IFS if BAD IN                 | x    |
| 1   | IFS if BAD CAS_IN             | x    |
| 2   | Use Uncertain as Good         | x    |
| 3   | IFS for only selected output  | x    |
| 4   | Reserved                      |      |
| 5   | Reserved                      |      |
| 6   | Reserved                      |      |
| 7   | Reserved                      |      |
| 8   | Reserved                      |      |
| 9   | Reserved                      |      |
| 10  | Reserved                      |      |
| 11  | IFS if BAD IN_1               | x    |
| 12  | Keep last value if not select | x    |
| 13  | IFS for only selected output  | x    |
| 14  | Use OUT for BKCAL_OUT         | x    |
| 15  | Use OUT_1 for BKCAL_OUT       | x    |

**MO\_STATUS\_OPTS**

| Bit | Meaning         | MAI | MAO | MDI | MDO |
|-----|-----------------|-----|-----|-----|-----|
| 0   | IFS if BAD IN_1 | x   | x   | x   | x   |
| 1   | IFS if BAD IN_2 | x   | x   | x   | x   |
| 2   | IFS if BAD IN_3 | x   | x   | x   | x   |
| 3   | IFS if BAD IN_4 | x   | x   | x   | x   |
| 4   | IFS if BAD IN_5 | x   | x   | x   | x   |
| 5   | IFS if BAD IN_6 | x   | x   | x   | x   |
| 6   | IFS if BAD IN_7 | x   | x   | x   | x   |
| 7   | IFS if BAD IN_8 | x   | x   | x   | x   |
| 8   | Reserved        |     |     |     |     |
| 9   | Reserved        |     |     |     |     |
| 10  | Reserved        |     |     |     |     |
| 11  | Reserved        |     |     |     |     |
| 12  | Reserved        |     |     |     |     |
| 13  | Reserved        |     |     |     |     |
| 14  | Reserved        |     |     |     |     |
| 15  | Reserved        |     |     |     |     |

## MO\_OPTS

| Bit | Meaning                            | MAI | MAO | MDI | MDO |
|-----|------------------------------------|-----|-----|-----|-----|
| 0   | Fault state to value 1             | x   | x   | x   | x   |
| 1   | Fault state to value 2             | x   | x   | x   | x   |
| 2   | Fault state to value 3             | x   | x   | x   | x   |
| 3   | Fault state to value 4             | x   | x   | x   | x   |
| 4   | Fault state to value 5             | x   | x   | x   | x   |
| 5   | Fault state to value 6             | x   | x   | x   | x   |
| 6   | Fault state to value 7             | x   | x   | x   | x   |
| 7   | Fault state to value 8             | x   | x   | x   | x   |
| 8   | Use fault state value on restart 1 | x   | x   | x   | x   |
| 9   | Use fault state value on restart 2 | x   | x   | x   | x   |
| 10  | Use fault state value on restart 3 | x   | x   | x   | x   |
| 11  | Use fault state value on restart 4 | x   | x   | x   | x   |
| 12  | Use fault state value on restart 5 | x   | x   | x   | x   |
| 13  | Use fault state value on restart 6 | x   | x   | x   | x   |
| 14  | Use fault state value on restart 7 | x   | x   | x   | x   |
| 15  | Use fault state value on restart 8 | x   | x   | x   | x   |

## CT\_OPTS

| Bit | Meaning             | ADT | DDT |
|-----|---------------------|-----|-----|
| 0   | CT_VAL_1 to OUT_1   | x   | x   |
| 1   | CT_VAL_2 to OUT_2   | x   | x   |
| 2   | CT_VAL_3 to OUT_3   | x   | x   |
| 3   | CT_VAL_4 to OUT_4   | x   | x   |
| 4   | CT_VAL_5 to OUT_5   | x   | x   |
| 5   | CT_VAL_6 to OUT_6   | x   | x   |
| 6   | CT_VAL_7 to OUT_7   | x   | x   |
| 7   | CT_VAL_8 to OUT_8   | x   | x   |
| 8   | CT_VAL_9 to OUT_9   | x   | x   |
| 9   | CT_VAL_10 to OUT_10 | x   | x   |
| 10  | CT_VAL_11 to OUT_11 | x   | x   |
| 11  | CT_VAL_12 to OUT_12 | x   | x   |
| 12  | CT_VAL_13 to OUT_13 | x   | x   |
| 13  | CT_VAL_14 to OUT_14 | x   | x   |
| 14  | CT_VAL_15 to OUT_15 | x   | x   |
| 15  | CT_VAL_16 to OUT_16 | x   | x   |

## Hardware Configuration Block Options

### MODULE\_STATUS\_R0\_3

| Bit | Meaning                           |
|-----|-----------------------------------|
| 0   | Status of module in rack 0 slot 0 |
| 1   | Status of module in rack 0 slot 1 |
| 2   | Status of module in rack 0 slot 2 |
| 3   | Status of module in rack 0 slot 3 |
| 4   | Status of module in rack 1 slot 0 |
| 5   | Status of module in rack 1 slot 1 |
| 6   | Status of module in rack 1 slot 2 |
| 7   | Status of module in rack 1 slot 3 |

| Bit | Meaning                           |
|-----|-----------------------------------|
| 0   | Status of module in rack 2 slot 0 |
| 1   | Status of module in rack 2 slot 1 |
| 2   | Status of module in rack 2 slot 2 |
| 3   | Status of module in rack 2 slot 3 |
| 4   | Status of module in rack 3 slot 0 |
| 5   | Status of module in rack 3 slot 1 |
| 6   | Status of module in rack 3 slot 2 |
| 7   | Status of module in rack 3 slot 3 |

### MODULE\_STATUS\_R4\_7

| Bit | Meaning                           |
|-----|-----------------------------------|
| 0   | Status of module in rack 4 slot 0 |
| 1   | Status of module in rack 4 slot 1 |
| 2   | Status of module in rack 4 slot 2 |
| 3   | Status of module in rack 4 slot 3 |
| 4   | Status of module in rack 5 slot 0 |
| 5   | Status of module in rack 5 slot 1 |
| 6   | Status of module in rack 5 slot 2 |
| 7   | Status of module in rack 5 slot 3 |

| Bit | Meaning                           |
|-----|-----------------------------------|
| 0   | Status of module in rack 6 slot 0 |
| 1   | Status of module in rack 6 slot 1 |
| 2   | Status of module in rack 6 slot 2 |
| 3   | Status of module in rack 6 slot 3 |
| 4   | Status of module in rack 7 slot 0 |
| 5   | Status of module in rack 7 slot 1 |
| 6   | Status of module in rack 7 slot 2 |
| 7   | Status of module in rack 7 slot 3 |

## MODULE\_STATUS\_R8\_11

| Bit | Meaning                           |
|-----|-----------------------------------|
| 0   | Status of module in rack 8 slot 0 |
| 1   | Status of module in rack 8 slot 1 |
| 2   | Status of module in rack 8 slot 2 |
| 3   | Status of module in rack 8 slot 3 |
| 4   | Status of module in rack 9 slot 0 |
| 5   | Status of module in rack 9 slot 1 |
| 6   | Status of module in rack 9 slot 2 |
| 7   | Status of module in rack 9 slot 3 |

| Bit | Meaning                            |
|-----|------------------------------------|
| 0   | Status of module in rack 10 slot 0 |
| 1   | Status of module in rack 10 slot 1 |
| 2   | Status of module in rack 10 slot 2 |
| 3   | Status of module in rack 10 slot 3 |
| 4   | Status of module in rack 11 slot 0 |
| 5   | Status of module in rack 11 slot 1 |
| 6   | Status of module in rack 11 slot 2 |
| 7   | Status of module in rack 11 slot 3 |

## MODULE\_STATUS\_R12\_14

| Bit | Meaning                            |
|-----|------------------------------------|
| 0   | Status of module in rack 12 slot 0 |
| 1   | Status of module in rack 12 slot 1 |
| 2   | Status of module in rack 12 slot 2 |
| 3   | Status of module in rack 12 slot 3 |
| 4   | Status of module in rack 13 slot 0 |
| 5   | Status of module in rack 13 slot 1 |
| 6   | Status of module in rack 13 slot 2 |
| 7   | Status of module in rack 13 slot 3 |

| Bit | Meaning                            |
|-----|------------------------------------|
| 0   | Status of module in rack 14 slot 0 |
| 1   | Status of module in rack 14 slot 1 |
| 2   | Status of module in rack 14 slot 2 |
| 3   | Status of module in rack 14 slot 3 |
| 4   |                                    |
| 5   |                                    |
| 6   |                                    |
| 7   |                                    |

**ELEM\_MISMAT\_BS\_1[4]**

| Bit | ELEM_MISMAT_BS_1[0] | ELEM_MISMAT_BS_1[1] | ELEM_MISMAT_BS_1[2] | ELEM_MISMAT_BS_1[3] |
|-----|---------------------|---------------------|---------------------|---------------------|
| 0   | Rack0 Slot0 (LSB)*  | Rack2 Slot0 (LSB)*  | Rack4 Slot0 (LSB)*  | Rack6 Slot0 (LSB)*  |
| 1   | Rack0 Slot1         | Rack2 Slot1         | Rack4 Slot1         | Rack6 Slot1         |
| 2   | Rack0 Slot2         | Rack2 Slot2         | Rack4 Slot2         | Rack6 Slot2         |
| 3   | Rack0 Slot3         | Rack2 Slot3         | Rack4 Slot3         | Rack6 Slot3         |
| 4   | Rack1 Slot0         | Rack3 Slot0         | Rack5 Slot0         | Rack7 Slot0         |
| 5   | Rack1 Slot1         | Rack3 Slot1         | Rack5 Slot1         | Rack7 Slot1         |
| 6   | Rack1 Slot2         | Rack3 Slot2         | Rack5 Slot2         | Rack7 Slot2         |
| 7   | Rack1 Slot3         | Rack3 Slot3         | Rack5 Slot3         | Rack7 Slot3         |

**ELEM\_MISMAT\_BS\_2[4]**

| Bit | ELEM_MISMAT_BS_2[0] | ELEM_MISMAT_BS_2[1] | ELEM_MISMAT_BS_2[2] | ELEM_MISMAT_BS_2[3] |
|-----|---------------------|---------------------|---------------------|---------------------|
| 0   | Rack8 Slot0 (LSB)*  | Rack10 Slot0 (LSB)* | Rack12 Slot0 (LSB)* | Rack14 Slot0 (LSB)* |
| 1   | Rack8 Slot1         | Rack10 Slot1        | Rack12 Slot1        | Rack14 Slot1        |
| 2   | Rack8 Slot2         | Rack10 Slot2        | Rack12 Slot2        | Rack14 Slot2        |
| 3   | Rack8 Slot3         | Rack10 Slot3        | Rack12 Slot3        | Rack14 Slot3        |
| 4   | Rack9 Slot0         | Rack11 Slot0        | Rack13 Slot0        | Rack15 Slot0        |
| 5   | Rack9 Slot1         | Rack11 Slot1        | Rack13 Slot1        | Rack15 Slot1        |
| 6   | Rack9 Slot2         | Rack11 Slot2        | Rack13 Slot2        | Rack15 Slot2        |
| 7   | Rack9 Slot3         | Rack11 Slot3        | Rack13 Slot3        | Rack15 Slot3        |

**MO\_FSTATE\_OPTS\_16[2]**

| Bit | Meaning                       |
|-----|-------------------------------|
| 0   | Fault state to point 1 (LSB)* |
| 1   | Fault state to point 2        |
| 2   | Fault state to point 3        |
| 3   | Fault state to point 4        |
| 4   | Fault state to point 5        |
|     |                               |
| ..  | ..                            |
| 14  | Fault state to point 15       |
| 15  | Fault state to point 16       |

**MO\_FSTATE\_OPTS\_64[8]**

| Bit | Meaning                       |
|-----|-------------------------------|
| 0   | Fault state to point 1 (LSB)* |
| 1   | Fault state to point 2        |
| 2   | Fault state to point 3        |
| 3   | Fault state to point 4        |
| 4   | Fault state to point 5        |
| ..  | ..                            |
| 62  | Fault state to point 63       |
| 63  | Fault state to point 64       |

**MO\_POWERUP\_OPTS\_16[2]**

| Bit | Meaning   |
|-----|---|
| 0   | Use fault state value for point 1 on restart (LSB)* |
| 1   | Use fault state value for point 2 on restart        |
| 2   | Use fault state value for point 3 on restart        |
| 3   | Use fault state value for point 4 on restart        |
| 4   | Use fault state value for point 5 on restart        |
|     |   |
| ..  | ..  |
| 14  | Use fault state value for point 15 on restart       |
| 15  | Use fault state value for point 16 on restart       |

**MO\_POWERUP\_OPTS\_64[8]**

| Bit | Meaning   |
|-----|---|
| 0   | Use fault state value for point 1 on restart (LSB)* |
| 1   | Use fault state value for point 2 on restart        |
| 2   | Use fault state value for point 3 on restart        |
| 3   | Use fault state value for point 4 on restart        |
| 4   | Use fault state value for point 5 on restart        |
|     |   |
| ..  | ..  |
| 62  | Use fault state value for point 15 on restart       |
| 63  | Use fault state value for point 16 on restart       |

**FSTATE\_ST\_16[2]**

| Bit | Meaning                                      |
|-----|--|
| 0   | Fault state active status for point 1 (LSB)* |
| 1   | Fault state active status for point 2        |
| 2   | Fault state active status for point 3        |
|     |  |
| ..  | ..   |
| 14  | Fault state active status for point 15       |
| 15  | Fault state active status for point 16       |

**FSTATE\_ST\_64[8]**

| Bit | Meaning                                      |
|-----|--|
| 0   | Fault state active status for point 1 (LSB)* |
| 1   | Fault state active status for point 2        |
| 2   | Fault state active status for point 3        |
|     |  |
| ..  | ..   |
| 62  | Fault state active status for point 63       |
| 63  | Fault state active status for point 64       |



**MAP\_MM\_16**

| Bit | MAP_MM_64[0]   | MAP_MM_64[1]   |
|-----|----------------|----------------|
| 0   | Point 1 (LSB)* | Point 9 (LSB)* |
| 1   | Point 2        | Point 10       |
| 2   | Point 3        | Point 11       |
| 3   | Point 4        | Point 12       |
| 4   | Point 5        | Point 13       |
| 5   | Point 6        | Point 14       |
| 6   | Point 7        | Point 15       |
| 7   | Point 8        | Point 16       |

**MAP\_MM\_64**

| Bit | MAP_MM_64[0]   | MAP_MM_64[1]   | MAP_MM_64[2]    | MAP_MM_64[3]    |
|-----|----------------|----------------|-----------------|-----------------|
| 0   | Point 1 (LSB)* | Point 9 (LSB)* | Point 17 (LSB)* | Point 25 (LSB)* |
| 1   | Point 2        | Point 10       | Point 18        | Point 26        |
| 2   | Point 3        | Point 11       | Point 19        | Point 27        |
| 3   | Point 4        | Point 12       | Point 20        | Point 28        |
| 4   | Point 5        | Point 13       | Point 21        | Point 29        |
| 5   | Point 6        | Point 14       | Point 22        | Point 30        |
| 6   | Point 7        | Point 15       | Point 23        | Point 31        |
| 7   | Point 8        | Point 16       | Point 24        | Point 32        |

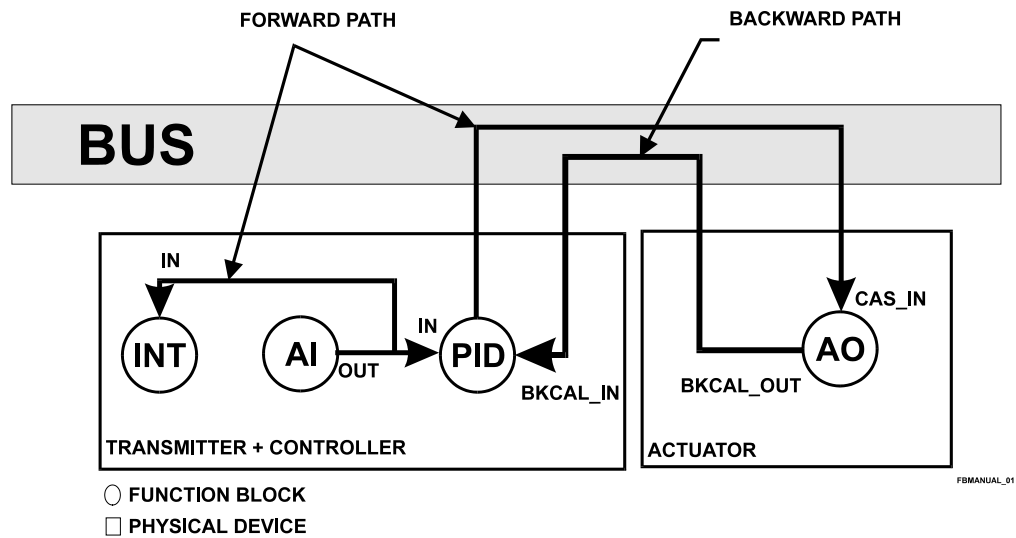
| Bit | MAP_MM_64[4]    | MAP_MM_64[5]    | MAP_MM_64[6]    | MAP_MM_64[7]    |
|-----|-----------------|-----------------|-----------------|-----------------|
| 0   | Point 33 (LSB)* | Point 41 (LSB)* | Point 49 (LSB)* | Point 57 (LSB)* |
| 1   | Point 34        | Point 42        | Point 50        | Point 58        |
| 2   | Point 35        | Point 43        | Point 51        | Point 59        |
| 3   | Point 36        | Point 44        | Point 52        | Point 60        |
| 4   | Point 37        | Point 45        | Point 53        | Point 61        |
| 5   | Point 38        | Point 46        | Point 54        | Point 62        |
| 6   | Point 39        | Point 47        | Point 55        | Point 63        |
| 7   | Point 40        | Point 48        | Point 56        | Point 64        |



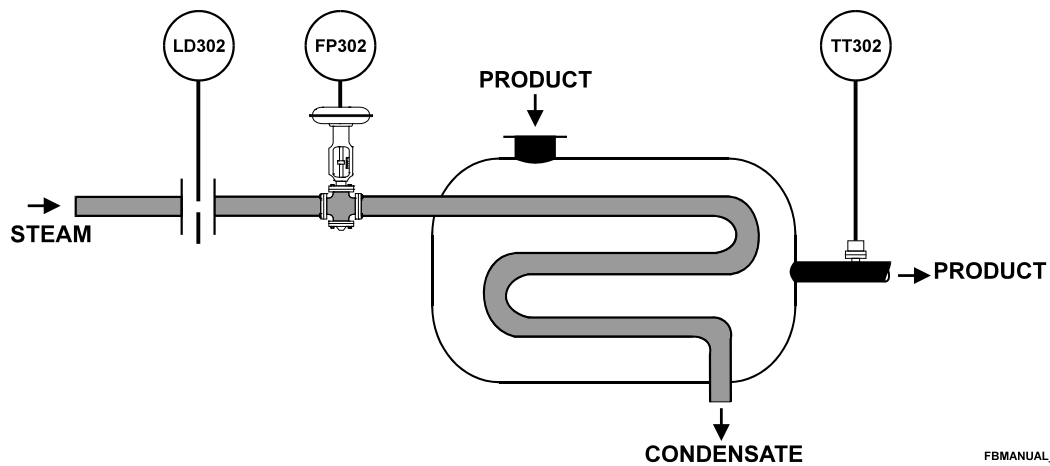
## Chapter 3

### EXAMPLES

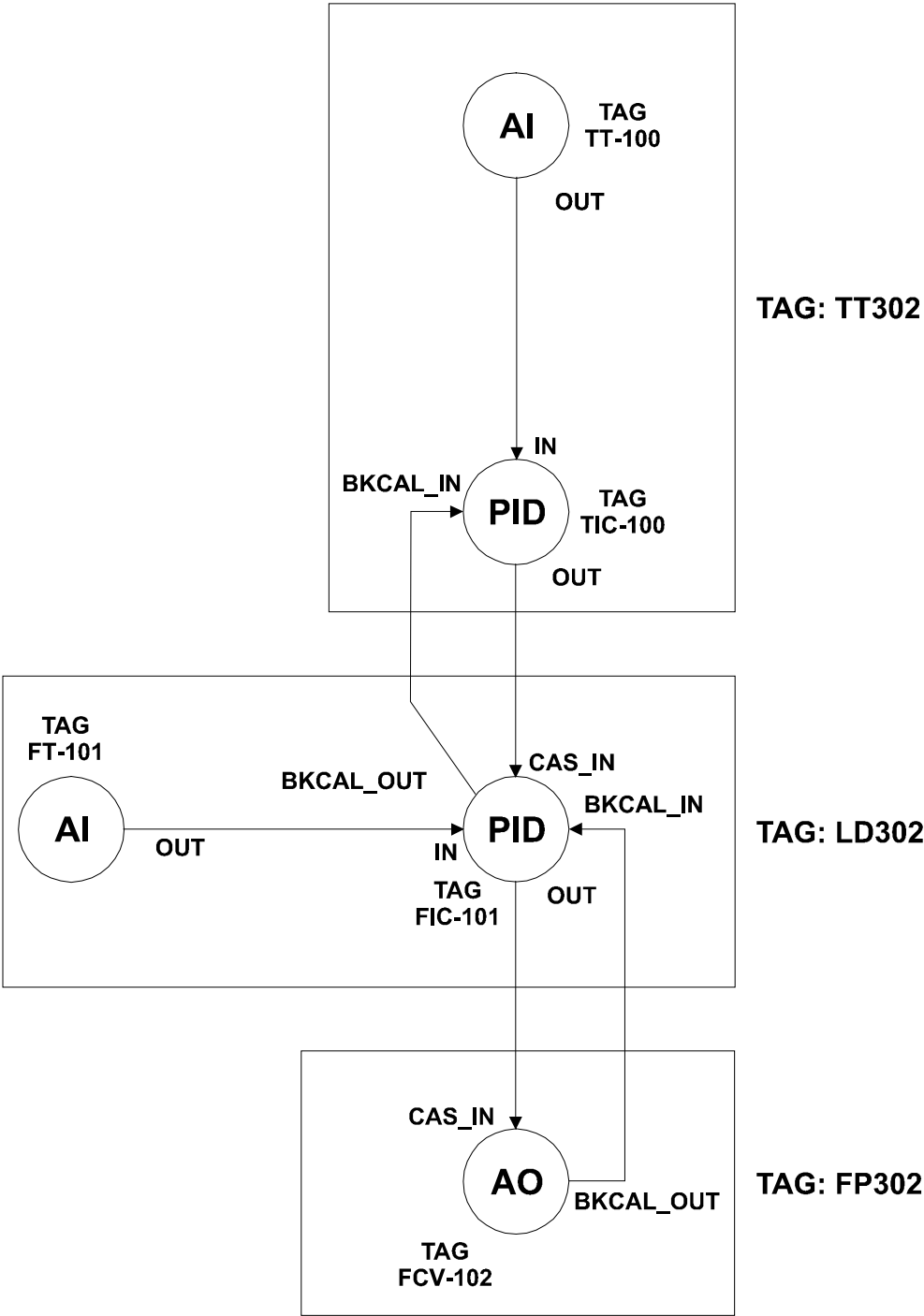
#### Simple Control Application



#### Cascade Control



Corresponding Configuration

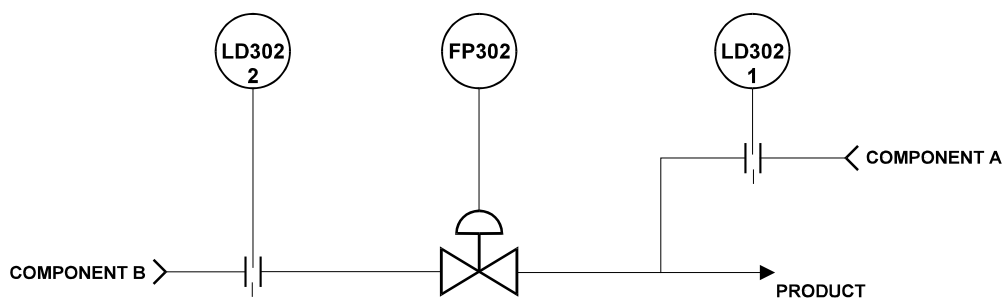


FBMANUAL\_03

## Parameterization

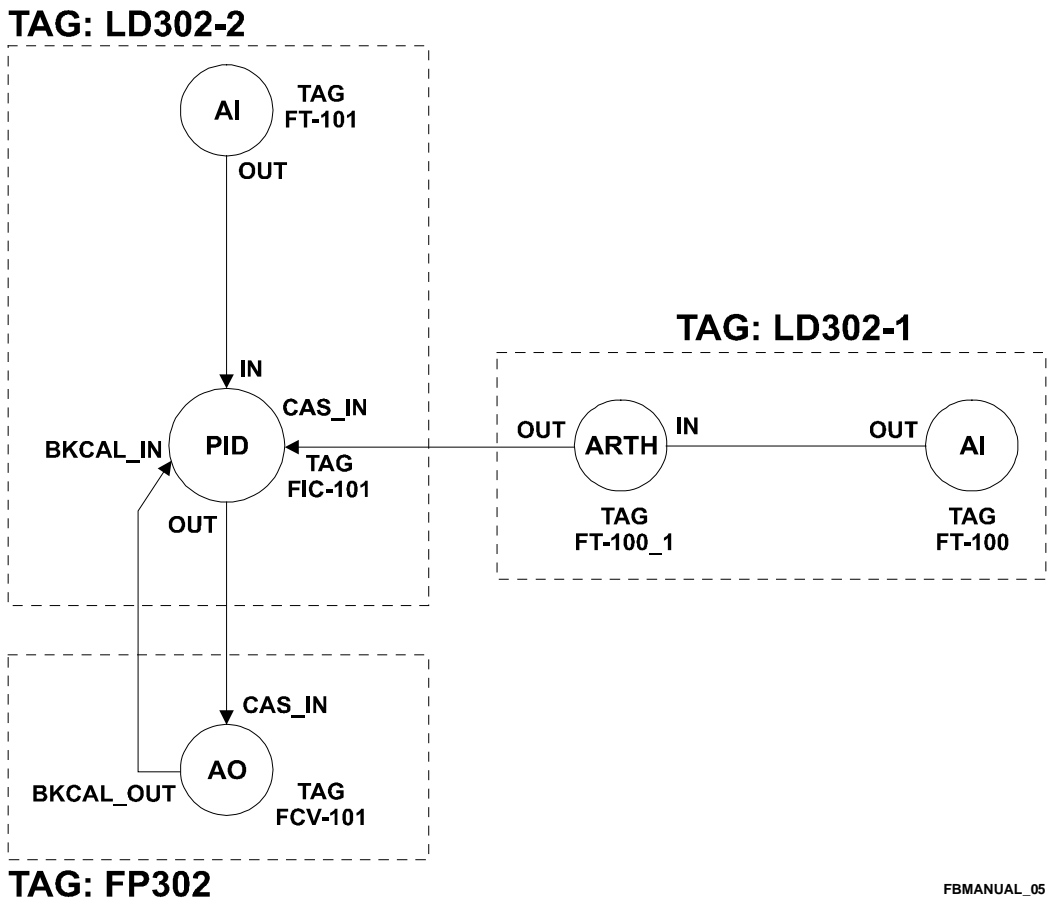
| Device | Block | Block Tag | Parameter | Element | Value                 |
|--------|-------|-----------|-----------|---------|-----------------------|
| TT302  | AI    | TT-100    | MODE_BLK  | Target  | AUTO                  |
|        |       |           | MODE_BLK  | Target  | AUTO                  |
|        | PID   | TIC-100   | PV_SCALE  |         | 0-600 °C              |
|        |       |           | OUT_SCALE |         | 0-200 kg/h            |
| LD302  | AI    | FT-101    | MODE_BLK  | Target  | AUTO                  |
|        |       |           | L_TYPE    |         | Indirect, square root |
|        |       |           | XD_SCALE  |         | 0-200 in H2O          |
|        |       |           | OUT_SCALE |         | 0-200 kg/h            |
|        | PID   | FIC_101   | MODE_BLK  | Target  | CAS                   |
|        |       |           | PV_SCALE  |         | 0-200 kg/h            |
|        |       |           | OUT_SCALE |         | 0-100%                |
| FP302  | AO    | FCV-102   | MODE_BLK  | Target  | CAS                   |
|        |       |           | PV_SCALE  |         | 0-100%                |
|        |       |           | XD_SCALE  |         | 3-15 psi              |

## Ratio Control



FBMANUAL\_04

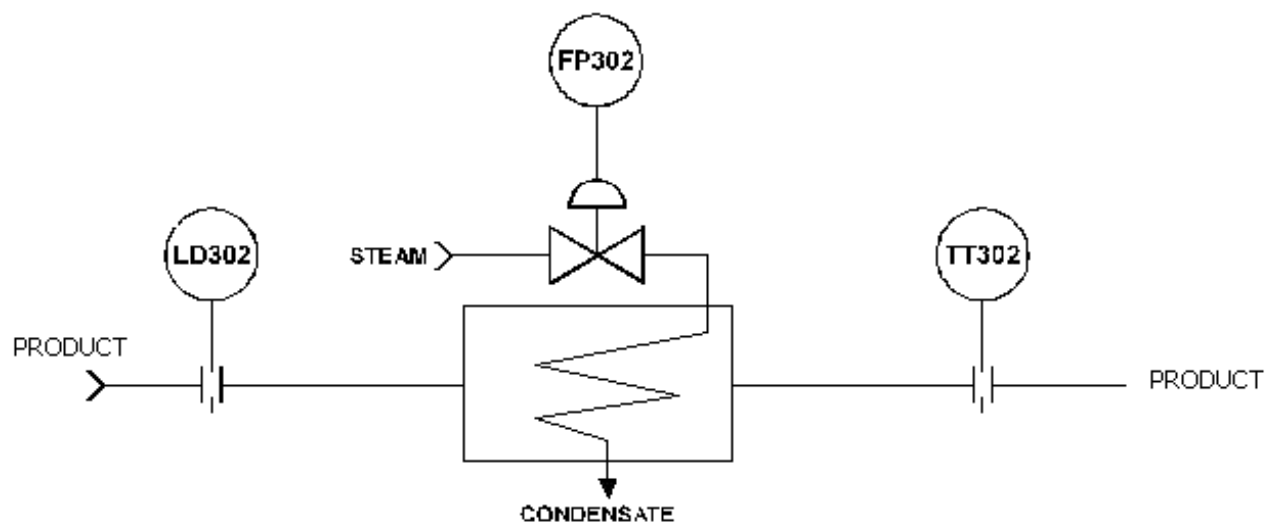
Corresponding Configuration



Parameterization

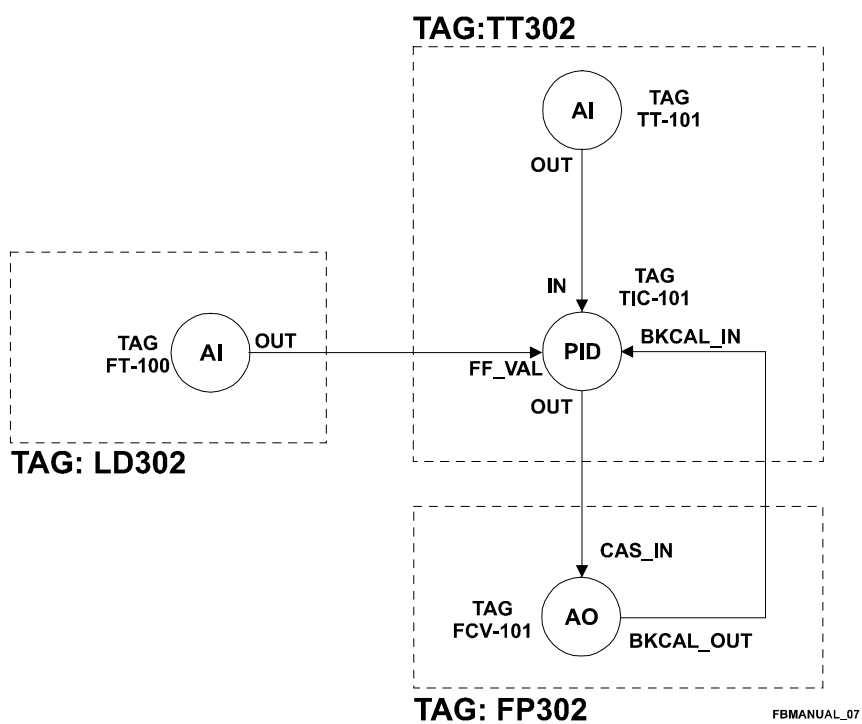
| Device  | Block | Block Tag | Parameter  | Element | Value                                  |
|---------|-------|-----------|------------|---------|--|
| LD302-1 | AI    | FT-100    | MODE_BLK   | Target  | AUTO                                   |
|         |       |           | MODE_BLK   | Target  | AUTO                                   |
|         |       |           | ARITH_TYPE |         | 7                                      |
|         |       |           | GAIN       |         | Adjusted by user to the desired rate p |
|         |       |           | RANGE_LO   |         | 0                                      |
|         |       |           | RANGE_HI   |         | -10 (for g = 1)                        |
| LD302-2 | PID   | FIC-101   | MODE_BLK   | Target  | Cas                                    |
|         |       |           | PV_SCALE   |         | 0-200 in H2O                           |
|         |       |           | OUT_SCALE  |         | 0-100 %                                |
|         |       |           | XD_SCALE   |         | 3-15 psi                               |

## Feedforward Control



FBMANUAL\_06

### Corresponding Configuration

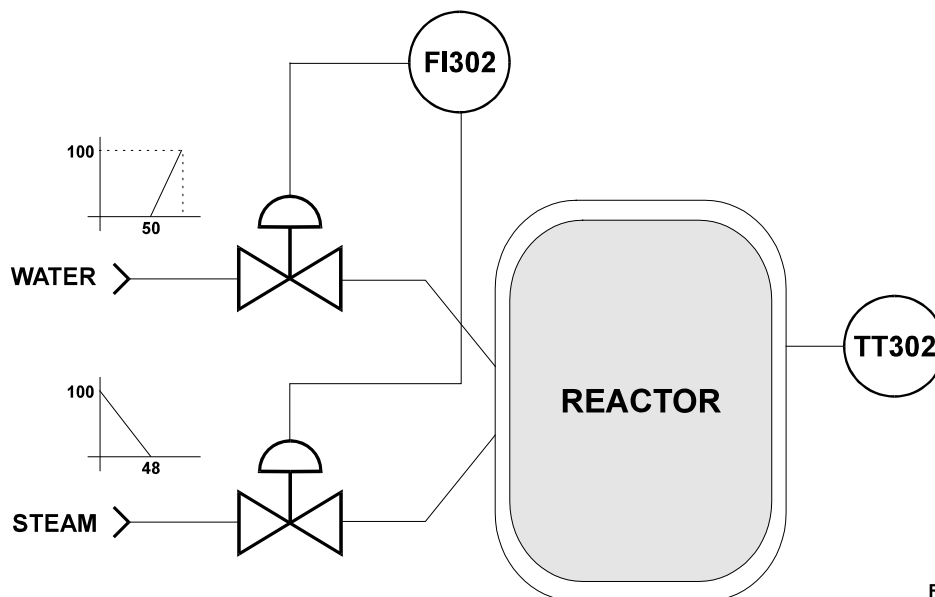


FBMANUAL\_07

## Parameterization

| Device | Block | Block Tag | Parameter | Element | Value                     |
|--------|-------|-----------|-----------|---------|---------------------------|
| TT302  | AI    | TT-101    | MODE_BLK  | Target  | AUTO                      |
|        | PID   | TIC-101   | MODE_BLK  | Target  | AUTO                      |
|        |       |           | PV_SCALE  |         | 0-600 °C                  |
|        |       |           | FF_SCALE  |         | 0-500 GAL/min             |
|        |       |           | FF_GAIN   |         | 0.1                       |
| LD302  | AI    | FT-100    | MODE_BLK  | Target  | AUTO                      |
|        |       |           | L_TYPE    |         | Indirect, square root     |
|        |       |           | XD_SCALE  |         | 0-125 in H <sub>2</sub> O |
|        |       |           | OUT_SCALE |         | 0-500 GAL/min             |
| FP302  | AO    | FCV-101   | MODE_BLK  | Target  | CAS                       |
|        |       |           | PV_SCALE  |         | 0-100%                    |
|        |       |           | XD_SCALE  |         | 3-15 psi                  |

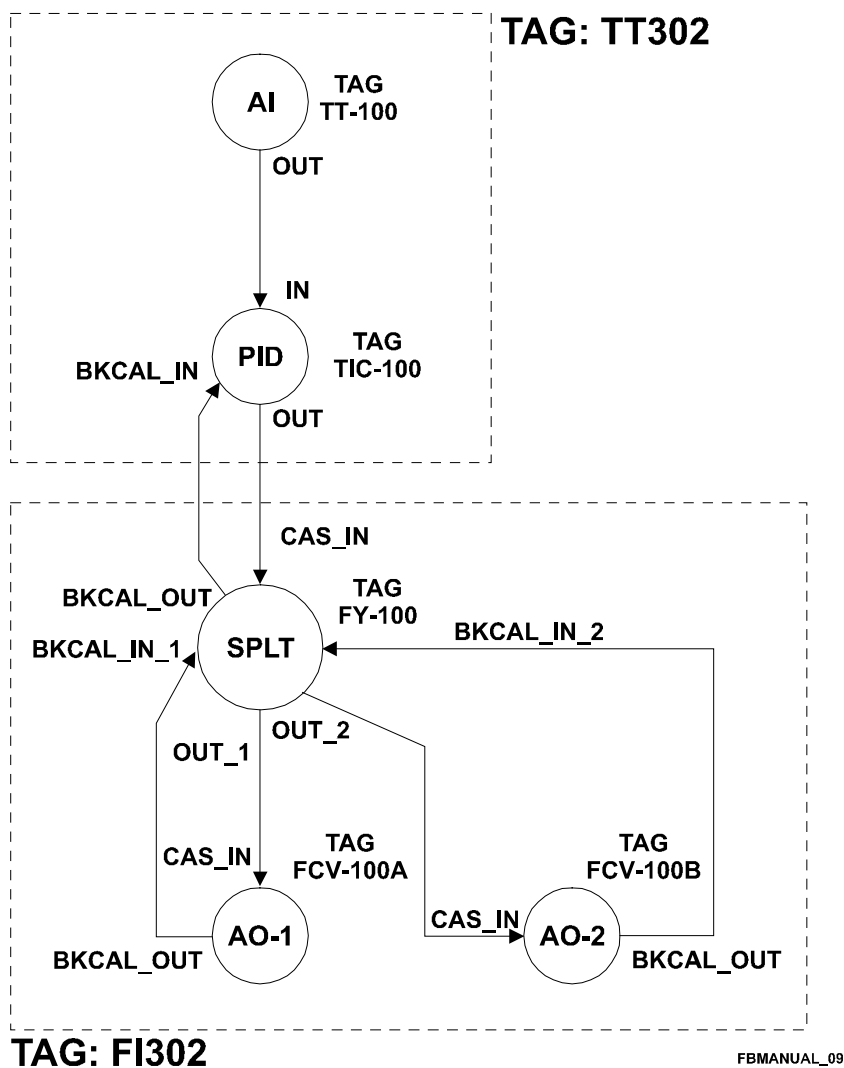
## Split Range Control



FBMANUAL\_08



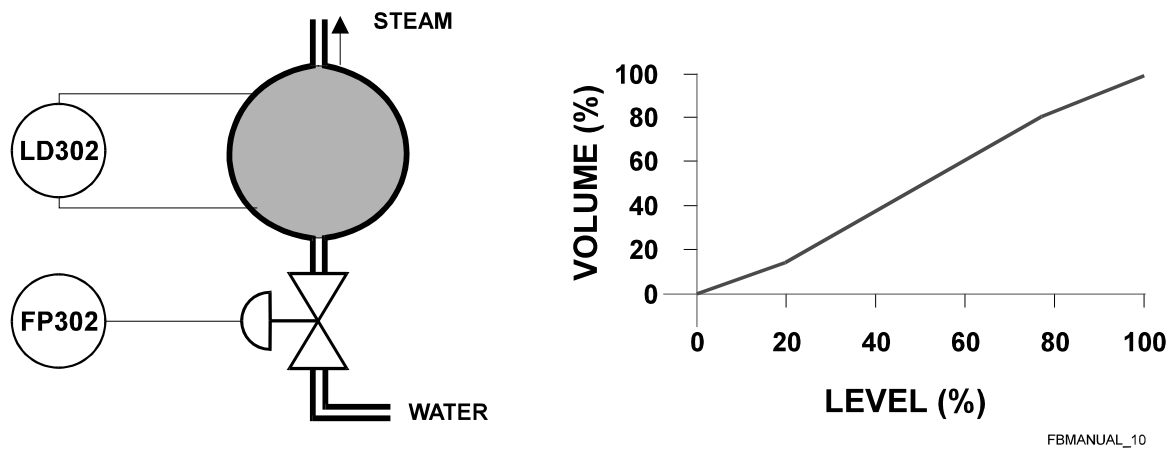
## Corresponding Configuration



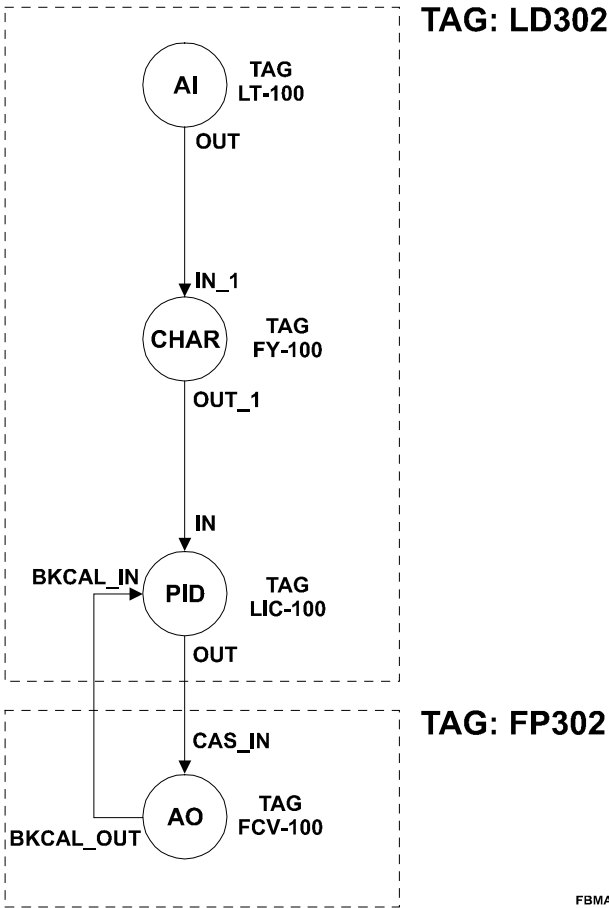
## Parameterization

| Device | Block | Block Tag | Parameter | Element | Value          |
|--------|-------|-----------|-----------|---------|----------------|
| TT302  | AI    | TT-100    | MODE_BLK  | Target  | AUTO           |
|        |       |           |           |         |                |
|        |       |           |           |         |                |
|        |       |           |           |         |                |
|        | PID   | TIC-100   | MODE_BLK  | Target  | AUTO           |
|        |       |           | PV_SCALE  |         | 0-600 °C       |
|        |       |           | OUT_SCALE |         | 0-100 %        |
| FI302  | SPLT  | FY-100    | MODE_BLK  | Target  | Cas            |
|        |       |           | LOCKVAL   |         | Yes            |
|        |       |           | IN_ARRAY  |         | 0, 48, 50, 100 |
|        |       |           | OUT_ARRAY |         | 100, 0, 0, 100 |
|        | AO    | FCV-100A  | MODE_BLK  | Target  | Cas            |
|        |       |           | PV_SCALE  |         | 0-100 %        |
|        |       |           | XD_SCALE  |         | 4-20 mA        |
|        | AO 2  | FCV-100B  | MODE_BLK  | Target  | Cas            |
|        |       |           | PV_SCALE  |         | 0-100 %        |
|        |       |           | XD_SCALE  |         | 4-20 mA        |

Level Control



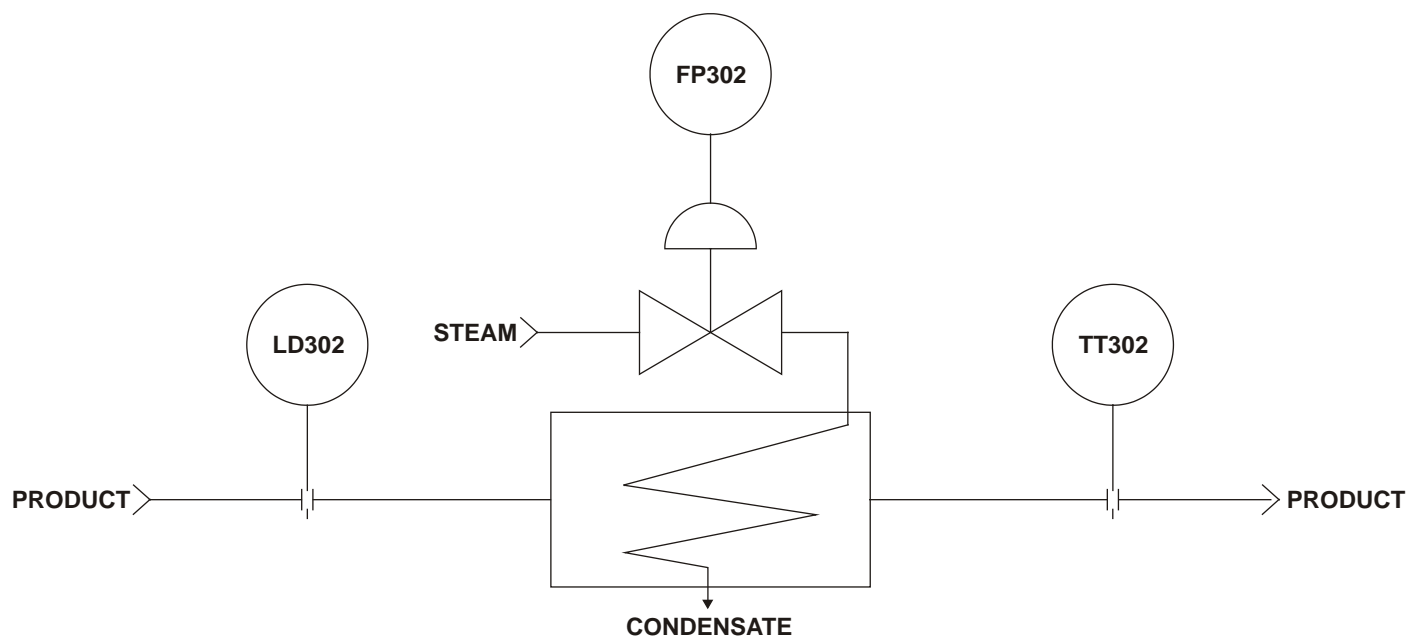
Corresponding Configuration



## Parameterization

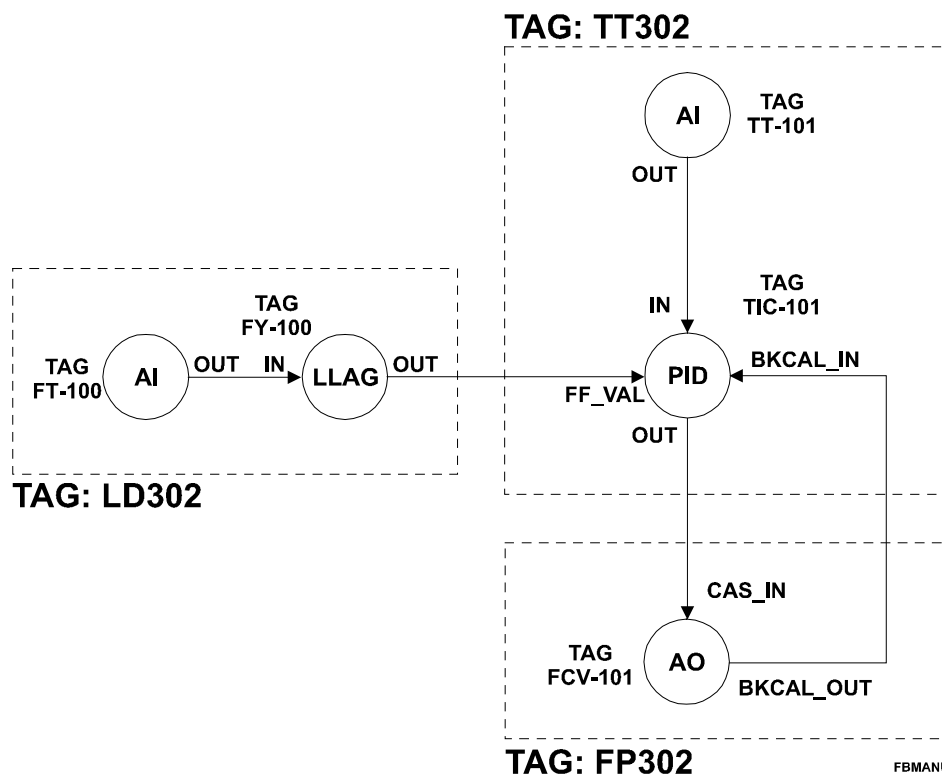
| Device | Block | Block Tag | Parameter | Element | Value                                  |
|--------|-------|-----------|-----------|---------|--|
| LD302  | AI    | LT-100    | MODE_BLK  | Target  | AUTO                                   |
|        | CHAR  | FY-100    | MODE_BLK  | Target  | AUTO                                   |
|        |       |           | X_RANGE   |         | inH2O                                  |
|        |       |           | Y_RANGE   |         | gal                                    |
|        |       |           | CURVE_X   |         | 0,40,80,100,120,160,200                |
|        |       |           | CURVE_Y   |         | 0, 14.23, 37.35, 50, 62.64, 85.76, 100 |
|        | PID   | LIC-100   | MODE_BLK  | Target  | AUTO                                   |
|        |       |           | PV_SCALE  |         | 0-100 gal                              |
|        |       |           | OUT_SCALE |         | 0-100 %                                |
| FP302  | AO    | FCV-100   | MODE_BLK  | Target  | Cas                                    |
|        |       |           | PV_SCALE  |         | 0-100 %                                |
|        |       |           | XD_SCALE  |         | 3-15 psi                               |

### ***Rate Control Loop with Lead - Lag***



FBMANUAL\_12

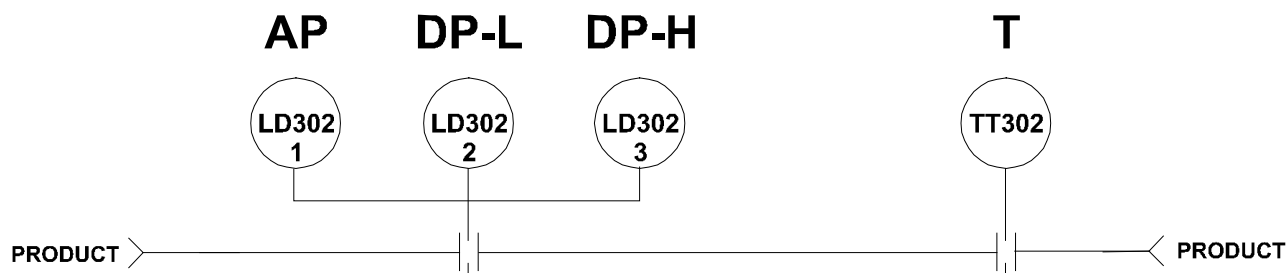
## Corresponding Configuration



## Parameterization

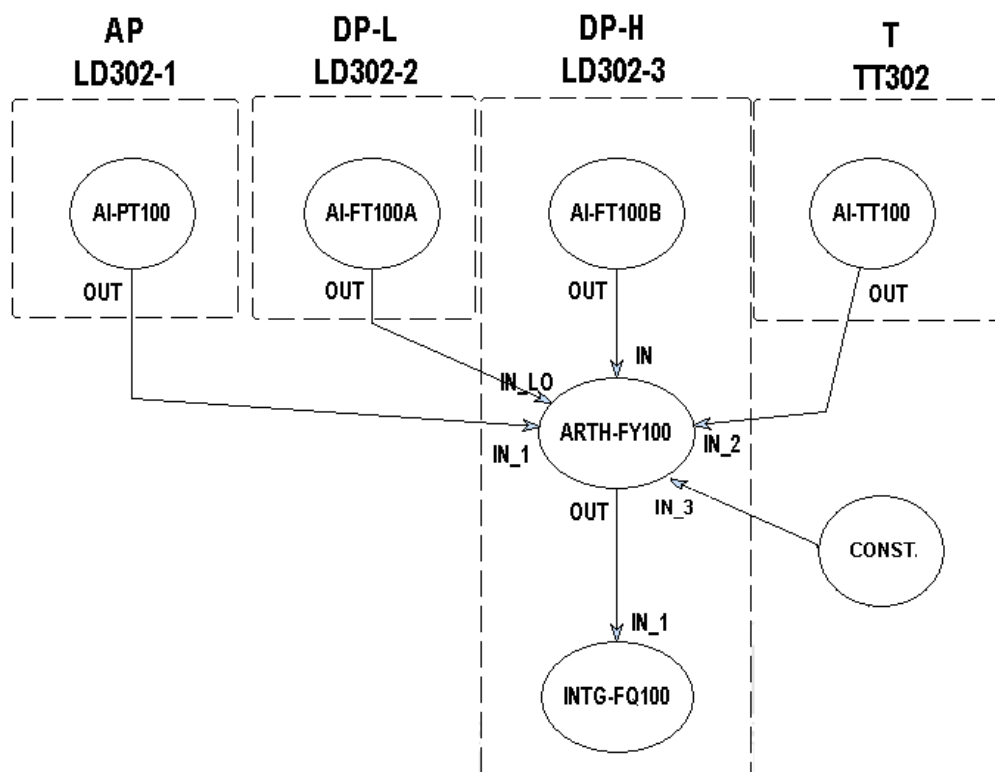
| Device | Block | Block Tag | Parameter | Element | Value                 |
|--------|-------|-----------|-----------|---------|-----------------------|
| TT302  | AI    | TT-101    | MODE_BLK  | Target  | AUTO                  |
|        |       |           |           |         |                       |
|        | PID   | TIC-101   | MODE_BLK  | Target  | AUTO                  |
|        |       |           | PV_SCALE  |         | 0-100 gal             |
|        |       |           | OUT_SCALE |         | 0-100 %               |
|        |       |           | FF_SCALE  |         | 0-500 GAL/min         |
| LD302  | AI    | FT-100    | MODE_BLK  | Target  | AUTO                  |
|        |       |           | XD_SCALE  |         | 0-125 inH2O           |
|        |       |           | OUT_SCALE |         | 0-500GAL/min          |
|        |       |           | L_TYPE    |         | Indirect, square root |
|        | LLAG  | FY-100    | MODE_BLK  | Target  | AUTO                  |
|        |       |           | OUT_RANGE |         | GAL/min               |
|        |       |           | LEAD_TIME |         | 60                    |
|        |       |           | LAG_TIME  |         | 60                    |
| FP302  | AO    | FCV-101   | MODE_BLK  | Target  | Cas                   |
|        |       |           | PV_SCALE  |         | 0-100 %               |
|        |       |           | XD_SCALE  |         | 3-15 psi              |

## Flow Compensation Configuration with Totalization



FBMANUAL\_14

### Corresponding Configuration

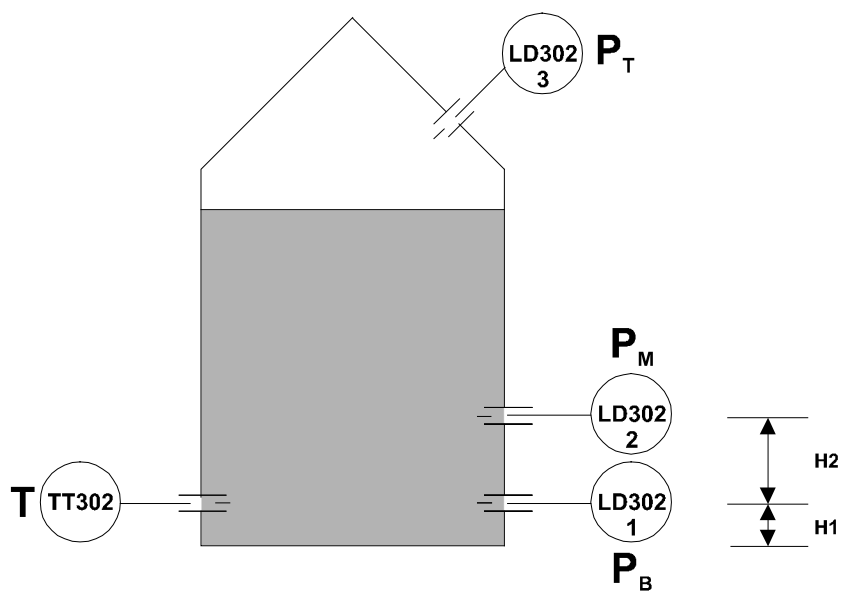


### Parameterization

| Device  | Block | Block Tag | Parameter | Element | Value          |
|---------|-------|-----------|-----------|---------|----------------|
| LD302-1 | AI    | PT-100    | MODE_BLK  | Target  | AUTO           |
|         |       |           | L_TYPE    |         | Direct         |
|         |       |           | XD_SCALE  | UNIT    | Pa             |
| LD302-2 | AI    | FT-100A   | MODE_BLK  | Target  | AUTO           |
|         |       |           | XD_SCALE  |         | 0-20 H2O       |
|         |       |           | OUT_SCALE |         | 0-156 Cutf/min |
|         |       |           | L_TYPE    |         | SQR ROOT       |
| LD302-3 | AI    | FT-100B   | MODE_BLK  | Target  | AUTO           |
|         |       |           | XD_SCALE  |         | 0-200 in H2O   |

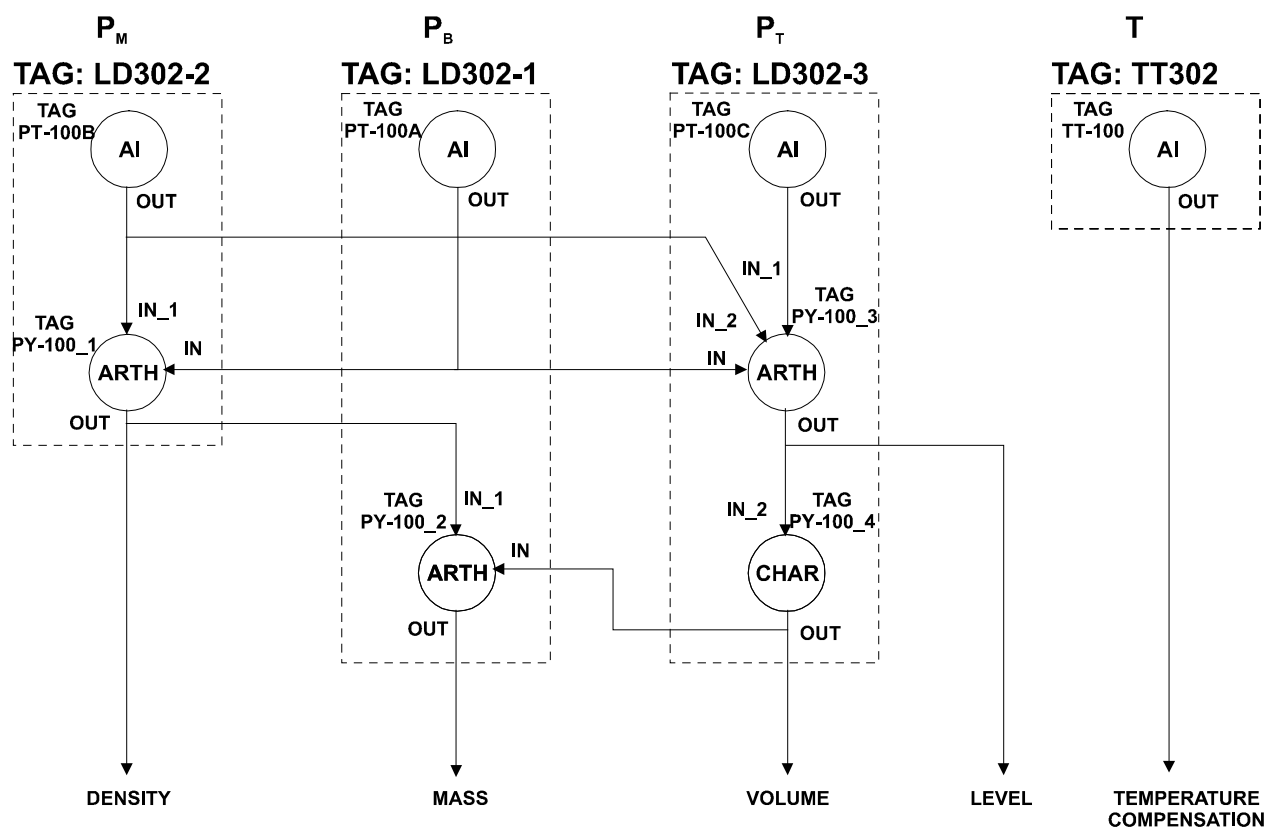
| Device | Block | Block Tag | Parameter   | Element | Value                             |
|--------|-------|-----------|-------------|---------|-----------------------------------|
|        | ARTH  | FY-100    | OUT_SCALE   |         | 0-495 Cutf/min                    |
|        |       |           | L_TYPE      |         | SQR ROOT                          |
|        |       |           | MODE_BLK    | Target  | AUTO                              |
|        |       |           | PV_RANGE    |         | GAL/min                           |
|        |       |           | OUT_RANGE   |         | GAL/min                           |
|        |       |           | ARITH_TYPE  |         | 2 (flow comp. square root – AGA3) |
|        |       |           | GAIN_IN_1   |         | 1                                 |
|        |       |           | GAIN        |         | 1                                 |
|        |       |           | RANGE_LO    |         | 400                               |
|        |       |           | RANGE_HI    |         | 600                               |
|        |       |           | COMP_HI_LIM |         | + INF                             |
|        |       |           | COMP_LO_LIM |         | - INF                             |
|        | INT   | FQ-100    | MODE_BLK    | Target  | AUTO                              |
|        |       |           | OUT_RANGE   |         | GAL                               |
| TT302  | AI    | TT-100    | MODE_BLK    | Target  | AUTO                              |
|        |       |           | XD_SCALE    | UNIT    | K                                 |

## Hydrostatic Tank Gauging



FBMANUAL\_16

## Corresponding Configuration



FBMANUAL\_17

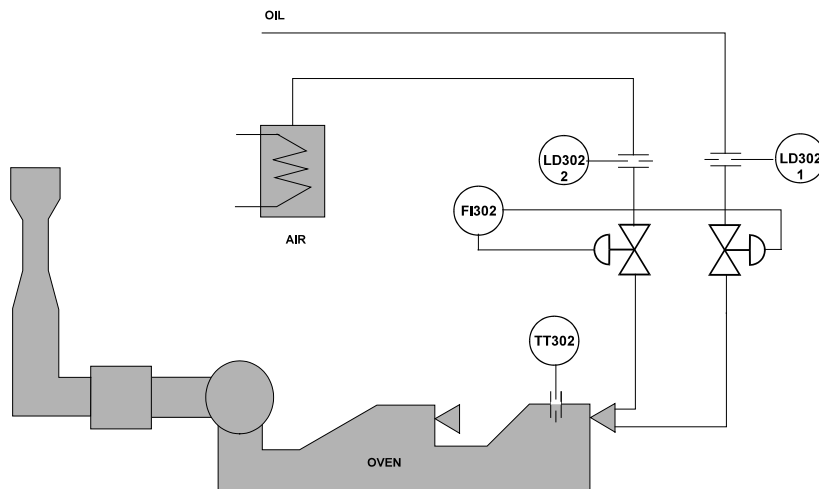
## Parameterization

| Device  | Block | Block Tag | Parameter   | Element | Value                      |
|---------|-------|-----------|-------------|---------|----------------------------|
| LD302-2 | AI    | PT-100B   | MODE_BLK    | Target  | AUTO                       |
|         |       |           | XD_SCALE    | UNIT    | Pa                         |
|         |       |           | OUT_SCALE   |         | 0-156 Cutf/min             |
|         |       |           | L_TYPE      |         | SQR ROOT                   |
|         | ARTH  | PY-100_1  | MODE_BLK    | Target  | AUTO                       |
|         |       |           | OUT_RANGE   |         | Kg/m <sup>3</sup>          |
|         |       |           | ARITH_TYPE  |         | 7 (traditional summer)     |
|         |       |           | GAIN_IN_1   |         | 1                          |
|         |       |           | GAIN        |         | 1/ (H2*g)                  |
|         |       |           | RANGE_LO    | Target  | 20                         |
|         |       |           | RANGE_HI    |         | -10                        |
| LD302-1 | AI    | PT-100A   | MODE_BLK    | Target  | AUTO                       |
|         |       |           | XD_SCALE    | UNIT    | Pa                         |
|         | ARTH  | PY-100_2  | MODE_BLK    | Target  | AUTO                       |
|         |       |           | PV_RANGE    |         | m <sup>3</sup>             |
|         |       |           | OUT_RANGE   |         | Ton                        |
|         |       |           | ARITH_TYPE  |         | 5 (traditional mult. div.) |
|         |       |           | GAIN_IN_2   |         | 1                          |
|         |       |           | GAIN        |         | 1                          |
|         |       |           | COMP_HI_LIM |         | + INF                      |
|         |       |           | COMP_LO_LIM |         | - INF                      |

| Device  | Block | Block Tag | Parameter  | Element | Value                 |
|---------|-------|-----------|------------|---------|-----------------------|
| LD302-3 | AI    | PT-100C   | MODE_BLK   | Target  | AUTO                  |
|         |       |           | XD_SCALE   | UNIT    | Pa                    |
|         | ARTH  | PY-100_3  | MODE_BLK   | Target  | AUTO                  |
|         |       |           | PV_RANGE   |         | mH2O                  |
|         |       |           | OUT_RANGE  |         | m                     |
|         |       |           | ARITH_TYPE |         | 9 (HTG comp. level)   |
|         |       |           | GAIN       |         | H2                    |
|         |       |           | BIAS       |         | H1                    |
|         |       |           | RANGE_LO   |         | 20                    |
|         |       |           | RANGE_HI   |         | -10                   |
|         | CHAR  | PY-100_4  | MODE_BLK   | Target  | AUTO                  |
|         |       |           | X_RANGE    |         | m                     |
|         |       |           | Y_RANGE    |         | M <sup>3</sup>        |
|         |       |           | CURVE_X    |         | 0 ,20,40,50,70,80,100 |
|         |       |           | CURVE_Y    |         | 0,20,40,50,70,80,100  |
| TT302   | AI    | TT-100    | MODE_BLK   | Target  | AUTO                  |



## Combustion Control with Double Cross Limits



FBMANUAL\_18

This type of control tries to keep the air-fuel ratio strictly within the limits. A sudden change on the load would require a corresponding air and fuel variation.

The master controller supplies Setpoint values to air and fuel flow controllers while it is stabilized.

During the transitions, the air flow determines the maximum upper and lower limits that the fuel flow cannot exceed. The same occurs for the air flow, whose limits are fixed by those of the fuel flow.

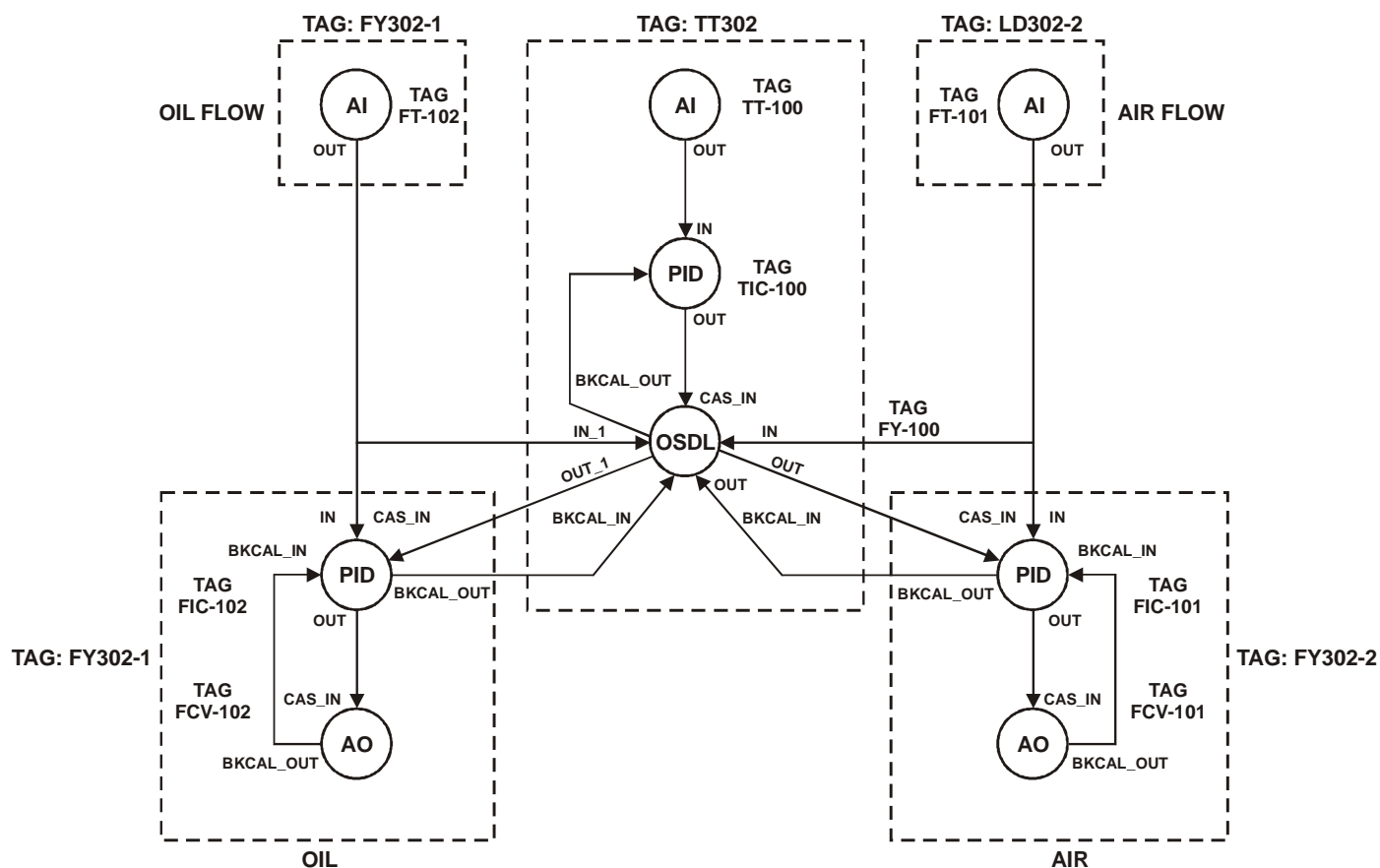
In this way, even when there is a large shift in the master signal the air/fuel ratio is maintained very close to the desired value.

The “double cross limits” prevents that the fastest variable unbalance the desired ratio. This strategy is implemented using the OSDL Block, that generates the setpoint for the air and fuel controllers based on the output of the master controller, air flow ( $Q_a \rightarrow IN$  parameter) and fuel flow ( $Q_c \rightarrow IN\_1$  parameter).

This configuration allows the air flow setpoint to vary just between  $(Q_c - LO\_BIAS)$  and  $(Q_c + HI\_BIAS)$  and the fuel flow setpoint to vary just between  $(Q_a - LO\_BIAS\_1)$  and  $(Q_a + HI\_BIAS\_1)$ .

When the double crossed limit is interfered with, then an unexpected change in the consumption upsets the desired ratio and in the same way when there is a transient in the master signal of the air/fuel flow it is able to be maintained very close to the desired ratio.

## Corresponding Configuration

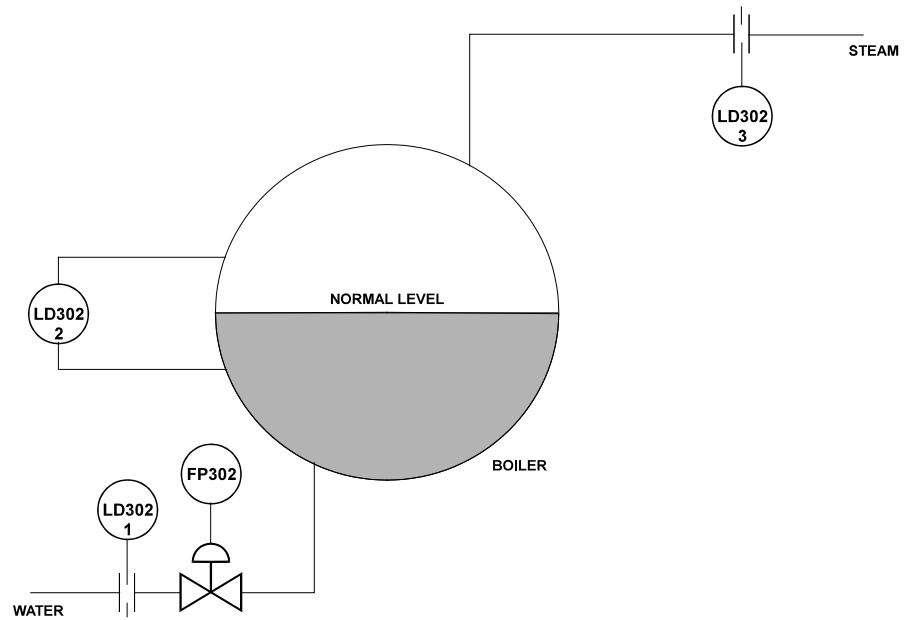


## Parameterization

| Device  | Block | Block Tag | Parameter    | Element       | Value           |
|---------|-------|-----------|--------------|---------------|-----------------|
| LD302-1 | AI    | FT-102    | MODE_BLK     | Target        | AUTO            |
|         |       |           | XD_SCALE     |               | 0-300 mm H2O    |
|         |       |           | OUT_SCALE    |               | 0-100 %         |
| LD302-2 | AI    | FT-101    | MODE_BLK     | Target        | AUTO            |
|         |       |           | XD_SCALE     |               | 0-200 inH2O     |
|         |       |           | OUT_SCALE    |               | 0-100 %         |
| TT302   | AI    | TT-100    | MODE_BLK     | Target        | AUTO            |
|         |       |           | XD_SCALE     |               | 0-600 °C        |
|         |       |           | L_TYPE       |               | direct          |
|         | PID   | TIC-100   | MODE_BLK     | Target        | AUTO            |
|         |       |           | PV_SCALE     |               | 0-600 °C        |
|         |       |           | OUT_SCALE    |               | 0-100 %         |
|         |       |           | CONTROL_OPTS | Direct-acting | Reverse         |
|         | OSDL  | FY-100    | MODE_BLK     | Target        | CAS             |
|         |       |           | OUT_TYPE     |               | Dynamic limiter |
|         |       |           | HI_GAIN      |               | 1               |
|         |       |           | HI_BIAS      |               | 5%              |
|         |       |           | LO_GAIN      |               | 1               |
|         |       |           | LO_BIAS      |               | 2%              |
|         |       |           | HI_GAIN_1    |               | 1               |
|         |       |           | HI_BIAS_1    |               | 2%              |

| Device  | Block | Block Tag | Parameter    | Element       | Value   |
|---------|-------|-----------|--------------|---------------|---------|
|         |       |           | LO_GAIN_1    |               | 1       |
|         |       |           | LO_BIAS_1    |               | 5%      |
|         |       |           | GAIN         |               | 1       |
|         |       |           | GAIN_1       |               | 1       |
| FY302-1 | PID   | FIC_102   | MODE_BLK     | Target        | CAS     |
|         |       |           | PV_SCALE     |               | 0-100 % |
|         |       |           | OUT_SCALE    |               | 0-100 % |
|         |       |           | CONTROL_OPTS | Direct-acting | Reverse |
|         | AO    | FCV-102   | MODE_BLK     | Target        | CAS     |
|         |       |           | PV_SCALE     |               | 0-100 % |
|         |       |           | XD_SCALE     |               | 0-100 % |
| FY302-2 | PID   | FIC-101   | MODE_BLK     | Target        | CAS     |
|         |       |           | PV_SCALE     |               | 0-100 % |
|         |       |           | OUT_SCALE    |               | 0-100 % |
|         |       |           | CONTROL_OPTS | Direct-acting | Reverse |
|         | AO    | FCV-101   | MODE_BLK     | Target        | CAS     |
|         |       |           | PV_SCALE     |               | 0-100 % |
|         |       |           | XD_SCALE     |               | 0-100 % |

### 3 Element Boiler Level/Feedwater Control

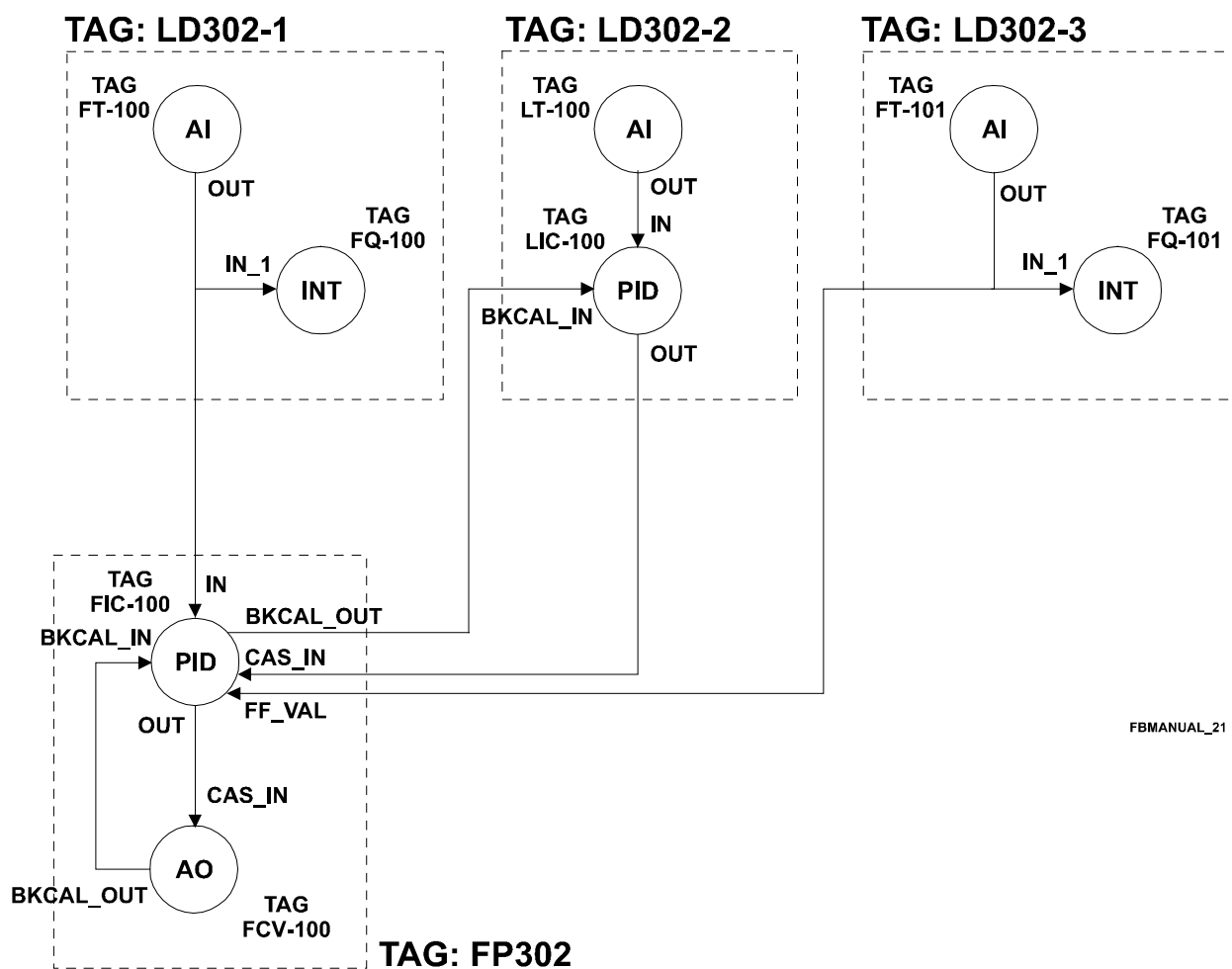


FBMANUAL\_20

This control loop uses feed forward control combined with cascade control. In this case the feed forward steam flow correction is done for steam flow and the feedback is done through the transmitter and the level controller LIC-100, whilst the feed water flow is maintained by the secondary cascade control loop of water.

In this loop the drum level controller LIC-100 provides the setpoint for the feedwater controller FIC-100 in cascade. Any disturbance in feed water flow is corrected by a feed forward arrangement in FIC-100. By connecting FT-101 to the FF\_VAL input of FIC-100, any change in flow adjusts the FIC-100 output directly. The FF\_SCALE is set -100 to +100 % to provide a fixed 50 % bias, giving a 50 % setpoint when load and manipulated flow are perfectly matched.

## Corresponding Configuration



## Parameterization

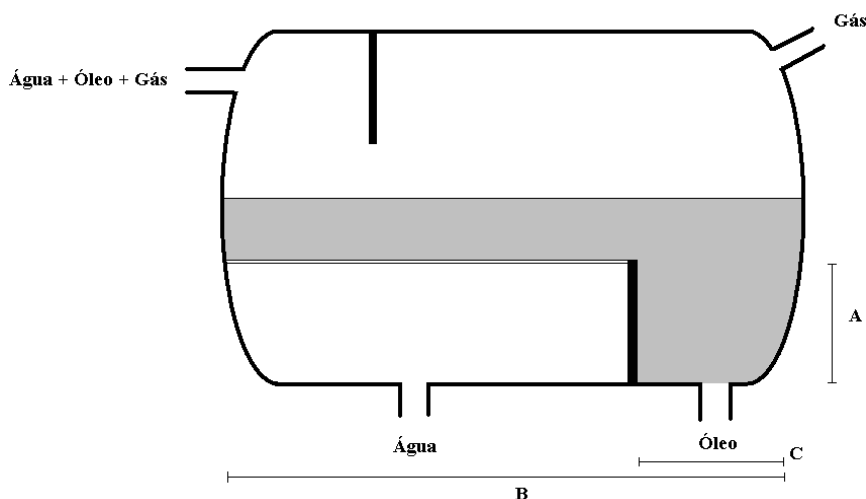
| Device  | Block | Block Tag | Parameter    | Element       | Value                    |
|---------|-------|-----------|--------------|---------------|--------------------------|
| LD302-2 | AI    | LT-100    | MODE_BLK     | Target        | AUTO                     |
|         |       |           | XD_SCALE     |               | 642 -140 mmH2O           |
|         |       |           | OUT_SCALE    |               | 0-100 %                  |
|         | PID   | LIC-100   | MODE_BLK     | Target        | AUTO                     |
|         |       |           | PV_SCALE     |               | 0-100 %                  |
|         |       |           | OUT_SCALE    |               | 0-150 Ton/hr             |
|         |       |           | CONTROL_OPTS | Direct-acting | Reverse                  |
| LD302-3 | AI    | FT-101    | MODE_BLK     | Target        | AUTO                     |
|         |       |           | XD_SCALE     |               | 0-9500 mm H2O            |
|         |       |           | OUT_SCALE    |               | 0-150 Ton/hr             |
|         |       |           | L_TYPE       |               | Indirect, Square Root    |
|         | INTG  | FQ-101    | MODE_BLK     | Target        | AUTO                     |
|         |       |           | TIME_UNIT1   |               | Hours                    |
|         |       |           | OUT_RANGE    |               | Ton                      |
| LD302-1 | AI    | FT-100    | MODE_BLK     | Target        | AUTO                     |
|         |       |           | XD_SCALE     |               | 0-3500 mmH2O             |
|         |       |           | OUT_SCALE    |               | 0-150 m <sup>3</sup> /hr |

| Device | Block | Block Tag | Parameter    | Element       | Value                    |
|--------|-------|-----------|--------------|---------------|--------------------------|
|        | INTG  | FQ-100    | L_TYPE       |               | Indirect, Square Root    |
|        |       |           | MODE_BLK     | Target        | AUTO                     |
|        |       |           | TIME_UNIT1   |               | Hours                    |
|        |       |           | OUT_RANGE    |               | m <sup>3</sup>           |
| FP302  | PID   | FIC-100   | MODE_BLK     | Target        | CAS                      |
|        |       |           | PV_SCALE     |               | 0-150 m <sup>3</sup> /hr |
|        |       |           | OUT_SCALE    |               | 0-100 %                  |
|        |       |           | CONTROL_OPTS | Direct-acting | Reverse                  |
|        |       |           | FF_SCALE     |               | -100 to + 100 %          |
|        |       |           | FF_GAIN      |               | 1                        |
|        | AO    | FCV_100   | MODE_BLK     | Target        | CAS                      |
|        |       |           | PV_SCALE     |               | 0-100 %                  |
|        |       |           | XD_SCALE     |               | 3-15 psi                 |

## Application of the Flexible Mathematical Function Block

The FMTH will be used to solve the following problem:

It is necessary to obtain the volume of oil in a cylindrical tank represented by dark region in the figure below

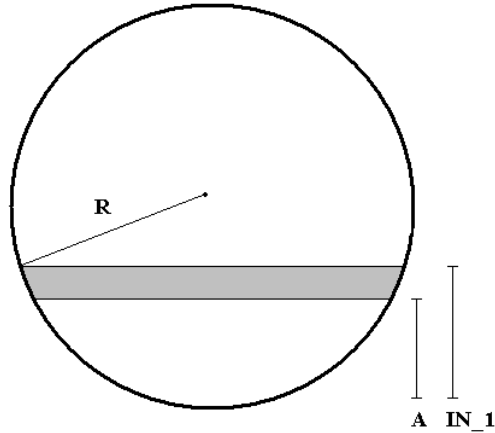


Initially the considerations are:

- The A, B and C lengths are known;
- The Cylindrical Ray – R is known;
- There is a sensor to measure the length that is the oil, and in this configuration this value is entering by the IN\_1 input;
- The interface between water and oil has inconsiderable thickness ;
- The interface is always in the height A.

There are three situations:

- **Situation 1:** The oil height is low than the ray of transversal section and higher than the A height:



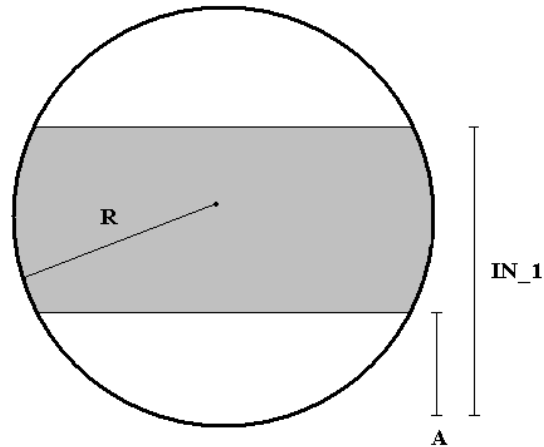
The area of the dark region of the figure above is given by:

$$Ar = R^2 * \left( \arccos\left(\frac{R - In\_1}{R}\right) - \arccos\left(\frac{R - A}{R}\right) \right) + \left( (R - A) * \sqrt{R^2 - (R - A)^2} - (R - In\_1) * \sqrt{R^2 - (R - In\_1)^2} \right)$$

So, the oil volume will be:

$$V = Ar * B + \left( R^2 * \arccos\left(\frac{R - A}{R}\right) - (R - A) * \sqrt{R^2 - (R - A)^2} \right) * C$$

- **Situation 2:** The oil height is higher than the ray of the transversal section



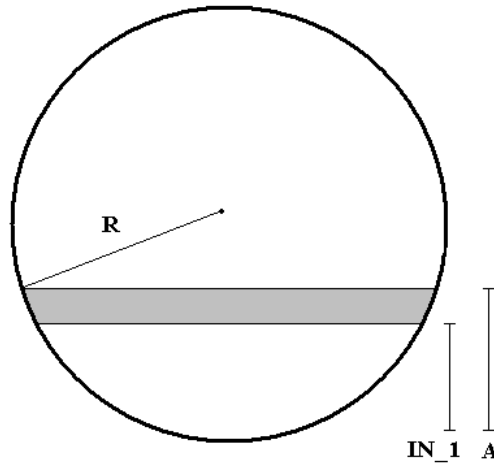
The area of the dark region is:

$$Ar = R^2 * \left( 2\pi - \arccos\left(\frac{In\_1 - R}{R}\right) - \arccos\left(\frac{R - A}{R}\right) \right) + \left( (R - A) * \sqrt{R^2 - (R - A)^2} + (In\_1 - R) * \sqrt{R^2 - (In\_1 - R)^2} \right)$$

So, the oil volume will be:

$$V = Ar * B + \left( R^2 * \arccos\left(\frac{R - A}{R}\right) - (R - A) * \sqrt{R^2 - (R - A)^2} \right) * C$$

- **Situation 3:** The oil height is low than A height



Similarly to the Situation 1, it has

$$Ar = R^2 * \left( \arccos\left(\frac{R-A}{R}\right) - \arccos\left(\frac{R-In_1}{R}\right) \right) + \left( (R-In_1) * \sqrt{R^2 - (R-In_1)^2} - (R-A) * \sqrt{R^2 - (R-A)^2} \right)$$

So, the oil volume will be:

$$V = \left( R^2 \arccos\left(\frac{R-A}{R}\right) - (R-A) * \sqrt{R^2 - (R-A)^2} \right) * C - Ar * C$$

Through the expressions aboe, the block can be configured of the following way:

$$AA2 = AA11 * a \cos((AA7) / AA5) - (AA7) * (AA11 - (AA7) * (AA7))^{0.5}$$

$$AA7 = if(AI1 \leq AA5; AA5 - AI1; AI1 - AA5)$$

$$AA3 = if(AI1 \leq AA6; AA1 - AA2; if(AI1 \leq AA5; AA2 - AA1; AA8 - AA1 - AA2))$$

$$AO1 = if(AI1 \leq AA6; (AA1 - AA3) * AA10; AA3 * AA9 + AA1 * AA10)$$

Where:

- AA1 = AA11\*acos((AA5-AA6)/AA5)-(AA5-AA6)\*(AA11-(AA5-AA6)\*(AA5-AA6))^0.5
- AA5 = R
- AA6 = A
- AA8 =  $\pi * R^2$
- AA9 = B
- AA10 = C
- AA11 =  $R^2$

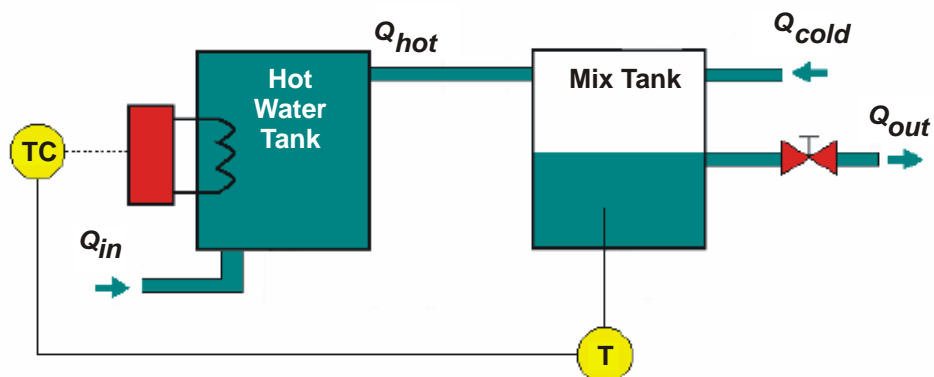
## Smith Predictor and Transfer Function Block Application

In the next process, there is a mix between fluids that are water in different temperatures. The left fluid is sent by the first tank (Hot water Tank) and the other fluid is in the second tank (Mix Tank). The fluid that is sent to the first tank is heat by electrical resistances and it is transferred to the next tank (mix tank). The mix tank receives a cold water flow. The cold water flow ( $Q_{cold}$ ) and the hot water flow ( $Q_{hot}$ ) are kept constant.

The purpose is to control the mix tank temperature by setting the resistances power.

A great delay between the actuation and the system measurement was identified through experimental tests. This delay occurs due to the distance between the first and the second tanks.

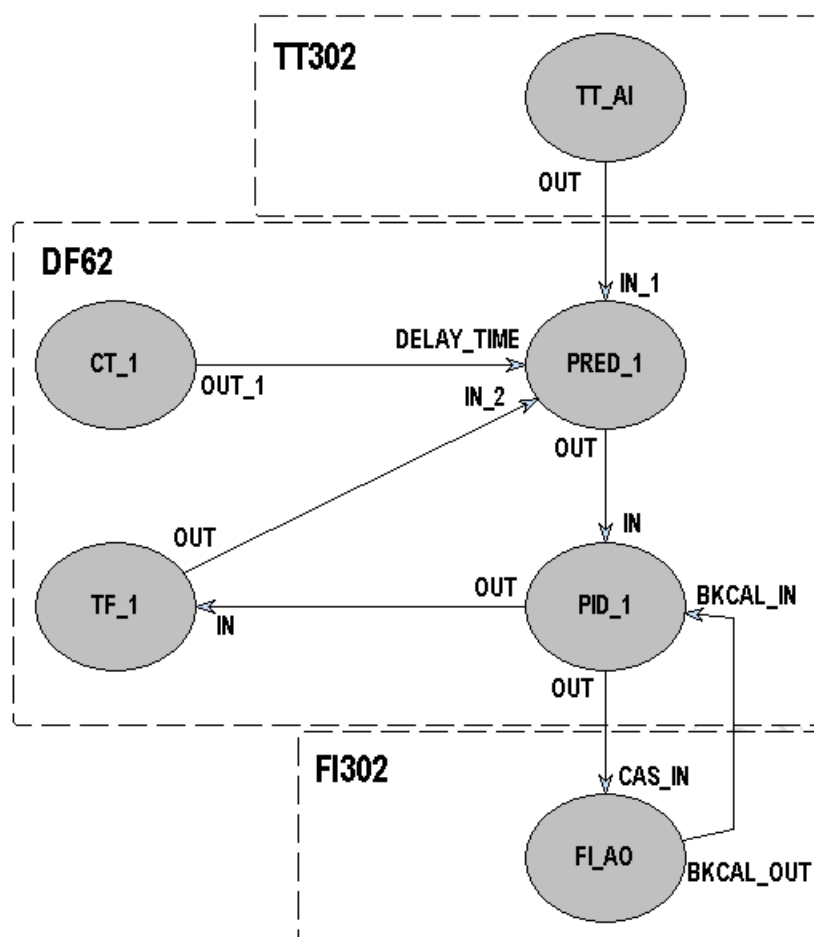




The transfer function for the system is:

$$G(s) = \frac{0.89}{s + 0.89} * e^{-10s}$$

### Corresponding Configuration



## Parameterization

| Device | Block | Block Tag | Parameter     | Element | Value           |
|--------|-------|-----------|---------------|---------|-----------------|
| TT302  | AI    | TT-AI     | MODE_BLK      | Target  | AUTO            |
|        |       |           | OUT_SCALE     |         | 0-100%          |
|        |       |           | L_TYPE        |         | Indirect        |
| FI302  | AO    | FI-AO     | MODE_BLK      | Target  | Cas             |
| DF62   | PID   | PID_1     | MODE_BLK      | Target  | AUTO            |
|        |       |           | SP            |         | 30%             |
|        |       |           | GAIN          |         | GAIN            |
|        |       |           | RESET         |         | RESET           |
|        | CT    | CT_1      | MODE_BLK      | Target  | AUTO            |
|        |       |           | CT_VAL_1      |         | 10              |
|        | TF    | TF_1      | MODE_BLK      | Target  | AUTO            |
|        |       |           | TF_A          |         | 0               |
|        |       |           | TF_B          |         | 0               |
|        |       |           | TF_C          |         | 0.89            |
|        |       |           | TF_D          |         | 0               |
|        |       |           | TF_E          |         | 1               |
|        |       |           | TF_F          |         | 0.89            |
|        | PRED  | PRED_1    | MODE_BLK      | Target  | AUTO            |
|        |       |           | PRED_SELECTOR |         | Smith Predictor |