

**MANUAL**  
INSTRUÇÕES | OPERAÇÃO | MANUTENÇÃO

# MANUAL DE INSTRUÇÕES DOS BLOCOS FUNCIONAIS



MAR/22 - VERSÃO 2

**smar**  
Technology Company

# MANUAL DE INSTRUÇÕES DOS BLOCOS FUNCIONAIS



Consulte nossos  
representantes



Rua Dr. Antônio Furlan Junior, 1028 - Sertãozinho, SP - CEP: 14170-480  
orcamento@smar.com.br | +55 (16) 3946-3599 | www.smar.com.br

© Copyright 2022, Nova Smar S/A. Todos os direitos reservados. - 2022  
Especificações e informações estão sujeitas a modificações.  
Informações atualizadas dos endereços estão disponíveis em nosso site.

**smar**  
Technology Company

# ÍNDICE

<b>SEÇÃO 1 - INTRODUÇÃO À APLICAÇÃO DO BLOCO FUNCIONAL.....</b>	<b>1.1</b>
GERAL.....	1.1
BLOCO FUNCIONAL.....	1.1
BLOCO TRANSDUTOR.....	1.1
BLOCO FÍSICO.....	1.1
DEFINIÇÕES DO BLOCO FUNCIONAL.....	1.1
TROCA DE DADOS CÍCLICA.....	1.1
DADOS DE CONFIGURAÇÃO (CFG_DATA).....	1.2
EXEMPLO DE CONFIGURAÇÃO.....	1.4
ACESSO À INFORMAÇÃO.....	1.5
ESTRUTURA DA APLICAÇÃO DO BLOCO FUNCIONAL.....	1.5
OBJETO DO BLOCO.....	1.5
PARÂMETROS DOS BLOCOS.....	1.5
PARÂMETRO DOS IDENTIFICADORES.....	1.6
USO DO PARÂMETRO.....	1.6
CONTAINED.....	1.6
SAÍDA.....	1.6
ENTRADA.....	1.6
RELAÇÕES DE PARÂMETRO.....	1.7
STATUS DO PARÂMETRO.....	1.7
COMPOSIÇÃO DO STATUS.....	1.7
EXEMPLO: CONVERSÃO DAS ENUMERAÇÕES PARA STATUS.....	1.11
EXEMPLO: CONVERSÃO DE NÚMERO PARA ENUMERAÇÕES.....	1.12
TRATAMENTO DO PARÂMETRO CHANNEL.....	1.12
CANAL DOS BLOCOS FUNCIONAIS DE ENTRADA.....	1.12
CANAL DOS BLOCOS FUNCIONAIS DE SAÍDA.....	1.13
EXEMPLOS DE USO DO CANAL.....	1.13
CÁLCULO DE SAÍDA.....	1.13
CONTROLE DE CASCATA REMOTA.....	1.14
PARÂMETROS DO MODO.....	1.15
A) MODO DE OPERAÇÃO.....	1.15
B) ELEMENTOS DO MODO DE UM BLOCO FUNCIONAL.....	1.16
C) PRIORIDADE DE MODO.....	1.16
D) MODO DE CÁLCULO.....	1.16
E) INFORMAÇÃO ESPECÍFICA PARA DESENVOLVEDORES DE DRIVERS.....	1.17
PARÂMETROS DE ESCALA.....	1.17
EXEMPLO PARA USO DE PARÂMETROS DE ESCALA.....	1.18
TRATAMENTO DO ESTADO DE FALHA (FAIL SAFE).....	1.18
CONDIÇÕES QUE ATIVAM O <i>FAIL SAFE</i> .....	1.19
AÇÕES DE <i>FAIL SAFE</i> (FALHA DE SEGURA).....	1.19
DIAGNÓSTICO.....	1.19
DIAGNÓSTICO DE CARACTERÍSTICAS DO EQUIPAMENTO.....	1.20
DIAGNÓSTICO DO ATUADOR.....	1.20
ALARMES E EVENTOS – PROCESSAMENTO DE ALERTA.....	1.20
A) PARÂMETRO DE ALARME (PARÂMETRO X_ALM).....	1.20
B) LIMITE DE ALARME (PARÂMETRO X_LIM).....	1.20
C) HISTERESE DE ALARME (PARÂMETRO ALARM_HYS).....	1.20
D) CHAVE DE ALERTA (PARÂMETRO ALERT_KEY).....	1.20
E) RESUMO DE ALARME (PARÂMETRO ALM_SUM).....	1.21
F) ATUALIZAÇÃO DO EVENTO ( <i>UPDATE EVENT</i> ).....	1.21
DEFINIÇÃO E TIPOS DE ESTRUTURA DE DADOS.....	1.21
OBJETO DO BLOCO – DS-32.....	1.21
VALOR & STATUS - ESTRUTURA DO PONTO FLUTUANTE – DS-33.....	1.22
VALOR & STATUS - ESTRUTURA DISCRETA – DS-34.....	1.22
ESTRUTURA DE ESCALA – DS-36.....	1.22
ESTRUTURA DE MODO – DS-37.....	1.22
ESTRUTURA DE ALARME DE PONTO FLUTUANTE–DS-39.....	1.23
ESTRUTURA DE ALARME DISCRETA–DS-40.....	1.23
ESTRUTURA DE ATUALIZAÇÃO DO ALARME–DS-41.....	1.23

ESTRUTURA ÍNDICE DE ALARME–DS-42 .....	1.23
SIMULATE – ESTRUTURA DO PONTO FLUTUANTE–DS-50 .....	1.23
SIMULATE - ESTRUTURA DISCRETA–DS-51 .....	1.24
ESTRUTURA DE GRUPO–DS-67 .....	1.24
<b>SEÇÃO 2 - BIBLIOTECA DE BLOCOS .....</b>	<b>2.1</b>
BLOCO FÍSICO – PHY .....	2.1
DESCRIÇÃO .....	2.1
PARÂMETRO FACTORY_RESET .....	2.1
MEMÓRIA NÃO VOLÁTIL .....	2.1
BLOQUEIO DE GRAVAÇÃO VIA SOFTWARE .....	2.1
DIAGNÓSTICOS .....	2.2
SELETOR DO NÚMERO IDENTIFICADOR .....	2.2
MODOS SUPORTADOS .....	2.2
PARÂMETROS .....	2.2
ENTRADA ANALÓGICA – AI .....	2.3
GERAL .....	2.3
ESQUEMÁTICO .....	2.4
DESCRIÇÃO .....	2.4
SIMULAÇÃO .....	2.4
MODOS SUPORTADOS .....	2.4
STATUS .....	2.4
CÍCLICO – CFG_DATA .....	2.4
PARÂMETROS .....	2.5
SAÍDA ANALÓGICA – AO .....	2.6
GERAL .....	2.6
ESQUEMÁTICO .....	2.6
DESCRIÇÃO .....	2.6
AUMENTAR PARA FECHAR ( <i>INCREASE TO CLOSE</i> ) .....	2.6
SIMULAÇÃO .....	2.6
PARÂMETROS DE <i>READBACK</i> .....	2.7
MODOS SUPORTADOS .....	2.7
CÍCLICO – CFG_DATA .....	2.7
PARÂMETROS .....	2.8
TOTALIZADOR - TOT .....	2.9
GERAL .....	2.9
ESQUEMÁTICO .....	2.9
DESCRIÇÃO .....	2.9
TOTALIZAÇÃO DA VAZÃO .....	2.10
<i>RESET</i> E <i>PRESET</i> .....	2.10
INICIANDO O BLOCO .....	2.10
MODOS SUPORTADOS .....	2.10
CÍCLICO – CFG_DATA .....	2.11
PARÂMETROS .....	2.11
SAÍDA DISCRETA – DO .....	2.12
GERAL .....	2.12
DESCRIÇÃO .....	2.13
SIMULAÇÃO .....	2.13
MODOS SUPORTADOS .....	2.13
STATUS HANDLING .....	2.13
CÍCLICO CFG_DATA .....	2.14
PARÂMETROS .....	2.14
ATRIBUTOS DOS PARÂMETROS DO BLOCO DE FUNÇÃO DE SAÍDA DISCRETA .....	2.15
ENTRADA DISCRETA – DI .....	2.15
GERAL .....	2.15
DESCRIÇÃO .....	2.16
SIMULAÇÃO .....	2.16
MODOS SUPORTADOS .....	2.16
STATUS HANDLING .....	2.17
CÍCLIC CFG_DATA .....	2.17
PARÂMETROS .....	2.17
ATRIBUTOS DOS PARÂMETROS DO BLOCO DE FUNÇÃO DE SAÍDA DISCRETA .....	2.18
DESCRIÇÕES DE BITSTRINGS .....	2.19
DIAGNÓSTICOS (BLOCO <i>PHYSICAL</i> ) .....	2.19
<i>CHECK_BACK</i> (BLOCO DE SAÍDA ANALÓGICA) .....	2.20
CONFIGURAÇÕES E DISPONIBILIDADES DE FB .....	2.20

## INTRODUÇÃO À APLICAÇÃO DO BLOCO FUNCIONAL

### **Geral**

As aplicações que utilizam os blocos funcionais são definidas como aplicações de planta ou fábrica, que executam um ou mais monitoramentos automáticos e funções de controle.

### **Bloco Funcional**

Os blocos funcionais representam as funções de automação básicas executadas por aplicações dos blocos funcionais. Cada bloco funcional processa parâmetros de entrada de acordo com um algoritmo específico e um conjunto interno de parâmetros de controle. Eles fornecem os parâmetros de saída, disponíveis internamente, para serem usados na mesma aplicação ou para outras aplicações dos blocos funcionais.

### **Bloco Transdutor**

Os blocos transdutores isolam os blocos funcionais dos equipamentos específicos de E/S, tais como sensores, atuadores e interruptores. Os blocos transdutores controlam o acesso às E/S dos equipamentos por uma interface independente do equipamento definida pelo uso de blocos funcionais. Os blocos transdutores também executam funções, tais como calibração e linearização, nos dados de E/S para convertê-los para uma representação independente. Sua interface para blocos funcionais é definida como um ou mais canais de E/S independentes das implementações.

### **Bloco Físico**

Os blocos físicos (*resource*) são usados para definir as características de hardware específicas das aplicações do bloco funcional. Similar aos blocos transdutores, eles isolam os blocos funcionais do hardware físico por conterem um conjunto de implementação independente dos parâmetros de hardware.

### **Definições do Bloco Funcional**

Os blocos funcionais são definidos por suas entradas, saídas, parâmetros de controle e pelo algoritmo que opera sobre estes parâmetros. Os blocos funcionais são identificados usando um nome (Tag) e um índice numérico.

Os Tags fornecem uma referência simbólica aos blocos funcionais. Eles não podem ser ambíguos dentro de um sistema fieldbus. Os índices numéricos pré-determinados são números atribuídos para otimizar o acesso aos blocos funcionais. Ao contrário do que faz os Tags do bloco funcional, que são globais, os índices numéricos pré-determinados têm somente um significado dentro da aplicação que contém o bloco funcional.

Os parâmetros do bloco funcional definem as entradas, saídas e os dados usados para controlar a operação dos blocos funcionais. Eles são visíveis e acessíveis na rede. Os parâmetros adicionais, chamados "*contained within*" são parâmetros usados para definir os dados privados de um bloco funcional. Embora visíveis na rede não podem participar de trocas de dados cíclicas.

### **Troca de Dados Cíclica**

A troca de dados cíclica indica que um parâmetro da entrada de um bloco funcional obtém seu valor, ciclicamente, dos parâmetros de saída específicos de um outro bloco funcional no outro equipamento. Não há nenhuma ligação interna entre os blocos funcionais no dispositivo.

Em geral, um bloco funcional do equipamento transmissor, ou atuador, troca dados ciclicamente com o controlador (por exemplo, um PLC mestre). Tipicamente, o transmissor obtém os dados do sensor e o equipamento controlador requisita estes dados, faz cálculos e envia o resultado para um atuador que tomará algumas ações no processo.

Para configurar quais informações serão trocadas, o mestre obtém a informação dos equipamentos consultando o arquivo GSD. Para cada equipamento há um arquivo GSD. Este arquivo possui um número identificador que é único e que identifica o equipamento para o mestre da rede. O arquivo GSD contém toda a informação do equipamento, por exemplo, tipo de transmissor, quantos e que tipo de blocos o equipamento possui e a possível configuração cíclica que o equipamento suporta. Um exemplo de arquivo GSD é mostrado na Figura 1.1.

```
;
;   GSD file for LD303 - Pressure Transmitter
;   smar0895.GSD
;
#Profibus_DP
GSD_Revision           = 2
Vendor_Name            = "SMAR"
Model_Name             = "LD303"
Revision               = "1.0"
Ident_Number           = 0x0895 ; 0x9740
:
:
:
;Modules for Analog Input
Module = "Analog Input (short) " 0x94
EndModule
Module = "Analog Input (long) " 0x42, 0x84, 0x08, 0x05
EndModule

;Module for Totalizer
Module = "Total " 0x41, 0x84, 0x85
      EndModule
Module = "Total_Settot " 0xC1, 0x80, 0x84, 0x85
EndModule
Module = "Total_Settot_Modetot " 0xC1, 0x81, 0x84, 0x85
EndModule

;Empty module
Module = "EMPTY_MODULE" 0x00
EndModule
```

Figura 1.1 - Exemplo do Arquivo GSD

## Dados de Configuração (CFG\_DATA)

Na troca de dados cíclica há diferentes parâmetros para os blocos funcionais (para maiores detalhes, o parâmetro CYCLIC\_CFG\_DATA deve ser verificado para cada bloco da função). As diferenças vêm das necessidades diferentes do usuário em relação à aplicação da informação (com ou sem a realimentação da posição real da saída) e o modo de integração na tarefa do controle (com ou sem cascata remota). Durante a configuração, o operador escolhe a combinação de parâmetros e as ferramentas concatenar uma *string* de configuração interna (na figura 1, a *string* de configuração ou o *identifier byte* são aqueles números definidos em cada módulo).

O mestre busca pela configuração, no arquivo GSD, suportada por um bloco específico. Por exemplo, de acordo com o arquivo GSD da figura 1, o usuário pode configurar a saída do bloco de Entrada Analógica "Analog Input (short)" ou a saída do Totalizador (total) ou de ambos.

No arquivo GSD há uma seção onde são definidas todas as configurações possíveis suportadas pelo equipamento. Cada configuração permitida é iniciada com a palavra "Module" e concluída com a palavra "EndModule". Na descrição de cada "Module" há um nome da *string* e alguns números da configuração dos dados (CFG\_DATA). Estes números são internos ao mestre e descrevem as combinações dos parâmetros (quantos parâmetros, tipos de dados, comprimento, etc.) do bloco funcional. Por exemplo:

Module = Total_Settot "	0xC1, 0x80, 0x84, 0x85,
-----	-----
String descrevendo o parâmetro da configuração cíclica	Dados de Configuração (CFG_DATA)

Há um módulo especial "EMPTY\_MODULE" indicando que aquele bloco funcional específico não participará da troca de dados cíclica.

**Para cada bloco funcional, que suporta as trocas cíclicas de dados, é necessário configurar uma (e somente uma) combinação cíclica de configuração para este bloco OU o "EMPTY\_MODULE" (indicando que o usuário não deseja usar o bloco).**

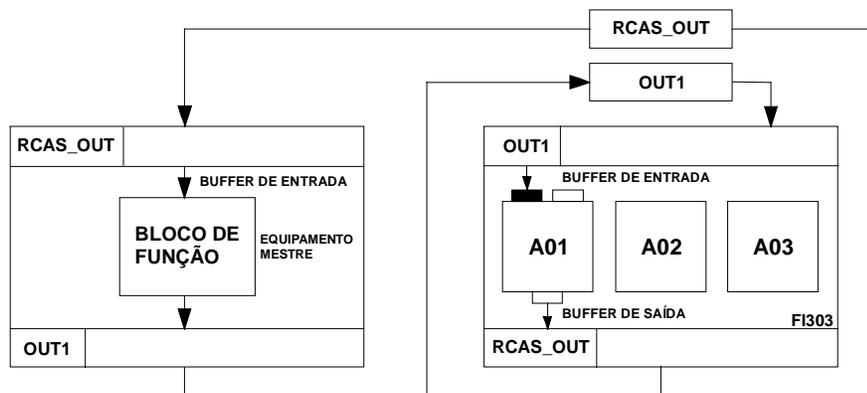
De acordo com o Profibus-PA profile 3.0, há dois tipos de configuração para uma combinação dos parâmetros. O *byte Identifier* (ou *short identifier*) e o *Extended Identifier Format* (ou *long identifier*). Algumas combinações têm somente um tipo de configuração e outras têm ambos. Os equipamentos Smar suportam ambos os tipos de configuração. Assim, no exemplo acima o usuário pode escolher "Analog Input (short)" ou "Analog Input (long)" e terá o mesmo resultado para a configuração. Na mesma configuração o usuário pode, ainda, combinar ambos tipos de configuração: longo (*long*) e curto (*short*).

Se houver mais de um bloco funcional com mais de um parâmetro cíclico, os elementos de dados ficarão concatenados na mesma entrada ou no frame dos dados de saída (isto depende da configuração). A ordem dos parâmetros para um bloco funcional no frame dos dados de entrada e de saída é do mais baixo para o mais alto do índice relativo na tabela dos blocos funcionais (veja o exemplo na figura 3). Se houver mais de um bloco funcional do mesmo tipo dentro de um equipamento, por exemplo: 3 blocos funcionais AI, a ordem dos parâmetros cíclicos na entrada e na saída do frame dos dados da saída será a mesma da ordem do bloco funcional, no diretório do exemplo do equipamento (veja o exemplo na figura 2).

**Todos os blocos funcionais do equipamento precisam ser configurados no "Config data" na mesma ordem do diretório no equipamento (veja na tabela "FB set and FB type availability" a ordem dos blocos para cada dispositivo).**

Por exemplo, no equipamento LD303 que tem um bloco funcional AI e um TOT, a ordem para o "Config data" deve ser AI e TOT, respectivamente. No equipamento IF303 que tem 3 AI e 3 TOT, a ordem do "Config data" deve ser AI1, AI2, AI3, TOT1, TOT2, TOT3, respectivamente.

As figuras a seguir ilustram as trocas de dados entre o equipamento mestre e o equipamento escravo. Na Figura 1.2, no equipamento FI303 (3 blocos AO) só o primeiro AO é configurado (deve ser observado que a ordem da configuração necessita ser respeitada). Assim, na configuração, o usuário precisa indicar os blocos (AO2 e AO3) que não são usados. Para isto, os blocos não utilizados devem ser configurados para "Empty Module". Para o bloco AO1, é escolhido "RCAS\_IN + RCAS\_OUT" para a troca de dados, aonde os dados da saída do bloco funcional do mestre vão para a entrada RCAS\_IN do equipamento escravo. E a saída RCAS\_OUT do bloco funcional AO1 vai para a entrada do bloco funcional do Mestre.



**Figura 1.2 – Exemplo da troca cíclica de dados - Dados da Configuração "RCAS\_IN + RCAS\_OUT" + "Empty\_Module" + "Empty\_Module" para o equipamento FI303**

Na Figura 1.3, para o equipamento FY303 (1 bloco AO), é escolhida a troca de dados “SP + RB + RCAS\_IN + RCAS\_OUT + POS\_D + CB”, onde há 2 entradas e 4 saídas para serem transportadas do escravo para o mestre. Nesse caso, a ordem do buffer de entrada e de saída depende do índice relativo do parâmetro do bloco. Por exemplo, os dados de entrada para o escravo OUT1 e OUT2 (no frame) respeitam a ordem ascendente do bloco AO. O SP recebe OUT1 e o RCAS\_IN recebe OUT2 (veja a tabela de parâmetros do bloco para verificar o índice relativo do parâmetro). O mesmo ocorre com as 4 saídas do bloco AO. A ordem do buffer da saída é READBACK, RCAS\_OUT, POS\_D e CHECKBACK.

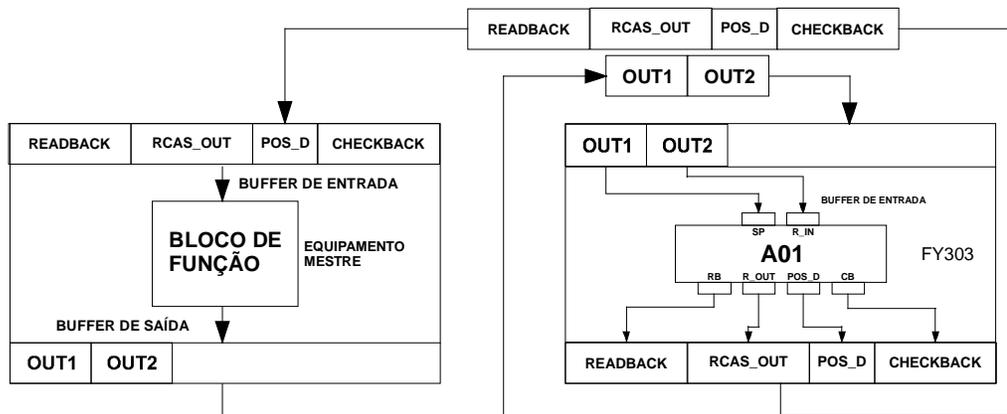


Figura 1.3 – Exemplo da troca cíclica de dados - Dados de Configuração “SP + RB + RCAS\_IN + RCAS\_OUT + POS\_D + CB” para o equipamento FY303

## Exemplo de Configuração

- 1) Considerando o transmissor LD303, que tem 2 blocos funcionais: 1 AI e 1 TOT. A seguinte configuração é válida:
  - Configurando a saída do bloco AI e a saída do bloco TOTAL: " Analog Input (short) " e "Total" ou "0x94, 0x41, 0x84, 0x85"
  - Configurando somente a saída do bloco AI: " Analog Input (long) " e "Empty module" ou "0x42, 0x84, 0x08, 0x05, 0x00"
  - Configurando somente a saída do bloco TOTAL: "Empty Module" e "Total\_Settot" ou "0x00, 0xC1, 0x80, 0x84, 0x85"
- 2) Considerando o transmissor FI303 que tem 3 blocos funcionais, 3 blocos AO. As seguintes configurações são válidas:
  - Configurando a entrada RCAS\_IN, RCAS\_OUT e somente a saída CHECKBACK do AO1 (primeiro bloco de Entrada Analógica):  
"RCAS\_IN + RCAS\_OUT + CB ", "EMPTY\_MODULE" e "EMPTY\_MODULE" ou "0x97,0xA4,0x00,0x00".
  - Não configura o AO1, configurando a entrada SP, Readback e as saídas PosD do AO2, e o SP do AO3, resulta:  
"EMPTY\_MODULE", " SP + RB + POS\_D " e "SP" ou "0x00,0x96,0xA4,0xA4"
  - Configurando a entrada RCAS\_IN e a saída RCAS\_OUT dos 3 AOs (AO1,AO2 e AO3):  
"RCAS\_IN+ RCAS\_OUT", "RCAS\_IN + RCAS\_OUT", "RCAS\_IN + RCAS\_OUT" ou "0xB4,0xB4,0xB4"

### Notas:

- No exemplo 1, a ordem nos blocos funcionais no LD303 é: AI e TOT, respectivamente. Assim, na configuração é necessário escolher uma configuração válida para os blocos AI e TOT. Se o usuário escolher primeiro configurar o bloco TOT, a conexão cíclica não será estabelecida. No exemplo 2 é necessário configurar os blocos AO1, AO2 e AO3, respectivamente, nesta ordem.

- Alguns configuradores usam a descrição do módulo no arquivo GSD, por exemplo, “*Analog Input (short)*.” Outros usam o número do identificador, por exemplo, “0x94.” Assim no exemplo são mostrados ambos os tipos de configuração.
- Na configuração, usando os números do identificador, todos os blocos funcionais já foram incluídos na *string*, por exemplo, “0x00,0x96,0xA4,0xA4” do exemplo 2, tem “0x00” (módulo vazio) para o primeiro bloco, “0x96,0xA4” (SP, READBACK, POS\_D) para o segundo bloco, e “0xA4” (SP) para o terceiro bloco.
- O uso de identificadores curtos ou longos têm o mesmo efeito.

## Acesso à Informação

A informação do bloco funcional pode ser agrupada para acesso, dependendo de como ela será usada. Por enquanto, há somente a definição de acesso para fins de visualização dinâmica dos dados de operação.

Para suportar o acesso da informação pela interface do operador, durante execução do bloco funcional, dois níveis de acesso a rede são definidos: um para o tráfego operacional e um para o tráfego de supervisão. O operador do tráfego na interface é transferido como tráfego de supervisão para prevenir a interferência no tempo crítico da operação dos blocos funcionais.

## Estrutura da Aplicação do Bloco Funcional

As aplicações dos blocos funcionais são modeladas como configurações de blocos funcionais coordenadas para executar uma configuração relacionada às operações.

O modelo do bloco funcional é um algoritmo em tempo real que transforma parâmetros de entrada em parâmetros de saída. Sua operação é controlada através dos parâmetros de configuração do controle.

Os equipamentos, transmissor e atuador têm blocos funcionais que são modelados para fornecer ou obter os valores de processo dos blocos funcionais do equipamento controlador.

A operação entre os blocos funcionais e os diferentes equipamentos é modelada através das trocas de dados entre o parâmetro de entrada de um bloco funcional e um parâmetro de saída do outro. Os blocos funcionais podem ser limitados juntos, internamente, e através dos equipamentos. As interfaces entre os blocos funcionais localizadas na mesma aplicação do bloco funcional são localmente definidas. Estas interfaces entre os blocos funcionais em diferentes equipamentos usam as tarefas de comunicação.

Para suportar a operação dos blocos funcionais, a arquitetura do bloco funcional também suporta os blocos transducer, resource e objetos de display.

O processo de aplicação do bloco funcional representa a aplicação do bloco funcional como uma configuração integrada destes componentes acessados para sua interface de rede.

## Objeto do Bloco

Um objeto de bloco representa a unidade lógica de processamento composta dos parâmetros de entrada, processamento e de controle e um algoritmo associado.

Durante a operação do sistema, um guia de referência rápido conhecido como índice numérico é usado para fins de acesso ao bloco. O índice numérico de um bloco é único somente dentro da aplicação onde o bloco funcional existe.

O algoritmo do bloco é identificado por seu tipo e pelo seu nível de revisão. Esta informação indica como a execução do algoritmo é afetada pelos parâmetros de controle.

## Parâmetros dos Blocos

Os parâmetros definem as entradas, saídas e dados de controle para um bloco. Seus relacionamentos um com o outro e com o algoritmo do bloco são mostrados abaixo.

## Parâmetro dos Identificadores

Os nomes dos parâmetros são únicos dentro de um bloco. Dentro de um sistema, um parâmetro pode ser facilmente identificado qualificando o nome com o tag do bloco.

### Uso do Parâmetro

Os parâmetros são definidos para um bloco com um propósito específico. Cada um é definido para ser usado com uma entrada, uma saída ou um parâmetro de controle. Os parâmetros de controle, também, são chamados parâmetros *contained* porque eles podem não ser conectados ciclicamente com os parâmetros nos outros blocos. Cada tipo de uso está definido a seguir:

#### Contained

Um parâmetro "*contained*" é um parâmetro cujo valor é configurado, ajustado por um operador ou equipamento de um nível mais elevado, ou calculado. Não pode ser linkado a outra entrada ou saída do bloco da função. O parâmetro de modo é um exemplo de um parâmetro contido comum para todos os blocos.

#### Saída

Um parâmetro de saída é um parâmetro que pode ser conectado ciclicamente a um parâmetro de entrada do outro bloco funcional. Em geral, os parâmetros de saída contêm o *status* do bloco. O *status* da saída indica a qualidade do valor do parâmetro e o modo do bloco quando ele foi gerado.

O valor de um parâmetro de saída pode não ser obtido de uma fonte externa para o bloco. Ele pode, ou não, ser gerado pelo algoritmo do bloco.

Os valores de determinados parâmetros de saída são dependentes do valor do parâmetro do modo do bloco. Estes parâmetros de saída podem ser referidos como parâmetros de saída *mode-controlled*.

Os blocos cujo propósito é gerar uma saída simples contêm um parâmetro projetado como parâmetro de saída primário. As saídas primárias são usadas por outros blocos, com finalidade de controle ou de cálculo. Estes blocos contêm também parâmetros de saída secundários, tais como os parâmetros de alarme e parâmetros de evento que representam um suporte para o parâmetro de saída primário.

#### Entrada

Um parâmetro de entrada obtém seu valor de uma fonte externa para o bloco. Um parâmetro de entrada pode ser conectado ciclicamente a um parâmetro de saída de outro bloco funcional. Seu valor pode ser usado pelo algoritmo do bloco.

Em geral, os valores dos parâmetros de entrada são acompanhados por um *status*. Quando um parâmetro de entrada é conectado ciclicamente a um parâmetro de saída, os *status* serão fornecidos como *status* do parâmetro de saída (quando o parâmetro tem um *status*). Quando ele não é conectado ciclicamente a um parâmetro de saída, os *status* indicarão que o valor não foi fornecido por um parâmetro de saída. Quando um valor esperado do parâmetro de entrada não é recebido, os serviços suportados pelos blocos funcionais responsáveis pela entrega dos dados configuram os *status* do parâmetro de entrada indicando falha.

Se um parâmetro de entrada não é conectado ciclicamente a algum parâmetro de saída, então ele será tratado como um valor constante pela aplicação do bloco funcional. A diferença entre os parâmetros de entrada não ciclicamente conectados e os parâmetros *contained* é que os parâmetros de entrada têm capacidade para suportar uma conexão cíclica e os parâmetros *contained* não a têm.

Os blocos cuja finalidade é transformar ou operar numa única entrada, conterão um parâmetro projetado como entrada primária. Um parâmetro de entrada de alguns tipos de bloco é designado como parâmetro de entrada primária. Entradas primárias são usadas para controle ou cálculo. Estes blocos também podem conter parâmetros de entrada secundárias, que suportam o processamento feito no parâmetro de entrada primária.

## Relações de Parâmetro

A execução do bloco envolve entradas, saídas, parâmetros contained e o algoritmo do bloco. O tempo de execução para um algoritmo do bloco é definido como um parâmetro do bloco. Seu valor depende de como o bloco foi implementado.

Os parâmetros de entrada compostos são usados pelo algoritmo em conjunto com o estado da aplicação do bloco da função que contém o bloco para determinar se o algoritmo pode atingir o modo *target* (modo desejado) estabelecido para ele. O parâmetro de TARGET\_MODE indica qual modo de operação é desejado para o bloco. Normalmente é ajustado por um equipamento de controle ou por um operador.

Sob determinadas circunstâncias operacionais, um bloco pode não estar apto para funcionar no modo requisitado. Nestes casos, o modo *actual* reflete o modo que o bloco deve estar para conseguir realizar as funções requisitadas. A comparação do modo *actual* com o modo *target* indica se o objetivo foi alcançado.

Os valores para o parâmetro Modo de um bloco são definidos pelo parâmetro de Modos Permitidos para o bloco. Assim, os modos disponíveis para controlar um bloco podem variar para os blocos.

Assim que o modo *actual* é determinado, há a execução do bloco e as saídas são geradas.

## Status do Parâmetro

O argumento *status* para os parâmetros de entrada e de saída mostra para os outros blocos o *status* atual deste bloco que está enviando dados ou valores .

O campo *Status* é composto por três partes: *Quality*, *Sub-status* e *Limits*.

**Quality** – indica a qualidade do valor do parâmetro.

- *Good Cascade* – A qualidade do valor é boa e pode fazer parte da estrutura em cascata.
- *Good Non-cascade* – A qualidade do valor é boa, mas o bloco não suporta um caminho em cascata.
- *Uncertain* – A qualidade do valor está abaixo do normal, mas o valor ainda pode ser útil.
- *Bad* – O valor não é útil.

**Sub-status** – O *sub-status* é um complemento do estado da qualidade e leva a informação para inicializar ou parar um controle em cascata, alarmes e outros. Há diferentes configurações do *sub-status* para cada qualidade.

**Limits** – Fornece informação se o valor associado está limitado ou não, bem como a direção. Os limites são classificados como: *Not Limited*, *High Limited*, *Low Limited*, *Constant*.

Quando um parâmetro de entrada é “conectado” num parâmetro de saída pela troca de dados cíclicos, a estrutura toda (*status* e valor) é recebida do barramento. Se a entrada não estiver “conectada”, então os *status* e os valores podem ser configurados manualmente pelo usuário.

## Composição do Status

O *status* tem a seguinte composição:

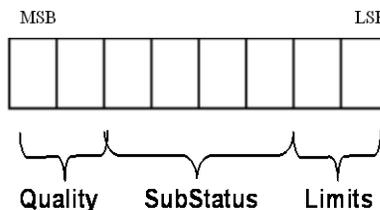


Figura 1.4 – Composição do Status

*Quality*, *Sub-status* e *Limits* são os componentes do *status* e são definidos como se segue:

**Quality** - A qualidade usada será determinada pela condição de mais alta prioridade:

- 0 = *Bad*
- 1 = *Uncertain*
- 2 = *Good Non Cascade*
- 3 = *Good Cascade*

**Sub-status** – Os valores do sub-status no status attribute são definidos conforme Tabela 1.1.

**Limit** – As seguintes condições de limit sempre estarão disponíveis no status Atributo.

- 0 = *Not Limited*
- 1 = *Low Limited*
- 2 = *High Limited*
- 3 = *Constant*

Exemplos de status:

0xC1 (em hexadecimal) significa “*Good-Cascade Non Specific and Low Limited*”

0xCF(em hexadecimal) significa “*Good-Cascade Not invited and Constant*”

0x4E(em hexadecimal) significa “*Uncertain Initial Value and High Limited*”

<b>Quality</b>	<b>Sub-Status</b>	<b>Limit</b>	<b>Valor Hexa</b>	<b>Valor Decimal</b>	<b>Não em Cascata</b>	<b>Caminho para Frente na Cascata</b>	<b>Caminho de Retorno na Cascata</b>
GoodNC	0 = ok ( <b>lowest priority</b> )	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x80	128	X		
GoodNC	1 = <i>Active Update Event</i>	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x84	132	X		
GoodNC	2 = <i>Active Advisory Alarm</i>	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x88	136	X		
GoodNC	3 = <i>Active Critical Alarm</i>	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x8C	140	X		
GoodNC	4 = <i>Unacknowledged Update Event</i>	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x90	144	X		
GoodNC	5 = <i>Unacknowledged Advisory Alarm</i>	0 – <i>Not Limited</i> 1 – <i>Low Limited</i> 2 – <i>High Limited</i> 3 - <i>Constant</i>	0x94	148	X		

GoodNC	6 = Unacknowledged Critical Alarm	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x98	152	X		
GoodNC	8 = Initiate Fail Safe (IFS)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xA0	160	X		
GoodNC	9 = Maintenance required	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xA4	164	X		
Uncertain	0 = Non-specific	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x40	64	X		
Uncertain	1 = Last Usable Value	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x44	68	X		
Uncertain	2 = Substitute	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x48	72	X		
Uncertain	3 = Initial Value	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x4C	76	X		
Uncertain	4 = Sensor Conversion not Accurate	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x50	80	X		
Uncertain	5 = Engineering Unit Range Violation	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x54	84	X		
Uncertain	6 = Sub-normal	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x58	88	X		

Uncertain	7 = Configuration Error	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x5C	92	X		
Uncertain	8 = Simulated Value	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x60	96	X		
Uncertain	9 = Sensor Calibration	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x64	100	X		
GoodC	0 = ok	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xC0	192		X	X
GoodC	1 = Initialization Acknowledged(IA)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xC4	196		X	
GoodC	2 = Initialization Request(IR)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xC8	200			X
GoodC	3 = Not Invited (NI)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xCC	204			X
GoodC	5 = Do Not Selected(NS)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xD4	212		X	
GoodC	6 = Local Override(LO)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xD8	216			X
GoodC	8 = Initiate Fail Safe (IFS)	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0xE0	224		X	

<i>Bad</i>	0 = Non-specific	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x00	0	X	X	X
<i>Bad</i>	1 = Configuration Error	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x04	4	X	X	X
<i>Bad</i>	2 = Not Connected	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x08	8			
<i>Bad</i>	3 = Device Failure	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x0C	12	X	X	X
<i>Bad</i>	4 = Sensor Failure	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x10	16	X	X	X
<i>Bad</i>	5 = No Communication, with last usable value	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x14	20			
<i>Bad</i>	6 = No Communication, with no usable value	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x18	24			
<i>Bad</i>	7 = Out of Service ( <b>highest priority</b> )	0 – Not Limited 1 – Low Limited 2 – High Limited 3 - Constant	0x1C	28			

Tabela 1.1 – Composição dos Status

### Exemplo: Conversão das Enumerações para Status

A seguinte fórmula é usada para obter o número da enumeração de um atributo para um determinado status:

$$\text{Estado decimal do valor} = 64 \times \text{Quality} + 4 \times \text{Sub-Status} + \text{Limit}$$

Por exemplo, considerando o seguinte status:

“Uncertain - Initial Value - High Limited”

Onde:

$Quality = "Uncertain" = 1$

$Sub-status = "Initial Value" = 3$

$Limit = "High Limited" = 2$

Aplicando a fórmula:

$Decimal Value Status = 64 \times 1 + 4 \times 3 + 2 = 78$  (em decimal) 0x4E (em Hexadecimal)

## Exemplo: Conversão de Número para Enumerações

Há muitas formas para converter o número enumerado à *string* de *status*. Abaixo, são mostradas duas formas para fazê-lo:

- 1) Expressando o número em binário.

$Status$  do valor em hexadecimal = 78 = 0x4E = 01001110 (em binário)

Dividindo este número binário nos campos *Quality*, *Sub-Status* e *Limit*:

$Quality = 01 = 1 = "Uncertain"$

Sub-estado = 0011 = 3 = "Valor Inicial"

$Limit = 10 = 2 = "Limitado em Alto"$

O estado correspondente é "*Uncertain - initial value - high limited.*"

- 2) Usando o valor do estado em formato decimal.

$Decimal Value Status = 78$

Dividido o número por 64. O quociente será o *Quality* e armazenado o resto:

$Quality = 78 / 64 = 1$

Resto = 14

Dividido o resto por 4. O quociente será o *Sub-Status* e o resto será o limite:

$SubStatus = 14 / 4 = 3$

$Limit = 2$

## Tratamento do Parâmetro CHANNEL

Os blocos funcionais de Entrada Analógica, Saída Analógica e Totalizador são conectados ao bloco transdutor através do parâmetro *CHANNEL*.

O formato do parâmetro *CHANNEL* é *Unsigned16* e seu valor representa um ponteiro relacionado ao bloco transdutor e a seu parâmetro. Consiste em 2 elementos:

- TransducerID (o primeiro Byte é 1 para o primeiro transdutor, 2 para o segundo, etc. (Depende da ordem no diretório).
- Índice relativo do parâmetro do bloco transdutor usado (Segundo *Byte*).

Opcionalmente, o parâmetro *CHANNEL* pode ser desconectado ajustando seu valor para zero (0x00).

## Canal dos Blocos Funcionais de Entrada

Tipicamente, um bloco transdutor de um transmissor tem três parâmetros que podem ser conectados ao bloco funcional de entrada: Valor Primário (PV), Valor Secundário1 (SV1) e Valor Secundário 2 (SV2).

O bloco funcional conectado ao transdutor pode ser conectado com cada uma destas saídas. Embora ele dependa do equipamento e do bloco funcional. Há algumas regras que são descritas abaixo:

- O bloco Totalizador só pode ser configurado com o parâmetro PV;
- Tipos diferentes de blocos funcionais (por exemplo, AI ou TOT), podem ser configurados com o mesmo valor de *CHANNEL*, mas blocos funcionais do mesmo tipo (por exemplo, AI e AI ou TOT e TOT) não podem referenciar ao mesmo transducer ID no *CHANNEL*;

- O índice relativo dos parâmetros de saída do transdutor (PV, SV1, SV2) é diferente para cada tipo de equipamento.

## Canal dos Blocos Funcionais de Saída

O Transdutor de um equipamento atuador só tem um parâmetro de referência, e o Índice Relativo do Bloco Funcional do parâmetro de canal deverá ser 0 (zero).

O bloco AO tem dois canais, sendo um para conectar o transdutor (AO→TRD) e *OUT\_CHANNEL*, onde o bloco AO envia o valor calculado ao transdutor. O outro canal, é utilizado para conectar o TRD ao AO (TRD→AO) e *IN\_CHANNEL*, onde o transdutor envia a posição ao parâmetro *readback* do bloco de saída analógico. Para os equipamentos Smar não são necessários os 2 canais, portanto é recomendado o mesmo valor para ambos os canais.

## Exemplos de Uso do Canal

- 1) Configurando o canal do equipamento LD303:

Considerando que o equipamento LD303 possui 1 transdutor e 2 blocos funcionais: 1 AI e 1 TOT. Os possíveis canais destes blocos funcionais podem ser:

- a) Há apenas um bloco transdutor, portanto o primeiro byte é 1.
- b) O bloco TOT deve ser configurado com o parâmetro PV. Portanto o segundo byte do canal tem de ser PV. (No equipamento LD303, o PV do transdutor possui o índice relativo 18 (0x12 relativo)).
- c) No bloco AI, o segundo byte do canal pode ser PV, SV1 ou SV2 (índice relativo SV1 = 29 (0x1D), índice relativo SV1 = 31(0x1F)).

O canal de AI e TOT será:

*AI.CHANNEL* = 0x011D (se a saída do transdutor escolhido for SV1) ou 0x011F (Se a saída for SV2) ou 0x0112 (se o parâmetro escolhido for PV)

*TOT.CHANNEL* = 0X0112 (saída PV)

- 2) Configurando o canal do equipamento FI303

Considerando que o equipamento FI303 possui 3 transdutores e 3 blocos funcionais AO. Os possíveis canais deste FBs podem ser:

- a) Para este caso de 3 transdutores, o primeiro byte pode ser 1 (para o primeiro transdutor), 2 (para o segundo transdutor), 3 (para o terceiro transdutor).
- b) O segundo byte para o equipamento atuador será sempre zero (0).

Os canais dos blocos AO seriam:

*AO1.IN\_CHANNEL* = *AO1.OUT\_CHANNEL* = 0X0100 (o primeiro AO configurado com o primeiro transdutor)

*AO2.IN\_CHANNEL* = *AO2.OUT\_CHANNEL* = 0X0200 (o segundo AO configurado com o segundo transdutor)

*AO3.IN\_CHANNEL* = *AO3.OUT\_CHANNEL* = 0X0300 (o terceiro AO configurado com o terceiro transdutor)

## Cálculo de Saída

Quando o modo atual é AUTO ou RCAS, o algoritmo é executado normalmente. Este cálculo é específico para cada tipo de bloco funcional. Se o equipamento está em modo "manual", a saída é seguida de um valor fornecido pelo usuário (Man ou LO).

No modo manual o estado da saída será de "*Uncertain Simulate Value*" indicando que a saída está sendo escrita.

## Controle de Cascata Remota

Em uma cascata remota, o bloco de controle superior fornece o valor do *status* da saída, que se torna a entrada da cascata remota para o bloco inferior.

O bloco inferior em cascata remota fornece um valor de saída que é comunicado ao bloco superior como retorno de cálculo.

O seguinte exemplo mostra o processo de inicialização da cascata remota.

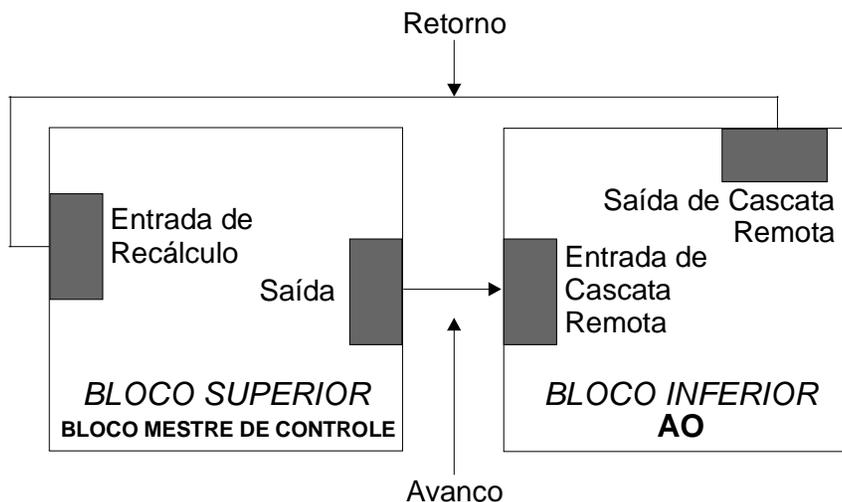


Figura 1.5 – Exemplo da Cascata Remota

Quatro passos para completar a inicialização de cascata remota:

**1. Not cascade mode** – O bloco AO está no modo Auto e será indicado para o próximo bloco superior (no equipamento mestre) que a saída do mestre não está sendo usada. Sendo assim, o bloco (AO) envia o valor atual usando (SP *value*) e o estado *GoodC-Not Invited* de volta.

AO

Target\_Mode = Auto

RCAS\_IN.Status = Qualquer estado (porque o bloco não está no modo cascata)

MODE\_BLK.Actual = Auto

BKCAL\_OUT.Status = *GoodC-not Invited*

**2. Initialize** – O usuário muda o modo *Target* do bloco inferior (AO) para RCas e o bloco AO envia o estado de “GoodC-IR” na saída RCAS\_OUT. O valor de RCAS\_OUT é o valor inicial para que o bloco superior inicie os cálculos. O bloco AO aguarda o estado “*Good initiate acknowledge*” na entrada RCAS\_IN. Esta entrada é conectada à saída do bloco imediatamente superior. O bloco AO continua executando o último valor válido de SP.

AO

Target\_Mode = Rcas

RCAS\_IN.Status = *Good Cascade* e substates diferentes de *Initiate Acknowledge*

MODE\_BLK.Actual = Auto

BKCAL\_OUT.STATUS = *GOODC-INICIALIZATION REQUEST (IR)*

**3. Initialization complete** – O bloco AO vai para Rcas, pois o bloco superior enviou um estado “GoodC-IA” para a entrada RCAS\_IN. Agora, o bloco AO começa a ler o valor de SP da entrada RCAS\_IN.

AO

Target\_Mode = Rcas

RCAS\_IN.STATUS = *GOOD CASCADE – INITIATE ACKNOWLEDGE (IA)*

MODE\_BLK.Actual = Rcas

RCAS\_OUT.Status = *GoodC-Ok*

**4. Cascade complete** – O bloco superior muda o *status* da saída de *GoodC-IA* para *GoodC-ok*. O bloco AO lê o valor SP da entrada RCAS\_IN.

AO  
 Target\_Mode = RCas  
 RCAS\_IN.Status = Good Cascade- Ok  
 MODE\_BLK.Actual = RCas  
 RCAS\_OUT.Status = Good Cascade - Ok

**Nota:**

O bloco inferior não pode ser executado em RCas: o modo designado do bloco inferior não é RCas ou há alguma condição forçando o bloco inferior para uma prioridade maior como condições de *Fail Safe* (Veja detalhes na seção Parâmetros do Modo).

## Parâmetros do Modo

### a) Modo de Operação

A operação do bloco é resumida para cada tipo de modo conforme abaixo:

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

O bloco não está sendo avaliado. A saída é mantida no último valor ou, em caso de falta de alimentação, pode ser programado para manter um valor.

#### **Local Override (LO):**

O bloco de saída não está sendo calculado, embora possa estar limitado. Aplica-se ao bloco de controle que suporta parâmetro de entrada rastreado. Quando o bloco está em LO, a saída segue o valor estabelecido pelo usuário localmente (através de atuações de chaves magnéticas). O usuário não pode alterar as saídas do host remoto.

#### **Manual (Man):**

A saída do bloco não está sendo calculada, embora possa estar limitada. Neste modo, o operador pode ajustar diretamente as saídas do bloco.

#### **Automático (Auto):**

O algoritmo normalmente calcula a saída do bloco. Se o bloco tiver um setpoint ele será usado com um valor local, que pode ser gravado pelo operador através de um equipamento de interface local.

A saída do bloco é calculada usando a entrada do bloco transdutor, no caso de um bloco funcional, e usando um valor de setpoint fornecido por um servidor ou um operador através de uma interface em caso de um bloco funcional de saída.

#### **Remote Cascade (RCas):**

O setpoint do bloco está sendo ajustado por um aplicativo de controle através do parâmetro de cascata remota RCAS\_IN. O algoritmo normal calcula a saída do bloco baseado naquele setpoint.

Os modos “automático” são Auto e Rcas, que calculam a saída primária usando o algoritmo normal. Os modos “manual” são LO e Man.

Modo	Fonte SP	Fonte de Saída
O/S	Usuário	Usuário
LO	Usuário	Usuário
Man	Usuário	Usuário
Auto	Usuário	Algoritmo do Bloco
Rcas	Aplicação de controle sendo executada no equipamento de interface	Algoritmo do Bloco

**Tabela 1.2 – Parâmetros do Modo**

## b) Elementos do Modo de um Bloco Funcional

O modo é composto por dois parâmetros: TARGET\_MODE e MODE\_BLK que são definidos para todos os blocos. Estes parâmetros são explicados abaixo:

**TARGET\_MODE** - este é o modo solicitado pelo operador. Apenas um modo da lista de modos permitidos pelo parâmetro pode ser solicitado. Isto é checado pelo próprio equipamento.

**MODE\_BLK** - este parâmetro é calculado pelo algoritmo baseado nas entradas e no modo *Target*. Entretanto, o usuário não pode escrever nos atributos deste parâmetro. É definido como tendo três elementos:

- **Actual** - Este é o modo atual do bloco. Pode ser diferente do TARGET\_MODE baseado nas condições operacionais e na configuração do bloco, como *status* do parâmetro de entrada e configuração de bypass. Seu valor sempre é calculado como parte de execução do bloco.
- **Permitted** - Este define os modos que são permitidos para uma determinada instância do bloco. É como uma lista de modos selecionados dos modos suportados.
- **Normal** - Este é o modo que o bloco deve ser ajustado para condições operacionais normais. O atributo normal é usado como lembrete. Ele não afeta o cálculo do algoritmo.

A execução de um bloco funcional é controlada pelo parâmetro de modo. O usuário configura o TARGET\_MODE e indica qual o modo de operação é desejado para o bloco. Assim, o algoritmo avalia se o bloco pode ser executado no modo requisitado ou no modo mais próximo da prioridade. O modo atual reflete no modo de operação do bloco.

## c) Prioridade de modo

O conceito de prioridade é usado quando o bloco calcula o modo atual.

Modo	Descrição	Prioridade
O/S	Fora de Serviço	7- mais alto
LO	Passagem Local	5
Man	Manual	4
Auto	Automático	3
Rcas	Cascata Remota	1 - mais baixo

Tabela 1.3 – Prioridade do Modo

## d) Modo de Cálculo

O modo atual será calculado da seguinte forma:

- Cada modo tem algumas condições que fazem com que o modo atual tenha prioridade mais alta que o modo designado;
- Partindo do modo de mais alta prioridade (O/S) é analisado suas condições correspondentes. Se estão presentes, então o modo atual será este. Caso contrário, é necessário checar as condições para o modo de prioridade mais inferior (LO, Man, Auto e Rcas) até o modo designado, exclusivo. Por exemplo, se o modo designado é RCas, é necessário checar as condições de O/S, LO, Man e Auto, nesta ordem. Se todas essas condições forem falsas, o modo atual será o modo designado.

Modo	Condições
O/S	- O modo designado é O/S
LO	- O modo designado é O/S
Man	- O modo designado é Man. - O modo designado é RCas, na execução anterior, o modo atual era Man e o estado de entrada RCAS_IN não é Good_IA.
Auto	- O modo designado é Auto - O modo designado é RCas e na execução anterior o modo atual era LO ou O/S. - O objetivo e na execução anterior o modo atual era Rcas e o bloco vai para fail safe. (RCAS_IN.status é Good_IFS ou o estado de RCAS_IN é ruim na maior parte do tempo do que FSAFE_TIME) - O modo designado é Rcas e na execução anterior o modo atual era Auto e o estado de entrada era RCAS_IN não é Good_IA.
RCas	- O modo designado é Cas e a inicialização da cascata completou-se. (O status de entrada RCAS_IN é GOOD_IA). - O modo atual da última execução foi Rcas.

Tabela 1.4 - Modo de Cálculo

### e) Informação específica para desenvolvedores de drivers

Internamente, cada atributo do modo é designado dentro do *bitstring* da seguinte maneira (o *bitstring* é o mesmo para o TARGET\_MODE e para os elementos Atual, Normal e MODE\_BLK).

Modo	Valor HEX	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
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
Rcas	0x02	0	0	0	0	0	0	1	0

Tabela 1.5 – Modo Bitstring

O elemento suportado do MODE\_BLK é a lógica OU com os respectivos modos suportados. Por exemplo, o bloco de entrada analógico possui o modo suportado = O/S, AUTO e MAN, assim o modo suportado será:

Modo	Valor HEX	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
O/S,Auto e Man	0x98	1	0	0	1	1	0	0	0

Tabela 1.6 – Modo suportado Bitstring do bloco AI

## Parâmetros de Escala

O parâmetro escala define a faixa operacional e as unidades de engenharia associadas ao parâmetro. Também define o número de dígitos à direita do ponto decimal, o qual deve ter uma interface com um equipamento para exibir este parâmetro.

A informação da escala é usada para dois propósitos:

- Os equipamentos que possuem display necessitam ter conhecimento da faixa para as *bar graphs* e tendências, assim como os códigos de unidade.
- Os blocos de controle precisam saber a faixa para uso interno da porcentagem do span. Assim, as constantes de ajuste podem permanecer adimensionáveis.

O bloco AI usa o OUT\_SCALE para converter o valor do transdutor para a unidade de engenharia utilizada na aplicação.

O bloco AO usa o OUT\_SCALE para converter o valor de SP à unidade de engenharia esperada pelo bloco transdutor de saída, que também, são as unidades de engenharia do valor de readback.

Os campos seguintes formam a escala:

- Unidades de Engenharia a 100% da escala - O valor que representa o limite superior da faixa em unidade de engenharia.
- Unidades de Engenharia a 0% da escala - O valor que representa o limite inferior da faixa em unidade de engenharia.
- Índice das Unidades – Descrição do índice dos códigos para unidades de Engenharia.
- Ponto Decimal - O número de dígitos à direita do ponto decimal que devem ser usados por um equipamento de interface na indicação do parâmetro especificado.

## Exemplo para Uso de Parâmetros de Escala

O algoritmo AO trabalha internamente com valores em percentagem do span. O bloco AO converte o SP em percentagem (PV\_SCALE), calcula a percentagem de saída e, então, converte-a para a unidade de engenharia na saída (OUT\_SCALE).

1. O AO leva o SP e converte para percentagem da PV\_SCALE:

$$SP\% = \frac{(SP - EU\_0) * 100}{(EU\_100 - EU\_0)} \quad [PV\_SCALE]$$

PV\_SCALE:  
EU at 100% = 20  
EU at 0% = 4  
Unidade = mA  
Casa Decimal = 2

SP = 15 mA

Os valores de SP em percentagem são:

$$SP\% = \frac{(15 - 4) * 100}{(20 - 4)}$$

SP% = 68.75%

2. O valor de saída é convertido de percentagem para unidades de engenharia do OUT\_SCALE:

$$OUT = \frac{SP\%}{100} * (EU\_100\% - EU\_0\%) + EU\_0\% \quad [OUT\_SCALE]$$

OUT\_SCALE:  
EU at 100% = 15  
EU at 0% = 3  
Unidade = psi  
Casa Decimal = 2

O valor de saída deste exemplo é:

$$OUT = \frac{68.75}{100} * (15 - 3) + 3$$

OUT = 11.25 psi

## Tratamento do Estado de Falha (Fail Safe)

O *Fail Safe* é um estado especial, que permite ao bloco funcional atuar quando for detectado uma situação anormal.

Uma situação anormal ocorre quando houver uma entrada não usual, por exemplo um sensor com defeito ou perda de comunicação entre os blocos funcionais por um período maior que o especificado (FSAFE\_TIME).

Quando a condição que ativou o estado de falha (*Fault State*) for normalizada, o *Fault State* é zerado e o bloco retorna à operação normal.

## Condições que ativam o *Fail Safe*

Quando os blocos funcionais de entrada ou saída detectam uma condição anormal, o bloco entra no modo *FAIL\_SAFE*. Estas situações anormais são detectadas através de diferentes formas nos blocos funcionais de entrada e saída.

Os blocos funcionais de entrada, por exemplo, AI e TOT são conectados a um bloco superior (bloco transdutor) pelo canal. Quando a saída do transdutor tem um estado ruim, (por exemplo, sensor avariado) a condição de *FAIL\_SAFE* do bloco é ativada.

Os blocos funcionais de saída, por exemplo o bloco AO, recebem os valores de entrada de um dos blocos superiores através da “conexão cíclica”. Estes blocos superiores são blocos funcionais de controle e, geralmente, são equipamentos de controle. Neste caso, o *Fail Safe* é ativado quando uma das seguintes condições é detectada:

- Perda de comunicação de RCAS\_IN por um tempo que superior ao especificado em *FSAFE\_TIME*;
- Perda de comunicação de SP durante um tempo que seja superior ao especificado em *FSAFE\_TIME*;
- O *status* IFS na entrada RCAS\_IN quando o modo designado é RCas;
- O *status* IFS no SP quando o modo designado é Auto.

## Ações de *Fail Safe* (Falha de Segura)

As ações que um bloco de entrada ou de saída pode realizar, quando o bloco está em *Fail Safe*, devem ser selecionadas pelo usuário através do parâmetro *FSAFE\_TYPE* nos blocos AI e AO ou usando o parâmetro de *FAIL\_TOT* no bloco TOT.

No parâmetro *FSAFE\_TYPE* as seguintes opções estão disponíveis:

- Use *FSAFE\_VALUE* – Neste caso, os blocos AI e AO usam o valor de segurança fornecido pelo parâmetro *FSAFE\_VALUE* para o cálculo quando o *Fail Safe* está ativo. O *status* da saída vai para “*Uncertain, substitute value*”;
- Use *Last Usable Value* – Neste caso, os blocos AI e AO usam o último valor usável para o cálculo do algoritmo. O *status* será “*Uncertain Last Usable Level*”. Se ainda não houver um valor satisfatório, use o Valor Inicial na saída. O *status* será “*Uncertain Initial Value*”;
- Use *wrong value* (apenas para o bloco AI) – O bloco AI usa os valores e estados errados para o cálculo;
- Use o *ACTUATOR\_ACTION* (apenas para bloco AO) – O bloco AO entra em posição de segurança baseado em um parâmetro discreto *ACTUATOR\_ACTION* no bloco transdutor.

No parâmetro *FAIL\_TOT* (só usado no bloco Totalizador) as seguintes opções estão disponíveis:

- *Hold* - Pára a totalização no último valor. O *status* da saída vai para “*Uncertain non-specific*”;
- *Memory* - Usa o último valor válido para a totalização. O *status* será “*Uncertain Last Usable Value*”. Se não houver um *status* válido na memória, deve ser usado o valor inicial para a totalização. O *status* será “*Uncertain, Initial Value*”;
- *Run* - A totalização é continuada (Reiniciada). O valor e o *status* incorretos são usados para a saída.

## Diagnóstico

Para fornecer a informação sobre o equipamento para o controle e a interface humana existem alguns parâmetros de diagnóstico no equipamento.

Os parâmetros de diagnóstico têm um *datatype* de *bitstring* e há um parâmetro máscara que indica qual diagnóstico é suportado pelo equipamento.

O *bitstring* de diagnóstico e seu diagnóstico suportado são mostrados na “Descrição de *BitString*.”

## Diagnóstico de Características do Equipamento

No bloco *Physical*, o parâmetro *DIAGNOSIS* possui a informação sobre os “alertas” no equipamento (por exemplo, *device not initialized*, *power up*, *factory init*, *hardware failure*, etc.). O parâmetro *DIAGNOSIS\_MASK* tem o diagnóstico suportado pelo equipamento.

## Diagnóstico do Atuador

No bloco AO existe um parâmetro especial de saída com a finalidade de diagnóstico. O parâmetro *CHECKBACK* é um parâmetro *bitstring* somente leitura que possui o resumo das informações principais sobre o bloco funcional e o bloco transdutor.

O *CHECKBACK* pode ser utilizado para configuração de uma comunicação cíclica e a informação discreta fornecida pelo parâmetro pode ser usada por um equipamento de controle para detectar situações anormais com o atuador.

## Alarmes e Eventos – Processamento de Alerta

Os alarmes e os eventos, conhecidos como alertas, representam as mudanças de estado dentro das aplicações do bloco funcional.

### a) Parâmetro de Alarme (Parâmetro X\_ALM)

O parâmetro de alarme é fornecido num bloco para capturar a informação dinâmica associada ao alarme. O parâmetro de alarme é formado pelos seguintes campos:

- *Unacknowledged* - Não usado.
- *Alarm State* - Este campo indica se o alerta está ativo ou não. O *Alarm State* tem os seguintes estados:
  - 0 – não ativo;
  - 1 – ativo.
- *Time stamp* – Não usado.
- *Subcode* – Não usado.
- *Value* – O valor do parâmetro associado no instante do alerta.

O estado de alarme é desativado quando o bloco vai para o modo *Out of Service*.

### b) Limite de Alarme (Parâmetro X\_LIM)

Um alarme analógico ocorre quando um valor atinge ou excede um limite. Por exemplo, para um alarme configurado para limite alto, a condição de alarme se tornará verdadeira quando o valor analógico for superior a este limite. O estado do alarme mantém-se verdadeiro até que o valor do alarme seja inferior ao limite menos a histerese.

O tipo de alarme pode ser desabilitado ajustando o parâmetro limite de alarme para  $\pm$  INF, que é o default para todos os limites de alarme.

O parâmetro de saída do bloco funcional será comparado ao limite de alarme.

### c) Histerese de Alarme (Parâmetro ALARM\_HYS)

A saída (*OUT*) deverá retornar dentro dos limites do alarme para que a condição de alarme seja removida. No bloco AI, a histerese de alarme é expressa em porcentagem do span de saída (*OUT\_SCALE*). No bloco TOT, a histerese é expressa em unidades de engenharia.

### d) Chave de Alerta (Parâmetro ALERT\_KEY)

É um número de identificação da unidade da planta. Esta informação pode ser usada no mestre para selecionar alarmes, etc.

### e) Resumo de Alarme (Parâmetro ALM\_SUM)

O parâmetro Alarm Summary resume o estado de até 16 alarmes de processo do mesmo bloco. Para cada alarme são mantidos os estados atuais, estados desconhecidos, estados não reportados e estados desabilitados.

Há quatro atributos:

- *Current Alarms* – O estado ativo de cada alarme.
- *Unacknowledged* – O estado não respondido de cada alarme.
- *Unreported* – O estado não reportado de cada alarme.
- *Disabled* – O estado desabilitado de cada alarme.

### f) Atualização do Evento (*Update Event*)

O *Update Event* é um alarme especial que ocorre quando o parâmetro estático é alterado. Quando o parâmetro estático é alterado, o bit correspondente no *ALARM\_SUM* é fixado e a saída do bloco vai para “*Good (NC) Active Update Event*” (se o estado atual tem menos prioridade que ele) e o bloco permanece neste estado por 10 segundos. Após isto, o bloco retorna à situação normal (Saída com o último estado e o *bit Update\_Event* no parâmetro *ALARM\_SUM* é “zerado”).

## Definição e Tipos de Estrutura de Dados

Nesta seção são definidos toda estrutura de dados e tipos de dados usados no sistema.

Código do Tipo de Dados	Tipo de Dado	Tamanho	Descrição
1	<i>Boolean</i>	1	Verdadeiro ou falso
2	<i>Integer8</i>	1	
3	<i>Integer16</i>	2	
4	<i>Integer32</i>	4	
5	<i>Unsigned8</i>	1	
6	<i>Unsigned 16</i>	2	
7	<i>Unsigned 32</i>	4	
8	<i>Floating Point</i>	4	
9	<i>VisibleString</i>	1,2,3,...	Eles são um byte por caractere e inclui os caracteres ASCII de 7 bits.
10	<i>OctetString</i>	1,2,3,...	<i>Octet strings</i> são binárias
-	Dados	-	
-	<i>TimeOfDay</i>	-	
-	<i>TimeDifferenc</i>	-	
-	<i>BitString</i>	-	
-	<i>DataTimeValu</i>	-	

Tabela 1.7 – Definição e Tipos de Estrutura de Dados

### Objeto do Bloco – DS-32

Esta estrutura de dados consiste nos atributos de um bloco.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Reserved</i>	<i>Unsigned8</i>	1
2	<i>Block Object</i>	<i>Unsigned8</i>	1
3	<i>Parent Class</i>	<i>Unsigned8</i>	1
4	<i>Class</i>	<i>Unsigned8</i>	1
5	<i>DD REFERENCE</i>	<i>Unsigned32</i>	4
6	<i>DD REVISION</i>	<i>Unsigned16</i>	2
7	<i>Profile</i>	<i>OctetString</i>	2
8	<i>Profile Revision</i>	<i>Unsigned16</i>	2
9	<i>Execution Time</i>	<i>Unsigned8</i>	1
10	<i>Number_of_Parameters</i>	<i>Unsigned16</i>	2
11	<i>ADDRESS OF VIEW_1</i>	<i>Unsigned16</i>	2
12	<i>Number of Views</i>	<i>Unsigned8</i>	1

Tabela 1.8 – Objeto do Bloco DS-32

### Valor & Status - Estrutura do Ponto Flutuante – DS-33

Esta estrutura de dados consiste no valor e *status* dos parâmetros de ponto flutuante, que podem ser entradas ou saídas.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Value</i>	<i>Float</i>	4
2	<i>Status</i>	<i>Unsigned8</i>	1

Tabela 1.9 – Estrutura do Ponto Flutuante Ds-33

### Valor & Status - Estrutura Discreta – DS-34

Esta estrutura de dados consiste de valor e *status* dos parâmetros de valores discretos.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Value</i>	<i>Unsigned8</i>	1
2	<i>Status</i>	<i>Unsigned8</i>	1

Tabela 1.10 – Estrutura Discreta DS-34

### Estrutura de Escala – DS-36

Esta estrutura de dados consiste nos dados estáticos usados para escalar valores de pontos flutuantes para propósitos de indicação no display.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>EU at 100%</i>	<i>Float</i>	4
2	<i>EU at 0%</i>	<i>Float</i>	4
3	<i>Units Index</i>	<i>Unsigned16</i>	2
4	<i>Decimal Point</i>	<i>Integer8</i>	1

Tabela 1.11 – Estrutura de Escala DS-36

### Estrutura de Modo – DS-37

Esta estrutura de dados consiste em *bitstrings* para os modos: atual, permitido e normal.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Actual</i>	<i>Bitstring</i>	1
2	<i>Permitted</i>	<i>Bitstring</i>	1
3	<i>Normal</i>	<i>Bitstring</i>	1

Tabela 1.12 – Estrutura de Modo DS-37

### Estrutura de Alarme de Ponto Flutuante–DS-39

Esta estrutura de dados consiste em dados que descrevem alarmes de pontos flutuantes.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Unacknowledged</i>	<i>Unsigned8</i>	1
2	<i>Alarm State</i>	<i>Unsigned8</i>	1
3	<i>Time Stamp</i>	<i>Time Value</i>	8
4	<i>Subcode</i>	<i>Unsigned16</i>	2
5	<i>Value</i>	<i>Float</i>	4

**Tabela 1.13 – Estrutura de Alarme de Ponto Flutuante DS-39**

### Estrutura de Alarme Discreta–DS-40

Esta estrutura de dados consiste em dados que descrevem alarmes discretos.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Unacknowledged</i>	<i>Unsigned8</i>	1
2	<i>Alarm State</i>	<i>Unsigned8</i>	1
3	<i>Time Stamp</i>	<i>Time Value</i>	8
4	<i>Subcode</i>	<i>Unsigned16</i>	2
5	<i>Value</i>	<i>Unsigned8</i>	1

**Tabela 1.14 – Estrutura de Alarme Discreta DS-40**

### Estrutura de Atualização do Alarme–DS-41

Esta estrutura de dados consiste em dados que descrevem um alarme de revisão estático.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Unacknowledged</i>	<i>Unsigned8</i>	1
2	<i>Update State</i>	<i>Unsigned8</i>	1
3	<i>Time Stamp</i>	<i>Time Value</i>	8
4	<i>Subcode</i>	<i>Unsigned16</i>	2
5	<i>Relative Index</i>	<i>Unsigned16</i>	2

**Tabela 1.15 – Estrutura de Atualização de Alarme DS-41**

### Estrutura Índice de Alarme–DS-42

Esta estrutura de dados consiste em dados que mostram 16 alertas.

E	Nome de elemento	Tipo de Dado	Tamanho
1	<i>Current</i>	<i>Bit String</i>	2
2	<i>Unacknowledged</i>	<i>Bit String</i>	2
3	<i>Unreported</i>	<i>Bit String</i>	2
4	<i>Disabled</i>	<i>Bit String</i>	2

**Tabela 1.16 – Estrutura do Índice de Alarme**

### Simulate – Estrutura do Ponto Flutuante–DS-50

Esta estrutura de dados consiste de uma simulação, valores de ponto flutuante, estado e uma simulação discreta de habilitação/deshabilitação (enable/disable).

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Simulate Status</i>	<i>Unsigned8</i>	1
2	<i>Simulate Value</i>	<i>Float</i>	4
3	<i>Simule En/Disable</i>	<i>Unsigned8</i>	1

**Tabela 1.17 – Simulate – Estrutura do Ponto Flutuante – DS50**

### Simulate - Estrutura Discreta-DS-51

Esta estrutura de dados consiste de uma simulação, valor discreto do transdutor e uma simulação discreta de habilitação/desabilitação (enable/disable).

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>Simulate Status</i>	<i>Unsigned8</i>	1
2	<i>Simulate Value</i>	<i>Unsigned8</i>	1
5	<i>Simule En/Disable</i>	<i>Unsigned8</i>	1

**Tabela 1.18 – Simulate – Estrutura Discreta DS-51**

### Estrutura de Grupo-DS-67

Esta estrutura de dados contém uma estrutura do parâmetro Batch.

E	Nome do elemento	Tipo de Dado	Tamanho
1	<i>BATCH_ID</i>	<i>Unsigned32</i>	4
2	<i>RUP</i>	<i>Unsigned16</i>	2
3	<i>OPERATION</i>	<i>Unsigned16</i>	2
4	<i>PHASE</i>	<i>Unsigned16</i>	2

**Tabela 1.19 – Estrutura de Grupo DS-67**

# BIBLIOTECA DE BLOCOS

A Tabela 2.1 fornece uma descrição dos blocos.

BLOCO	DESCRIÇÃO
RES	<i>PHYSICAL</i> – Este bloco contém informação específica para o hardware associado com o recurso.
AI	<i>ANALOG INPUT</i> – Este bloco obtém a informação de entrada do bloco transdutor e disponibiliza para outros blocos funcionais. Possui uma escala de conversão, “filtro” e mecanismo de fail safe (falha segura).
AO	<i>ANALOG OUTPUT</i> – O bloco AO fornece um valor para um bloco transdutor de saída. Fornece valores, escalas de conversão, mecanismo de <i>fail safe</i> e outros recursos.
TOT	<i>TOTALIZER</i> – Integra uma variável em função do tempo. Possui uma escala de conversão, “filtro” e mecanismo de <i>fail safe</i> .

Tabela 2.1 - Biblioteca de Blocos

## Bloco Físico – PHY

### Descrição

Este bloco contém dados que são específicos ao hardware associado ao equipamento. Todo dado é modelado como *Contained*, portanto não há entradas ou saídas para este bloco. O dado não é processado como em um bloco funcional.

Este conjunto de parâmetros é o mínimo requerido para a aplicação de blocos funcionais associado com o recurso em que bloco PHY reside. Alguns parâmetros que poderiam estar neste conjunto, por exemplo, dados de calibração e temperatura ambiente, são mais relacionados ao bloco transducer."

### Parâmetro FACTORY\_RESET

Este parâmetro permite níveis de inicialização para o equipamento. Eles são:

Opção	Nome	Descrição
1	<i>Restart with defaults</i> (Reinicia com os padrões)	Para limpar a memória de configurações. Funciona como uma inicialização de fábrica.
2506	<i>Restart processor</i> (Reinicia processador)	É uma forma de acionar o botão Reset no processador associado ao equipamento
2712	<i>Restart bus address</i> (Reinicia endereços no barramento)	Inicia endereços para o valor padrão, 126.

Tabela 2.2 – Parâmetros do FACTORY\_RESET

### Memória não volátil

Os equipamentos Smar não suportam armazenamento cíclico de parâmetros voláteis na memória não-volátil. Por outro lado, os equipamentos Smar têm um mecanismo para armazenar os parâmetros não voláteis numa memória não volátil durante o desligamento do equipamento e estes serão recuperados posteriormente ao religar o equipamento.

### Bloqueio de gravação via software

Se o parâmetro *WRITE\_LOCKING* estiver bloqueado (“*locked*”), ele impedirá qualquer alteração externa no banco de dados do equipamento. As conexões de bloco e resultados de cálculos continuarão normalmente, mas a configuração será bloqueada. Se ele for “*Write Unlocked*”, a configuração será novamente permitida.

## Diagnósticos

O parâmetro *DIAGNOSIS* possui uma avaliação dos pontos principais para que o equipamento funcione devidamente.

## Seletor do Número Identificador

O parâmetro *IDENT\_NUMBER\_SELECTOR* permite ao usuário selecionar um número diferente do identificador.

Se o perfil do equipamento é mudado para *Ident Number*, o equipamento deve interagir com as características do perfil do arquivo GSD.

## Modos Suportados

AUTO.

## Parâmetros

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento / Modo	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Este parâmetro será incrementado sempre que houver mudança em parâmetros estáticos do bloco.
2	TAG_DESC	OctString(32)		Espaços	NA	S	Título do bloco. Este parâmetro deve ser único na configuração.
3	STRATEGY	Unsigned16		0	Nenhuma	S	É um valor fornecido pelo usuário para identificar uma configuração.
4	ALERT_KEY	Unsigned8	1 a 255	0	Nenhuma	S	
5	TARGET_MODE	Unsigned8	AUTO	AUTO	Nenhuma	S	Contém o modo desejado para o bloco.
6	MODE_BLK	DS-37			NA	D/RO	Ver parâmetros de modo
7	ALARM_SUM	DS-42		0,0,0,0	Nenhuma	D/RO	Contém os estados presentes dos alarmes de bloco.
8	SOFTWARE_REVISION	VisibleString(16)	Configurado pela Fábrica		Nenhuma	S/RO	Número da revisão do <i>software</i> no equipamento de campo.
9	HARDWARE_REVISION	VisibleString(16)	Configurado pela Fábrica		Nenhuma	S/RO	Número da revisão do hardware associado ao resource.
10	DEVICE_MAN_ID	Unsigned16	Configurado pela Fábrica		Nenhuma	S/RO	Número de identificação do fabricante.
11	DEVICE_ID	VisibleString(16)	Configurado pela Fábrica		Nenhuma	S/RO	Número do modelo do fabricante associado ao equipamento.
12	DEV_SER_NUM	VisibleString(16)	Configurado pela Fábrica		Nenhuma	S/RO	Número de série do equipamento de campo.
13	DIAGNOSIS	Octetstring(4)			Nenhuma	D/RO	<i>Bitstring</i> indicando o diagnóstico do equipamento. Ver tópico Diagnóstico.
14	DIAGNOSIS_EXTENSION	Octetstring(6)			Nenhuma	D/RO	Não usado.
15	DIAGNOSIS_MASK	Octetstring(4)			Nenhuma	D/RO	<i>Bitstring</i> indicando os tipos de diagnósticos suportados pelo aparelho.
16	DIAGNOSIS_MASK_EXTENSION	Octetstring(6)			Nenhuma	D/RO	Não usado.
17	DEVICE_CERTIFICATION	VisibleString(32)			NA	S/RO	Certificações do equipamento de campo.

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento / Modo	Descrição
18	WRITE_LOCKING	Unsigned16	0: Escrita Bloqueada 2457: Escrita Desbloqueada	2457	NA	S	Se bloqueado, nenhuma mudança é permitida, exceto para limpar o WRITE_LOCK. Entradas cíclicas no bloco serão atualizadas continuamente.
19	FACTORY_RESET	Unsigned16	1: Reinicia com padrão 2506: Reinicia o processador 2712: Recupera o endereço padrão para o equipto.	0	NA	S	Permite que o equipamento reinicie manualmente. Vários níveis de reinicialização são possíveis.
20	DESCRIPTOR	Octetstring(32)				S	É uma descrição fornecida pelo usuário do bloco na aplicação.
21	DEVICE_MESSAGE	Octetstring(32)			Nenhuma	S	Mensagem do bloco fornecida pelo usuário na aplicação.
22	DEVICE_INSTALL_DATE	Octetstring(16)				S	Data da instalação do equipamento.
23	LOCAL_OP_ENA	Unsigned8		1	Nenhuma	N	Não usado.
24	IDENT_NUMBER_SELECTOR	Unsigned8	0: Profile specific Ident_Num 1: Mnf specific Ident_Number 2: Mnf specific Ident_Number of V2.0 3: Ident_Number of Multi_Variable device			S	Permite que o usuário mude o IDENT_NUMBER do equipamento.
25	HW_WRITE_PROTECTION	Unsigned8			Nenhuma	D/RO	Não usado.

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RO – Somente leitura; D – dinâmico; N – não-volátil; S – estático

Tabela 2.3 – Parâmetros do Bloco Físico

## Entrada Analógica – AI

### Geral

O bloco de entrada analógico usa os dados de entrada do bloco transdutor, selecionada por um canal, e disponibiliza estes dados em sua saída para outros *Blocos Funcionais*.

## Esquemático

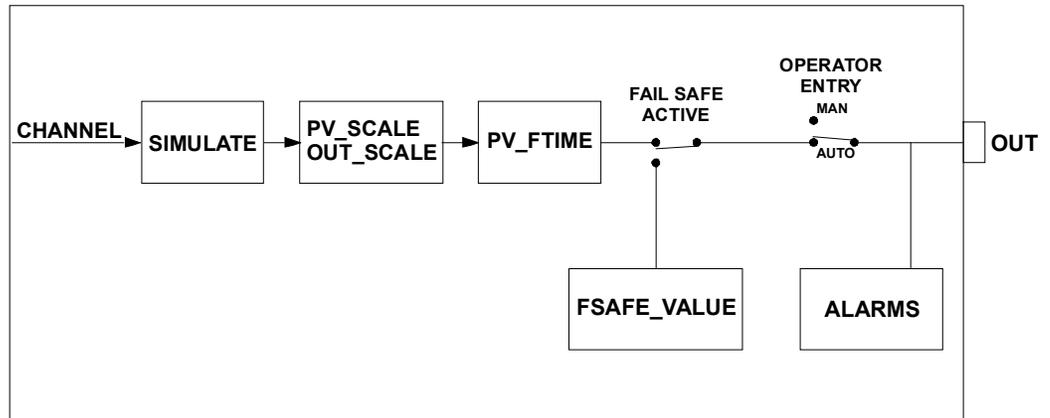


Figura 2.1 – Bloco de Entrada Analógica

## Descrição

O bloco AI realiza uma conversão de escala do valor TRD.PRIMARY\_VALUE utilizando os parâmetros PV\_SCALE e OUT\_SCALE, sendo que a unidade de engenharia em OUT\_SCALE é utilizada apenas para indicação na IHM.

Opcionalmente, um filtro pode ser aplicado ao valor do sinal de processo, que é a constante de tempo PV\_FTIME. Considerando as mudanças na entrada, este é o tempo em segundos para que a PV atinja 63,2% do valor final. Se o valor PV\_FTIME for zero, o filtro é desabilitado.

## Simulação

O parâmetro SIMULATE é usado para fins de diagnóstico e verificação. Quando ativo, o valor do transdutor e o status serão sobrepostos pelo status e valor simulados.

A estrutura SIMULATE é composta pelos seguintes atributos:

- Simulate Value and Status
- Simulate Enable

Quando a simulação é habilitada, o parâmetro de entrada do transdutor será calculado baseada no Valor/Estado atribuído ao parâmetro SIMULATE. Caso contrário, ele será um outro fornecido pelo bloco transdutor.

## Modos Suportados

O/S, MAN e AUTO.

## Status

O bloco AI não suporta o modo cascata. Portanto, o status de saída não possui um sub-status de cascata.

## Cíclico – CFG\_DATA

Configuração suportada	Config Curto (Byte Identificador)	Config Longo (formato Identificador Extendido)
OUT	0x94	0x42,0x84,0x08,0x05

Tabela 2.4 – Cíclico CFG\_DATA

## Parâmetros

Idx	Parâmetro	Tipo Dado (comp.)	Faixa Válida/ Opções	Valor Padrão	Unid.	Armazenamento/ Modo	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Este parâmetro será incrementado sempre que ocorrer mudanças em parâmetros estáticos do bloco.
2	TAG_DESC	OctString(32)		Spaces	NA	S	Título do bloco. Este parâmetro deve ser único na configuração.
3	STRATEGY	Unsigned16		0	Nenhuma	S	É um valor fornecido pelo usuário para identificar uma configuração.
4	ALERT_KEY	Unsigned8	1 a 255	0	Nenhuma	S	
5	TARGET_MODE	Unsigned8	O/S, MAN e AUTO	AUTO	Nenhuma	S	Contém o modo desejado para o bloco.
6	MODE_BLK	DS-37			NA	D/RO	Ver parâmetros de modo
7	ALARM_SUM	DS-42		0,0,0,0	Nenhuma	D/ RO	Contém os estados presentes dos alarmes de bloco.
8	BATCH	DS-67		0,0,0,0	Nenhuma	S	Para uso em sistema fieldbus distribuído . É utilizado para identificar canais usados e disponíveis. Não há algoritmo relacionado.
10	OUT	DS-33	OUT_SCALE		OUT	D / Man	Valor analógico calculado como o resultado da execução da função.
11	PV_SCALE	2 Floats		100,0	Trd output	S	Os valores de escala alto e baixo para o transdutor para um canal especificado.
12	OUT_SCALE	DS-36		100,0,-,-	OUT	S	Os valores de escala alto e baixo para o parâmetro OUT.
13	LIN_TYPE	Unsigned8		0	Nenhuma	S	Não usado.
14	CHANNEL	Unsigned16		0	Nenhuma	S	O número do canal lógico de hardware para o transdutor que será conectado a este bloco I/O.
16	PV_FTME	Float	Non-Negative	0	Sec	S	Constante de tempo de um filtro exponencial para a PV, em segundos.
17	FSAFE_TYPE	Unsigned8	0:Use FSAFE_VALUE 1:Use Last Usable Value 2:Use Wrong Value	1	E	S	Define a reação do equipamento na condição de Fail Safe.
18	FSAFE_VALUE	Float	OUT_SCALE	0	OUT	S	Reajusta o Valor para ajustar a saída quando está ativo o Fail Safe.
19	ALARM_HYS	Float	0 a 50 %	0.5%	%	S	Parâmetro de histerese de alarme. Para limpar alarme, o valor da PV deve retornar dentro dos limites do alarme mais a histerese.
21	HI_HI_LIM	Float	OUT_SCALE, +INF	INF	OUT	S	O ajuste para o alarme superior em unidades de Engenharia.
23	HI_LIM	Float	OUT_SCALE, +INF	INF	OUT	S	O ajuste para o alarme superior em unidades de Engenharia.
25	LO_LIM	Float	OUT_SCALE, -INF	INF	OUT	S	O ajuste para o alarme inferior em unidades de Engenharia.
27	LO_LO_LIM	Float	OUT_SCALE, -INF	INF	OUT	S	O ajuste para o alarme inferior em unidades de Engenharia.
30	HI_HI_ALM	DS-39				D	O estado do alarme superior.
31	HI_ALM	DS-39				D	O estado do alarme superior.
32	LO_ALM	DS-39				D	O estado do alarme inferior.
33	LO_LO_ALM	DS-39				D	O estado do alarme inferior.
34	SIMULATE	DS-50	0: Disable ; ≠0: Active are the Enable options.	Disable		S	Permite que o valor do transdutor seja fornecido manualmente quando habilitado o modo simulação.
35	OUT_UNIT_TEXT	OctString(16)			None	S	É usado quando a unidade específica da saída não está na lista de códigos. O usuário pode fornecer uma definição da unidade em texto.

Legenda: E – parâmetro Enumerado; NA – parâmetro Adimensional; RO – Somente Leitura; D – dinâmico; N – não-volátil; S – estático

Tabela 2.5 – Parâmetros do Bloco de Entrada Analógica

## Saída Analógica – AO

### Geral

O bloco de Saída Analógica é um *Bloco Funcional* usado por equipamentos que funcionam como elementos de saída em uma malha fechada, por exemplo válvulas, atuadores, posicionadores, etc. O bloco AO recebe o sinal de outro *Bloco Funcional* e passa seus resultados para um bloco transdutor de saída através de um canal interno de referência.

### Esquemático

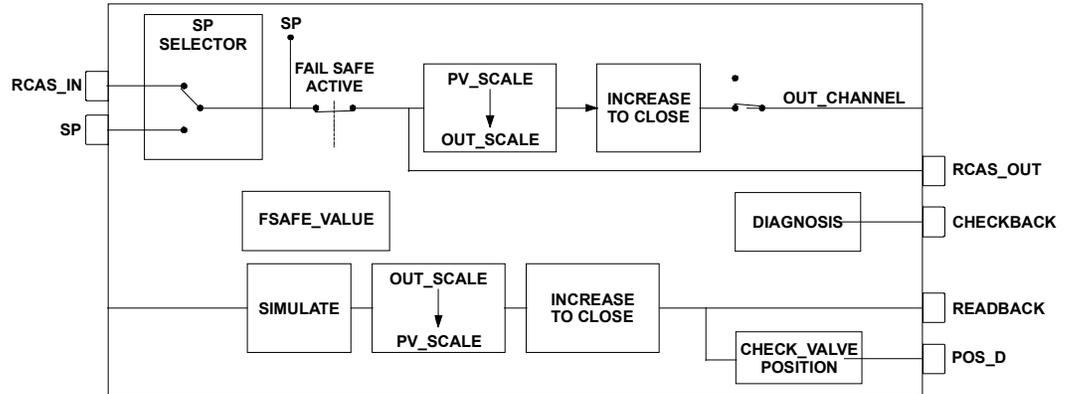


Figura 2.2 – Bloco de Saída Analógica

### Descrição

#### Tratamento de Valores de Entrada

O valor SP pode ser controlado automaticamente através de um controle automático de cascata remota ou manualmente por um operador. A PV\_SCALE é usada para a conversão de escalas de SP.

$$SP\% = \frac{SP}{(EU\_100\% - EU\_0\%) + EU\_0\%} [PV\_SCALE]$$

#### Tratamento de Valores de Saída

A escala de saída (OUT\_SCALE) é usada para converter porcentagem do span para o valor usado pelo transdutor. Isto permite que partes do SP span, cause movimento completo do span na saída.

$$OUT = SP\% * (EU\_100\% - EU\_0\%) + EU\_0\% [OUT\_SCALE]$$

#### Aumentar para fechar (*Increase to Close*)

O parâmetro INCREASE\_CLOSE permite que a saída seja invertida em relação ao span de entrada. Por exemplo, se o SP for 100. (PV\_SCALE=0-100%; OUT\_SCALE = 3-15Psi):

- Se o INCREASE\_CLOSE = 0 (“Incrementando”), SP convertido para OUT\_SCALE será 15 psi. Portanto, o tipo de atuador será: “ar para abrir”.
- Se o INCREASE\_CLOSE = 1 (“Decrementando”), SP convertido para OUT\_SCALE será 3 psi. Portanto, o tipo de atuador será: “ar para fechar”.

### Simulação

O parâmetro *SIMULATE* é usado para fins de diagnóstico e verificação. Quando ativo, o valor do transdutor e o status serão sobrepostos pelo valor e status simulados.

A estrutura *SIMULATE* é composta pelos seguintes atributos:

- *Simulate Value and Status*
- *Simulate Enable*

### Parâmetros de *Readback*

O valor de *readback* do bloco transdutor é composto de dois parâmetros: *READBACK* and *POS\_D*. O *READBACK* é o retorno analógico do transdutor, por exemplo a posição da válvula. O *POS\_D* é um status discreto: aberto, fechado ou posição intermediária.

Se o hardware suporta um valor de *readback*, como posição da válvula, o valor será lido pelo bloco transdutor. Se não for suportado, o valor/estado do transdutor é gerado do *AO\_OUT* pelo bloco transdutor.

Quando a simulação está ativa, os valores e estados de *readback* serão calculados baseados nos valores/estados atribuídos ao parâmetro *SIMULATE*. Caso contrário, será aquele fornecido pelo bloco transdutor.

### Modos Suportados

O/S, LO, MAN, AUTO e RCAS.

### Cíclico – *CFG\_DATA*

Configuração suportada	Config Curto (Byte Identificador)	Config Longo (formato Identificador Estendido)
RCASIN / RCASOUT	0xB4	0xC4,0x84,0x84,0x08,0x05,0x08,0x05
RCASIN / RCASOUT / CHECKBACK	0x97,0xA4	0xC5,0x84,0x87,0x08,0x05,0x08,0x05,0x0A
SP	0xA4	0x82,0x84,0x08,0x05
SP / CHECKBACK	0x92,0xA4	0xC3,0x84,0x82,0x08,0x05,0x0A
SP / READBACK / POSD	0x96,0xA4	0xC6,0x84,0x86,0x08,0x05,0x08,0x05,0x05,0x05
SP / READBACK / POSD / CHECKBACK	0x99,0xA4	0xC7,0x84,0x89,0x08,0x05,0x08,0x05,0x05,0x05,0x0A
SP / READBACK / RCASIN / RCASOUT / POSD / CHECKBACK	0x9E,0xA9	0xCB,0x89,0x8E,0x08,0x05,0x08,0x05,0x08,0x05,0x08,0x05,0x08,0x05,0x05,0x0A

**Tabela 2.6 – Cíclico – *CFG\_DATA***

## Parâmetros

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento/ Modo	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Este parâmetro será incrementado sempre que houver mudança nos parâmetros estático do bloco.
2	TAG_DESC	OctString(32)		Spaces	NA	S	Título do bloco. Este parâmetro deve ser único na configuração.
3	STRATEGY	Unsigned16		0	Nenhuma	S	É um valor fornecido pelo usuário para identificar uma configuração.
4	ALERT_KEY	Unsigned8	1 a 255	0	Nenhuma	S	
5	TARGET_MODE	Unsigned8	O/S, LO, MAN, AUTO, and RCAS.	AUTO	Nenhuma	S	Contém o modo desejado para o bloco.
6	MODE_BLK	DS-37			NA	D/RO	Ver parâmetros de Modo
7	ALARM_SUM	DS-42		0,0,0,0	Nenhuma	D/RO	Contém os estados presentes dos alarmes de bloco.
8	BATCH	DS-67		0,0,0,0	Nenhuma	S	Para uso em sistema fieldbus distribuído, para identificar canais usados e disponíveis. Não há algoritmo relacionado.
9	SP	DS-33	PV_SCALE		PV	N/Auto	O set point analógico pode ser ajustado manualmente, automaticamente através da interface ou qualquer outro equipamento de campo.
11	PV_SCALE	DS-36		0-100%	PV	S	Valores de escala altos e baixos para o parâmetro SP.
12	READBACK	DS-33	PV_SCALE		PV	D/RO	Indica o readback da posição atual do transdutor.
14	RCAS_IN	DS-33			PV	D	Valor de setpoint remoto e estado fornecido por um Host supervisor para um controle analógico ou bloco de saída.
21	IN_CHANNEL	Unsigned16		0	Nenhuma	S	O número do canal de hardware lógico <b>VINDO</b> do transdutor que é conectado ao bloco I/O. Veja manuseio de canais.
22	OUT_CHANNEL	Unsigned16		0	Nenhuma	S	O número do canal de hardware lógico <b>PARA</b> o transdutor que é conectado ao bloco I/O. Veja manuseio de canais.
23	FSAFE_TIME	Float	Positive	0	Segundos	S	Período da detecção da falha até a ação do bloco, se existir a condição.
24	FSAFE_TYPE	Unsigned8	0:Use FSAFE_VALUE 1:Use Last Usable Value 2:Goes to ACTUATOR_ACTION position	1	E	S	Define a reação do equipamento em uma condição de <i>Fail Safe</i> .
25	FSAFE_VALUE	Float	OUT_SCALE	0	OUT	S	Valor de Preset para ajustar a Saída, quando o <i>Fail Safe</i> estiver ativo e o FSAFE_TYPE = 0.
27	RCAS_OUT	DS-33			PV	D/RO	O valor e estado requisitado por um bloco superior para que o bloco superior possa evitar <i>reset windup</i> e proporcionar transferência sem interferência em uma malha fechada de controle.
31	POS_D	DS-34	0: not initialized 1: closed 2: opened 3: intermediate	0	E	D/RO	Posição da válvula.
32	SETP_DEVIATION	Float		0	PV	D/RO	Diferença entre SP e o <i>Readback</i> .

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento/ Modo	Descrição
33	CHECK_BACK	OctString(3)		0	Bitwise	D/RO	Informação do estado do equipamento. Ver Opções de <i>Check Back</i> .
34	CHECK_BACK_M ASK	OctString(3)		0	Bitwise	S/RO	Informações de <i>Check Back</i> . Suportadas
35	SIMULATE	DS-50	0: Disable ; ≠0: Active are the Enable options.	Disable		S	Permite que o valor de readback seja fornecido manualmente quando o modo simula é habilitado. Neste caso o valor simulado e o estado serão o valor PV.
36	INCREASE_CLOS E	Unsigned8	0: Rising 1: Falling	0	E	S	Direção do posicionador em modo automático.
37	OUT	DS-33	OUT_SCALE		OUT	N/Man	O resultado do valor de saída para o bloco transdutor.
38	OUT_SCALE	DS-36		0-100%	Trd input	S	Valores de escala altos e baixos para o transdutor para um canal especificado.

Legenda: E – parâmetro Enumerado; NA – parâmetro Adimensional; RO –Somente Leitura; D – dinâmico; N – não-volátil; S – estático

Tabela 2.7 – Parâmetros do Bloco Saída Analógica

## Totalizador - TOT

### Geral

O bloco funcional Totalizador obtém os dados do bloco transdutor selecionados através dos números de canal e os totaliza ao longo do tempo. Este bloco é usado, normalmente, para totalizar vazão, ou seja, medir a massa ou o volume durante um certo período de tempo, ou totalizar a potência, medindo a energia total.

### Esquemático

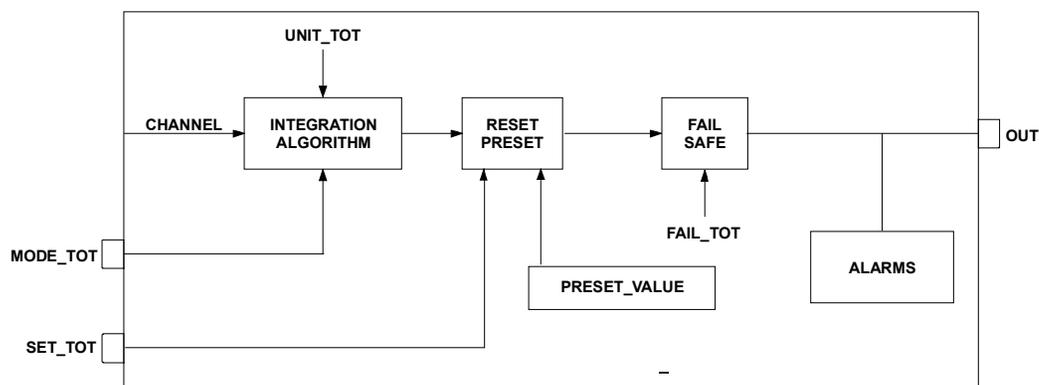


Figura 2.3 – Bloco Totalizador

### Descrição

O bloco funcional Totalizador integra uma variável (por exemplo: taxa de vazão ou alimentação) em função do tempo para a quantidade correspondida (por exemplo: volume, massa ou distância).

A unidade de medida do totalizador é fornecida pelo bloco transdutor. Internamente, as unidades de tempo são convertidas em unidades por segundo. Cada taxa, multiplicada pelo tempo de execução do bloco, fornece a quantidade de massa, volume e incremento de energia por execução do bloco.

O TOTAL é a quantidade totalizada. A unidade de engenharia usada na saída é o UNIT\_TOT. A unidade de saída deve ser compatível com a unidade da entrada, fornecida pelo transdutor através do canal. Portanto, se a taxa da entrada é vazão em massa (por exemplo: Kg/s, g/min, ton/h) a unidade de saída deverá ser massa (por exemplo: kg, g, ton, lb, etc.).

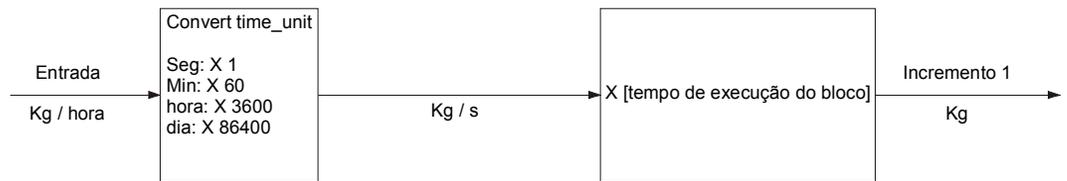


Figura 2.4 – Cálculo de incremento com taxa de entrada

## Totalização da Vazão

Para a distinguir a vazão positiva da negativa, o bloco totalizador adota um sinal negativo como indicador de vazão negativa.

O valor líquido de vazão é obtido somando os dois incrementos. O valor do incremento líquido terá um sinal positivo ou negativo, para indicar a direção da vazão. Para integrar a diferença entre o fluxo de entrada e o fluxo de saída, por exemplo, o segundo pode ser designado negativo. A direção da vazão a ser considerada na totalização é definida em MODE\_TOT. As seguintes opções estão disponíveis:

- *Positive only* – somente vazão positiva é totalizada. Os valores negativos serão admitidos como zero;
- *Negative only* – somente a vazão negativa é totalizada. Os valores positivos serão admitidos como zero;
- *Balanced* – Os valores negativos e positivos serão totalizados;
- *Hold* – O valor totalizado será constante.

## Reset e Preset

O parâmetro especial SET\_TOT é usado para resetar ou presetar a integração. Este parâmetro é sensível a nível. Enquanto o SET\_TOT está em "Reset", a saída será zero. E enquanto o SET\_TOT é "Preset", a saída será o valor do parâmetro PRESET\_TOT. Portanto, a integração só é realizada quando o parâmetro volta para "Totalize".

## Iniciando o Bloco

Para inicializar o bloco totalizador para que o funcionamento seja apropriado, os seguintes passos devem ser seguidos:

- O Parâmetro Canal deve ser ajustado para PV;
- O parâmetro LINEARIZATION\_TYPE do bloco transdutor deve ser ajustado para "Square Root";
- O parâmetro PRIMARY\_VALUE\_UNIT do bloco transdutor deve ser ajustado para uma unidade válida de fluxo;
- O parâmetro UNIT\_TOT do bloco TOT deve ser ajustado para uma unidade de massa ou volume equivalente à unidade de entrada.

## Modos Suportados

O/S, AUTO

## Cíclico – CFG\_DATA

Configuração suportada	Configuração Longa (Formato Identificador estendido)
TOTAL	0x41,0x84,0x85
TOTAL / SET_TOT	0xC1,0x80,0x84,0x85
TOTAL / SET_TOT / MODE_TOT	0xC1,0x81,0x84,0x85

Tabela 2.8 – Cíclico – CFG\_DATA

## Parâmetros

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa Válida/ Opções	Valor Padrão	Unidades	Armaze- namento/ Modo	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Este parâmetro será incrementado sempre que ocorrer mudanças nos parâmetros estáticos do bloco.
2	TAG_DESC	OctString(32)		Espaços	Na	S	Título do bloco. Este parâmetro deve ser único na configuração.
3	STRATEGY	Unsigned16		0	Nenhuma	S	É um valor fornecido pelo usuário para identificar uma configuração.
4	ALERT_KEY	Unsigned8	1 to 255	0	Nenhuma	S	
5	TARGET_MODE	Unsigned8	O/S, AUTO.	AUTO	Nenhuma	S	Contém o modo desejado para o bloco.
6	MODE_BLK	DS-37			Na	D/RO	Ver parâmetros de Modo.
7	ALARM_SUM	DS-42		0,0,0,0	Nenhum	D/RO	Contém os estados dos alarmes de bloco.
8	BATCH	DS-67		0,0,0,0	Nenhum	S	Para uso em sistema fieldbus distribuído para identificar canais usados e disponíveis. Não há algoritmo relacionado.
10	TOTAL	DS-33				N/RO	Em modo manual este parâmetro deve ser do tipo READ/WRITE.
11	UNIT_TOT	Unsigned16			OUT	S	Unidade de engenharia da saída.
12	CHANNEL	Unsigned16		0	Nenhum	S	O número do canal lógico de hardware do transdutor que é conectado neste bloco I/O.
13	SET_TOT	Unsigned8	0: Totalize 1: Reset 2: Preset	Totaliza	E	N	Reinicia a saída do totalizador, ou ajusta a saída para PRESET_TOT. É sensível à nível.
14	MODE_TOT	Unsigned8	0: Balanced 1: Positive only 2: Negative only 3: Hold	Balanceado	E	N	Define o tipo de contagem (positiva ou negativa ou mantém o último valor) da integração.
15	FAIL_TOT	Unsigned8	0: Run 1: Hold 2: Memory	Run	E	S	Define o procedimento na condição de fail safe.
16	PRESET_TOT	Float		0	OUT	S	Valor da saída quando o SET_TOT está em Preset.
17	ALARM_HYS	Float		0	OUT	S	Parâmetro de histerese de alarme. Para limpar o alarme, o valor da PV deve retornar inferior ao limite do alarme somada à histerese.
18	HI_HI_LIM	Float		INF	OUT	S	Ajuste de alarme muito alto em unidades de Engenharia.

Idx	Parâmetro	Tipo de Dado (comp.)	Faixa Válida/ Opções	Valor Padrão	Unidades	Armazenamento/ Modo	Descrição
19	HI_LIM	Float		INF	OUT	S	Ajuste de alarme alto em unidades de Engenharia.
20	LO_LIM	Float		INF	OUT	S	Ajuste de alarme baixo em unidades de Engenharia.
21	LO_LO_LIM	Float		INF	OUT	S	Ajuste de alarme muito baixo em unidades de Engenharia.
22	HI_HI_ALM	DS-39				D	Estado de alarme muito alto.
23	HI_ALM	DS-39				D	Estado de alarme alto.
24	LO_ALM	DS-39				D	Estado de alarme baixo
25	LO_LO_ALM	DS-39				D	Estado de alarme muito baixo.

Legenda: E – parâmetro Enumerado; NA – parâmetro Adimensional; RO –Somente Leitura; D – dinâmico; N – não-volátil; S – estático

Tabela 2.9 – Parâmetros do Bloco Totalizador

## Saída Discreta – DO

### Geral

O bloco Saída Discreta é um Bloco Funcional usado, por exemplo, em válvulas discretas, saídas a relé, saída de transistores etc.

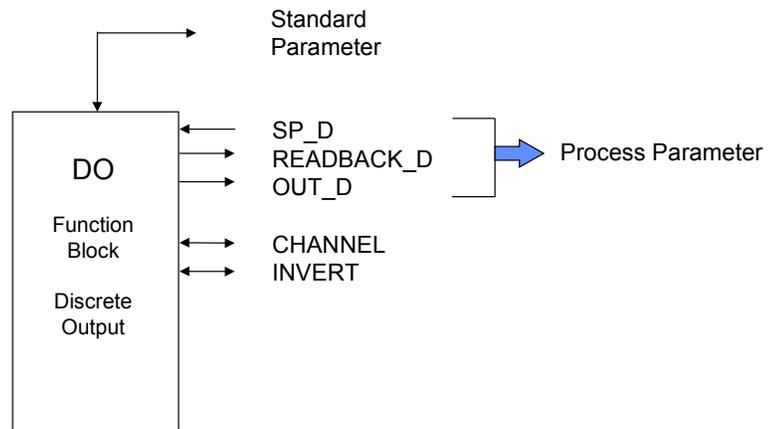


Figura 2.5 – Bloco de Saída Discreta

O diagrama esquemático deste bloco é mostrado na Figura 2.6

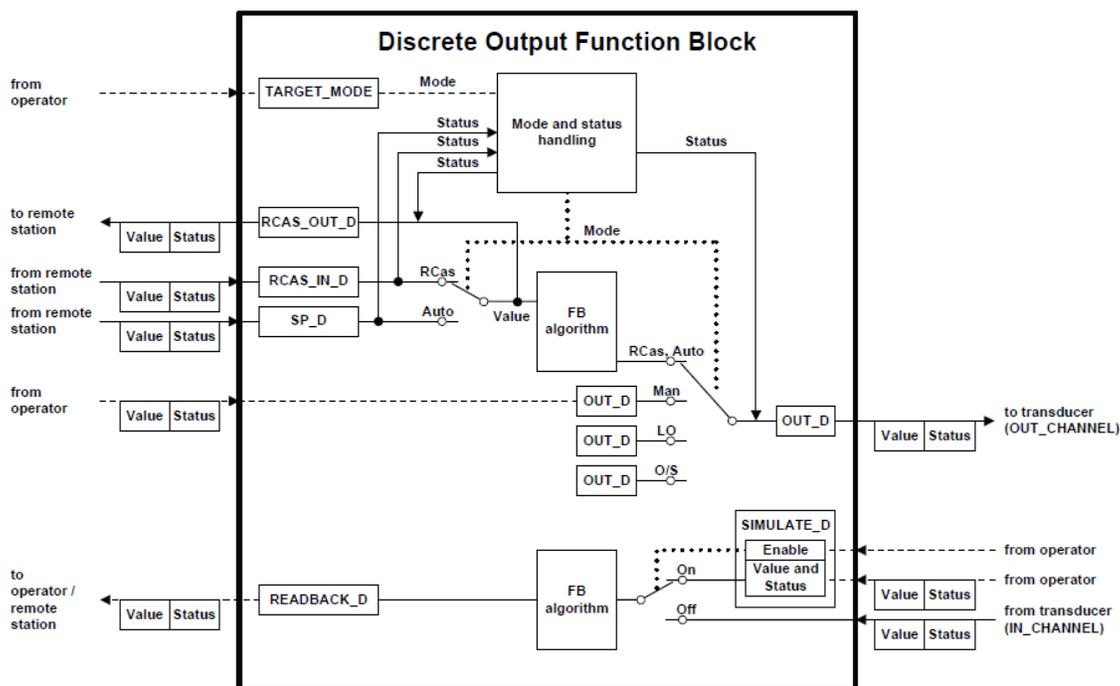


Figura 2.6 – Diagrama Esquemático do Bloco de Função Saída Discreta

## Descrição

O Bloco Transducer recebe o valor de OUTD\_D que vem do DO\_Block. O Modo *Target* é configurado pelo operador de acordo com o permitido pelo *Mode Block*. O bloco DO recebe o valor do setpoint discreto do *Host* de acordo com *Mode Block*.

## Simulação

O parâmetro SIMULATE\_D é usado para diagnóstico e propósitos de saída. Quando ele está ativado, o valor de readback e o status serão substituídos pelo valor simulado e status. A estrutura do SIMULATE\_D é composta pelos seguintes atributos:

- Valor simulado e Status;
- Simulação habilitada.

Quando a simulação está habilitada, o bloco Transducer e o bloco DO serão desconectados.

## Modos Suportados

O/S, MAN, RCAS, LO e AUTO.

## Status Handling

O bloco DO suporta modo cascata. Então, o status saída tem um sub-status cascata.

## Cíclico CFG\_DATA

Bloco de Função	Parâmetro	Byte Identificador	Formato Identificador Estendido
DO block	SP_D / READBACK_D		0xC1,0x81,0x81,0x83
	SP_D / CHECK_BACK_D		0xC1,0x81,0x82,0x92
	SP_D / READBACK_D / CHECK_BACK_D		0xC1,0x81,0x84,0x93
	RCAS_IN_D / RCAS_OUT_D		0xC1,0x81,0x81,0x8C
	RCAS_IN_D / RCAS_OUT_D / CHECK_BACK_D		0xC1,0x81,0x84,0x9C
	SP_D / READBACK_D / RCAS_IN_D / RCAS_OUT_D / CHECK_BACK_D		0xC1,0x83,0x86,0x9F

Tabela 2.10 - Cíclico CFG\_DATA

## Parâmetros

Parâmetro	Descrição
IN_CHANNEL	Se refere à ativação do bloco Transducer e seus parâmetros que fornecem os valores discretos ao elemento de controle final.
CHECKBACK	Informação detalhada do equipamento, bit codificado. Mais de uma mensagem ao mesmo tempo.
CHECK_BACK_MASK	Definição de bits de informação suportados pelo CHECK_BACK 0 = não suportado 1 = suportado
FSAFE_TIME	Período da detecção da falha (SP_D =Mau ou RCAS_IN <> Bom) até a ação do bloco, se a condição existir.
FSAFE_TYPE	Define a reação do equipamento, se a falha do setpoint real utilizado ainda é detectada após FSAFE_TIME ou se o seu o status de referência usado é Initiate Fail Safe. O cálculo de ACTUAL MODE é AUTO respectivamente. 0 = valor FSAVE_VALUE é usado como status do setpoint do OUT_D = UNCERTAIN – Valor substituído; 1 = armazenar o status do último setpoint válido do OUT_D = UNCERTAIN – Último valor usável ou BAD – sem comunicação, no último valor válido; 2 = atuador vai para posição de fail-safe definida pelo ACTUATOR_ACTION, status de OUT_D = BAD – não especificado
FSAFE_VAL_D	OUT_D é usado se FSAFE_TYPE = 0 e FSAFE está ativado.
INVERT	Indica se SP_D deverá ser logicamente invertido antes de escrever OUT_D no modo AUTO ou RCAS. Lista de valores válidos: 0 = não inverter 1 = inverter
OUT_D	Este parâmetro é a variável de processo do bloco de saída discreta em AUTO e modo RCAS e é o valor especificado pelo operador/engenheiro em MAN e LO.
READBACK_D	Pode ser a posição atual do elemento final de controle e seus sensores.
RCAS_IN_D	Target Setpoint e status fornecido pelo Host supervisor para o bloco de Saída Discreta usado em MODE RCAS.
RCAS_OUT_D	Setpoint do bloco de função e status fornecido por um Host supervisor para monitoração / cálculo para permitir a ação a ser tomada em condições limitadas ou alterar o modo.
SIMULATE	Para comissionamento e razões de manutenção, é possível simular o READBACK, definindo o valor eo status. Isso significa que o bloco Transducer e o bloco OD serão desligados.

Parâmetro	Descrição
SP_D	Setpoint do bloco de função usado em MODE AUTO.

Tabela 2.11 – Parâmetros do Bloco de Função de Saída Discreta

### Atributos dos Parâmetros do Bloco de Função de Saída Discreta

A Tabela 2.12 apresenta, de forma geral, os parâmetros e seus atributos definidos na classe B.

Idx	Parâmetro	Tipo de Objeto	Tipo de Dado	Arma-zena-mento	Ta-ma-nho	Acesso	Tipo de Transporte	Valor Default
9	SP_D	Record	DS-34	D	2	r,w	l/a, cyc	-
10	OUT_D	Record	DS-34	D	2	r,w	C/a	-
12	READBACK_D	Record	DS-34	D	2	r	O/a, cyc	-
14	RCAS_IN_D	Record	DS-34	D	2	r,w	l/a, cyc	-
17	CHANNEL	Simple	Unsigned16	S	2	r,w	C/a	-
18	INVERT	Simple	Unsigned 8	S	1	r,w	C/a	0
19	FSAVE_TIME	Simple	Float	S	4	r,w	C/a	0
20	FSAVE_TYPE	Simple	Unsigned 8	S	1	r,w	C/a	2
21	FSAVE_VAL_D	Simple	Unsigned 8	S	1	r,w	C/a	0
22	RCAS_OUT_D	Record	DS-34	D	2	r	O/a, cyc	-
24	SIMULATE	Record	DS-51	S	3	r,w	C/a	disable
33	CHECK_BACK	Simple	OctetString	D	3	r	C/a, cyc	-
34	CHECK_BACK_MASK	Simple	OctetString	Cst	3	r	C/a	-
35-44	reserved by PNO							
45	first manufacture specific parameter							

Tabela 2.12 - Atributos dos Parâmetros do Bloco de Função de Saída

## Entrada Discreta – DI

### Geral

O bloco Saída Discreta é um Bloco Funcional usado, por exemplo, indutivos, ópticos, capacitivos, ultrassônicos e chaves de proximidade.

O bloco de Entrada Discreta leva os dados da Entrada Discreta para o bloco Transducer, selecionando via número do canal e torna-lo disponível para outros blocos e suas entradas.

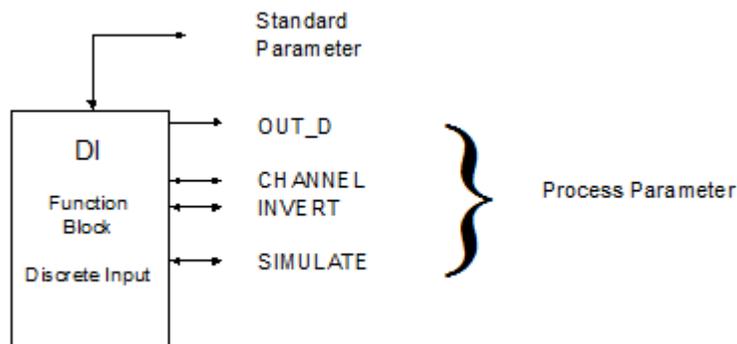


Figura 2.7 – Bloco de Entrada Discreta

O diagrama esquemático é mostrado na Figura 2.8.

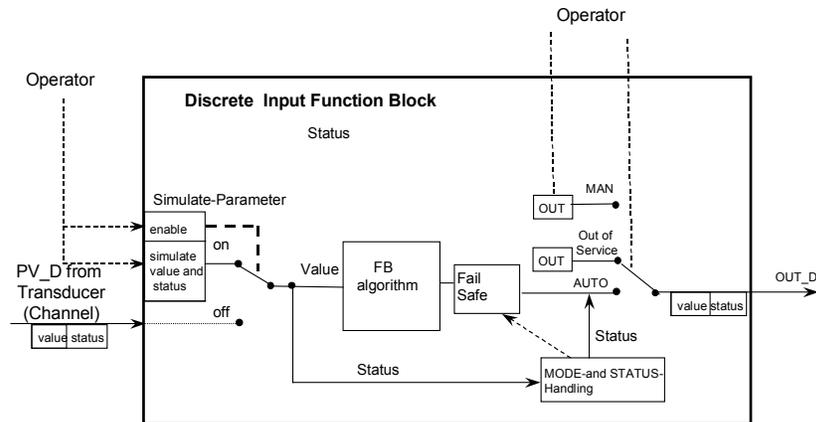


Figura 2.8 – Diagrama Esquemático do Bloco de Função Entrada Discreta

## Descrição

O status da variável de processo do bloco Transducer está visível ao bloco Transducer. O Modo Target é configurado pelo operador de acordo com o que é permitido pelo Mode Block.

O Status (OUT\_D) é acoplado com o valor OUT\_D a partir do bloco.

## Simulação

O parâmetro SIMULATE\_D é usado para diagnóstico e propósitos de saída. Quando ele está ativado, o valor de readback e o status serão substituídos pelo valor simulado e status. A estrutura do SIMULATE\_D é composta pelos seguintes atributos:

- Valor simulado e Status;
- Simulação habilitada.

Quando a simulação está habilitada, o parâmetro Transducer Input será calculado baseado no atributo Simulate Value/Status do parâmetro SIMULATE\_D. Caso contrário, será fornecido pelo bloco Transducer.

## Modos Suportados

O/S, MAN e AUTO.

## Status Handling

O bloco DI não suporta o modo cascata. Então, o status de saída não terá o sub-status cascata.

## Cíclic CFG\_DATA

Bloco de Função	Parâmetro	Byte Identificador	Formato Identificador Estendido
Discrete Input (DI)	OUT_D	0x91	

Tabela 2.13 – Cíclico CFG\_DATA

## Parâmetros

Parâmetros	Descrição
CHANNEL	Se refere à ativação do bloco Transducer que fornece o valor medido no Bloco de Função.
INVERT	Indica se o valor de entrada de PV_D deverá ser logicamente invertido antes de armazenar em OUT_D. Lista de valores válidos : 0 = not inverted 1 = invert
OUT_D	É a saída do bloco de função. O valor é especificado pelo operador em MODE MAN.
FSAFE_TYPE	Define a reação do equipamento, se a falha é detectada. 0 = valor FSAVE_VALUE é usado como OUT_D Status – UNCERTAIN_Initial Value; 1 = armazenar o último valor de OUT_D value Status - UNCERTAIN_LastUsableValue , se for inválido; 2 = OUT_D teve o valor calculado errado e o status = BAD
FSAFE_VALUE	Valor default para o parâmetro OUT_D, se a falha é detectada.
SIMULATE	Para propósitos de comissionamento e teste o valor de entrada do bloco Transducer no bloco de função Saída Discreta pode ser modificado. Assim, os blocos Transducer e DI serão desconectados.

Tabela 2.14 – Descrição dos Parâmetros do bloco Saída Discreta

**Atributos dos Parâmetros do Bloco de Função de Saída Discreta**

Relative Index	Parameter Name	Object type	Data type	Store (S;D;N)	Size	Access	Param. Usage/ Kind of Transport	Default Values
Function Block Parameter for Discrete Input DI								
10	OUT_D	Record	DS-34	D	2	r,w	O/cyc	measured of the value, state
14	CHANNEL	Simple	Unsigned16	S	2	r,w	C/a	-
15	INVERT	Simple	Unsigned 8	S	1	r,w	C/a	0
20	FSAVE_TYPE	Simple	Unsigned 8	S	1	r,w	C/a	1
21	FSAVE_VAL_D	Simple	Unsigned 8	S	1	r,w	C/a	0
24	SIMULATE	Record	DS-51	S	3	r,w	C/a	disable
25-34	reserved by PNO							
35	first manufacture specific parameter							

**Tabela 2.15 - Atributos dos Parâmetros do Bloco de Função de Saída Discreta**

## Descrições de BitStrings

### DIAGNÓSTICOS (BLOCO *PHYSICAL*)

Bit	Mnemônico	Significado
0	DIA_HW_ELECTR	Falha de hardware dos componentes eletrônicos.
1	DIA_HW_MECH	Falha de hardware dos componentes mecânicos.
2	DIA_TEMP_MOTOR	Temperatura do motor muito alta.
3	DIA_TEMP_ELECTR	Temperatura eletrônica muito alta.
4	DIA_MEM_CHKSUM	Erro de memória.
5	DIA_MEASUREMENT	Falha na medida.
6	DIA_NOT_INIT	Equipamento não inicializado.
7	DIA_INIT_ERR	Falha de Auto-Calibração.
8	DIA_ZERO_ERR	Erro de ponto Zero.
9	DIA_SUPPLY	Falha da Fonte de Alimentação.
10	DIA_CONF_INVALID	Configuração Inválida.
11	DIA_WARM_START	Reinício em andamento (Alimentado). Este bit vai para verdadeiro quando alimentado e será automaticamente reiniciado após 10 segundos.
12	DIA_COLD_START	Nova inicialização em andamento (Inic. De fábrica). Este bit vai para verdadeiro quando a inicialização de fábrica ocorre e será automaticamente reiniciado após 10 segundos.
13	DIA_MAINTAINANCE	Manutenção requisitada.
14	DIA_CHARACTER	Caracterização Inválida.
15	IDENT_NUMBER_VIOLATION	Vai para 1 se o Ident_Number do dado cíclico for transferido e o valor do parâmetro do bloco físico IDENT_NUMBER_SELECTOR forem diferentes.
37	EXTENSION AVAILABLE	Mais informação de diagnóstico está disponível. A informação deste diagnóstico estará em DIAGNOSIS_EXT.

Nota: Para mais detalhes sobre os parâmetros, consulte o manual específico de cada equipamento.

**Tabela 2.16 – Diagnóstico no Bloco Physical**

**CHECK\_BACK (BLOCO DE SAÍDA ANALÓGICA)**

Bit	Mnemônico	Significado	CHECK_BACK_MASK
0	CB_FAIL_SAFE	Equipamento no campo com <i>Fail Safe</i> ativo.	Usado
1	CB_REQ_LOC_OP	Requisita operação local.	Não usado
2	CB_LOCAL_OP	Equipamento de campo sob controle local, chave <i>LOCKED OUT</i> em funcionamento.	Não usado
3	CB_OVERRIDE	Emergência override ativo.	Não usado
4	CB_DISC_DIR	Posição de realimentação diferente da esperada.	Usado
5	CB_TORQUE_D_OP	Limite de torque na direção <i>OPEN</i> excedido.	Não usado
6	CB_TORQUE_D_CL	Limite de torque na direção <i>CLOSE</i> excedido.	Não usado
7	CB_TRAV_TIME	Indica o estado de monitoramento do curso do equipamento, caso esteja em <i>YES</i> , o tempo do curso do atuador excedeu-se.	Usado
8	CB_ACT_OPEN	Atuador se move na direção de abertura.	Usado
9	CB_ACT_CLOSE	Atuador se move na direção de fechamento.	Usado
10	CB_UPDATE_EVT	Alerta gerado por mudança em dados estáticos (bloco funcional e Bloco Transdutor).	Usado
11	CB_SIMULATE	Simulação de valores de processo habilitados.	Usado
13	CB_CONTR_ERR	Malha de controle interna interrompida.	Não usado
14	CB_CONTR_INACT	Posicionador inativo (OUT status = BAD).	Usado
15	CB_SELFTEST	Equipamento em autoteste.	Usado
16	CB_TOT_VALVE_TRAV	Indica que o limite total de percurso da válvula está excedido.	Usado
17	CB_ADD_INPUT	Indica que uma entrada adicional (por exemplo: para diagnóstico) está ativa.	Usado

Nota: Para mais detalhes sobre os parâmetros, consulte o manual específico de cada equipamento.

Tabela 2.17 – Check Back no Bloco de Saída Analógico

**Configurações e Disponibilidades de FB**

Ordem no Diretório	Blocos	LD		TT		DT		IF		FI		TP		FY		FP		LD293	
		FB tipo	Qtt																
1	PHY	Sim	1																
2	AO									Sim	3			Sim	1	Sim	1		
2	AI	Sim	1	Sim	2	Sim	1	Sim	3			Sim	1					Sim	1
3	TOT	Sim	1					Sim	3			Sim	1						
4	TRD-LD	Sim	1																
4	TRD-TP											Sim	1						
4	TRD-TT			Sim	2														
4	TRD-IF							Sim	3										
4	TRD-FI									Sim	3								
4	TRD-FP																Sim	1	
4	TRD-FY													Sim	1				
4	TRD-DT					Sim	1												
4	TRD-LD293																	Sim	1
5	TRD-DSP	Sim	1	Sim	1			Sim	1										

Nota 1 – A coluna “FB type” indica qual *Bloco Funcional* está disponível para cada tipo de equipamento.

Nota 2 – A coluna “Qtt” indica o número de *Blocos Funcionais* instanciada na inicialização de fábrica para cada tipo de equipamento.

Nota 3 – A coluna “Order into the directory” indica a ordem interna dos blocos no equipamento. (Esta informação é útil na configuração cíclica–CFG\_data config). Alguns blocos possuem o mesmo número que outros, pois o equipamento não possui os *Blocos Funcionais* AO e AI e o transdutor é específico para cada equipamento.

Tabela 2.18 – Configurações e Disponibilidade FB