

# smar - PROFIBUS

FIRST IN FIELD BUS

AGO / 02



## MANUAL DE INSTRUÇÕES DOS BLOCOS DE FUNÇÃO



# smar

**web: [www.smar.com.br](http://www.smar.com.br)**

**Especificações e informações estão sujeitas a modificações sem prévia consulta.  
Para atualizações mais recentes veja o site da smar acima.**

## **BRASIL**

**Smar Equipamentos Ind. Ltda.**  
Rua Dr. Antonio Furlan Jr., 1028  
Sertãozinho SP 14170-480  
Tel.: +55 16 3946-3599  
Fax: +55 16 3946-3554  
e-mail: insales@smar.com.br

## **ALEMANHA**

**Smar GmbH**  
Rheingaustrasse 9  
55545 Bad Kreuznach  
Germany  
Tel.: + 49 671-794680  
Fax: + 49 671-7946829  
e-mail: infoservice@smar.de

## **EUA**

**Smar International Corporation**  
6001 Stonington Street, Suite 100  
Houston, TX 77040  
Tel.: +1 713 849-2021  
Fax: +1 713 849-2022  
e-mail: sales@smar.com

## **CHINA**

**Smar China Corp.**  
3 Baishiqiao Road, Suite 30233  
Beijing 100873, P.R.C.  
Tel.: +86 10 6849-8643  
Fax: +86-10-6894-0898  
e-mail: info@smar.com.cn

## **MEXICO**

**Smar Mexico**  
Cerro de las Campanas #3 desp 119  
Col. San Andrés Atenco  
Tlalnepantla Edo. Del Méx - C.P. 54040  
Tel.: +53 78 46 00 al 02  
Fax: +53 78 46 03  
e-mail: ventas@smar.com

## **FRANÇA**

**Smar France S. A. R. L.**  
42, rue du Pavé des Gardes  
F-92370 Chaville  
Tel.: +33 1 41 15-0220  
Fax: +33 1 41 15-0219  
e-mail: smar.am@wanadoo.fr

## **CINGAPURA**

**Smar Singapore Pte. Ltd.**  
315 Outram Road  
#06-07, Tan Boon Liat Building  
Singapore 169074  
Tel.: +65 6324-0182  
Fax: +65 6324-0183  
e-mail: info@smar.com.sg

## **HOLANDA**

**Smar Nederland**  
De Oude Wereld 116  
2408TM Alphen aan den Rijn  
Tel: +31 172 494 922  
Fax: +31 172 479 888  
e-mail : info@smarnederland.nl

## **REINO UNIDO**

**Smar UK Ltd**  
3, Overhill Road - Cirencester  
Gloucestershire -  
GL7 2LG  
Tel: +44 (0)797 0094138  
Fax: +44 (0)797 4747502  
e-mail: info@smarUK.co.uk

## **Smar Research Corporation**

4250 Veterans Memorial Hwy. Suite 156  
Holbrook , NY 11741  
Tel: +1-631-737-3111  
Fax: +1-631-737-3892  
e-mail: sales@smarresearch.com

# ÍNDICE

<b>INTRODUÇÃO PARA APLICAÇÃO DO BLOCO DE FUNÇÃO .....</b>	<b>1.1</b>
Geral .....	1.1
Bloco de função .....	1.1
Bloco Transdutor.....	1.1
Bloco físico .....	1.1
Definições Bloco de função .....	1.1
Troca de Dados cíclicos .....	1.2
Dados de configuração (CFG-DADOS) .....	1.3
Exemplo de Configuração .....	1.5
Acesso a Informações .....	1.5
<b>ESTRUTURA DA APLICAÇÃO DO BLOCO DE FUNÇÃO .....</b>	<b>1.6</b>
Objeto do Bloco .....	1.6
Parâmetros dos Blocos.....	1.6
parâmetro dos Identificadores .....	1.6
Uso do parâmetro .....	1.6
Contained.....	1.7
Saída.....	1.7
Entrada .....	1.7
Relações de parâmetros.....	1.8
Estado do parâmetro .....	1.8
Composição de Estado.....	1.9
Exemplo: conversão das enumerações para status .....	1.10
Exemplo: conversão de número para enumerações .....	1.11
Manipulação do canal .....	1.11
Canal dos Blocos de Função de Entrada .....	1.11
Canal dos Blocos de Função de Saída.....	1.11
Exemplo de uso do Canal.....	1.12
Cálculo de saída .....	1.12
Controle em Cascata Remoto.....	1.13
Parâmetro do modo .....	1.14
a) Modo de Operação.....	1.14
b) Elementos do Modo de um Bloco Funcional .....	1.15
c) Prioridade de Modo.....	1.15
d) Modo de Cálculo.....	1.16
e) Informação Específica para desenvolvedores de drivers .....	1.16
Parâmetros de escala.....	1.17
Exemplo Usando Parâmetros de Escala .....	1.17
Manipulação em Fail Safe (Falha Segura) .....	1.18
Condições que ativam a Fail Safe .....	1.18
Ações de Fail Safe (Falha Segura).....	1.18
Diagnóstico .....	1.19
Diagnóstico de Características do Equipamento .....	1.19
Diagnóstico do Actuator.....	1.19
Processando um Alerta – Alarmes e Eventos .....	1.19
a) Parâmetro de Alarme (parâmetro X_ALM) .....	1.19
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) Índice de alarme (parâmetro ALM_SUM) .....	1.20
f) Atualização do Evento (Update Event).....	1.20
Definição e Tipos de Estrutura de Dados .....	1.21
Objeto do Bloco-DS-32 .....	1.21
Valor & Estado-Estrutura de Ponto Flutuante-DS-33.....	1.21
Valor & Estado-Estrutura Discreta-DS-34 .....	1.22
Estrutura Escalar-DS-36 .....	1.22
Estrutura do modo-DS-37 .....	1.22
Estrutura de Alarme flutuante-DS-39 .....	1.22
Estrutura de Alarme Discreta-DS-40.....	1.22
Estrutura de Atualização do Alarme-DS-41 .....	1.23

Estrutura Índice de Alarme–DS-42 .....	1.23
Simula–Estrutura do Ponto Flutuante–DS-50 .....	1.23
Simula–Estrutura Discreta–DS-51 .....	1.23
Estrutura de grupo–DS-67 .....	1.23
<b>BIBLIOTECA DE BLOCO .....</b>	<b>2.1</b>
<b>BLOCO FÍSICO - PHY.....</b>	<b>2.1</b>
Descrição .....	2.1
Parâmetro FACTORY_RESET .....	2.1
Memória Não-volátil.....	2.1
Bloqueio de gravação via software .....	2.1
Diagnósticos .....	2.1
Seletor do Número identificador.....	2.2
Modos Suportados .....	2.2
Parâmetros .....	2.2
<b>ENTRADA ANALÓGICA - AI.....</b>	<b>2.4</b>
Geral .....	2.4
Diagrama .....	2.4
Descrição .....	2.4
Simulação .....	2.4
Modos Suportados .....	2.4
Manuseio de Estados.....	2.4
Cíclico - CFG_DATA .....	2.5
Parâmetros .....	2.5
<b>SAÍDA ANALÓGICA - AO.....</b>	<b>2.6</b>
Geral .....	2.6
Diagrama .....	2.6
Descrição .....	2.6
Tratamento de Valores de Entrada.....	2.6
Tratamento de Valores de Saída .....	2.6
Aumentar para fechar (Increase to Close).....	2.7
Simulação .....	2.7
Parâmetros de Readback .....	2.7
Modos Suportados .....	2.7
Cíclico - CFG_DATA .....	2.8
Parâmetros .....	2.8
<b>TOTALIZADOR - TOT .....</b>	<b>2.10</b>
Geral .....	2.10
Diagrama .....	2.10
Descrição .....	2.10
Totalização da Vazão .....	2.10
Reset e Preset.....	2.11
Iniciando o Bloco .....	2.11
Modos Suportados .....	2.11
Cíclico - CFG_DATA .....	2.11
Parâmetros .....	2.12
<b>DESCRIÇÃO DE BITSTRINGS.....</b>	<b>2.13</b>
Diagnósticos (Bloco Físico).....	2.13
Check_Back - Bloco Analógico de Saída .....	2.13
<b>FB Ajustes e Disponibilidade de FB .....</b>	<b>2.14</b>

## **Introdução a Aplicação do Bloco de Função**

### **Geral**

As aplicações com os blocos de função são definidas como aplicações de planta ou fábrica, que executam um ou mais monitoramentos automático e funções de controle.

### **Bloco de função**

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

### **Bloco Transdutor**

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

### **Bloco Físico**

Os blocos físicos são usados para definir as características de hardware específicas das aplicações do bloco de função. Similar aos blocos transdutores, eles isolam os blocos de função do hardware físico por conterem uma configuração de parâmetros de hardware de implementações independentes.

### **Definições do Bloco de função**

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

Os Tags fornecem uma referência simbólica aos blocos de função. Eles não são ambíguos dentro de um sistema fieldbus. Os deslocamentos numéricos predeterminados são números atribuídos para otimizar o acesso aos blocos de função. Ao contrário do que faz os Tags do bloco de função, que são globais, os deslocamentos numéricos predeterminados têm significado somente dentro da aplicação que contém o bloco de função.

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

## Troca de dados cíclica

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

Em geral, um bloco de função 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 sobre os equipamentos consultando o arquivo GSD. Cada equipamento tem este arquivo o qual possui um número identificador único e que identifica o equipamento ao mestre. Este arquivo 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 mesmo suporta. Um exemplo de arquivo GSD é obtido na figura 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 - Exemplo do arquivo GSD**

## Dados de Configuração (CFG\_DATA)

Na troca de dados cíclica há um parâmetro diferente para os blocos de função (veja o Cyclic\_CFG\_DATA de cada bloco da função). As diferenças vêm das necessidades diferentes do usuário em relação a necessidade do espaço 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 concatenam um 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 da entrada analógico "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 "end\_module". 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 de função. 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 de função específico não participará da troca de dados cíclica.

**Para cada bloco de função que suportam 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).**

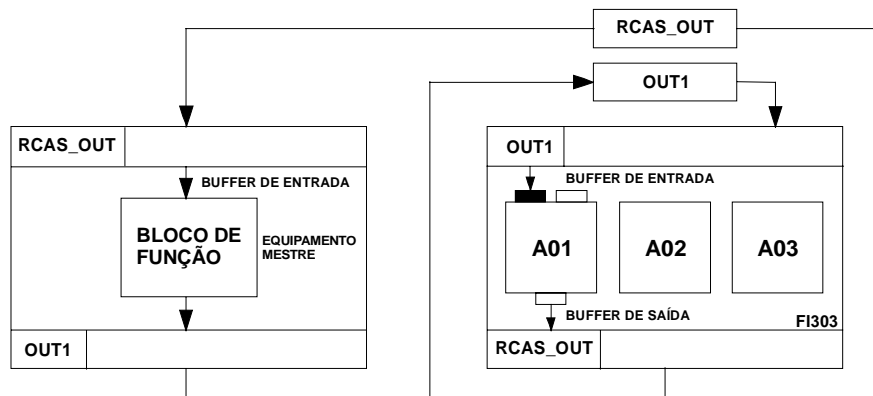
De acordo com o Profibus-PA profile 3.0, há dois tipos de configuração para uma combinação dos parâmetros. O byte Identifier (ou short identifier) e o Extended Identifier Format (ou long identifier). Algumas combinações tem somente um tipo de configuração e outras tem ambos. O equipamento da Smar suporta 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 combinar ambos tipos de configuração: longo (long) e curto (short).

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

**Todos os blocos de função 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).**

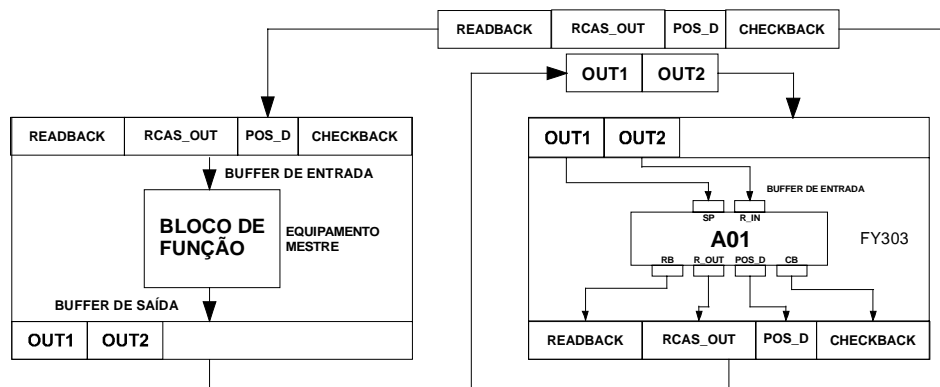
Por exemplo, no equipamento do LD303 que tem um um bloco de função AI e um TOT, a ordem para o "Config data" deve ser AI e TOT, respectivamente. No equipamento IF303 que tem 3 AI e 3 TOT, a ordem do "Config data" deve ser AI1, AI2, AI3, TOT1, TOT2, TOT3, respectivamente.

As Figuras abaixo ilustram as trocas de dados entre o equipamento mestre e o equipamento escravo. Na Figura 2, no equipamento FI303 (3 blocos AO) só o Primeiro AO é configurado (observe que a ordem da configuração precisa ser respeitada). Assim, na configuração, o usuário precisa indicar os blocos (AO2 e AO3) que não são usados. Para isto, configure-os para "empty module". Para o bloco AO1 escolha "RCAS\_IN + RCAS\_OUT" para a troca de dados, aonde os dados da saída do bloco de função do mestre vão para a entrada RCAS\_IN do equipamento escravo. E a saída RCAS\_OUT do bloco de função AO1 vai para a entrada do bloco de função do Mestre.



**Figura 2 – Exemplo da troca cíclica de dados - Dados da Configuração “Rcas\_In + Rcas\_Out” + “Empty\_Module” + “Empty\_Module” para equipamento FI303**

Na Figura 3, no equipamento FY303 (1 bloco AO), é escolhida a troca de dados “SP + RB + RCASIN + RCASOUT + POSD + 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 o OUT1 e o RCAS\_IN recebe o 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 3 – Exemplo da troca cíclica de dados - Dados de Configuração “SP + RB +RCASIN + RCASOUT + POSD + CB” para o equipamento FY303**



## Exemplo de Configuração

1) Considerando o transmissor LD303, que tem 2 blocos de função, 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:  
"Analog Input (long)" e "Total\_Settot" ou  
"0x00, 0xC1, 0x80, 0x84, 0x85"

2) Considerando o transmissor FI303 que tem 3 blocos de função, 3 blocos AO. As seguintes configurações são válidas:

- Configurando a entrada RcasIn, RcasOut 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 + POSD " e "SP" ou "0x00,0x96,0xA4,0xA4"
- Configurando a entrada RCASIN e a saída RCASOUT das 3 AOs (AO1,AO2 e AO3):  
"RcasIn + RcasOut", "RcasIn + RcasOut", "RcasIn + RcasOut" ou "0xB4,0xB4,0xB4"

### Notas:

- No exemplo 1, a ordem nos Blocos de Função 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 (curto)." 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 de função 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,POSD) para o segundo bloco, e "0xA4" (SP) para o terceiro bloco.
- O identificador curto ou longo dá o mesmo efeito.

## Acesso a Informação

A informação do Bloco de função 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 de função, 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 de função.

## ***Estrutura da Aplicação do Bloco de Função***

As aplicações dos blocos de função são modeladas como configurações dos blocos de função coordenados para executar uma configuração relacionada as operações.

O modelo do Bloco de função 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 de função que são modelados para fornecer ou obter o valor de processo com os blocos de função do equipamento controlador.

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

Para suportar a operação dos blocos de função, a arquitetura do bloco de função também fornece os blocos transducer, resource e objetos display.

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

## ***Objeto do Bloco***

Um objeto de bloco representa uma 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 rápida 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 de função 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 como 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 seu nome com um tag do seu 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 de função. Em geral, os parâmetros de saída contêm o estado. O estado 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 cuja finalidade 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 de função. Seu valor pode ser usado pelo algoritmo do bloco.

Em geral, os valores dos parâmetros de entrada são acompanhados pelo estado (status). Quando um parâmetro de entrada é conectado ciclicamente a um parâmetro de saída, os estados serão fornecidos como estado do parâmetro de saída (quando o parâmetro tem um estado). Quando ele não é conectado ciclicamente a um parâmetro de saída, os estados 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 de função responsáveis pela entrega dos dados configura o estado do parâmetro de entrada indicando uma 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 de função. A diferença entre os parâmetros de entrada não ciclicamente conectado e os parâmetros contidos é que os parâmetros de entrada têm capacidade para suportar uma conexão cíclica e os parâmetros contained não a têm.

Os blocos cujo fim 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 propósito de 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 composto 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 conseguir o modo target (modo desejado) estabelecido. 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 determinada circunstância operacional, um bloco pode não ser capaz de funcionar no modo requisitado. Em tais casos, o modo real reflete o modo capaz de consegui-lo. A comparação do real com o exibido na tela indica se o objetivo foi alcançado.

Os valores para o parâmetro modo para um bloco são definidos pelo Parâmetro Modo Permitido. Assim, os modos disponíveis para controlar um bloco, podem variar com cada bloco.

Assim que o modo real é determinado, a execução do bloco progride e as saídas são geradas.

## Estado do parâmetro

O argumento estado (status) da entrada e dos parâmetros de saída é feito para mostrar para os outros blocos o estado atual da fonte do bloco.

O campo de estado é composto de três partes: Qualidade, Sub-estados e Limites.

**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 e 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.

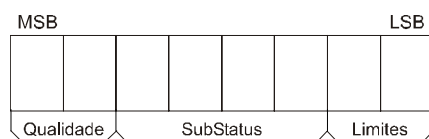
**Sub-status** – O sub-status é um complemento do estado da qualidade e leva a informação para inicializar ou parar um controle em cascata, alarmes e outros. Há diferentes configurações do sub-status para cada qualidade.

**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 (estado e valor) é recebida do barramento. Se a entrada não é “conectada”, então os estados e os valores podem ser configurados manualmente pelo usuário.

## Composição do Estado

O estado tem a seguinte composição:



Quality, sub-status e limits o componentes do estado são definidos como se segue:

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

- 0 = Ruim
- 1 = Incerto
- 2 = Bom (Não-cascata)
- 3 = Bom (Cascata)

**Sub-status-** Os valores do sub-status no status attribute são definidos conforme tabela 1.

**Limit** – As seguintes condições de limit sempre estarão disponíveis no estado atributo.

- 0 = Não limitada
- 1 = Baixo Limite
- 2 = Alto Limite
- 3 = Constante

Exemplos:

0xC1 (em hexadecimal) é o estado “Good-Cascade Non Specific and Low Limited”

0xCF(em hexadecimal) é o estado “Good-Cascade Not invited and Constant”

0x4E(em hexadecimal) é o estado “ Uncertain Initial Value and High Limited”

Qualidade	Sub-estado	Valor Hexa	Valor Decimal	Não em Cascata	Caminho de avanço da cascata	caminho de recuo da cascata
GoodNC	0 = ok (prioridade mais baixa)	0x80	128	X		
GoodNC	9 = requer manutenção	0xA4	164	X		
GoodNC	1 = Evento de atualização ativo	0x84	132	X		
GoodNC	2 = Alarme de alerta ativo	0x88	136	X		
GoodNC	3 = Alarme crítico ativo	0x8c	140	X		
GoodNC	4 = Atualização de evento não reconhecido	0x90	144	X		
GoodNC	5 = Alarme de alerta não reconhecido	0x94	148	X		
GoodNC	6 = Alarme crítico não reconhecido	0x98	152	X		
GoodNC	8 = início de Fail-Safe (IFS)	0xA0	160	X		
Uncertain	0 = Não especificado	0x40	64	X		
Uncertain	1 = Último valor utilizável	0x44	68	X		
Uncertain	2 = Substituto	0x48	72	X		
Uncertain	3 = Valor inicial	0x4c	76	X		
Uncertain	4 = Conversão de sensor sem precisão	0x50	80	X		
Uncertain	5 = Violada faixa de Unidades de Engenharia	0x54	84	X		

Qualidade	Sub-estado	Valor Hexa	Valor Decimal	Não em Cascata	Caminho de avanço da cascata	caminho de recuo da cascata
Uncertain	6 = Sub-normal	0x58	88	X		
Uncertain	7 = Erro de configuração	0x5C	92	X		
Uncertain	9 = Calibração de sensor	0x64	100	X		
Uncertain	8 = Valor simulado	0x60	96	X		
GoodC	0 = ok	0xC0	192		X	X
GoodC	1 = Reconhece inicialização (IA)	0xC4	196		X	
GoodC	2 = Requisita inicialização (IR)	0xC8	200			X
GoodC	3 = NÃO Convidado (NI)	0xCC	204			X
GoodC	5 = Não Selecionado (NS)	0xD0	208		X	
GoodC	6 = Override(LO local)	0xD8	216			X
GoodC	8 = Iniciada Fail Safe (IFS)	0xE0	224		X	
Bad	0 = Não-específico	0x00	0	X	X	X
Bad	1 = Erro de configuração	0x04	4	X	X	X
Bad	2 = Não Conectado	0x08	8			
Bad	4 = Falha do sensor	0x10	16	X	X	X
Bad	3 = Falha do equipamento	0x0C	12	X	X	X
Bad	5 = Nenhuma comunicação, com último valor utilizável	0x14	20			
Bad	6 = Nenhuma comunicação, com o valor utilizável	0x18	24			
Bad	7 = Fora de Serviço (a mais alta prioridade)	0x1c	28			

### **Exemplo: conversão das enumerações para status.**

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

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

Por exemplo, considerando o seguinte estado:

“Uncertain - Initial Value - High Limited”

onde:

Quality = “Uncertain” = 1

Sub-estado = “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 estado. Abaixo, são mostradas duas formas para fazê-lo:

1) Expressando o número em binário.

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

Dividindo este número binário em Quality, Sub-Status e Limit fields:

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

## Manipulação do canal

A função bloco de entrada analógica, saída analógica e o totalizador são conectados ao bloco do transdutor através do CANAL.

O CANAL é um parâmetro unsigned16 e seu valor representa um ponteiro relacionado ao bloco transdutor e a seu parâmetro. Consiste em 2 elementos:

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

Opcionalmente, o canal pode ser desconectado ajustando seu valor em zero (0x00).

## Canal dos Blocos de Função de Entrada

Tipicamente, em um bloco transdutor de um equipamento, o transmissor tem três parâmetros que podem ser conectados com o bloco de função de entrada: Valor Primário (PV), Valor Secundário1 (SV1), Valor Secundário 2 (SV2).

O bloco de função conectado ao transdutor pode ser conectado com cada uma destas saídas. Embora ele dependa do equipamento e do Bloco de Função. Há algumas regras descritas abaixo:

- O bloco Totalizador só pode ser configurado com o parâmetro PV.
- Tipos diferentes de blocos de função (por exemplo, AI ou TOT), podem ser configurados com o mesmo valor de canal, mas blocos de função do mesmo tipo (por exemplo, AI e AI ou TOT e TOT) não podem referenciar ao mesmo transducer ID no canal
- O índice relativo dos parâmetros de saída do transdutor (PV, SV1, SV2) é diferente para cada tipo de aparelho.

## Canal dos blocos de função de Saída

O Transdutor de um equipamento atuador só tem um parâmetro de referência, e o Índice Relativo do bloco de Função 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, está utilizando 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 da Smar não são necessários 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 de função: 1 AI e 1 TOT. Os possíveis canais destes Blocos de Função podem ser:

1. Há apenas um bloco transdutor, portanto o primeiro byte é 1.
2. 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 de 18(0x12 relativo).
3. No bloco AI, o segundo byte do canal pode ser PV, SV1 ou SV2 (índice relativo SV1 = 29 (0x1D), índice relativo SV2 = 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 FI 303

Considerando que o equipamento FI 303 possui 3 transdutores e 3 blocos de função AO. Os possíveis canais deste FBs podem ser:

1. 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).
2. 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 de função. Se o equipamento está em modo "manual", a saída é seguida de um valor fornecido pelo usuário (Man ou LO).

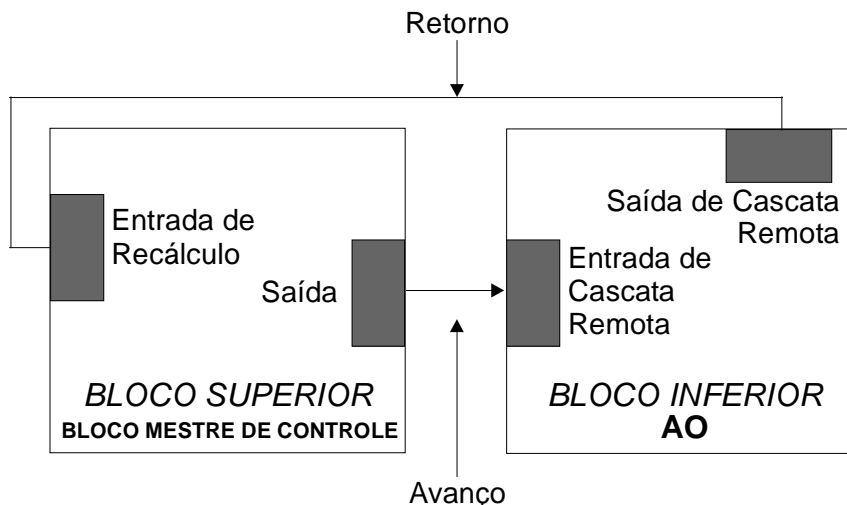
Em modo manual o estado da saída será de "Uncertain Simulate Value" indicando que a saída está sendo escrita.



## Controle Remoto em Cascata

Em uma cascata remota, o bloco de controle superior fornece um valor de estado da saída, a qual 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 4—Exemplo da Cascata Remota**

Quatro passos para completar uma inicialização de cascata remota:

**1. Not cascade mode** – O bloco AO está no modo Auto e será indicado para o próximo bloco superior (no aparelho 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 target mode 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 Acknowlegde
MODE_BLK.Actual = Auto
BKCAL_OUT.STATUS = GOODC-INICIALIZATION REQUEST (IR)
```

**3. Inicialização completa** – O bloco AO vai para RCas porque o bloco acima 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. **Cascata completa** – O bloco de cima muda o estado 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

**Notas:**

- 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 ser limitado. Aplica-se ao bloco de controle que suporta na entrada um parâmetro de busca. Quando o bloco está em LO, sua saída está buscando o valor estabelecido (através de chaves magnéticas). O usuário não pode mudar as saídas do servidor remoto.

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

A saída do bloco não está sendo calculada, porém este valor na saída pode ser limitado. Estes limites podem ser ajustados na saída pelo operador.

#### **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 de função, e usando um valor de setpoint fornecido por um servidor ou um operador através de uma interface em caso de um bloco de função 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, Man.

Modo	Fonte SP	Fonte de Saída
O/S	Usuário	Usuário
LO	Usuário	Usuário
Man	Usuário	Usuário
Auto	Usuário	Bloqueio do Algoritmo
Rcas	Controle da Aplicação que é executada em um equipamento de interface	Bloqueio do Algoritmo

### b) Elementos do Modo de um Bloco Funcional

O modo é composto por dois parâmetros. Target\_mode e mode\_blk, que estão definidos em todos os blocos. Estes parâmetros são explicados abaixo:

**TARGET\_MODE** - este é o modo solicitado pelo operador. Apenas um modo permitido 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. Assim, o usuário não pode gravar nos atributos deste parâmetro. É definido como tendo três elementos:

- **Actual** - Este é o modo corrente do bloco, que pode ser diferente do TARGET\_MODE baseado nas condições operacionais e na configuração do bloco como estado de parâmetro de entrada e configuração de bypass. Seu valor sempre é calculado como parte de execução do bloco.
- **Permitted** - define os modos que são permitidos para um determinado instante no 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 de função será controlada pelo parâmetro modo. O usuário configura o TARGET\_MODE, que indica qual o modo de operação é desejado para o bloco. Assim, o algoritmo avalia se o bloco pode ser executado na modalidade requisitada ou na modalidade mais próxima em 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

**Tabela1–Prioridade do Modo**

### d) Modo de cálculo

O modo atual será calculado na seguinte forma:

- Cada modo tem algumas condições que fazem com que o modo atual a ser de 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 o 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 de 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.

### e) Informação específica para desenvolvedores de drivers

Internamente, cada atributo do modo 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).

	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 2–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:

	Valor HEX	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
O/S,AutoMan	0x98	1	0	0	1	1	0	0	0

**Tabela 3–Modo suportado Bitstring do bloco AI**

## Parâmetros de escala

O parâmetro escala define a faixa operacional e as unidades de engenharia associadas com um 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: mostrar os equipamentos, para conhecer a faixa para 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 sintonia podem permanecer sem dimensão.

O bloco de função 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 Usando Parâmetros de Escala

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

1. O AO leva o SP e converte para porcentagem da PV\_SCALE:  
 $SP\% = (SP - EU_0) \cdot 100 / (EU_{100} - EU_0) [PV\_SCALE]$

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

SP = 15 mA

Os valores de SP em porcentagem são:  
 $SP\% = (15 - 4) \cdot 100 / (20 - 4) = 68.75\%$

2. O valor de saída é convertido de porcentagem para unidades de engenharia do  
 OUT\_SCALE:  
 $OUT = SP\% / 100 (EU_{100} - EU_0) + EU_0 [OUT\_SCALE]$

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

Ponto Decimal = 2

O valor de saída deste exemplo é:  
OUT= 68.75/100 (15-3) + 3 = 11.25 PSI

## **Manipulação em Fail Safe (Falha de Segura)**

O Fail Safe é um estado especial, que permite o bloco de função atuar quando for detectado uma situação anormal.

Uma situação anormal ocorre quando houver uma entrada não usual, por exemplo sensor com defeito ou perda de comunicação entre os blocos de função 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 volta à operação normal.

## **Condições que ativam o Fail Safe**

Quando os Blocos de Função de entrada ou saída detecta uma condição anormal, o bloco entra em modo FAIL\_SAFE. Estas situações anormais são detectadas através de diferentes formas nos blocos de função de entrada e saída.

Os blocos de função de entrada, por exemplo, AI e TOT são conectados com 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 de função de saída, por exemplo: bloco AO recebe os valores de entrada de um dos blocos superiores através da “conexão cíclica.” Estes blocos superiores são blocos de função de controle e, geralmente, são equipamentos de controle. Neste caso, o fail safe é ativado quando uma das seguintes condições é alcançada:

- Perda de comunicação de RCAS\_IN por um tempo que excede FSAFE\_TIME;
- Perda de comunicação para SP durante um tempo que excede FSAFE\_TIME;
- O estado IFS na entrada RCAS\_IN quando o modo designado é RCas;
- O estado 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 podem tomar, 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 seguro fornecido pelo parâmetro FSAFE\_VALUE para o cálculo quando o Fail Safe está ativo. O estado da saída vai para “Uncertain, substitute value”
- Use Last Usable Value – Neste caso, os blocos AI e AO usam o último valor utilizável para o cálculo do algoritmo. O estado será “Uncertain Last Usable Level”. Se ainda não houver um valor satisfatório, use o Valor Inicial na saída. O estado será “Uncertain Initial Value”.
- “Use the wrong value” (apenas para o bloco AI) – o 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 de função) as seguintes opções estão disponíveis:

- Hold - Para a totalização no último valor. O estado da saída vai para “Uncertain non-specific”.
- Memory - Usa o último valor válido para a totalização. O estado será de “Uncertain Last Usable Value”. Se não houver um estado válido na memória use o valor inicial para a totalização. O estado será “Uncertain, Initial Value”
- Run - A totalização é continuada (Reiniciada). Usa o valor e estado incorreto para a saída.

## Diagnóstico

Para fornecer 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 físico, o parâmetro DIAGNOSIS possui a informação sobre os “alertas” no equipamento (por exemplo, device not initialized, power up, factory init, hardware failure, etc.). O 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 de bitstring que possui o resumo das informações principais sobre o bloco de função e o bloco transdutor.

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

## Processando um Alerta - Alarmes e Eventos

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

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

O parâmetro de alarme é fornecido num bloco para capturar a informação dinâmica associada com um alarme. Os seguintes campos formam o parâmetro de alarme:

- Unacknowledged - Não usado.
- Alarm State - Este campo indica se o alerta está ativado ou não. O Alarm State terá os seguintes estados: 0 – não ativo; ≠ 0 – ativo.

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

- Time stamp–Não usado.
- Subcode– Não usado.
- Value – O valor do parâmetro associado no instante do alerta.

**b) Limite de alarme (parâmetro X\_LIM)**

Um alarme analógico ocorre quando um valor se encontra ou excede um limite. O alarme é verdadeiro, para um alarme alto quando o valor analógico é maior que o limite. O estado do alarme mantém-se verdadeiro até que seu valor cai abaixo do limite menos a histerese.

O tipo de alarme pode ser desabilitado ajustando o parâmetro limite de alarme para mais ou menos infinito, que é o default para todos os limites de alarme.

O parâmetro de saída do bloco de função, que será comparado ao limite de alarme.

**c) Histerese de alarme (parâmetro ALARM\_HYS)**

A saída (OUT), deverá retornar dentro dos limites do alarme antes de desfazer o alarme. 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) Índice de alarme (parâmetro ALM\_SUM)**

O parâmetro Alarm Summary, mostra 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 relatado 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.

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

### Objeto do Bloco –DS-32

Esta estrutura de dados consistem nos atributos de um bloco.

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

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

Esta estrutura de dados consiste no valor e estado dos parâmetros de ponto flutuante, que são entradas ou saídas

E	Nome do elemento	Tipo de Dado	Tamanho
1	Value	Float	4
2	Status	Unsigned8	1

**Valor & Estado - Estrutura Discreta–DS-34**

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

E	Nome do elemento	Tipo de Dado	Tamanho
1	Value	Unsigned8	1
2	Status	Unsigned8	1

**Estrutura escalar–DS-36**

Esta estrutura de dados consiste nos dados estáticos usados para escalar valores de pontos flutuantes para propósitos de exibição.

E	Nome do elemento	Tipo de Dado	Tamanho
1	EU à 100%	Float	4
2	EU à 0%	Float	4
3	Units Index	Unsigned16	2
4	Decimal Point	Integer8	1

**Estrutura do modo–DS-37**

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

E	Nome do elemento	Tipo de Dado	Tamanho
1	Actual	Bitstring	1
2	Permitted	Bitstring	1
3	Normal	Bitstring	1

**Estrutura de alarme 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	Unacknowledged	Unsigned8	1
2	Alarm State	Unsigned8	1
3	Time Stamp	Time Value	8
4	Subcode	Unsigned16	2
5	Value	Float	4

**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	Unacknowledged	Unsigned8	1
2	Alarm State	Unsigned8	1
3	Time Stamp	Time Value	8
4	Subcode	Unsigned16	2
5	Value	Unsigned8	1

**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	Unacknowledged	Unsigned8	1
2	Update State	Unsigned8	1
3	Time Stamp	Time Value	8
4	Subcode	Unsigned16	2
5	Relative Index	Unsigned16	2

**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	Current	Bit String	2
2	(Unacknowledged	Bit String	2
3	Unreported	Bit String	2
4	Disabled	Bit String	2

**Simula – 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/desabilitação (enable/disable).

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

**Simula - 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	Simulate Status	Unsigned8	1
2	Simulate Value	Unsigned8	1
5	Simule En/Disable	Unsigned8	1

**Estrutura de grupo–DS-67**

Esta estrutura de dados contém uma estrutura do parâmetro Batch.

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



# Biblioteca de Blocos

A tabela 2.1 dá uma descrição dos blocos

BLOCO	DESCRIÇÃO	PÁGINA NÚMERO
RES	PHYSICAL – Este bloco contém informação específica para o hardware associado com o recurso.	2.2
AI	ANALOG INPUT – Este bloco toma a informação de entrada do bloco transdutor e disponibiliza para outros blocos de função. Possui uma escala de conversão, “filtro” e mecanismo de fail safe (falha segura).	2.5
AO	ANALOG OUTPUT – O bloco AO fornece um valor para um bloco transdutor de saída. Fornece valores, escalas de conversão, mecanismo de fail safe e outros recursos.	2.8
TOT	TOTALIZER – Integra uma variável em função do tempo. Possui uma escala de conversão, “filtro” e mecanismo de fail safe.	2.12

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 conforme Contained, portanto não haverá entradas ou saídas para este bloco. O dado não é processado como em um bloco de função.

O intuito destes parâmetros é ser pouco requisitado pelo bloco de função.

### Parâmetro **FACTORY\_RESET**

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

Parâmetro	Nome	Descrição
1	Restart with defaults (Reinicia com os padrões)	para limpar memória de configurações, funciona como uma inicialização de fábrica.
2506	Restart processor (Reinicia processador)	é uma forma de acionar o botão reset no processador associado com o equipamento
2712	Restart bus address (Reinicia endereços no barramento)	inicia endereços para o valor padrão, 126.

### 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 é 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 o 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

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa/Opções	Valor Padrão	Unidades	Armazenamento / Modo*	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Será incrementado cada vez que houver mudança num parâmetro estático no bloco físico.
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 software 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	Bitstring indicando o diagnóstico do equipamento. Ver diagnóstico.
14	DIAGNOSIS_EXTENSION	Octetstring(6)			Nenhuma	D/RO	Não usado.
15	DIAGNOSIS_MASK	Octetstring(4)			Nenhuma	D/RO	Bitstring 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.
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.

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento / Modo*	Descrição
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.

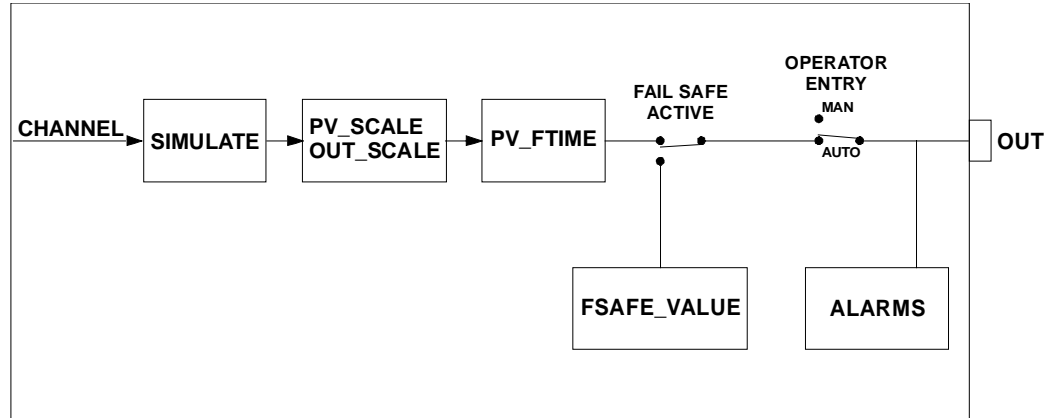
**\*Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RO – Somente leitura; D – dinâmico; N – não-volátil; S -estático**

## ENTRADA ANALÓGICA – AI

### Geral

O bloco de entrada analógico usa os dados de entrada do bloco transdutor, selecionada por um canal, e disponibiliza sua saída para outros blocos de função.

### Diagrama



### Descrição

O bloco transdutor fornece a unidade da PV para a entrada analógica e, quando a unidade PV é mudada no transdutor, a unidade de escala PV\_SCALE é convertida.

Opcionalmente, um filtro pode ser aplicado ao valor do sinal de processo, o qual a constante de tempo é PV\_FTIME. Considerando as mudanças na entrada, este é o tempo em segundos para que a PV alcance 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 avaliação. Quando ativo, o valor do transdutor e o estado serão sobrepostos pelo estado e pelo valor simulado.

A estrutura SIMULATE é composta pelos seguintes atributos:

- Simulate valve 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.

### Manuseio de Estados

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



**Cíclico – CFG\_DATA**

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

**Parâmetros**

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa Válida/ Opções	Valor Padrão	Unidades	Armazenamento/ Modo*	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Será incrementado cada vez que ocorrer mudanças no parâmetro estático no 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 para identificar canais usados e disponíveis. Não há algoritmo relacionado.
10	OUT	DS-33	OUT_SCALE		OUT	D / Man	O 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 à 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, a quantidade de 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.

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa Válida/ Opções	Valor Padrão	Unidades	Armazenamento/ Modo*	Descrição
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.

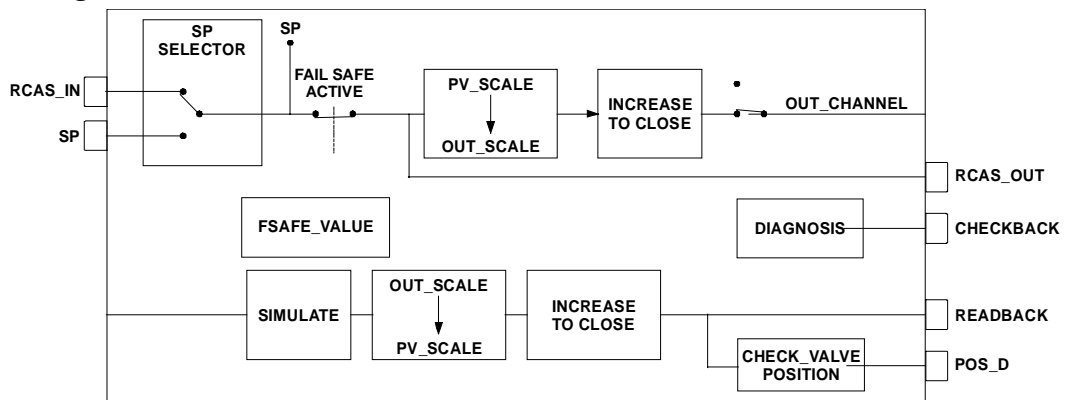
Legenda: E – parâmetro Enumerado; NA – parâmetro Adimensional; RO – Somente Leitura; D – dinâmico; N – não-volátil; S – estático

## SAÍDA ANALÓGICA – AO

### Geral

O Bloco de saída analógico é um bloco de função 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 de função e passa seus resultados para um bloco transdutor de saída através de um canal interno de referência.

### Diagrama



### 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.

$$SP\% = SP / (EU_{100\%} - EU_{0\%}) + EU_{0\%} [PV\_SCALE]$$

#### 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 (“Subindo”), SP, convertido para OUT\_SCALE, será 15 psi. Portanto, o tipo de atuador será: “ar para abrir”.
- Se o INCREASE\_CLOSE = 1 (“Descendo”), 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 checagem. Quando ativo, o valor do transdutor e o estado serão sobrepostos pelo valor e estado de simulado.

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 estado discreto: abrir, fechar 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,0x08,0x05,0x08,0x05,0x08,0x05,0x08,0x05,0x05,0x0A

**Parâmetros**

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento /Modo*	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Será incrementado cada vez que houver mudança em um parâmetro estático no bloco físico.
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 a condição existir.
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 Fail Safe.
25	FSAFE_VALUE	Float	OUT_SCALE	0	OUT	S	Valor de Preset para ajustar a Saída, quando o Fail Safe estiver ativo e o FSAFE_TYPE = 0.

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa/ Opções	Valor Padrão	Unidades	Armazenamento /Modo*	Descrição
27	RCAS_OUT	DS-33			PV	D/RO	O valor e estado requisitado por um bloco superior para que o bloco superior possa evitar reset windup 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 Readback.
33	CHECK_BACK	OctString(3)		0	Bitwise	D/RO	Informação do estado do equipamento. Ver Opções de Check Back.
34	CHECK_BACK_MASK	OctString(3)		0	Bitwise	S/RO	Informações de Check Back. 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_CLOSE	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.

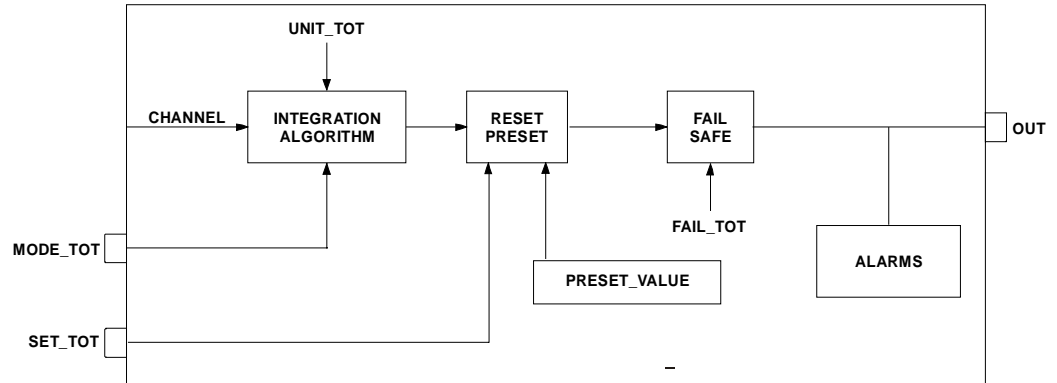
\*Legenda: E – parâmetro Enumerado; NA – parâmetro Adimensional; RO –Somente Leitura; D – dinâmico; N – não-volátil; S – estático

## TOTALIZADOR - TOT

### Geral

O bloco de função totalizador leva os dados do bloco transdutor selecionados através dos números de canal e o integra ao longo do tempo. Este bloco, normalmente, é usado para totalizar o fluxo ou seja, medir a massa ou o volume durante um certo período de tempo, ou totalizar a potência, medindo a energia total.

### Diagrama



### Descrição

O bloco de função totalizador integra uma variável (por exemplo: taxa de fluxo 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 fluxo de massa (por exemplo: Kg/s, g/min, ton/h) a unidade de saída deverá ser massa (por exemplo: kg, g, ton, lb, etc.).

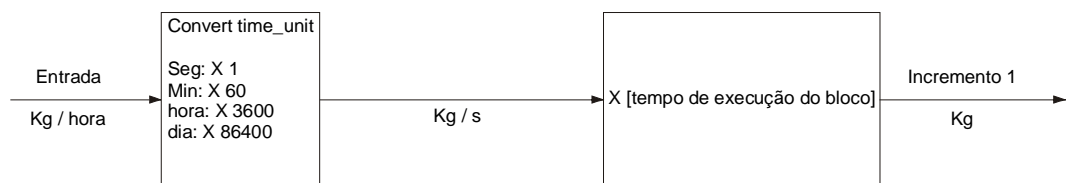


Fig 2.1 – Cálculo de incremento com taxa de entrada.

### Totalização da Vazão

Para a distinguir a vazão positiva da negativa, o bloco totalizador adota um sinal negativo como indicação 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. Quando o SET\_TOT está em “Reset” a saída será zero. Enquanto este parâmetro ter o valor de “Reset”, o bloco será o valor do reset antigo. Quando o SET\_TOT é “Preset”, a saída recebe o valor do parâmetro PRESET\_TOT e enquanto o valor for “Preset” a saída permanece a mesma. Portanto, a integração só é realizada quando o parâmetro volta para “Totalize”.

### **Iniciando o Bloco**

Para inicializar o bloco totalizador para funcionar apropriadamente, os seguintes passos devem ser seguidos:

- O Parâmetro Canal deve ser ajustado para PV;
- O Primary\_Value\_Type do bloco transdutor deve ser ajustado para “Flow”;
- O Linearization\_Type do bloco transdutor deve ser ajustado para “Square Root”;
- O Primary\_Value\_Unit do bloco transdutor deve ser ajustado para uma unidade válida de fluxo;
- O UNIT\_TOT do bloco transdutor deve ser ajustado para uma unidade de massa ou volume correspondente 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 / SETTOT	0xC1,0x80,0x84,0x85
TOTAL / SETTOT / MODETOT	0xC1,0x81,0x84,0X85

## Parâmetros

Índice	Parâmetro	Tipo de Dado (extensão)	Faixa Válida/ Opções	Valor Padrão	Unidades	Armazenamento / Modo*	Descrição
1	ST_REV	Unsigned16		0	Nenhuma	S/RO	Será incrementado cada vez que ocorrer mudanças no parâmetro estático no 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			OUT	N/RO	O valor primário analógico calculado como o resultado da execução da função. É o resultado da integração.
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 ao limite do alarme mais a histerese.
18	HI_HI_LIM	Float		INF	OUT	S	Ajuste de alarme muito alto em unidades de Engenharia.
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



## DESCRIÇÕES DE BITSTRINGS

### DIAGNÓSTICOS (BLOCO FÍSICO)

Bit	Mnemônico	significado	DIAGNOSIS_MASK
0	DIA_HW_ELECTR	Falha de hardware dos componentes eletrônicos	Não usado
1	DIA_HW_MECH	Falha de hardware dos componentes mecânicos	Não usado
2	DIA_TEMP_MOTOR	Temperatura do motor muito alta	Não usado
3	DIA_TEMP_ELECTR	Temperatura eletrônica muito alta	Não usado
4	DIA_MEM_CHKSUM	Erro de memória	Não usado
5	DIA_MEASUREMENT	Falha na medida	Não usado
6	DIA_NOT_INIT	Equipamento não inicializado	Não usado
7	DIA_INIT_ERR	Falha de Auto-Calibração	Não usado
10	DIA_ZERO_ERR	Erro de ponto Zero	Não usado
11	DIA_SUPPLY	Falha da Fonte de Alimentação	Não usado
12	DIA_CONF_INVALID	Configuração Inválida	Não usado
13	DIA_WARM_START	Reinício em andamento (Alimentado). Este bit vai para verdadeiro quando alimentado e será automaticamente reiniciado após 10 segundos.	Usado
14	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.	Usado
15	DIA_MAINTAINANCE	Manutenção requisitada	Não usado
16	DIA_CHARACTER	Caracterização Inválida	Não usado
17	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.	Não usado
37	EXTENSION_AVAILABLE	Mais informação de diagnóstico está disponível. A informação deste diagnóstico estará em DIAGNOSIS_EXT	Não usado

### CHECK\_BACK (BLOCO ANALÓGICO DE SAÍDA)

Bit	Mnemônico	significado	CHECK_BACK_MASK
0	CB_FAIL_SAFE	Equipamento no campo com Fail Safe ativo.	Usado
1	CB_REQ_LOC_OP	Requisita operação local.	Não usado
2	CB_LOCAL_OP	Equipamento de campo sob controle local, chave LOCKED OUT 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 OPEN excedido.	Não usado
6	CB_TORQUE_D_CL	Limite de torque na direção CLOSE excedido.	Não usado
7	CB_TRAV_TIME	Indica o estado de monitoramento do curso do equipamento, caso esteja em YES, o tempo do curso do atuador excedeu-se.	Não usado
8	CB_ACT_OPEN	Atuador se move na direção de abertura.	Não usado
9	CB_ACT_CLOSE	Atuador se move na direção de fechamento.	Não usado
10	CB_UPDATE_EVT	Alerta gerado por mudança em dados estáticos (Bloco de Função 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).	Não usado

Bit	Mnemônico	significado	CHECK_BACK_MASK
15	CB_SELFTEST	Equipamento em auto-teste.	Não usado
16	CB_TOT_VALVE_TRAV	Indica que o limite total de percurso da válvula está excedido.	Não usado
17	CB_ADD_INPUT	Indica que uma entrada adicional (por exemplo: para diagnóstico) está ativa	Não usado

## FB ajustes e disponibilidade de FB

Ordem no diretório	Blocos	LD		TT		DT		IF		FI		TP		FY		FP		LD293	
		FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt	FB tipo	Qtt
1	PHY	Sim	1	Sim	1	Sim	1	Sim	1	Sim	1	Sim	1	Sim	1	Sim	1	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	Sim	1	Sim	1	Sim	1	Sim	1	Sim	1

Nota 1 – A coluna "FB type" indica qual bloco de função está disponível para cada tipo de equipamento.

Nota 2 – A coluna "Qtt" indica o número de blocos de função instanciada na inicialização de fábrica para cada tipo de aparelho.

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 aparelho não possui os blocos de função AO e AI e o transdutor é específico para um equipamento.