PROFIBUS: O MODELO DE BLOCOS FUNCIONAIS

INTRODUÇÃO

O Profibus é um protocolo digital utilizado em sistemas de controle, que permite a conexão com interoperabilidade de diver sos equipamentos e fabricantes. Possui uma s érie de vantagens em relação à tecnologia 4-20 mA, onde r esumidamente pode-se citar, dentre outras:

- Fácil cabeamento com redução de custos;
- Simples operação, através da sala de controle;
- Aplicações em área classificadas;
- Altas taxas de comunicação no Profibus-DP;
- Poderosas ferramentas de configuração/parametrização e gerenciamento de ativos;
- Tecnologia aberta e em contínua evolução.

Trata-se de um protocolo padronizado, totalmente aberto, onde sua organização de usuários mantem o desenvolvimento e manutenção da tecnologia, combinand o os interesses de usuários e fabricantes. Este padrão é garantido segundo as normas EN 50170 e EN 50254. e foi firmemente estabelecido com a IEC 61158, desde janeiro de 2000.

O Profibus-PA é uma das variantes da tecnologia Profibus, dedicada à área de processos e amplamente utilizada em comunicação digital bidirecional, permitindo a implementação de sistemas de controle de processos tecn ologicamente avançados.

O Profibus-PA permite serviços cíclicos e acíclicos. Os serviços cíclicos são utilizados para a transmissão de dados pertinentes às medições e comandos de atuação co m informações de valor/status. Os serviços acíclicos são utilizados pelas ferramentas de configuração, man utenção e diagnóstico, durante a operação.

ARQUITETURA DO SISTEMA DE CONTROLE DE PROCESSO

Na prática existem diversos fabricantes de sistemas de controle, assim como várias possibilidades de arquiteturas, mas basicamente deve-se atentar para:

- O número de estações *host* e estações de engenharia;
- O número de controladores;
- A hierarquia da comunicação;
- As atribuições dos dispositivos e equipamentos de campo aos seus respectivos controladores;
- O método de conexão dos equipamentos de campo;
- As condições envolvendo áreas à prova de explosão, segurança intrínseca, emissões eletromagnéticas, condições ambientais, distribuição de cabeamento, aterramento etc.

A figura 1 mostra uma arquitetura típica, onde se tem o controlador Profibus, estações de engenharia, ferramentas de parametrização, acopladores e outros elementos da rede.

Image not found or type unknown

Figura 1 – Arquitetura típica Profibus.

Um sistema Profibus pode ser operado e monitorado independ entemente de equipamentos e fabricantes se todas as funcionalidades e parametrizações, bem como as formas de acesso a estas informações, forem padrões. Estes padrões são determinados pelos *profiles* (perfis) do Profibus.

Os *profiles* especificam como os fabricantes devem implementar os objetos de comunicação, variáveis e parâmetros, segundo a classe de funcionamentos dos equipamentos. E ainda existe a classificação dos próprios parâmetros:

- Valores dinâmicos d e processo: dizem respeito às variáveis de processo, cuja informação é descrita nos arquivos GSD (*Generic Station Description*), que serão lidas ciclicamente pelos mestres Classe 1 e também aciclicamente pelos mestres Classe 2.
- Mestre Classe 1: responsável pelas operações cíclicas (leituras/e scritas) e controle das malhas abertas e fechadas do siste ma.
- Mestre Classe 2: responsável pelos acessos acíclicos dos parâmetros e funções dos equipamentos PA (estações de engenharia, por exemplo P+F Pactware ou Siemens Simatic PDM).

Atualmente, o Profibus-PA está definido segundo o PROFILE 3, onde se tem informações para vários tipos de equipamentos, com o transmissores e posicionadores de válvulas.

Estes equipamentos são implementados segundo o modelo de blocos funcionais (Function Blocks), onde um agrupamento de parâmetros garante acesso uniforme e sistemático das informações.

Vários blocos e funções são necessários, dependendo do modo e fase de operação. Basicamente, tem-se os seguintes blocos:

- **Blocos Funcionais de Entradas e Saídas Analógicas:** estes blocos des crevem funcionalidades durante a operação, tais como, troc as de dados cíclicos de entrada/saída, condições de alarmes e limites;
- **Bloco Físico (*Physic al Block*):** traz informações de identificação do eq uipamento, assim como pertinentes ao hardware e ao software;
- **Blocos Transdutores (*Transducer Blocks*):** fazem o acondicionamento de informações dos sensores que serão utilizadas pelos blocos funcionais, assim como informações para disparo de atuações em elementos finais de contr ole, como por exemplo em um posicionador de válvulas. Normalmente um equipamento de entrada (u m transmissor de pressão, por exemplo) possui um bloco transdutor (TRD) que está amarrado via canal a um bloco de entrada analógica (AI), enquanto um equipamento de saída (um posicionador de válvulas, por exemplo) possui um bloco de saída analógic a (AO) que recebe um valor de *setpoint* e o disponibiliza vi a canal a um bloco transdutor (TRD) que acionar á o elemento final.

Existem alguns e quipamentos que possuem vários blocos AIs e AOs. São chamados de equipamentos multicanais e possibilitam vários blocos TRDs associados ao hardware.

O Profibus-PA ainda diferencia os *profiles* em classes:

- Equipamento Classe A: inclui informações somente dos blocos físico e de funções. Nesse tipo de classe, o equipamento está limitado ao básico necessário para operação : variável do processo (valor e status), unidade e *tag*.
- Equipamento Class e B: possui funções estendidas de informações dos blocos físico, transdutor e de funções.

Uma característica poderosa suportada pelo PROFILE 3 é a definiçã o de cada equipamento segundo os arquivos GSD. Estes arquivos garantem que qualquer sistema Profibus possa integrar o equipamento, independentemente de suas características. Com isto cada fabricante pode desenvolver suas particularidades em formas de blocos funcionais que vão além do que está definido no *profile*.

Isto agrega valor ao s equipamentos e torna possível a competição de desenvolvimento e oferta de características adicionais nos equipamentos pelos fabricantes, sendo que as particularidades específicas de cada equipamento podem ser acessadas via conceitos padrões de interfaces, baseado em EDDL (Linguagem Eletrônica Descritiva de Equipamentos) ou FDT (Ferramenta de Equipamento de Campo).

Através destas interfaces, o usuário ganha versatilidade e flexibilidade de configuração, parametrização, calibração e principalmente mecanismos de *download* e *upload* durante a fase de planejamento/comissionamento dos projetos.

Em geral o Profibus melhora a eficiência e reduz os custos durante o ciclo de vida de uma planta.

Podemos citar como benefícios:

- Projetos mais simples, em tempos mais curtos;
- Instalações otimizadas;
- Comissionamentos mais rápidos;
- Alta exatidão;
- Maior confiabilidade;
- Fácil expansão;
- Gerenciamento de ativos.

MODELO DE BLOCOS FUNCIONAIS

Em engenharia de processo é comum utilizar blocos funcionais pa ra descrever as características e funções de um ponto de medida ou ponto de manipulação num certo ponto de controle ou para representar uma aplicação de automação através da combinação destes tipos de blocos.

A especificação de dispositivos Profibus-PA utiliza este modelo de blocos de função para representar sequências funcionais, como mostrado na Figura 2.

Image not found or type unknown

Figura 2 - Modelo de blocos funcionais em um dispositivo Profibus-PA.

Os seguintes três tipos de blocos são utilizados:

- Bloco Físico (*Physical Block* – PB)

O bloco físico cont ém os dados característicos de um dispositivo, como o modelo do dispositivo, fabricante, versão, número de série etc. Há somente um bloco físico em cada dispositivo.

- Bloco Transdutor (*Transducer Block* – TB)

O bloco transdutor contém todos os dados requeridos para proce ssar um sinal não condicionado, obtido de um sensor, para passar ao bloco de função. Se este processame nto não for necessário, o bloco transdutor pode ser omitid o.
Dispositivos multifuncionais com dois ou mais sensores têm o correspondente número de blocos transdutores.

- Bloco de Função (*Function Block* – FB)

O bloco de função contém todos os dados para processamento final do valor medido antes da transmissão para o sistema de controle ou, por outro lado, para processamento de uma etapa antes do cenário do processo.

Os seguintes blocos de funções estão disponíveis:

- Bloco de Entrada Analógica (*Analog Input Block* – AI)

O bloco de entrada analógica fornece o valor medido do sensor e bloco transdutor para o sistema de controle.

- Bloco de Saída Analógica (*Analog Output Block* – AO)

O bloco de saída analógica fornece ao dispositivo o valor especificado pelo sistema de controle.

- Bloco de Totalização (*Totalizer Block* – TOT)

O bloco de totalização fornece ao dispositivo o valor totalizado especificado pelo sistema de controle.

- Bloco de Entrada Digital (*Digital Input Block* – DI)

O bloco de entrada digital fornece ao sistema de controle um valor digital do dispositivo.

- Bloco de Saída Digital (*Digital Output Block* – DO)

O bloco de saída digital fornece ao dispositivo um valor especificado pelo si stema de controle.

Os blocos são implementados pelos fabricantes como soluções de *software* nos dispositivos de campo e, levando como o conjunto, representam a funcionalidade do dispositivo. Como regra, vários blocos podem trabalhar em conjunto em uma aplicação. Veja a Figura 3.

Figura 3 - Modelo completo de blocos em um dispositivo Profibus-PA.

PHYSICAL BLOCK (PB)

O bloco físico contém os dados característicos de um dispositivo, como o modelo do dispositivo, fabricante,

versão, número de série etc. Há somente um bloco físico em cada dispositivo.

Figura 4 - Bloco físico.

As seguintes informações/f unções são fornecidas pelo *Physical Block*:

- *Tag;*
- Diagnósticos;
- Gerenciamento de proteção de escrita (*Write Locking*);
- *Factory Reset (War m/Cold reset);*
- *Seleção do GSD Id N umber (de acordo com o GSD do equipamento).*

BLOCO FÍSICO – Slot (0)					
Parâmetro	**Índice Relativo**	**Descrição**	**Tipo de dado**	**Armazenamento em memória e acesso R/W**	**Faixa de valores**
ST_REV	1	Este parâmetro será incrementado sempre que houver mudança em parâmetros estáticos (S) do bloco.	Unsigned16	S/RO	
TAG_DESC	2	*Tag* do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW	
STRATEGY	3	Valor fornecido pelo usuário que identifica uma configuração.	Unsigned16	S/RW	
ALERT_KEY	4	Estado atual de alarmes.	Unsigned8	S/RW	1 a 255
TARGET_MODE	5	Contém o modo desejado para o bloco.	Unsigned8	S/RW	AUTO
MODE_BLK	6	Modo corrente do bloco.	DS-37	D/RO	
ALARM_SUM	7	Contém os estados presentes dos alarmes de bloco.	DS-42	D/RO	
SOFTWARE_REVISON	8	Número da revisão de *software* do equipamento de campo.	Visible-String(16)	S/RO	

HARDWARE_REVISON	9	Número da revisão do hardware do equipamento de campo.	Visible-String(16)	S/RO	
DEVICE_MAN_ID	10	Número de identificação do fabricante.	Unsigned16	S/RO	
DEVICE_ID	11	Número do modelo do fabricante associado ao equipamento.	Visible-String(16)	S/RO	
DEV_SER_NUM	12	Número de série do equipamento de campo.	Visible-String(16)	S/RO	
DIAGNOSIS	13	*Bitstring* indicando o diagnóstico do equipamento.	Octet-String(4)	D/RO	
DIAGNOSIS_EXTENSION	14	Não usado.	Octet-String(6)	D/RO	
DIAGNOSIS_MASK	15	*Bitstring* indicando os tipos de diagnósticos suportados pelo equipamento de campo.	Octet-String(4)	D/RO	
DIAGNOSIS_MASK_EXTENSION	16	Não usado.	Octet-String(6)	D/RO	
DEVICE_CERTIFICATION	17	Certificações do equipamento de campo.	Visible-String(32)	S/RO	
WRITE_LOCKING	18	Se bloqueado, nenhuma mudança é permitida, exceto para se alterar o WRITE_LOCKING. Entradas cíclicas no bloco serão atualizadas continuamente.	Unsigned16	S/RW	0: Escrita Bloqueada 2457: Escrita Desbloqueada
FACTORY_RESET	19	Permite que o equipamento reinicie pelo comando do usuário. Vários níveis de reinicializações são possíveis.	Unsigned16	S/RW	1: Reinicia com padrão (default) 2506: Reinicia o processador 2712: Recupera o endereço padrão (126) para o equipamento.
DESCRIPTOR	20	É uma descrição fornecida pelo usuário para o bloco na aplicação.	Octet-String(32)	S/RW	

DEVICE_MESSAGE	21	Mensagem do bloco fornecida pelo usuário na aplicação.	Octet-String(32)	S/RW	
DEVICE_INSTALL_DATE	22	Data da instalação do equipamento.	Octet-String(16)	S/RW	
LOCAL_OP_ENA	23	Não usado.	Unsigned8	N/RW	
IDENT_NUMBER_SELECTOR	24	Permite que o usuário selecione o *Identifier Number* do equipamento.	Unsigned8	S/RW	0: Profile specific Ident_Num 1: Manufacturer specific Ident_Num 2: Manufacturer specific Ident_Num of V2.0 3: Ident_Num of Multi_Variable device
HW_WRITE_PROTECTION	25	Não usado.	Unsigned8	D/RO	
Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW - Escrita/Leitura; RO – Somente leitu Dinâmico; N – Não-volátil; S – Estático					

Tabela 1 – Bloco físico.

A seguir, tem-se um detalhamento dos parâmetros mais utilizados. Para mais detalhes e outros parâmetros, consulte a documentação da Organização Profibus International.

PARÂMETRO FACTORY_RESET

Em algumas aplicações é interessante que o usuário possa reinicializar o equipamento Profibus-PA através de sua estação de trabalho. O padrão Profibus permite algumas formas de comandos de inicialização do equipamento através do parâmetro FACTORY_RESET do *Physical Block:*

FACTORY_RESET		
Opção	**Tipo de Ação**	**Descrição**
1	*Restart with defaults* (Reinicia com os padrões).	Reinicia o equipamento com os valores padrões (*default*). Funciona como uma inicialização de fábrica.
2506	*Restart processor* (Reinicia o processador).	Reinicia o equipamento com os valores mais recentes em sua memória.
2712	*Restart bus address* (Reinicia com o endereço 126).	Inicia o equipamento com o endereço 126.

Tabela 2 – Parâmetros do FACTORY_RESET.

PARÂMETRO DE ESCRITA

Um equipamento Profibus-PA pode ser acessado pelo usuário via estação de trabalho, localmente através de uma interface d e ajuste local ou ainda por um mestre Profibus C lasse 1. Para evitar acessos não autorizados, pode-se prote ger o equipamento através do parâmetro *WRITE_LOCKING*, bloqueando a escrita acíclica.

Se o parâmetro *WRITE_LOCKING* estiver bloqueado ("*Locked*"), impedirá qualquer comando de escrita no equipamento. O s acessos cíclicos, assim como os cálculos feitos pelo equipamento continuarão normalmente, mas a escrit a acíclica será bloqueada. Quando em "*Write Unlocked*", a escrita acíclica (para configuração/parametrizaç ão) será novamente permitida.

SELEÇÃO DO NÚMERO IDENTIFICADOR (GSD IDENTIFIER NUMBER)

O parâmetro *IDENT_NUMBER_SELECTOR* permite ao usuário selecio nar um número válido e diferente do identificador.

A comunicação cíclica com o mestre Profibus Classe 1 utiliza-se do *I dent Number* para ter uma relação única entre mestre e escravo, associada ao arquivo GSD.

Se o perfil do equipamento é mudado de acordo com o *Ident Number*, o equipamento interagirá com as características do pe rfil associado ao arquivo GSD.

Após uma alteração do *Ident Number*, o equipamento deve ser reinicializado para que possa ter sua comunicação cíclica de acor do com o novo valor de *Ident Number*.

IDENT_NUMBER_SELECTOR	Permite a seleção do *Identifier Number* do equipamento.	*0: Profile specific Ident_Num* *1: Manufacturer specific Ident_Num* *2: Manufacturer specific Ident_Num of V2.0* *3: Ident Num of Multi Variable device*

Tabela 3 – IDENT_NUMBER_SELECTOR.

Ao energizar os equ ipamentos da *Vivace Process Instruments*, dentre as mensagens de inicialização, serão exibidas no LCD:

- IDSEL MANUF: indic a que o *Ident Num* está de acordo com o arquiv o GSD do equipamento Vivace;
- IDSEL PROFI: indica que o *Ident Num* está de acordo com o arquivo GSD padrão.

Quando em IDSEL PROFI, caso o arquivo GSD utilizado seja d o equipamento Vivace, haverá comunicação acíclica com ferramentas de configuração (por exemplo, *PD M* ou *Pactware*) e não haverá comunicação cíclica com o mestre Profibus Classe 1 (PLC).

DIAGNÓSTICOS

A tecnologia Profibus prevê alguns mecanismos de diagnósticos durante a comunicação cíclica. O *Physical Block* possui o parâmetro DIAGNOSIS que tem a inf ormação 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 EM EQUIPAMENTOS DE SAÍDA

No bloco AO existe um parâmetro especial de saída com a finalidade de diagnóstico para elementos de saída, como atuadores ou posicionadores de válvulas.

O parâmetro CHECKBACK é um parâmetro do tipo *bitstring* (somen te leitura) que possui o resumo das informações principais entre o bloco funcional AO e o bloco transdutor. O CHECKBACK pode ainda ser utilizado para configuração cíclica, o que será visto posteriormente.

DIAGNÓSTICOS (BLOCO FÍSICO)

Bit	Mnemônico	Descrição
0	DIA_HW_ELECTR	Falha de hardware: componentes eletrônicos.
1	DIA_HW_MECH	Falha de hardware: 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 medição.
6	DIA_NOT_INIT	Equipamento não inicializado.
7	DIA_INIT_ERR	Erro de inicialização
10	DIA_ZERO_ERR	Erro de zero.
11	DIA_SUPPLY	Falha da fonte de alimentação.
12	DIA_CONF_INVAL	Configuração inválida.
13	DIA_WARM_START	Reinicialização em andamento (alimentado). Este bit vai para "verdadeiro" quando alimentado e será automaticamente reiniciado após 10 segundos.
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.
15	DIA_MAINTAINANCE	Manutenção requerida.
16	DIA_CHARACT	Caracterização inválida.
17	IDENT_NUMBER_VIOLATION	Vai para 1 durante a troca de dados cíclicos se o valor do *Ident Number* do equipamento (de acordo com o parâmetro IDENT_NUMBER_SELECTOR) e o arquivo GSD forem diferentes.
37	EXTENSION_AVAILABLE	Mais informações de diagnóstico estão disponíveis, de acordo com o parâmetro DIAGNOSIS_EXT.

Tabela 4 – Bits de diagnóstico.

CHECK_BACK (BLOCO DE SAÍDA ANALÓGICA-AO)

Bit	Mnemônico	Descrição
0	CB_FAIL_SAFE	Equipamento de campo com *Fail Safe* (falha segura) ativa.
1	CB_REQ_LOC_OP	Requisita operação local.
2	CB_LOCAL_OP	Equipamento de campo em operação local, LOCKED OUT em funcionamento.
3	CB_OVERRIDE	Emergência *override* ativa.
4	CB_DISC_DIR	Posição de realimentação diferente da esperada.
5	CB_TORQUE_D_OP	Limite de torque na direção OPEN excedido.
6	CB_TORQUE_D_CL	Limite de torque na direção CLOSE excedido.
7	CB_TRAV_TIME	Indica que o tempo de curso do atuador foi excedido.
8	CB_ACT_OPEN	Atuador move-se na direção de abertura.
9	CB_ACT_CLOSE	Atuador move-se na direção de fechamento.
10	CB_UPDATE_EVT	Alerta gerado por mudança em parâmetros estáticos.
11	CB_SIMULATE	Simulação de valores de processo habilitada.
13	CB_CONTR_ERR	Malha de controle interno interrompida.
14	CB_CONTR_INACT	Posicionador inativo (OUT status = BAD).
15	CB_SELFTEST	Equipamento em auto-teste.
16	CB_TOT_VALVE_TRAV	Indica que o limite total de curso da válvula foi excedido.
17	CB_ADD_INPUT	Indica que uma entrada adicional (por exemplo, para diagnóstico) está ativa.

Tabela 5 – Bits do parâmetro CHECK_BACK.

DIAGNÓSTICOS CÍCLICOS

Os diagnósticos podem ser verificados ciclicamente ou aciclicamente, através de leituras via mestre Profibus-DP classe 1 e classe 2, respectivamente.

Os equipamentos de campo Profibus-PA disponibilizam 04 byt espadrões via *Physical Block.* Quando o bit mais significativo do 4º byte for “1”, o diagnóstico será estendido em mais 6 bytes. Estes bits de diagnósticos estarão descritos no arquivo GSD. Unit_Diag_bit está descrito no arquivo GSD do equipamento Profibus-PA.

DIAGNÓSTICOS					
				Physical Block	
Comprimento do byte de status	Tipo de status	Slot do Physical Block	Status Appears / Disappears	Diag. Padrão	Diag. Estendido (de acordo com o fabricante)
08 – Diag Padrão FE – Diag. Estendido	FE	01	01-Appears 02-Disappears	4 bytes Bit 55 (MSB do 4° byte) = 1 - Diag. estendido	6 bytes
Bytes de Diagnóstico					

	Byte 1								Byte 2							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Unit_Diag_Bit	31	30	29	28	27	26	25	24	39	38	37	36	35	34	33	32
	Byte 3								Byte 4							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Unit_Diag_Bit	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48

Bit 55: Quando em “1” indica que há diag. estendido.
Tabela 6 – Diagnósticos cíclicos.

ANALOG INPUT BLOCK (A I)

O bloco AI recebe um valor analógico do processo através do bloco transdutor (TRD) em seu canal e o processa de acordo com sua configuração, disponibilizando-o ao mestre Profibus Classe 1. Possui modo de operação, escala, filtro, limites, simulação e tratamento de erros.

O bloco AI suporta os seguintes modos:

- OOS (*Out of Service* – fo ra de serviço): neste modo, a saída não é atualiz ada pelo algoritmo do bloco;
- Man (Manual): neste mo do, o operador pode escrever valores na saída d o bloco;
- Auto (Automático): neste modo, a saída do bloco é resultado do processamento do bloco.

O bloco AI permit e ainda que o usuário simule um valor de processo, através do parâmetro SIMULATE. Este parâmetro é usado para facilitar testes, por exemplo, quan do se está fazendo o *startup* de malhas de controle

A estrutura SIMULATE é co mposta pelos seguintes atributos:

- Simulate Value and Stat us
- Simulate Enable

Quando ativo, o valor do bloco transdutor e o status serão s obrepostos pelo status e valor simulados. Quando desabilitado, o valor de processo será disponibilizado pe lo bloco transdutor ao bloco AI.

Figura 5 – Resumo dos parâmetros do Bloco AI.

Figura 6 – Bloco AI.

O bloco AI realiza uma conversão de escala do parâmetro TRD_PRIMARY_VALUE, utilizando os parâmetros PV_SCALE e OU T_SCALE, sendo que a unidade de engenharia e m OUT_SCALE é utilizada apenas para indicação.

Pode-se ainda aplicar um filtro ao valor de processo, de acordo co m o parâmetro de constante de tempo PV_FTIME. Considerando as mudanças na entrada, este é o tempo e m segundos para que a PV atinja 63,2% do valor final. Se o va lor PV_FTIME for zero, o filtro é desabilitado.

Figura 7 – Algoritmo Bloco AI.

LIMITES E ALERTAS

O bloco AI suporta em seu algoritmo a verificação de limites de al armes e alertas. O usuário pode configurar os limites de a larmes e alertas através dos parâmetros HI_ HI_LIM, LO_LO_LIM e HI_LIM, LO_LIM, indicando os limites superior extremo, inferior extremo, superior normal e inferior normal, respectivamente.

Um alerta ou um alarme altera a condição do status do parâ metro AI_OUT. O parâmetro ALARM_HYS indica a histerese da variação do parâmetro e deve ser confi gurado evitando falsos alertas e alarmes. Um alerta ou um alarme também altera a condição do parâmetro ALARM_SUM.

Figura 8 – Limites de alarmes e alertas.

TRATAMENTO DE ERROS – FALHA SEGURA

Se um erro aconte cer no processo de medição, aquisição do valor do bloco transdutor (TRD) ou perda de comunicação cíclica com o mestre Profibus Classe 1, o mesmo se propagará através do status da variável de interligação do c anal com o bloco AI.

Nesta condição, o usuário pode configurar o tipo de tratamento qu e deve acontecer com a saída do bloco, através do parâmetro FSAFE_TYPE. A saída irá para a condição de falha segura configurada se a situação de anormalidade permanecer por um período de tempo maior que o configurado no parâmetro FSAFE_TIME. Após a normalização da condição, o bloco AI retornará à oper ação normal.

CONDIÇÕES QUE ATIVAM A FALHA SEGURA

Quando os blocos f uncionais de entrada ou saída detectam uma co ndição de anormalidade, entram no modo

de falha segura (FAIL_SAFE), caso assim estiver configurado. Estas situações de anormalidade são detectadas através de diferentes formas, dependendo do tipo dos blocos funcionais (entrada ou saída).

Os blocos funcionais de entrada recebem (via canal) os valores e status dos blocos transdutores. Caso recebam uma condiçã o de status "*bad*" (ruim), por exemplo, numa falha de leitura de um sensor, a condição de FAIL_SAFE do b loco é ativada, se estiver configurada.

Figura 9 – Tratamento do *Fail Safe* no Bloco AI.

Os blocos funcionais de saída recebem os valores de entrada do mestre Profibus Classe 1 via comunicação cíclica e a ação de FAIL_SAFE é ativada quando uma das seguintes condições é detectada:

- Perda de comunicação na entrada RCAS_IN por um tempo superior ao especificado no parâmetro

FSAFE_TIME;

- Perda de comunicação na entrada SP por um tempo superior ao especificado no parâmetro

FSAFE_TIME;

- Quando em modo RCas, receber o parâmetro RCAS_IN com statusIFS (*initiate fail-safe*);
- Quando em modo Aut o, receber o parâmetro SP com statusIFS (*initiate fail-safe*).

AÇÕES DE FALHA SEGURA

Quando na condição de falha segura, as ações possíveis a um bloco de entrada ou saída podem 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:

- FSAFE_VALUE – Neste caso, os blocos AI e AO usam o valor de segura nça configurado pelo usuário no parâmetro FSAFE_VALUE co mo valor no parâmetro OUT, quando o *Fail Saf e* estiver ativo. O status da saída vai para *"Uncertain, substit ute value";*
- *Last Usable Value* – N este caso, os blocos AI e AO usam o último valor com status *good* (bom) para o cálculo de suas saídas. O st atus será "*Uncertain Last Usable Value*". Se ao o correr a falha ainda não houver um valor adequado, usará o Valor Inicial na saída. O status será "*Uncertain Initial Value*";
- *Wrong Value* (apenas para o bloco AI) – O bloco AI usa o próprio status e valor vindos do bloco transdutor (TRD);

- ACTUATOR_ACTION (apenas para bloco AO) – O bloco AO entra na co ndição de segurança baseado no parâmetro ACTUATOR_ACTION do bloco transdutor.

Figura 10 – Tratamento do *Fail Safe* no Bloco AO.

Para o Bloco TOT (Totalizador) tem-se o parâmetro FAIL_TOT e as seguintes opções:

- *Hold* - Para 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 válid o na memória, deve ser usado o valor inicial p ara a totalização. O status será "*Uncertain, Initial Value*";
- *Run* - A totalização é co ntinuada (reiniciada). O valor e o status incorret os são usados para a saída.

Figura 11 – Tratamento do *Fail Safe* no Bloco TOT.

BLOCO DE ENTRADA ANALÓGICA – AI
(Equipamentos com somente 1 AI: slot 1, se 2 AIs: Slot 1 e 2, se 3 AIs: slot 1, 2 e 3)

Parâmetro	Índice Relativo	Descrição	Tipo de dado	Armazenamento em memória e acesso R/W	Faixa de valores	Padrão (Default)
ST_REV	1	Este parâmetro será sempre incrementado quando ocorrer mudanças em parâmetros estáticos do bloco.	Un-signed16	S/RO		0
TAG_DESC	2	TAG do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW		Espaços
STRATEGY	3	É um valor fornecido pelo usuário para identificar uma configuração.	Un-signed16	S/RW		0
ALERT_KEY	4	Estado atual de alarmes.	Un-signed8	S/RW	1 a 255	0

TARGET_MODE	5	Contém o modo de operação desejado para o bloco.	Un-signed8	S/RW	O/S, MAN e AUTO	AUTO
MODE_BLK	6	Modo de operação corrente do bloco.	DS-37	D/RO		
ALARM_SUM	7	Contém os estados presentes dos alarmes de bloco.	DS-42	D/ RO		0,0,0,0
BATCH	8	Para uso em sistema distribuído. Utilizado para identificar canais usados e disponíveis. Não há algoritmo relacionado.	DS-67	S/RW		0,0,0,0
OUT	10	Valor analógico calculado como o resultado da execução do algoritmo do bloco funcional.	DS-33	D / Man	OUT_SCALE	
PV_SCALE	11	Os valores de escala Eu100% e Eu0% para o bloco transdutor de acordo com o canal especificado.	2 Floats	S/RW		100,0
OUT_SCALE	12	Os valores de escala Eu100% e Eu0%para o parâmetro OUT.	DS-36	S/RW		100,0 e 0.0 %
LIN_TYPE	13	Não usado.	Un-signed8	S/RW		0
CHANNEL	14	O número do canal lógico de hardware para o bloco transdutor que será conectado a este bloco funcional.	Un-signed16	S/RW		0
PV_FTIME	16	Constante de tempo de um filtro exponencial para a PV, em segundos.	Float	S/RW	Somente valores positivos.	0
FSAFE_TYPE	17	Define o tipo de ação do equipamento na condição de Fail Safe.	Un-signed8	S/RW	0:Use FSAFE_VALUE 1:Use Last Usable Value 2:Use Wrong Value	1
FSAFE_VALUE	18	Valor de segurança para a saída quando está ativo o Fail Safe.	Float	S/RW	OUT_SCALE	0

ALARM_HYS	19	Parâmetro de histerese de alarme. Para sair da condição de alarme, o valor da PV deve retornar dentro dos limites do alarme mais a histerese.	Float	S/RW	0 a 50 %	0.5%
HI_HI_LIM	21	O ajuste para o alarme superior em unidades de Engenharia.	Float	S/RW	OUT_SCALE, +INF	INF
HI_LIM	23	O ajuste para o alarme superior em unidades de Engenharia.	Float	S/RW	OUT_SCALE, +INF	INF
LO_LIM	25	O ajuste para o alarme inferior em unidades de Engenharia.	Float	S/RW	OUT_SCALE, -INF	INF
LO_LO_LIM	27	O ajuste para o alarme inferior em unidades de Engenharia.	Float	S/RW	OUT_SCALE, -INF	INF
HI_HI_ALM	30	O estado do alarme superior extremo.	DS-39	D/RW		
HI_ALM	31	O estado do alarme superior.	DS-39	D/RW		
LO_ALM	32	O estado do alarme inferior.	DS-39	D/RW		
LO_LO_ALM	33	O estado do alarme inferior extremo.	DS-39	D/RW		
SIMULATE	34	Permite que o valor do transdutor seja fornecido manualmente pelo usuário quando habilitado o modo simulação.	DS-50	S/RW	0: Disable; [1]0: Enable.	Disable
OUT_UNIT_TEXT	35	É usado quando a unidade desejada pelo usuário não está na lista de códigos de unidades. O usuário pode fornecer uma unidade em texto, a chamada unidade de usuário	Oct-String(16)	S/RW		

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW - Escrita/Leitura; RO – Somente leitura; D – dinâmico; N – Não-volátil; S – Estático; INF – Infinito

Tabela 7 – Parâmetros do Bloco AI.

ANALOG OUTPUT BLOCK – AO

O Bloco AO pode receber um valor de *setpoint* via comunicação cíclica ou mesmo via usuário quando seu modo de operação for automático (Auto). Neste caso, recebe valor e status através do parâmetro SP. Nesta condição é importante ressaltar que o status deve ser maior ou igual a *good* (0x80).
Quando seu modo de operação estiver em cascata (RCas), o valor e status do *setpoint* será fornecido no parâmetro RCAS_IN pelo mestre Profibus Classe 1, através da comunicação cíclica. Nesta condição, o status deve ser igual a *IA-Initialization Acknowledge* (0xC4).
Este valor de *setpoint* é processado de acordo com o algoritmo do bloco e disponibilizado ao bloco transdutor (TRD) que atuará no elemento final de controle, como por exemplo, em um posicionador de válvula.
O valor de entrada deve estar de acordo com a escala de entrada, configurada no parâmetro PV_SCALE. A saída do bloco AO é fornecida ao bloco TRD de acordo com a escala de saída, configurada no parâmetro OUT_SCALE.

AUMENTAR PARA FECHAR (INCREASE TO CLOSE)

O parâmetro INCREASE_CLOSE permite que a saída seja invertida em relação ao *span* de entrada. Define o movimento do atuador em relação ao *setpoint*:

- 0: crescente (aumento do *setpoint* leva à abertura da válvula);
- 1: decrescente (aumento do *setpoint* leva ao fechamento da válvula).

SIMULAÇÃO

O bloco AO permite ainda que o usuário simule um valor, através do parâmetro SIMULATE. Este parâmetro é usado para facilitar testes, por exemplo, quando se está fazendo o *startup* de malhas de controle. Quando habilitado, os valores do bloco transdutor e status serão sobrepostos pelo valor e status simulados pelo usuário.Veja a seguir o parâmetro READBACK.

A estrutura SIMULATE é composta pelos seguintes atributos:

- Simulate Value and Status
- Simulate Enable

READBACK

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

Se houver a condição de retorno via hardware, então tem-se a condição de *readback*, como a posição da válvula, e nesta condição o valor será lido pelo bloco transdutor. Se não houver tal condição de retorno, o valor/estado do bloco TRD é gerado usando-se o parâmetro AO_OUT.

Quando a simulação estiver habilitada, valor e status de *readback* serão fornecidos de acordo com o parâmetro *SIMULATE*. Caso contrário, serão aqueles fornecidos pelo bloco transdutor.

O Bloco AO suporta os seguintes modos:

- OOS *(Out of Service* – fora de serviço): neste modo, a saída não é atualizada pelo algoritmo do bloco.
- Man (Manual): neste modo, o usuário pode escrever valores na saída do bloco.
- Auto (Automático): neste caso, a saída do bloco é atualizada pelo algoritmo do bloco. Neste modo, tanto o controlador quanto o usuário podem escrever no SP.
- RCas(Cascata): neste caso o *setpoint* é calculado pelo controlador e enviado via comunicação cíclica ao parâmetro RCAS_IN. Uma transição do modo para RCas requer a execução de uma máquina de estado para que o bloco AO possa rodar adequadamente. Se RCas é o AO_TARGET_MODE, então através do parâmetro RCAS_OUT, o bloco setará um status igual a *Good (C) Initialization Request* e o controlador enviará o status do RCAS_IN igual a *Good (C) Initialization Acknowledge,* possibilitando que o parâmetro AO_MODE_BLK mude para RCas.
- LO (*Local Override*): Quando o bloco está em LO, a saída segue o valor estabelecido pelo usuário localmente (através de atuações locais, ajuste local). O usuário não pode alterar as saídas do *host* remoto.

TRATAMENTO DE ERROS – FALHA SEGURA

O bloco AO possui o mesmo tratamento de falha detalhado anteriormente para o bloco AI (veja no item anterior).

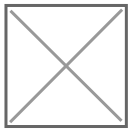


Figura 12 – Resumo dos parâmetros do Bloco AO.

Figura 13 – Bloco AO.

BLOCO DE SAÍDA ANALÓGICA – AO
(Equipamentos com somente 1 AO: slot 1, se 2 AOs: Slot 1 e 2, se 3 AOs: slot, 1, 2 e 3)

Parâmetro	Índice Relativo	Descrição	Tipo de dado	Armazenamento em memória e acesso R/W	Faixa de valores	Pa (D
ST_REV	1	Este parâmetro será sempre incrementado quando ocorrer mudanças em parâmetros estáticos do bloco.	Un-signed16	S/RO		0
TAG_DESC	2	*Tag* do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW		Es

STRATEGY	3	É um valor fornecido pelo usuário para identificar uma configuração.	Un-signed16	S/RW		0
ALERT_KEY	4	Estado atual de alarmes.	Un-signed8	S/RW	1 a 255	0
TARGET_MODE	5	Contém o modo de operação desejado para o bloco.	Un-signed8	S/RW	O/S, MAN, AUTO, RCAS e LO	O/
MODE_BLK	6	Modo de operação corrente do bloco.	DS-37	D/RO		
ALARM_SUM	7	Contém os estados presentes dos alarmes de bloco.	DS-42	D/ RO		0,0
BATCH	8	Para uso em sistema distribuído. É utilizado para identificar canais usados e disponíveis. Não há algoritmo relacionado.	DS-67	S/RW		0,0
SP	9	*Setpoint* utilizado em modo automático e que ainda pode ser alterado pelo usuário.	DS-33	N/Auto	PV_SCALE	
PV_SCALE	11	Valores de escala Eua100% e EU0% para o parâmetro SP.	DS-36	S/RW		0-1
READBACK	12	Indica o *readback* da posição atual do bloco transdutor.	DS-33	D/RO	PV_SCALE	
RCAS_IN	14	Valor de *setpoint* remoto fornecido pelo controlador quando em modo RCas.	DS-33	D/RW		
IN_CHANNEL	21	O número do canal de hardware lógico do transdutor que é conectado ao bloco AO.	Un-signed16	S/RW		0
OUT_CHANNEL	22	O número do canal de hardware lógico para o transdutor que é conectado ao bloco AO.	Un-signed16	S/RW		0
FSAFE_TIME	23	Período da detecção da falha até a ação do bloco, se a falha segura estiver configurada.	Float	S/RW	Valores maiores que zero	0
FSAFE_TYPE	24	Define a ação do equipamento em uma condição de falha segura.	Un-signed8	S/RW	0:Use FSAFE_VALUE 1:Use Last Usable Value 2:Goes to ACTUATOR_ACTION position	1: La Us Va

FSAFE_VALUE	25	Valor de falha segura para a saída do bloco AO, quando o *Fail Safe* estiver ativo e FSAFE_TYPE = 0.	Float	S/RW	OUT_SCALE	0
RCAS_OUT	27	O valor e status requisitado por um bloco no controlador a fim de evitar *reset windup* e proporcionar transferência sem interferência em uma malha fechada de controle.	DS-33	D/RO		
POS_D	31	Posição discreta da válvula.	DS-34	D/RO	0: not initialized 1: closed 2: opened 3: intermediate	0
SETP_DEVIATION	32	Diferença entre SP e o *Readback*.	Float	D/RO		0
CHECK_BACK	33	Informação do estado do equipamento. Ver opções de *Check Back*.	Oct-String(3)	D/RO		0
CHECK_BACK_MASK	34	Informações de *CheckBack*. suportadas	Oct-String(3)	S/RO		0
SIMULATE	35	Permite que o valor de readback seja fornecido manualmente pelo usuário.	DS-50	S/RW	0: Disable ¹0: Enable	Di
INCREASE_CLOSE	36	Direção da atuação quando em modo automático.	Un-signed8	S/RW	0: Rising 1: Falling	Ri
OUT	37	A saída do bloco para o bloco transdutor.	DS-33	N/Man/RW	OUT_SCALE	
OUT_SCALE	38	Valores de escala Eu100% e Eu0% para o bloco transdutor para um canal especificado.	DS-36	S/RW		0-1

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW - Escrita/Leitura; RO – Somente leitura; D
Dinâmico;
N – Não-volátil; S – Estático; INF – Infinito

Tabela 8 – Parâmetros do Bloco AO.

TOTALIZER BLOCK – TOT

O bloco Totalizador, assim como o bloco AI, recebe um valor de processo via canal com o bloco transdutor. Este valor é totalizado ao longo do tempo, por exemplo em medições de vazão mássica ou volumétrica, comuns em transmissores de pressão. Posteriormente, veremos as condições iniciais de configuração para que o bloco TOT tenha seu adequado funcionamento.

O bloco TOT suporta os seguintes modos de operação:

- Auto (Automático): neste modo a saída do bloco é calculada e disponibilizada via comunicação cíclica ao controlador, mestre Profibus Classe 1;

OOS (*Out of Service*): neste modo, o algoritmo do bloco não é processado.

Figura 14 - Algoritmo do Bloco TOT.

O algoritmo do bloco TOT é aplicado ao valor medido pelo bloco transdutor quando o modo estiver em Auto. Este algoritmo inclui o tratamento de erros e falhas, assim como visto para os blocos anteriores (veja detalhes na explicação para o bloco AI), seleção de sinais do valor a ser totalizado, bem como o tratamento de alarmes.

O bloco TOT totaliza a variável de processo em função do tempo e sua unidade é fornecida pelo bloco TRD. Seu algoritmo converte as unidades de tempo em unidades por segundo.

TOTALIZAÇÃO DA VAZÃO E SINAIS

A seleção do sinal dos valores totalizados é controlada pelo parâmetro MODE_TOT. A vazão resultante é obtida pela somatória dos valores, considerando-se seus sinais e o que for configurado no MODE_TOT:

- *Balanced* – Os valores negativos e positivos serão totalizados;
- *Positive only* – somente valores positivos são totalizados. Os valores negativos serão considerados como zero;
- *Negative only* – somente valores negativos são totalizados. Os valores positivos serão considerados como zero;
- *Hold* – O algoritmo é mantido com valor constante.

Figura 15 - Função MODE_TOT.

O parâmetro TOTAL é a quantidade totalizada pelo bloco, cuja unidade está de acordo com o parâmetro UNIT_TOT e deve ser compatível com a unidade da entrada, fornecida pelo bloco TRD.

RESET E PRESET

A totalização pode ser configurada pelo parâmetro SET_TOT, onde o usuário pode zerá-la com a opção

Reset, inicializá-la com um valor pré-definido pelo parâmetro PRESET_TOT, ao escolher a opção *Preset*, ou ainda iniciar a totalização, configurando o SET_TOT para a opção *Totalize*.

Figura 16 - Função SET_TOT.

CONDIÇÕES INICIAIS DE CONFIGURAÇÃO

Para que o bloco TOT possa funcionar adequadamente, o usuário deve fazer algumas configurações iniciais:

- O parâmetro CHANNEL deve ser configurado para PV, ou seja, o valor de processo servirá como valor entrada, vindo do bloco TRD;
- O parâmetro LINEARIZATION_TYPE do bloco TRD deve ser configurado para "*Square Root*";
- O parâmetro PRIMARY_VALUE_UNIT do bloco TRD deve ser configurado para uma unidade adequada de vazão;
- O parâmetro UNIT_TOT do bloco TOT deve ser configurado para uma unidade de massa ou volume equivalente à unidade de entrada, selecionada no parâmetro PRIMARY_VALUE_ UNIT do bloco TRD.

Figura 17 – Resumo dos parâmetros do Bloco TOT.

Figura 18 – Bloco TOT.

BLOCO TOTALIZADOR - TOT
(Equipamentos com somente 1 TOT: slot 2, se 2 TOTs: Slot 2 e 3, se 3 TOTs: slot 2, 3 e 4)

Parâmetro	Índice Relativo	Descrição	Tipo de dado	Armazenamento em memória e acesso R/W	Faixa de valores	Padrão (Default)
ST_REV	1	Este parâmetro será incrementado sempre que houver mudanças nos parâmetros estáticos do bloco.	Unsigned16	S/RO		0
TAG_DESC	2	TAG do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW		Espaços
STRATEGY	3	É um valor fornecido pelo usuário para identificar uma configuração.	Unsigned16	S/RW		0

ALERT_KEY	4	Estado atual de alarmes.	Unsigned8	S/RW	1 to 255	0
TARGET_MODE	5	Contém o modo desejado para o bloco.	Unsigned8	S/RW	O/S, AUTO.	AUTO
MODE_BLK	6	Modo corrente do bloco.	DS-37	D/RO		
ALARM_SUM	7	Contém os estados dos alarmes do bloco.	DS-42	D/RO		0,0,0,0
BATCH	8	Para uso em sistema distribuído para identificar canais usados e disponíveis. Não há algoritmo relacionado.	DS-67	S/RW		0,0,0,0
TOTAL	10	Saída do bloco TOT.	DS-33	N/RO		
UNIT_TOT	11	Unidade de engenharia da saída.	Unsigned16	S/RW		
CHANNEL	12	O número do canal lógico de hardware do bloco transdutor que é conectado ao TOT.	Unsigned16	S/RW		0
SET_TOT	13	Reinicia a saída do TOT, ou a ajusta de acordo com o valor em PRESET_TOT. E ainda, permite que o bloco totalize.	UnSigned8	N/RW	0: Totalize 1: Reset 2: Preset	Totalize
MODE_TOT	14	Define o tipo de sinal dos valores para a totalização (positiva, negativa ou último valor).	UnSigned8	N/RW	0: Balanced 1: Positive only 2: Negative only 3: Hold	Balanced
FAIL_TOT	15	Define a ação do bloco TOT na condição de *fail safe*.	UnSigned8	S/RW	0: Run 1: Hold 2: Memory	Run
PRESET_TOT	16	Valor da saída quando o SET_TOT for configurado para Preset.	Float	S/RW		0
ALARM_HYS	17	Parâmetro de histerese de alarme. Para sair da condição de alarme, o valor da PV deve ser inferior ao limite do alarme somado à histerese.	Float	S/RW		0
HI_HI_LIM	18	Ajuste de alarme muito alto em unidades de Engenharia.	Float	S/RW		INF
HI_LIM	19	Ajuste de alarme alto em unidades de Engenharia.	Float	S/RW		INF
LO_LIM	20	Ajuste de alarme baixo em unidades de Engenharia.	Float	S/RW		INF
LO_LO_LIM	21	Ajuste de alarme muito baixo em unidades de Engenharia.	Float	S/RW		INF
HI_HI_ALM	22	Estado de alarme muito alto.	DS-39	D/RW		
HI_ALM	23	Estado de alarme alto.	DS-39	D/RW		
LO_ALM	24	Estado de alarme baixo	DS-39	D/RW		
LO_LO_ALM	25	Estado de alarme muito baixo.	DS-39	D/RW		

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW – Escrita/Leitura; RO – Somente leitura; D – Dinâmico;
N – Não-volátil; S –Estático; INF – Infinito

Tabela 9 – Parâmetros do bloco TOT.

DIGITAL INPUT BLOCK – DI

O Bloco DI recebe um valor digital do processo através do bloco transdutor (TRD) em seu canal e o processa de acordo com sua configuração, disponibilizando ao mestre Profibus Classe 1, através do parâmetro OUT_D. Possui modo de operação, inversão, simulação e tratamento de erros.

O Bloco AI suporta os seguintes modos:

- OOS (*Out of Service* – fora de serviço): neste modo, a saída não é atualizada pelo algoritmo do bloco;
- Man (Manual): neste modo, o operador pode escrever valores na saída do bloco;
- Auto (Automático): neste modo, a saída do bloco é resultado do processamento do bloco.

O bloco DI permite ainda que o usuário simule um valor digital de processo, através do parâmetro SIMULATE_D. Este parâmetro é usado para facilitar testes, por exemplo, quando se está fazendo o *startup* de malhas de controle

A estrutura SIMULATE_D é composta pelos seguintes atributos:

- Simulate Value and Status
- Simulate Enable

Figura 19 – Simulação no bloco DI.

Quando ativo, o valor do bloco transdutor e seu status serão sobrepostos pelo valor e status simulados. Quando desabilitado, o valor de processo será disponibilizado pelo bloco transdutor ao bloco DI.

O bloco DI ainda permite através do parâmetro INVERT, que o nível lógico da entrada seja invertido, o que também acontece durante a simulação. Com o INVERT ativo, o parâmetro PV_D será invertido antes da saída OUT_D ser atualizada.

TRATAMENTO DE ERROS E FALHAS

O algoritmo do bloco DI inclui o tratamento de erros e falhas, assim como visto para os blocos anteriores (veja detalhes na explicação para o bloco AI):

Figura 20 – Falha segura no bloco DI.

Quando o bloco DI detecta uma condição de anormalidade, o modo de falha segura (FAIL_SAFE) é ativado, caso esteja configurado.

O bloco DI recebe (via canal) valores e status do bloco TRD. Caso receba uma condição de status "*bad*" (ruim), por exemplo, numa falha de leitura de um sensor *on/off*, a condição de FAIL_SAFE do bloco é ativada, se estiver configurada. Quando da condição de falha segura, as possíveis ações a serem realizadas pelo bloco podem ser selecionadas pelo usuário através do parâmetro *FSAFE_TYPE*.

No parâmetro *FSAFE_TYPE* as seguintes opções estão disponíveis:

- *FSAFE_VAL_D* – o bloco DI usa o valor discreto de segurança configurado pelo usuário no parâmetro *FSAFE_VAL_D* como valor no parâmetro *OUT_D*, quando o *Fail Safe* estiver ativo. O status da saída vai para *"Uncertain, substitute value";*
- *Last Usable Value* – o bloco DI usa o último valor com status *good* (bom) para o cálculo de sua saída. O status será "*Uncertain Last Usable Value*". Se, ao ocorrer a falha ainda não houver um valor adequado, usará o Valor Inicial na saída. O status será "*Uncertain Initial Value*";
- *Wrong Value* – o bloco DI usa status e valor vindos do bloco transdutor (TRD).

Figura 21 – Resumo dos parâmetros do Bloco DI.

Figura 22 – Bloco DI.

BLOCO DIGITAL DE ENTRADA - DI

(Equipamentos com somente 1 DI slot 1, se 2 DIs: Slot 1 e 2, se 3 DIs: slot 1, 2 e 3)

Parâmetro	**Índice Relativo**	**Descrição**	**Tipo de dado**	**Armazenamento em memória e acesso R/W**	**Faixa de valores**	**Padrão (Default)**
ST_REV	1	Este parâmetro será incrementado sempre que houver mudanças nos parâmetros estáticos do bloco.	Unsigned16	S/RO		0
TAG_DESC	2	TAG do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW		Espaços
STRATEGY	3	Valor fornecido pelo usuário para identificar uma configuração.	Unsigned16	S/RW		0
ALERT_KEY	4	Estado atual de alarmes.	Unsigned8	S/RW	1 to 255	0

TARGET_MODE	5	Contém o modo desejado para o bloco.	Unsigned8	S/RW	O/S, AUTO.	AUTO
MODE_BLK	6	Modo corrente do bloco.	DS-37	D/RO		
ALARM_SUM	7	Contém os estados dos alarmes do bloco.	DS-42	D/RO		0,0,0,0
BATCH	8	Para uso em sistema distribuído para identificar canais usados e disponíveis. Não há algoritmo relacionado.	DS-67	S/RW		0,0,0,0
OUT_D	10	Saída discreta do bloco DI.	DS-34	S/RW	0-1	
CHANNEL	14	Canal de hardware associado ao bloco TRD.	Unsigned16	S/RW		
INVERT	15	Indica se a PV_D deve ser invertida logicamente, antes de ser atribuída à saída OUT_D	Unsigned 8	S/RW	0: not inverted 1: inverted	0
FSAVE_TYPE	20	Permite configurar o tipo de *fail safe*.	Unsigned 8	S/RW	0: o valor em FSAFE_VAL_D é usado em OUT_D Status = UNCERTAIN-substitute value (**) 1: usar o último valor válido armazenado em OUT_D Status = UNCERTAIN-last usable value (se não houver nenhum valor válido, usará UNCERTAIN-Initial Value) 2: OUT_D usará o valor e status que vem do bloco TRD, mesmo que seja inadequado(Bad) Status	1
FSAVE_VAL_D	21	Valor discreto utilizado como valor de segurança quando o FAIL_SAFE_TYPE for 0.	Unsigned 8	S/RW		0
SIMULATE	24	Pemite que seja simulado um valor discreto de entrada	DS-51	S/RW	0: disable #0: enable	disable

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW - Escrita/Leitura; RO – Somente leitura; D – dinâmico; N – não-volátil;
S – Estático; INF – Infinito

Tabela 10 – Parâmetros do bloco DI.

DIGITAL OUTPUT BLOCK – DO

O Bloco DO pode receber um valor discreto de setpoint (SP_D) via comunicação cíclica ou via usuário, quando seu modo de operação for automático (Auto). Neste caso, recebe valor e status através do parâmetro SP_D. Nesta condição é importante ressaltar que o status deve ser maior ou igual a *good* (0x80).

Quando seu modo de operação estiver em cascata (RCas), o valor e status do setpoint são fornecidos via parâmetro RCAS_IN_D pelo mestre Profibus Classe 1, através da comunicação cíclica. Nesta condição, o status deve ser igual a *IA-Initialization Acknowledge* (0xC4).

Este valor de *setpoint* é processado de acordo com o algoritmo do bloco e disponibilizado ao bloco transdutor (TRD) que atuará no elemento final de controle, como por exemplo, acionando uma válvula *on/off*. Similar ao bloco AO, o bloco DO suporta os seguintes modos de operação: O/S, MAN, RCAS, LO e AUTO.

Assim como nos demais blocos, o bloco DO permite a simulação. Este parâmetro é usado para facilitar testes, por exemplo, quando se está fazendo o *startup* de malhas de controle.

A estrutura SIMULATE_D é composta pelos seguintes atributos:

- Simulate Value and Status
- Simulate Enable

Figura 23 – Simulação no bloco DO.

O bloco DO fornece o valor discreto, assim como o status ao bloco TRD através do parâmetro READBACK_D. Similarmente ao bloco DI, possui o parâmetro INVERT, onde pode-se inverter o valor discreto do *setpoint*, quer seja em modo Auto ou RCas.

TRATAMENTO DE ERROS - FALHA SEGURA

Assim como nos demais blocos, o bloco DO também possui tratamento de erros e falhas. Se há perda de comunicação cíclica com o mestre Profibus Classe 1 e o *setpoint* discreto não for mais atualizado, pode-se configurar o tipo de tratamento que deve acontecer com a saída do bloco DO, através do parâmetro FSAFE_TYPE.

A saída irá para a condição de falha segura configurada se a situação de anormalidade permanecer por um período de tempo maior que o configurado no parâmetro FSAFE_TIME. Após a normalização da condição, o bloco DO retornará à operação normal.

Figura 24 – Tratamento do *Fail Safe* no Bloco DO.

A condição de FAIL_SAFE_D é ativada quando uma das seguintes condições é detectada:

- Perda de comunicação na entrada RCAS_IN_D por um tempo superior ao especificado no parâmetro FSAFE_TIME;
- Perda de comunicação na entrada SP_D por um tempo superior ao especificado no parâmetro FSAFE_TIME;
- Em modo RCas, receber o parâmetro RCAS_IN_D com statusIFS (*initiate fail-safe*);
- Em modo Auto, receber o parâmetro SP_D com statusIFS (*initiate fail-safe*).

No parâmetro *FSAFE_TYPE* as seguintes opções estão disponíveis:

- *FSAFE_VALUE_D* – o bloco DO usa o valor de segurança configurado pelo usuário no parâmetro *FSAFE_VAL_D* como valor no parâmetro *OUT_D*, quando o *Fail Safe* estiver ativo. O status da saída vai para *"Uncertain, substitute value";*
- *Last Usable Value* – o bloco DO usa o último valor com status *good* (bom) para o cálculo de sua saída *OUT_D*. O status será "*Uncertain Last Usable Value*". Se ao ocorrer a falha ainda não houver um valor adequado, usará o Valor Inicial na saída. O status será "*Uncertain Initial Value*";
- ACTUATOR_ACTION – o bloco DO entra na condição de segurança baseado no parâmetro ACTUATOR_ACTION do bloco transdutor.

Figura 25– Resumo dos parâmetros do Bloco DO.

Figura 26– Bloco DO.

BLOCO DE SAÍDA DIGITAL - DO

(Equipamentos com somente 1 DO slot 1, se 2 DOs: Slot 1 e 2, se 3 DOs: slot 1, 2 e 3)

Parâmetro	**Índice Relativo**	**Descrição**	**Tipo de dado**	**Armazenamento em memória e acesso R/W**	**Faixa de valores**	**Padrão (Defaul**
ST_REV	1	Este parâmetro será incrementado sempre que houver mudanças nos parâmetros estáticos do bloco.	Unsigned16	S/RO		0

TAG_DESC	2	TAG do bloco. Este parâmetro deve ser único na configuração.	Oct-String(32)	S/RW		Espaços
STRATEGY	3	É um valor fornecido pelo usuário para identificar uma configuração.	Unsigned16	S/RW		0
ALERT_KEY	4	Estado atual de alarmes	Unsigned8	S/RW	1 to 255	0
TARGET_MODE	5	Contém o modo desejado para o bloco.	Unsigned8	S/RW	O/S, AUTO.	AUTO
MODE_BLK	6	Modo corrente do bloco	DS-37	D/RO		
ALARM_SUM	7	Contém os estados dos alarmes do bloco.	DS-42	D/RO		0,0,0,0
BATCH	8	Para uso em sistema distribuído para identificar canais usados e disponíveis. Não há algoritmo relacionado.	DS-67	S/RW		0,0,0,0
SP_D	9	Setpoint do bloco de função usado em MODE AUTO.	DS-34	D/RW	0 e 1	-
OUT_D	10	Saída discreta do bloco DO	DS-34	D/RW	0 e 1	-
READBACK_D	12	Pode ser a posição atual(discreta) do elemnto final de controle e seus sensores.	DS-34	D/RO	0 e 1	-
RCAS_IN_D	14	Setpoint e status desejado fornecido pelo controlado mestre Profibus ao bloco DO quando o modo de operação for RCas.	DS-34	D/RW	0 e 1	-
CHANNEL	17	Canal de hardware do bloco TRD associado ao bloco DO	Unsigned16	S/RW		-
INVERT	18	Utilizado para inverte o setpoint fornecido ao bloco DO: 0 = não inverter 1 = inverter	Unsigned 8	S/RW	0 e 1	Não inverter
FSAFE_TIME	19	Período da detecção da falha (SP_D =Bad ou RCAS_IN <> Good) até que haja ação do bloco, se a condição de falha segura existir.	Float	S/RW	maior que zero	0
FSAFE_TYPE	20	Define a ação do equipamento se houver uma falha e após FSAFE_TIME: 0 = valor FSAVE_VALUE é usado como status do setpoint do OUT_D = UNCERTAIN – Valor substituto; 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	Unsigned 8	S	0,1 e 2	2

FSAFE_VAL_D	21	Valor discreto utilizado na condição de falha segura, caso o FSAFE_TYPE seja configurado para FSAFE_VAL_D.	Unsigned 8	S/RW	0 e 1	0
RCAS_OUT_D	22	Valor discreto e status que retorna do bloco DO ao controlador.	DS-34	D/RO	0 e 1	-
SIMULATE	24	Para comissionamento e razões de manutenção, é possível simular o valor do READBACK, definindo-se o valor e status. Nesta condição de simulação, o valor do bloco DO é desprezado.	DS-51	S/RW	0 e 1, enable;disable	Disable

Legenda: E – Parâmetro Enumerado; NA – Parâmetro Adimensional; RW - Escrita/Leitura; RO – Somente leitura; D Dinâmico; N – Não-volátil; S – Estático; INF – Infinito

Tabela 11 – Parâmetros do bloco DO.

MODOS DE OPERAÇÃO DOS BLOCOS FUNCIONAIS

A tabela a seguir mostra os valores hexadecimais para os parâmetros TARGET_MODE e MODE_BLK (e para os seus elementos Actual, Permitted e Normal).

Modo	**Valor (HEX)**
Out of Service: O/S	0x80
Local Override: LO	0x20
Manual: Man	0x10
Automático: Auto	0x08
Cascata: Rcas	0x02

Tabela 12 – Mode Block.

GERAÇÃO DE STATUS

Uma das vantagens da tecnologia digital é poder atribuir, além do valor da medição, condições do estado das variáveis, indicando a qualidade, sub-qualidade e limites. A tabela a seguir detalha o byte de status:

Byte de Status							
Bit 2^7	Bit 2^6	Bit 2^5	Bit 2^4	Bit 2^3	Bit 2^2	Bit 2^1	Bit 2^0
Qualidade		Sub-qualidade				Limites	

Tabela 13 – Byte de Status.

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

- *Good Cascade* – A qualidade do valor é boa e pode ser utilizado para controle em cascata.
- *Good Non-cascade* – A qualidade do valor é boa e não pode ser utilizado para controle em cascata.
- *Uncertain* – A qualidade do valor está abaixo do normal, mas o valor ainda o valor pode ser usado.
- *Bad* – O valor não é útil.

Bad	$2^{7,6}$ = 00, *bad value*	O valor não é útil.
Uncertain	$2^{7,6}$ = 01, *uncertain value*	A qualidade do valor está abaixo do normal, mas o valor ainda o valor pode ser usado.

Good	$2^{7,6}$ = 10, *good value*	O valor é bom

Tabela 14 – Qualidade.

Bad	*Device Failure*: 2^5 a 2^2 = 0010 *Sensor Failure*: 2^5 a2^2 = 0100 *No Communication*: 2^5 a2^2 = 0101
Uncertain	*Last Usable Value*: 2^5 a 2^2 = 0001 – é utilizado quando o valor não está mais sendo atualizado. *Sensor Conversion not Accurate*: 2^5 a 2^2 = 0100 – é utilizado quando o valor do sensor está fora de seus limites.
Good	*Ok*: 2^5 a 2^2 = 0000 – o valor está bom. *Active Advisory Alarm*: 2^5 a 2^2 = 0010 – é utilizado quando o valor excede ou cai abaixo do valor de aviso de alarme.

Tabela 15 – Sub-status.

Not limited	$2^{1,0}$ =00- not limited – o valor medido está dentro de seus limites aceitáveis.
Low limited	$2^{1,0}$ =01- Lower limit – o valor medido está abaixo do limite inferior aceitável.
High limited	$2^{1,0}$ =10- Upper limit – o valor medido está acima do limite inferior aceitável.
Constant	$2^{1,0}$ =11- Constant – o valor medido não está sendo variado com o processo, por exemplo, está em manual.

Tabela 16 – Limites.

Sub-Qualidade – é o sub-status da qualidade, ou seja, 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 (Tabela 15).

Limites – 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* (Tabela 16).
Vejamos a seguir alguns valores do byte de status:

Qualidade	Sub-Status	Limite	Valor Hexa	Valor Decimal
GoodNC	**0 = ok**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x80**	**128**
GoodNC	**1 = Active Update Event**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x84**	**132**
GoodNC	**2 = Active Advisory Alarm**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x88**	**136**
GoodNC	**3 = Active Critical Alarm**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x8C**	**140**
GoodNC	**4 = Unacknowledged Update Event**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x90**	**144**
GoodNC	**5 = Unacknowledged Advisory Alarm**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x94**	**148**

GoodNC	6 = Unacknowledged Critical Alarm	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x98	152
GoodNC	8 = Initiate Fail Safe (IFS)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xA0	160
GoodNC	9 = Maintenance required	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xA4	164
Uncertain	0 = Non-specific	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x40	64
Uncertain	1 = Last Usable Value	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x44	68
Uncertain	2 = Substitute	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x48	72
Uncertain	3 = Initial Value	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x4C	76
Uncertain	4 = Sensor Conversion not Accurate	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x50	80
Uncertain	5 = Engineering Unit Range Violation	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x54	84
Uncertain	6 = Sub-normal	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x58	88
Uncertain	7 = Configuration Error	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x5C	92
Uncertain	8 = Simulated Value	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x60	96
Uncertain	9 = Sensor Calibration	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x64	100
GoodC	0 = ok	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xC0	192
GoodC	1 = Initialization Acknowledged (IA)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xC4	196
GoodC	2 = Initialization Request (IR)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xC8	200
GoodC	3 = Not Invited (NI)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xCC	204
GoodC	5 = Do Not Selected (NS)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xD4	212
GoodC	6 = Local Override (LO)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xD8	216
GoodC	8 = Initiate Fail Safe (IFS)	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0xE0	224
Bad	0 = Non-specific	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x00	0
Bad	1 = Configuration Error	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x04	4
Bad	2 = Not Connected	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x08	8
Bad	3 = Device Failure	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x0C	12
Bad	4 = Sensor Failure	0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant	0x10	16

Bad	**5 = No Communication, with last usable value**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x14**	**20**
Bad	**6 = No Communication, with no usable value**	**0 – Not Limited; 1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x18**	**24**
Bad	**7 = Out of Service**	**0 – Not Limited;1 – Low Limited; 2 – High Limited; 3 - Constant**	**0x1C**	**28**

Tabela 17 – Tabela de Status.

DEFINIÇÃO E TIPOS DE ESTRUTURA DE DADOS

A seguir estão detalhadas as estruturas e tipos de dados utilizados no Profibus-PA.

Código	Tipo de Dado	Tamanho	Descrição
1	*Boolean*	1	Verdadeiro ou falso
2	*Integer*8	1	Inteiro de 8 bits
3	*Integer*16	2	Inteiro de 16 bits
4	*Integer*32	4	Inteiro de 32 bits
5	*Unsigned*8	1	Inteiro sem sinal de 8 bits
6	*Unsigned* 16	2	Inteiro sem sinal de 16 bits
7	*Unsigned* 32	4	Inteiro sem sinal de 32 bits
8	*Floating Point*	4	Ponto flutuante IEEE 754
9	*VisibleString*	1,2,3,...	Um byte por caractere e inclui os caracteres ASCII de 7 bits.
10	*OctetString*	1,2,3,…	*Octetstring*s são bináros
-	Dados	-	
-	*TimeofDay*	-	
-	*TimeDifferenc*e	-	
-	*BitString*	-	
-	*DataTimeValu*e	-	

Tabela 18 – Definição e Tipos de Estrutura de Dados.

BLOCK OBJECT - DS-32

Esta estrutura de dados consiste nos atributos de um bloco.

E	Nome do elemento na estrutura	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

Tabela 19 – Block Object DS-32.

VALOR & STATUS - ESTRUTURA DO PONTO FLUTUANTE - DS-33

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Value*	*Float*	4
2	*Status*	*Unsigned*8	1

Tabela 20 – Estrutura do Ponto Flutuante DS-33.

VALOR & STATUS - ESTRUTURA DISCRETA - DS-34

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Value*	*Unsigned*8	1
2	*Status*	*Unsigned*8	1

Tabela 21 – Estrutura Discreta DS-34.

ESTRUTURA DE ESCALA - DS-36

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	EU at 100%	*Float*	4
2	EU at 0%	*Float*	4
3	*Units Index*	*Unsigned*16	2
4	*Decimal Point*	Integer8	1

Tabela 22 – Estrutura de Escala DS-36.

ESTRUTURA DE MODO - DS-37

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

Tabela 23 – Estrutura de Modo DS-37.

ESTRUTURA DE ALARME DO PONTO FLUTUANTE - DS-39

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Unacknowledged*	*Unsigned*8	1
2	*Alarm State*	*Unsigned*8	1
3	*Time Stamp*	*Time Value*	8
4	*Subcode*	*Unsigned*16	2
5	*Value*	*Float*	4

Tabela 24 – Estrutura de Alarme de Ponto Flutuante DS-39.

ESTRUTURA DE ALARME DISCRETA - DS-40

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Unacknowledged*	*Unsigned*8	1

2	*Alarm State*	*Unsigned*8	1
3	*Time Stamp*	*Time Value*	8
4	*Subcode*	*Unsigned*16	2
5	*Value*	*Float*	4

Tabela 24 – Estrutura de Alarme de Ponto Flutuante DS-39.

ESTRUTURA DE ALARME DISCRETA - DS-40

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Unacknowledged*	*Unsigned*8	1
2	*Alarm State*	*Unsigned*8	1
3	*Time Stamp*	*Time Value*	8
4	*Subcode*	*Unsigned*16	2
5	*Value*	*Unsigned*8	1

Tabela 25 – Estrutura de Alarme Discreta DS-40.

ESTRUTURA DE ATUALIZAÇÃO DO ALARME - DS-41

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Unacknowledged*	*Unsigned*8	1
2	*Update State*	*Unsigned*8	1
3	*Time Stamp*	*Time Value*	8
4	*Subcode*	*Unsigned*16	2
5	*Relative Index*	*Unsigned*16	2

Tabela 26 – Estrutura de Atualização de Alarme DS-41.

ESTRUTURA ÍNDICE DE ALARME - DS-42

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Current*	*Bit String*	2
2	*Unacknowledged*	*Bit String*	2
3	*Unreported*	*Bit String*	2
4	*Disabled*	*Bit String*	2

Tabela 27 – Estrutura do Índice de Alarme.

SIMULATE – ESTRUTURA DO PONTO FLUTUANTE - DS-50

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Simulate Status*	*Unsigned*8	1
2	*Simulate Value*	*Float*	4
3	*Simule En/Disable*	*Unsigned*8	1

Tabela 28 – Simulate - Estrutura do Ponto Flutuante DS-50.

SIMULATE – ESTRUTURA DISCRETA - DS-51

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*Simulate Status*	*Unsigned*8	1
2	*Simulate Value*	*Unsigned*8	1
5	*Simule En/Disable*	*Unsigned*8	1

Tabela 29 – Simulate - Estrutura Discreta DS-51.

ESTRUTURA DE GRUPO - DS-67

Esta estrutura de dados contém uma estrutura do parâmetro Batch.

E	Nome do elemento na estrutura	Tipo de Dado	Tamanho
1	*BATCH_ID*	*Unsigned*32	4
2	*RUP*	*Unsigned*16	2
3	*OPERATION*	*Unsigned*16	2
4	*PHASE*	*Unsigned*16	2

Tabela 30 - Estrutura de Grupo DS-67.

FORMATO IEEE-754
O formato *float* usado no Profibus está definido de acordo com o padrão IEEE 754:

- byte_MSB (byte 1) byte 2 byte 3 byte_LSB (byte 4)
- byte_MSB (byte 1) = exp
- byte 2 = mantissa
- byte 3 = mantissa
- byte_LSB (byte 4) = mantissa

Exemplo: C2 ED 40 00 00

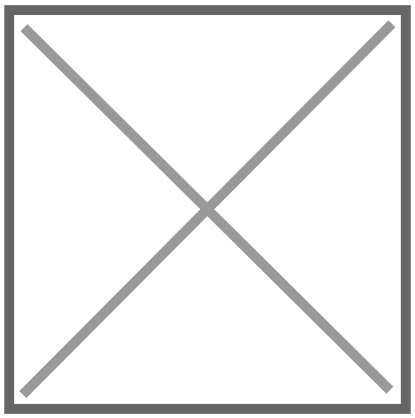
11000010 00000100 00000000 00000000

float= (-1) ^ bitsignal * [2 ^ (exp - 127) * (1 + mantissa)]

Onde: bitsignal é o bit mais significativo do byte_MSB (byte 1). Se este bit é 0, o número é positivo. Se for 1, é negativo. No exemplo anterior o número é negativo.

O valor de "exp" é a soma dos outros bits do byte_MSB (byte 1) e do bit mais significativo do byte 2.

Por exemplo:



- Bit 1 no sinal indica número negativo.
- Expoente 10000101 = 133_{10}
- Portanto, o valor antes da polarização era:

- $x + (2^{8-1}) -1 = 133$
- $x + 127 = 133$
- $x = 6$
- Com o bit escondido, temos a mantissa:
- 1.110110101×2^6 => deslocando o ponto em 6 casas: 1110110.101
  - 1110110= 118_{10}
  - 101 = 1*(2^-1) + 0*(2^-2) + 1*(2^-3) = 0.625
  - 118+0.625 = 118.625
- Como o bit de sinal representa um número negativo, temos: **-118.625**.

CONFIGURAÇÃO CÍCLICA

O protocolo Profibus-PA possui mecanismos contra falhas e erros de comunicação entre o equipamento da rede e o mestre. Por exemplo, durante a inicialização do equipamento estes mecanismos são utilizados para verificar os possíveis erros.

Após a energização (*power-up*) do equipamento de campo (escravo), pode-se trocar dados ciclicamente com o mestre Profibus classe 1 se a parametrização do escravo estiver correta. Estas informações são obtidas através dos arquivos GSD (arquivos fornecidos pelos fabricantes dos equipamentos que contém suas descrições).

Através dos comandos abaixo, o mestre executa todo o processo de inicialização com os equipamentos Profibus-PA:

- **Get_Cfg:** carrega a configuração dos escravos no mestre e verifica a configuração da rede;
- **Set_Prm:** escreve nos parâmetros dos escravos e executa os serviços de parametrização da rede;
- **Set_Cfg:** configura os escravos de acordo com as entradas e saídas;
- **Get_Cfg:** um outro comando, onde o mestre verifica a configuração dos escravos.

Todos estes serviços são baseados nas informações obtidas dos arquivos GSD dos equipamentos. De acordo com a tabela baixo, pode-se obter dos blocos funcionais quais os parâmetros que participam da troca de dados cíclicos. Um máximo de 244 bytes de entrada e 244 bytes de saída podem ser trocados entre mestre e escravo Profibus.

Em equipamentos que possuem mais de um bloco do mesmo tipo (por exemplo, 3 Blocos AIs e 3 Blocos TOTs), a ordem na configuração cíclica deverá ser AI_1, AI_2, AI_3, TOT_1,TOT_2 e TOT_3. Nos casos onde não se configura nenum bloco ciclicamente, é necessário preencher a configuração com o módulo vazio (Empty_Module = 0x00). Por exemplo, no caso onde existem 3 Blocos AIs e 3 Blocos TOTs, mas o usuário só irá utilizar AI_1, AI_3, TOT_1 e TOT_3, a ordem na configuração cíclica deverá ser AI_1, Empty_Module, AI_3, TOT_1, Empty_Module e TOT_3.

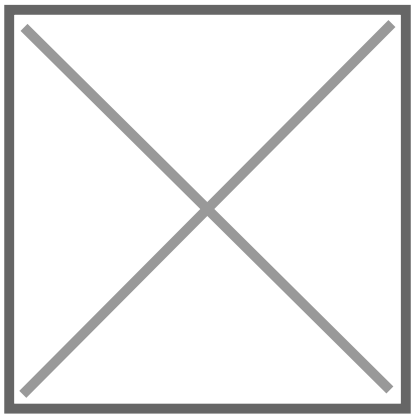


Tabela 31 - Blocos Funcionais e comunicação cíclica.

CONCLUSÃO

Com esta variedade de funcionalidades e aplicações, o protocolo PROFIBUS oferece toda a flexibilidade e autonomia que o usuário necessita para automatizar e otimizar processos, com facilidade de instalação, operação, manutenção e diagnósticos de variáveis que sejam importantes para o correto funcionamento da planta.

Este artigo não substitui os padrões IEC 61158 e IEC 61784 e nem os perfis e guias técnicos do PROFIBUS. Em caso de discrepância ou dúvida, os padrões IEC 61158 e IEC 61784, perfis, guias técnicos e manuais de fabricantes prevalecem.

O conteúdo deste artigo foi elaborado cuidadosamente. Entretanto, erros não podem ser excluídos e assim nenhuma responsabilidade poderá ser atribuída ao autor. Sugestões de melhorias podem ser enviadas ao email cesar.cassiolato@vivaceinstruments.com.br.

Sobre o autor

César Cassiolato é *Presidente e Diretor de Qualidade da Vivace Process Instruments. É também Conselheiro Administrativo da Associação PROFIBUS Brasil América Latina desde 2011, onde foi Presidente de 2006 a 2010, Diretor Técnico do Centro de Competência e Treinamento em PROFIBUS, Diretor do FDT Group no Brasil e Engenheiro Certificado na Tecnologia PROFIBUS e Instalações PROFIBUS pela Universidade de Manchester*

Referências

- *Manuais Vivace ProcessInstruments*
- *Artigos Técnicos César Cassiolato*
- *https://www.vivaceinstruments.com.br/*
- *Material de treinamento e artigos técnicos PROFIBUS - César Cassiolato*
- *Especificações técnicas PROFIBUS*
- *www.profibus.com*