PROFIBUS-PA: IDENTIFIER FORMAT E CONFIGURAÇÃO CÍCLICA

INTRODUÇÃO

O Profibus é um protocolo digital utilizado em sistemas de controle, que permite a conexão com interoperabilidade de diversos equipamentos e fabricantes. Possui uma série de vantagens em relação à tecnologia 4-20 mA, onde resumidamente 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.

Neste artigo veremos detalhes sobre os Identifier Formats e a Configuração Cíclica.

ARQUIVO GSD

O arquivo GSD é como se fosse um datasheet eletrônico do equipamento Profibus e que traz detalhes de revisão de hardware e software, bus timing do equipamento e informações sobre a troca de dados cíclicos.

As informações de troca de dados cíclicos para cada módulo permitido do equipamento são demarcadas pelas palavras-chaves “Module” e “EndModule” e permite que mestre e escravos entendam a troca de dados cíclicos, as informações requisitadas ou enviadas pelo mestre classe 1(PLC, por exemplo), de alta prioridade, e que é parte fundamental no controle e tomada de decisão.

Cada módulo possui um conjunto de Identifier Bytes ou Identifier Formats. Este artigo nos mostrará como cada identificador é interpretado pelo mestre classe 1.

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 GSDs (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 1, 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 nenhum 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 1 - Blocos Funcionais e comunicação cíclica.

ional	Parâmetro	Identificador	Identificador Especial ou E
(AI)	**OUT**	**0x94**	0x42, 0x84, 0x08, 0x05
t (AO)	**SP**	**0xA4**	0x82, 0x84, 0x08, 0x05
	SP READBACK POS_D	0x96, 0xA4	0xC6, 0x84, 0x86, 0x08, 0x0 0x05, 0x05, 0x05
	SP CHECK_BACK	0x92, 0xA4	0xC3, 0x84, 0x82, 0x08, 0x0
	SP READBACK POS_D CHECK_BACK	0x99, 0xA4	0xC7, 0x84, 0x89, 0x08, 0x0 0x05, 0x05, 0x05, 0x0A
	RCAS_IN RCAS_OUT	0xB4	0xC4, 0x84, 0x84, 0x08, 0x0 0x05
	RCAS_IN RCAS_OUT CHECK_BACK	0x97, 0xA4	0xC5, 0x84, 0x87, 0x08, 0x0 0x05, 0x0A
	SP READBACK RCAS_IN RCAS_OUT POS_D CHECK_BACK	0x9E, 0xA9	0xCB, 0x89, 0x8E, 0x08, 0x 0x05, 0x08, 0x05, 0x08, 0x0 0x05, 0x0A
(DI)	**OUT_D**	**0x91**	
ut (DO)	**SP_D**	**0xA1**	
	SP_D READBACK_D		0xC1, 0x81, 0x81, 0x83
	SP_D CHECK_BACK		0xC1, 0x81, 0x82, 0x92
	SP_D READBACK_D CHECK_BACK		0xC1, 0x81, 0x84, 0x93
	RCAS_IN_D RCAS_OUT_D		0xC1, 0x81, 0x81, 0x8C
	RCAS_IN_D RCAS_OUT_D CHECK_BACK		0xC1, 0x81, 0x84, 0x9C
	SP_D READBACK_D RCAS_IN_D RCAS_OUT_D CHECK_BACK		0xC1, 0x83, 0x86, 0x9F
	TOTAL	**-**	**0x41, 0x84, 0x85**
	TOTAL SET_TOT	-	0xC1, 0x80, 0x84, 0x85
	TOTAL	-	0xC1, 0x81, 0x84, 0x85

ENTENDENDO OS IDENTIFIER FORMATS

Os Identifier Formats são usados na configuração entre o mestre classe 1 e seus escravos. Após a energização (conhecida como power up) os equipamentos escravos estão prontos para a troca de dados cíclicos com o mestre classe 1, mas para isto, a parametrização no mestre para aquele escravo deve estar correta. Estas informações são obtidas através dos arquivos GSDs, que deve ser um para cada equipamento.

Todos estes serviços são baseados nas informações obtidas dos arquivos GSDs.
Existem 3 tipos de Identifier Formats, onde a principal diferença entre eles é a quantidade de bits e bytes que eles podem representar:

- Simples: podem representar 8 e 16 bits de dados
- Especial: podem representar 8 e 16 bits de dados e ainda tipos em formatos especiais
- Especial para PROFIBUS DP-V1: podem representar 8 e 16 bits de dados e tipos padrões definidos de acordo com o DP-V1.

Exemplo de Identifier Formats:

Modules for Analog Input
Module = "Analog Input (short) " 0x94
EndModule
Module = "Analog Input (long) " 0x42, 0x84, 0x08, 0x05
EndModule

Identifier Format simples

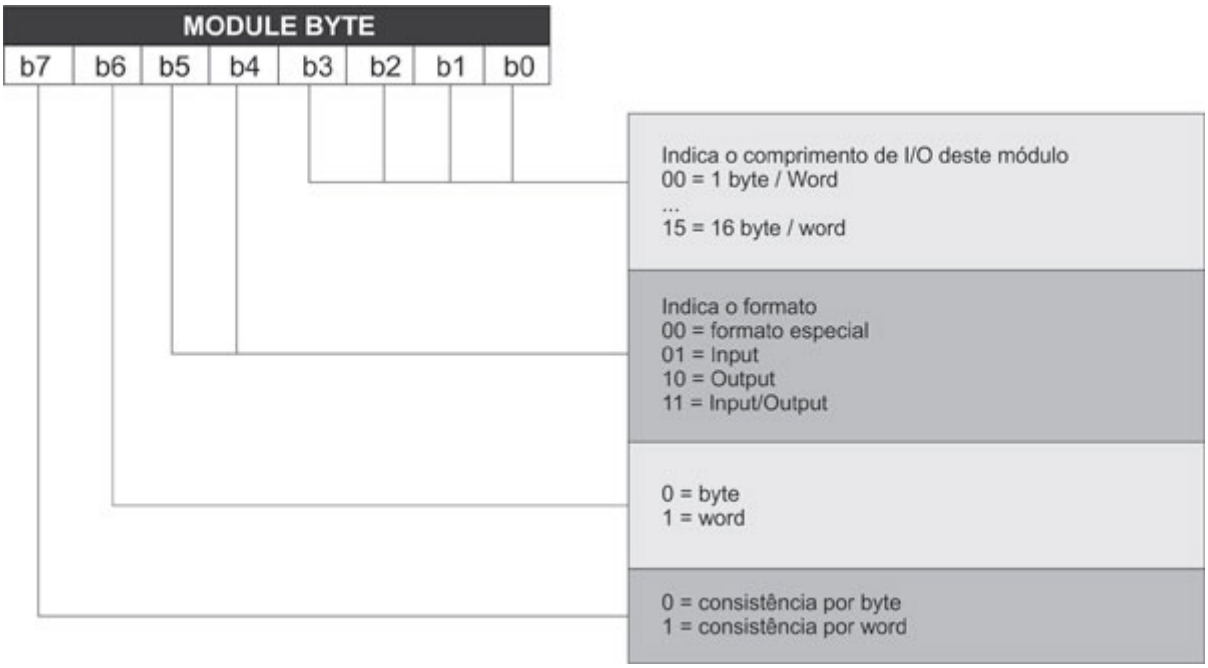


Figura 1 - Identifier Format simples

Quando o formato for especial, veja Identifier Formats especiais.
Exemplo:

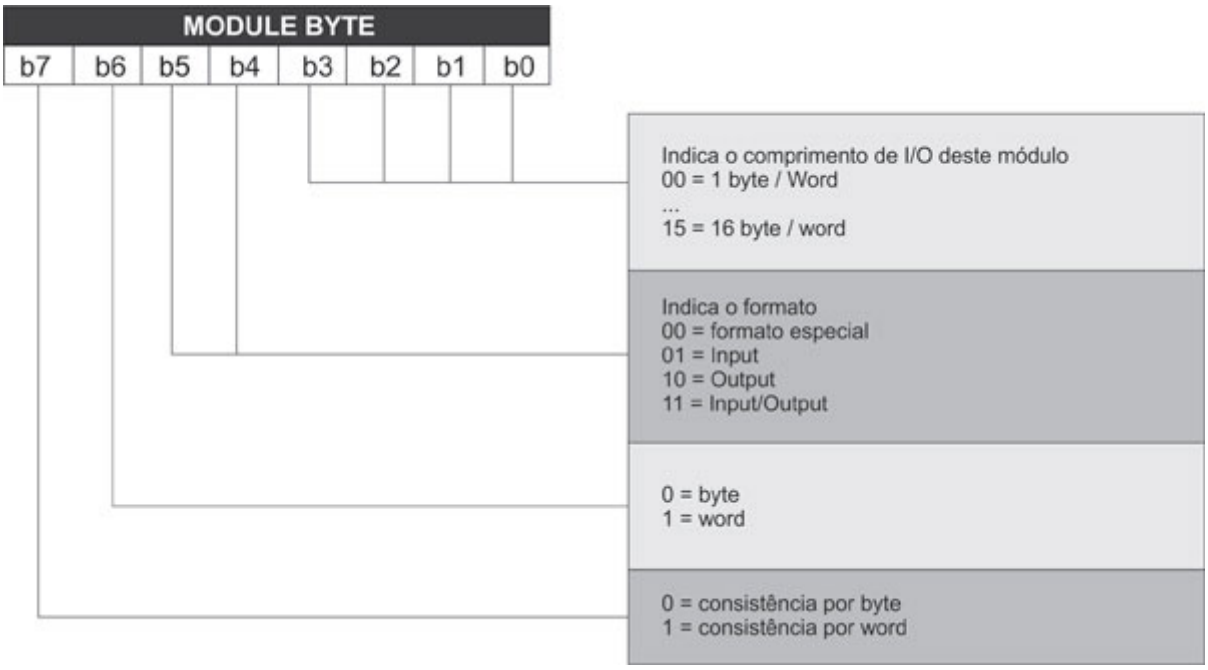


Figura 1a – Exemplo de Identifier Format simples

Identifier Format especial

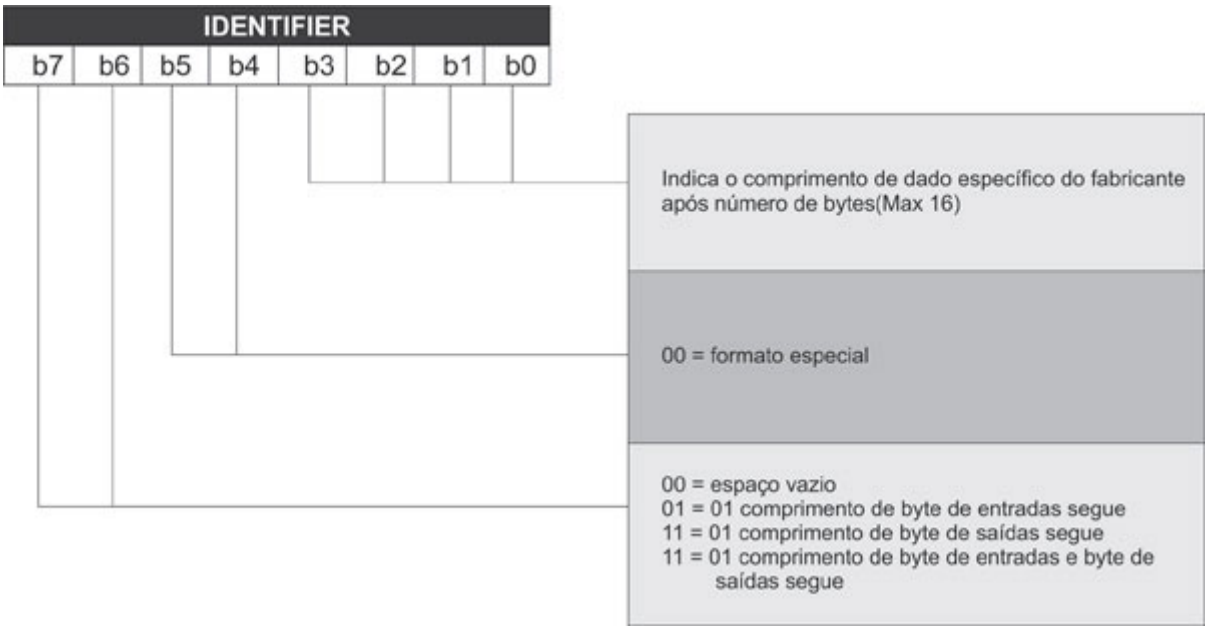


Figura 2 - Identifier Format especial

Comprimento de bytes para Identifier Format especial

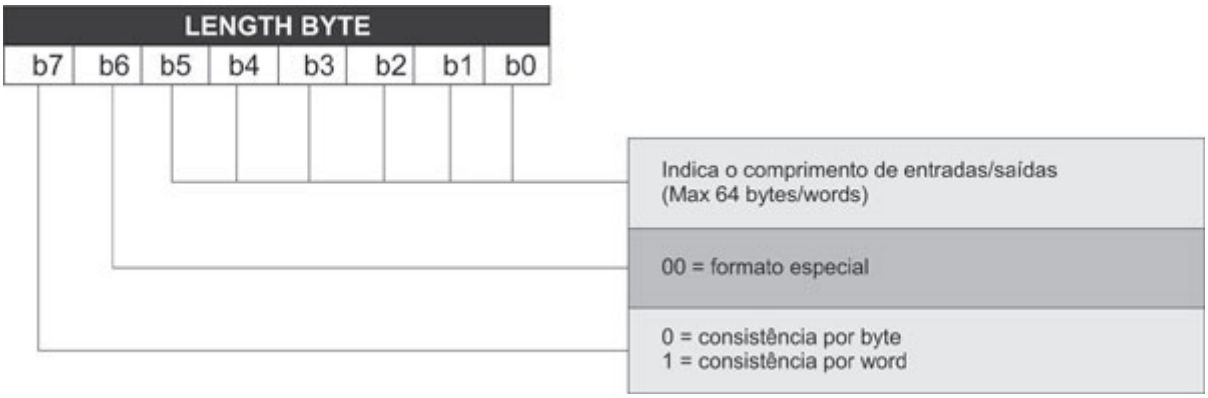


Figura 3 - Comprimento de bytes para Identifier Format especial

Note que através do formato especial pode-se descrever dados em 8 e 16 bits e que permitem módulos de até 64bytes/words.Vejamos o exemplo abaixo, onde para o campo de identificador temos 1 byte de comprimento de saída, 1 byte de comprimento de entrada em formato especial e um byte de dados de usuário.De acordo com o Length Byte 1, temos 64 words de saída com consistência em Word e de acordo com o Length Byte 2, temos 64 words de entrada com consistência em Word. Depois no Length Byte 2, temos um byte de dados de usuário que é específico e não pode ser descrito como padrão:

IDENTIFIER							
1	1	0	0	0	0	0	1
LENGTH BYTE 1							
0	1	1	1	1	1	1	1
LENGTH BYTE 2							
0	1	1	1	1	1	1	1
LENGTH BYTE 3							
X	X	X	X	X	X	X	X

Figura 3a – Exemplo de Identifier Format especial

Identifier Formats de acordo com PROFIBUS DP-V1.
Este formato pode representar 8 e 16 bits de dados e tipos padrões definidos de acordo com o DP-V1.A tabela a seguir lista alguns destes tipos.

Tipo de dados – Profibus DP-V1	Número de Bytes	Código
Boolean	1	1
Integer 8	1	2
Integer 16	2	3
Integer 32	3	4
Unsigned 8	1	5
Unsigned 16	2	6
Unsigned 32	4	7
Float Point	4	8
Visible String	1,2,3...	9
Octet String	1,2,3...	10

Tabela 2 – Alguns tipos de dados manuseados de acordo com o PROFIBUS DP-V1

Por dentro do Identifier Format de acordo com PROFIBUS DP-V1

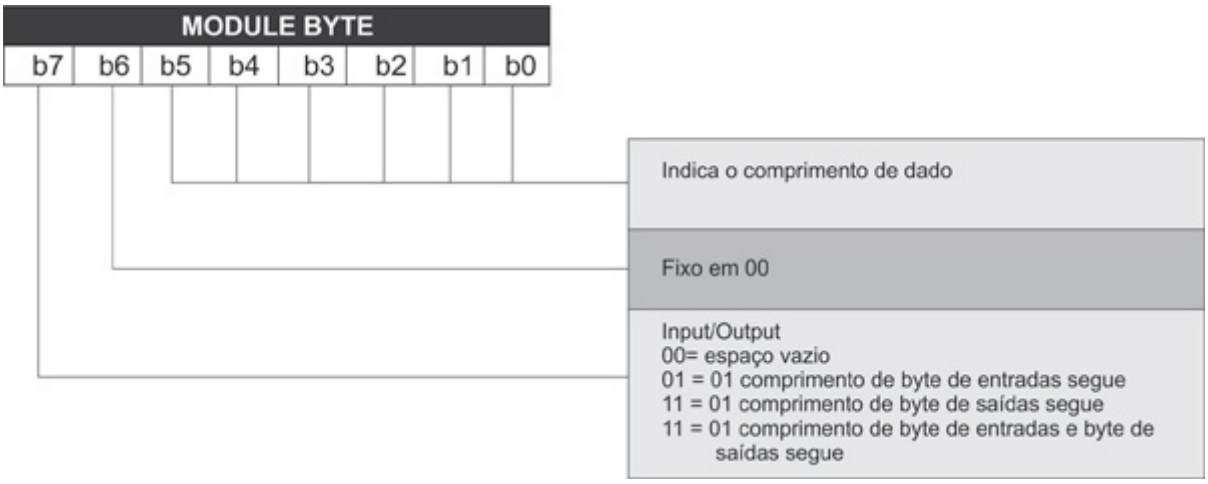


Figura 4 - Identifier Formats de acordo com PROFIBUS DP-V1

Comprimento de bytes para Identifier Format de acordo com PROFIBUS DP-V1

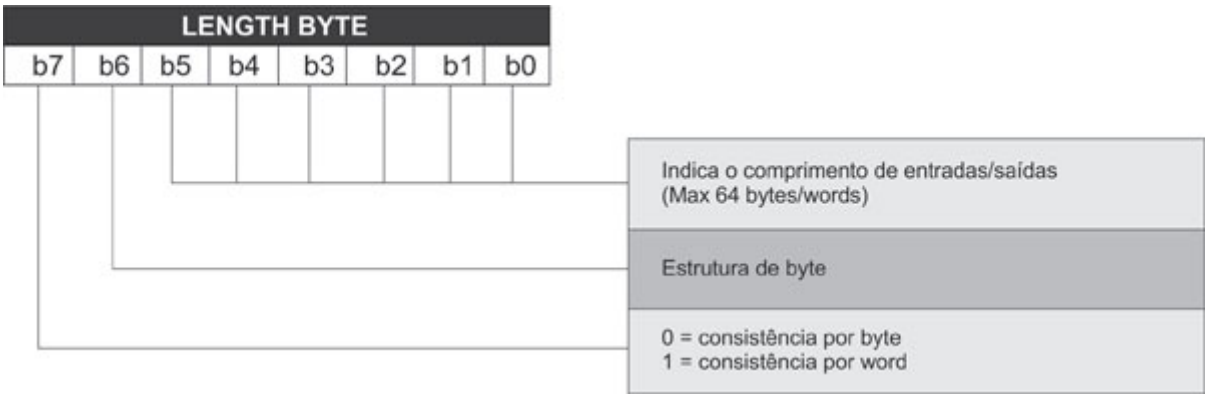


Figura 5 - Comprimento de bytes para Identifier Format de acordo com Profibus DP-V1

No início do artigo, foi dado um exemplo para o Bloco Analog Input (AI). Note que podemos defini-lo em dois formatos: short e long. Vejamos o exemplo abaixo, considerando o formato long:





Figura 6 – Exemplo de Identifier Format de acordo com o Bloco Analog Input (AI) do PROFIBUS DP-V1

Consideremos agora, um exemplo para o Bloco Analog Output (AO), onde temos a seguinte configuração possível entre várias:
Module = "eRCAS_IN + RCAS_OUT "
0xC4, 0x84, 0x84, 0x08, 0x05, 0x08, 0x05
EndModule



Figura 7 – Exemplo de Identifier Format de acordo com o Bloco Analog Output(AO) do PROFIBUS DP-V1

EXEMPLO: TROCA DE DADOS CÍCLICOS COM EQUIPAMENTOS PROFIBUS-PA DE SÁIDA

Equipamentos de Saída, por exemplo, posicionadores de válvulas, podem trocar dados cíclicos com o mestre Profibus DP Classe 1 de 7 maneiras diferentes, de acordo com os módulos descritos no arquivo GSD.

1. SP

(short) 0xA4
(extended format) 0x82,0x84,0x08,0x05

Com a esta configuração, o equipamento de saída recebe um setpoint (SP) do mestre Profibus como a posição desejada e não retorna nada ao mestre. Nesta condição, o bloco AO deve estar em automático (AUTO) e o status do SP deve ser pelo menos Good. Neste modo, além do mestre Profibus, o usuário pode atuar no SP, via comunicação acíclica.

Este setpoint é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
SP (Valor, float IEEE)				Status SP

2. RCAS_IN+RCAS_OUT

(short) 0xB4
(extended format) 0xC4,0x84,0x84,0x08,0x05,0x08,0x05

Com a esta configuração, o equipamento de saída recebe um setpoint (RCAS_IN) do mestre Profibus como posição desejada e retorna ao mestre o valor recebido, através do parâmetro RCAS_OUT. Nesta condição, o status deve ser igual a IA-Initialization Acknowledge (0xC4). Neste modo, somente o mestre Profibus pode atuar via comunicação cíclica.

Este RCAS_IN é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
RCAS_IN (Valor, float IEEE)				Status RCAS_IN

O retorno ao mestre, RCAS_OUT, também é em formato float IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
RCAS_OUT (Valor, float IEEE)				Status RCAS_OUT

3. SP+READBACK+POS_D

(short) 0x96,0xA4
(extended format) 0xC6,0x84,0x86,0x08,0x05.0x08,0x05,0x05,0x05

Com a esta configuração, o equipamento de saída recebe um setpoint (SP) do mestre Profibus como posição desejada e retorna à posição real (READBACK) e posição discreta (POS_D) ao mestre. Nesta condição, o bloco AO deve estar em automático (AUTO) e o status do SP deve ser pelo menos Good. Neste modo, além do mestre Profibus, o usuário pode atuar no SP, via comunicação acíclica.

Este setpoint é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
SP (Valor, float IEEE)				Status SP

O retorno ao mestre Profibus segue o formato abaixo. O READBACK é o retorno analógico do transdutor, a posição real da válvula. O POS_D é um status discreto: aberto, fechado ou posição intermediária.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
READBACK (Valor, float IEEE)				Status READ-BACK	POS_D (Valor)	Status POS_D

4. SP+CHECKBACK
(short) 0x92,0xA4
(extended format) 0xC3,0x84,0x82,0x08,0x05,0x0A

Com a esta configuração, o equipamento de saída recebe um setpoint (SP) do mestre Profibus como posição desejada e retorna à condição de diagnóstico (CHECKBACK) ao mestre. Nesta condição, o bloco AO deve estar em automático (AUTO) e o status do SP deve ser pelo menos Good. Neste modo, além do mestre Profibus, o usuário pode atuar no SP, via comunicação acíclica.

Este setpoint é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
SP (Valor, float IEEE)				Status SP

O retorno ao mestre Profibus DP, CHECKBACK, segue o formato:

Byte 1	Byte 2	Byte 3
CHECK BACK [0]	CHECK BACK [1]	CHECK BACK [2]

5. SP+READBACK+POS_D+CHECKBACK
(short) 0x99,0xA4
(extended format) 0xC7,0x84,0x89,0x08,0x05.0x08,0x05,0x05,0x05,0x0A

Com a esta configuração, o equipamento de saída recebe um setpoint (SP) do mestre Profibus como posição desejada e retorna à condição de diagnóstico (CHECKBACK) e a posição discreta ao mestre. Nesta condição, o bloco AO deve estar em automático (AUTO) e o status do SP deve ser pelo menos Good. Neste modo, além do mestre Profibus, o usuário pode atuar no SP, via comunicação acíclica.

Este setpoint é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
SP (Valor, float IEEE)				Status SP

O retorno ao mestre Profibus DP, READBACK e CHECKBACK, segue o formato:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
READBACK (Valor, float IEEE)				Status READ-BACK	POS_D (Valor)	Status POS_D	CHECK BACK [0]	CHECK BACK [1]	CHECK BACK [2]

6. RCAS_IN+RCAS_OUT+CHECKBACK
(short) 0x97,0xA4
(extended format) 0xC5,0x84,0x87,0x08,0x05,0x08,0x05,0x0A

Com a esta configuração, o equipamento de saída recebe um setpoint (RCAS_IN) do mestre Profibus como posição desejada e retorna ao mestre o valor recebido, através do parâmetro RCAS_OUT e as condições de diagnóstico no parâmetro CHECKBACK. Nesta condição, o status deve ser igual a IA-Initialization Acknowledge (0xC4). Neste modo, somente o mestre Profibus pode atuar via comunicação cíclica.

Este RCAS_IN é um float em formato IEEE-754:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
RCAS_IN (Valor, float IEEE)				Status RCAS_IN

O retorno ao mestre, RCAS_OUT e CHECKBACK, segue o formato:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
RCAS_OUT (Valor, float IEEE)				Status RCAS_ OUT	CHECK BACK [0]	CHECK BACK [1]	CHECK BACK [2]

7. SP+RB+RIN+ROUT+POS_D+CB
(Short) 0x9E,0xA9
(Extended format) 0xCB,0x89,0x8E,0x08,0x05.0x08,0x05,0x08,0x05,0x08,0x05,0x05,0x05,0x0A

Esta configuração cíclica é uma combinação completa das anteriores. Os valores de entrada para o equipamento são:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
SP (Valor, float IEEE)				Status SP	RCAS_IN (Valor, float IEEE)				Status RCAS_IN

Os valores de retorno ao mestre são:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
READBACK (Valor, float IEEE)				Status READ- BACK	RCAS_OUT (Valor, float IEEE)				Status RCAS_OUT

Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
POS_D (Valor)	Status POS_D	CHECK BACK [0]	CHECK BACK [1]	CHECK BACK [2]

CONCLUSÃO

Vimos através deste artigo a importância da interpretação dos Identifier Formats na tecnologia PROFIBUS e suas particularidades com a troca de dados cíclicos.

Com esta variedade de funcionalidades e aplicações, o protocolo PROFIBUS oferece toda a flexibilidade 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. Sempre que possível, consulte as regulamentações físicas, assim como as práticas de segurança de cada área.

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 e-mai 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 Process Instruments
- Artigos Técnicos César Cassiolato
- https://www.vivaceinstruments.com.br/
- Material de treinamento e artigos técnicos PROFIBUS - César Cassiolato
- Especificações técnicas PROFIBUS
- http://www.profibus.org.br/