

# Inversor de Frequência CFW-08 Manual da Comunicação DeviceNet Slave

03/2006

---

**Série:** CFW-08  
**Software:** versão 4.6X  
0899.5221 P/1

---

## Importante

Este manual adicional descreve as funções do protocolo DeviceNet no inversor de frequência CFW-08. Os inversores com esse protocolo devem ter no código inteligente a versão "A4" do cartão de controle conforme descrição exemplo:

MOD.: CFW080040B2024P0**A4Z**

As informações contidas no manual do usuário do CFW-08 para o cartão de controle "A1" (CFW-08 Plus) podem ser usadas para essa versão do cartão de controle (A4), porém com as seguintes modificações:

- Incluído o protocolo de comunicação DeviceNet.
- Excluídos os protocolos de comunicação serial (protocolos WEG, Modbus-RTU e IHM remota serial).

A descrição detalhada destas alterações é feita nos itens seguintes.

# Sumário

<b>Lista de Tabelas</b>	<b>5</b>
<b>Lista de Figuras</b>	<b>5</b>
<b>Sobre o manual</b>	<b>6</b>
Abreviações e Definições . . . . .	6
Documentos . . . . .	6
<b>1 Instalação</b>	<b>7</b>
1.1 Kit de comunicação . . . . .	7
1.2 Conexão com a rede . . . . .	7
1.3 Fonte de alimentação . . . . .	8
1.4 Cabos e terminações . . . . .	9
1.5 Arquivo de configuração . . . . .	11
<b>2 Descrição das principais características do CFW-08</b>	<b>12</b>
2.1 Tipos de mensagens suportadas . . . . .	12
2.2 Indicação de estados/erros . . . . .	12
<b>3 Formato dos telegramas de I/O</b>	<b>15</b>
3.1 Comando e Monitoramento . . . . .	15
3.1.1 Opções de Monitoramento . . . . .	15
3.1.2 Opções de Controle . . . . .	16
3.2 Instâncias 20/70 ( <i>Polled, Change of State e Cyclic</i> ) . . . . .	16
3.3 Instâncias 21/71 ( <i>Polled, Change of State e Cyclic</i> ) . . . . .	17
3.4 Instâncias 100/150 ( <i>Polled, Change of State e Cyclic</i> ) . . . . .	18
<b>4 Parâmetros relacionados ao DeviceNet</b>	<b>20</b>
4.1 Novos parâmetros . . . . .	20
4.1.1 P070 - Estado do controlador CAN . . . . .	20
4.1.2 P081 - Estado do mestre da rede . . . . .	20
4.1.3 P313 - Ação para erro de comunicação . . . . .	21
4.1.4 P700 - Protocolo CAN . . . . .	21
4.1.5 P701 - Endereço na rede CAN . . . . .	21
4.1.6 P702 - Taxa de comunicação . . . . .	22
4.1.7 P703 - Reset de bus-off . . . . .	22
4.1.8 P710 - Instâncias de I/O . . . . .	22
4.2 Parâmetros alterados . . . . .	23
4.2.1 P220 - Seleção da fonte local/remoto . . . . .	23
4.2.2 P221 - Seleção da referência de velocidade - situação local . . . . .	23
4.2.3 P222 - Seleção da referência de velocidade - situação remoto . . . . .	24
4.2.4 P229 - Seleção de comandos - situação local . . . . .	24
4.2.5 P230 - Seleção de comandos - situação remoto . . . . .	24
4.2.6 P231 - Seleção do sentido de giro - situação local e remoto . . . . .	24
<b>5 Classes de objetos suportadas</b>	<b>25</b>
5.1 Classe <i>Identity</i> (01h) . . . . .	25
5.2 Classe <i>Message Router</i> (02h) . . . . .	25
5.3 Classe <i>DeviceNet</i> (03h) . . . . .	25
5.4 Classe <i>Assembly</i> (04h) . . . . .	26

---

5.5	Classe <i>Connection</i> (05h)	27
5.5.1	Instância 1: <i>Explicit Message</i>	27
5.5.2	Instância 2: <i>Polled</i>	28
5.5.3	Instância 4: <i>Change of State/Cyclic</i>	29
5.6	Classe <i>Motor Data</i> (28h)	30
5.7	Classe <i>Control Supervisor</i> (29h)	31
5.8	Classe <i>AC/DC Drive</i> (2Ah)	32
5.9	Classe <i>Acknowledge Handler</i> (2Bh)	32
5.10	Classes WEG	33
<b>6</b>	<b>Outras funções</b>	<b>35</b>
6.1	Função <i>Automatic Device Replacement</i>	35
	<b>Glossário</b>	<b>36</b>

## Lista de Tabelas

1	Documentação técnica sobre CAN e DeviceNet . . . . .	6
2	Pinos do conector XC14 . . . . .	8
3	Dados para alimentação via rede . . . . .	9
4	Tamanho da rede x taxa de comunicação . . . . .	10
5	Instâncias 20/70 definidas pela ODVA . . . . .	16
6	Instâncias 21/71 definidas pela ODVA . . . . .	17
7	Instâncias 100/150 definidas pela WEG . . . . .	19
8	Classe <i>Identity</i> . . . . .	25
9	Classe <i>DeviceNet</i> . . . . .	25
10	Atributos das instâncias da classe <i>Assembly</i> . . . . .	26
11	Instâncias da classe <i>Assembly</i> . . . . .	26
12	Classe <i>Connection</i> - Instância 1: <i>Explicit Message</i> . . . . .	27
13	Classe <i>Connection</i> - Instância 2: <i>Polled</i> . . . . .	28
14	Classe <i>Connection</i> - Instância 4: <i>Change of State/Cyclic</i> . . . . .	29
15	Classe <i>Motor Data</i> . . . . .	30
16	Classe <i>Control Supervisor</i> . . . . .	31
17	Classe <i>AC/DC Drive</i> . . . . .	32
18	Classe <i>Acknowledge Handler</i> . . . . .	32
19	Classes WEG . . . . .	33
20	Parâmetros das classes WEG . . . . .	33

## Lista de Figuras

1	Kit KFB-DN-CFW-08 instalado no inversor. . . . .	7
2	Módulo de comunicação DeviceNet. . . . .	8
3	Conexão de um típico cabo DeviceNet . . . . .	9
4	Rede DeviceNet . . . . .	10
5	Detalhes do módulo de comunicação. . . . .	13

## Sobre o manual

Este documento descreve o funcionamento do protocolo DeviceNet para o inversor de frequência CFW-08. Ele deve ser utilizado em conjunto com o manual do usuário do produto.

## Abreviações e Definições

<b>ADR</b>	Automatic Device Replacement
<b>ASCII</b>	American Standard Code for Information Interchange
<b>CAN</b>	Controller Area Network
<b>CLP</b>	Controlador Lógico Programável
<b>IHM</b>	Interface Homem-Máquina
<b>MS</b>	Module Status
<b>NS</b>	Network Status
<b>ODVA</b>	Open DeviceNet Vendor Association



### Representação numérica

Números decimais são representados através de dígitos sem sufixo. Números hexadecimais são representados com a letra 'h' depois do número.

## Documentos

O protocolo DeviceNet para o CFW-08 foi desenvolvido tendo como base as seguintes especificações e documentos:

<i>Documento</i>	<i>Versão</i>	<i>Fonte</i>
DeviceNet Volume I DeviceNet Communication Model and Protocol	2.0	ODVA
DeviceNet Volume II DeviceNet Device Profiles and Object Library	2.0	ODVA

Tabela 1: Documentação técnica sobre CAN e DeviceNet

Para obter esta documentação, consulte a ODVA. Esta organização é responsável por manter, divulgar e atualizar informações relativas ao protocolo DeviceNet.

# 1 Instalação

A rede DeviceNet, assim como várias outras redes de comunicação industriais, exige certos cuidados para garantir uma baixa taxa de erros durante a sua operação. Isto deve-se ao fato do ambiente industrial ser em geral agressivo devido à interferência eletromagnética. A instalação deve seguir recomendações comuns a qualquer rede de comunicação de dados, tais como proximidade dos cabos de sinal/potência, aterramento, qualidade das emendas e conectores. Nas seções a seguir são apresentadas recomendações básicas para a instalação do inversor de frequência CFW-08.

## 1.1 Kit de comunicação

Para que o inversor de frequência CFW-08 possa participar de uma rede DeviceNet, é necessária a instalação do kit KFB-DN-CFW-08 (item WEG 417118222). Faz parte deste kit o módulo de comunicação DeviceNet, que contém além dos botões e do display encontrados na IHM padrão do produto, um conector de cinco vias e LEDs para sinalização de estados/erros (ver item 1.2). Também acompanha o kit uma bula com instruções detalhadas de instalação e remoção do módulo. Caso o inversor seja fornecido com o cartão de comunicação instalado, basta seguir o descrito neste manual para configuração e operação do equipamento em rede.

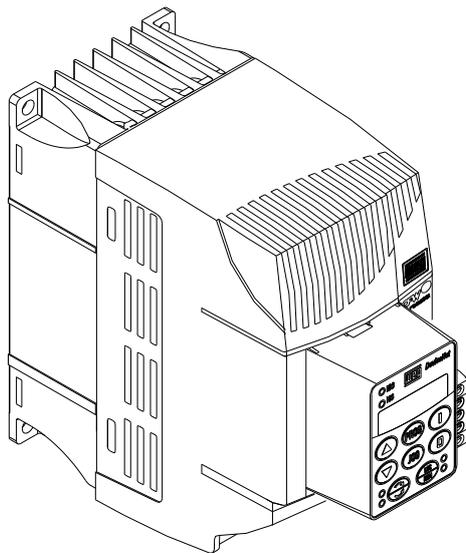


Figura 1: Kit KFB-DN-CFW-08 instalado no inversor.

## 1.2 Conexão com a rede

A interface para conexão com o barramento DeviceNet/CAN está disponível no conector XC14 do módulo de comunicação. A figura 2 ilustra a posição deste no módulo enquanto que a tabela 2 descreve a função de cada pino do conector XC14.

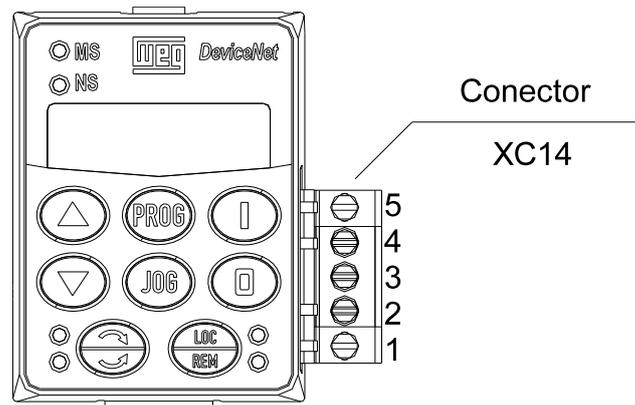


Figura 2: Módulo de comunicação DeviceNet.

<i>Conector XC14</i>			
5	V+	vermelho	<b>potência</b>
4	CAN_H	branco	<b>signal</b>
3	Blindagem ( <i>shield</i> )	-	
2	CAN_L	azul	<b>signal</b>
1	V-	preto	<b>potência</b>

Tabela 2: Pinos do conector XC14

Para efetuar a ligação, deve-se observar a correspondência dos sinais mostrados na tabela 2. Cada pino do conector dos diversos equipamentos ligados em rede deve apresentar os mesmos sinais (V- com V-, CAN\_L com CAN\_L, etc.). A blindagem do cabo (pino 3) não deve ser desprezada. Este pino está conectado internamente ao terra através de um circuito RC paralelo.



#### **NOTA!**

Deve-se também ligar o pino 5 (GND) do conector de sinais XC1 ao ponto de aterramento (PE) localizado no dissipador do produto.

Esta forma de ligação foi pensada tendo como principal objetivo a retirada de qualquer equipamento da rede sem prejuízos à mesma. A entrada e saída de nodos da rede deve ser totalmente transparente.

### **1.3 Fonte de alimentação**

Para alimentar o circuito responsável pela comunicação no CFW-08, é necessário fornecer uma tensão de alimentação entre os pinos 1 e 5 do conector da rede. Para evitar problemas de diferença de tensão entre os dispositivos, é recomendável que a rede seja alimentada em apenas um ponto e o sinal de alimentação seja levado a todos os dispositivos através do cabo, conforme figura 4. Também é recomendável fazer o cálculo da carga total na rede (o consumo de cada equipamento deve ser conhecido *a priori*) para se ter certeza de que a fonte é capaz de suprir a carga exigida. Se for necessária a instalação de mais de uma fonte, esta deverá estar referenciada ao mesmo ponto. A posição de instalação também é importante pois terá influência na distribuição de carga na rede, ou seja, deve-se fazer o balanceamento de corrente de saída de cada fonte de alimentação presente.

Não existe um tipo específico de fonte para ser utilizada na alimentação da linha. A rigor, qualquer fonte de 24V poderá ser usada, desde que seja dimensionada para suprir a corrente exigida pelos equipamentos e atenda às exigências dadas a seguir. Apesar disto, diversos fabricantes produzem fontes de alimentação certificadas pela ODVA. Para maiores informações, consulte o catálogo de produtos disponível no *site* da ODVA<sup>1</sup>.

Abaixo estão listadas as principais características destas fontes:

- capacidade de corrente compatível com os equipamentos instalados e também com o cabeamento utilizado.
- saída DC isolada da entrada AC.
- proteção contra sobrecorrente.
- boa regulação; 24V +/- 4%.

Além disso, recomenda-se também a instalação de fusíveis em cada um dos segmentos alimentados pela fonte.

Mais detalhes a respeito deste tópico podem ser obtidos no capítulo 10 do documento *DeviceNet Volume I*.

A tabela 3 apresenta os níveis de tensão e corrente exigidos pelo inversor CFW-08.

	<i>Mínimo</i>	<i>Máximo</i>	
<i>Tensão (Vcc)</i>	11	30	24 (recomendada)
<i>Corrente (mA)</i>	20	55	35 (média)

Tabela 3: Dados para alimentação via rede

## 1.4 Cabos e terminações

É recomendado a utilização de um cabo blindado com quatro fios - um par para os pinos 2 e 4 (CAN\_L e CAN\_H) e outro para os pinos 1 e 5 (V- e V+). A figura 3 ilustra esta conexão segundo a pinagem e código de cores disponível na tabela 2.

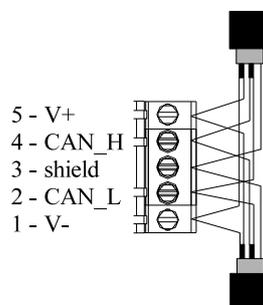


Figura 3: Conexão de um típico cabo DeviceNet

Para interligar os diversos nós da rede, recomenda-se a conexão do equipamento diretamente na linha principal. Apesar disto, derivações são permitidas. A tabela 4 mostra os limites

<sup>1</sup><http://www.odva.org>

destas derivações em função da taxa de comunicação utilizada. Durante a instalação dos cabos, deve-se evitar sua a passagem próxima a cabos de potência, pois devido a interferência eletromagnética, isto facilitará a ocorrência de erros durante a transmissão. Para evitar problemas de circulação de corrente por diferença de potencial entre diferentes aterramentos, é necessário que todos os dispositivos estejam conectados ao mesmo ponto de terra. Isto é garantido pelo circuito RC paralelo interno ao CFW-08 que interliga o pino 3 do conector DeviceNet ao terra local.

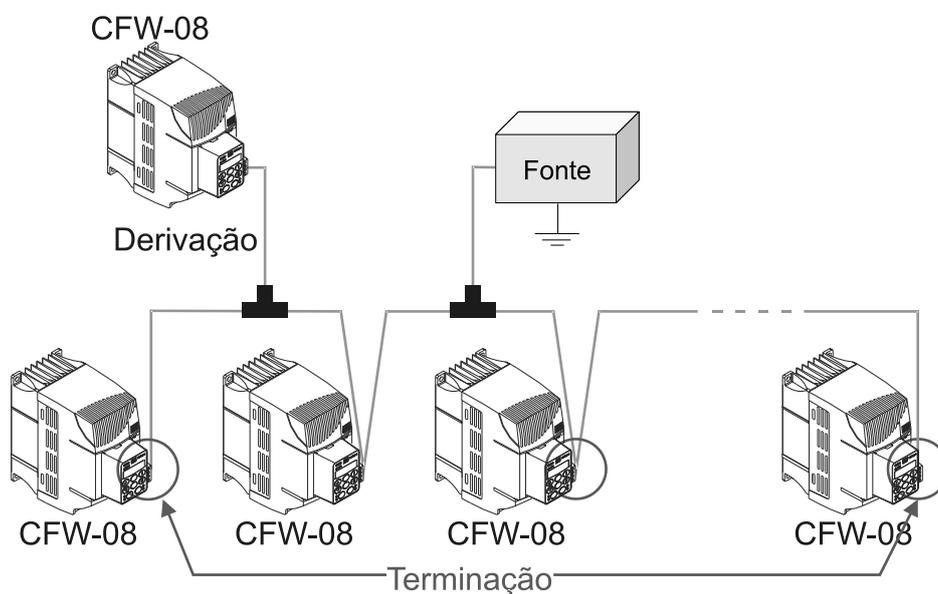


Figura 4: Rede DeviceNet

O cabo para a ligação dos sinais CAN\_L e CAN\_H deve ter impedância característica de aproximadamente  $120\Omega$ . O comprimento total é função da taxa de comunicação utilizada, como mostrado na tabela 4.

Taxa de comunicação	Comprimento da rede (máximo)	Derivação	
		Máximo	Cumulativo
125kbps	500m	6m	156m
250kbps	250m		78m
500kbps	100m		39m

Tabela 4: Tamanho da rede x taxa de comunicação

As extremidades da rede devem obrigatoriamente possuir um resistor de terminação a fim de evitar reflexões de sinal. A falta deste poderá provocar erros intermitentes na linha. Note que esta terminação deverá ser feita no cabeamento da rede entre os sinais CAN\_H e CAN\_L e não no equipamento em si. As características deste resistor são as seguintes:

- $121\Omega$
- 0,25W
- 1% de tolerância

## 1.5 Arquivo de configuração

Todo equipamento DeviceNet deve ser previamente registrado no *software* de configuração da rede para operar corretamente. Isto é feito com o auxílio de um arquivo **ASCII** com extensão **EDS** disponível no CD-ROM que acompanha o produto e também no *site* da WEG<sup>2</sup>. Neste arquivo estão codificadas instruções para que o mestre da rede possa comunicar-se com o escravo. A forma de registro é dependente do *software* de configuração e portanto uma consulta à documentação deste aplicativo talvez seja necessária.



### **NOTA!**

Este arquivo EDS não deve ser alterado pelo usuário.

---

<sup>2</sup><http://www.weg.com.br>

## 2 Descrição das principais características do CFW-08

O inversor de frequência CFW-08 atua como um escravo/servidor<sup>3</sup> numa rede DeviceNet, ou seja, ele recebe requisições (dados de saída) de um mestre<sup>4</sup>, processa os dados destas requisições e a seguir encaminha-os de volta ao mestre (dados de entrada). Este é um processo cíclico que ocorre enquanto a rede estiver ativa.

O CFW-08 também não é capaz de comunicar-se diretamente com outros escravos em um modo conhecido por *peer-to-peer*. É obrigatória a presença de um mestre na rede para gerenciar a comunicação. O CFW-08 funciona, portanto, como um dispositivo *Group 2 Only Server*<sup>5</sup>. Ou seja, ele não possui os serviços relacionados ao *Unconnected Message Manager (UCMM)*.

O inversor CFW-08 segue o perfil de dispositivo de um AC/DC Drive (*AC/DC Device Profile*). Este perfil define o formato dos dados trocados com o mestre através de mensagens do tipo *I/O* e representa a interface de operação com o *drive*.

### 2.1 Tipos de mensagens suportadas

O protocolo DeviceNet define dois tipos de mensagens para a operação e monitoramento do equipamento:

- I/O :** Tipo de telegrama síncrono dedicado à movimentação de dados prioritários entre um produtor e um ou mais consumidores. Podem ainda ser subdivididos em diferentes métodos de trocas de dados, como *Polled*, *Bit-Strobe*, *Change of State*, *Cyclic* e *Multicast*.
- Explicit :** Tipo de telegrama de uso geral e não prioritário utilizado principalmente em tarefas assíncronas como parametrização e configuração do equipamento.

O CFW-08 suporta ambos tipos de telegramas, *explicit* utilizado em geral para a configuração do *drive* via rede e *I/O* para troca de dados para operação do dispositivo. São suportados pelo CFW-08 os métodos *Polled*, *Change of State* e *Cyclic*. Estes podem ser utilizados separadamente ou então combinados. Cabe ao usuário avaliar e ponderar qual o mais eficiente para sua aplicação em particular.

### 2.2 Indicação de estados/erros

A indicação dos estados/erros do equipamento na rede DeviceNet é feita através de mensagens no display e de LEDs bicolores MS (*Module Status*) e NS (*Network Status*) localizados na IHM do produto, conforme figura 5.

O LED bicolor **MS** indica o estado do dispositivo em si:

<sup>3</sup>Também chamado de nodo.

<sup>4</sup>Este mestre é composto tipicamente por um módulo *scanner* instalado no CLP.

<sup>5</sup>Estes e outros termos estão explicados no glossário.

<b><i>Estado</i></b>	<b><i>Descrição</i></b>
Apagado	Sem alimentação.
Sólido verde	Operacional e em condições normais.
Intermitente vermelho/verde	Realizando auto-teste durante a inicialização.

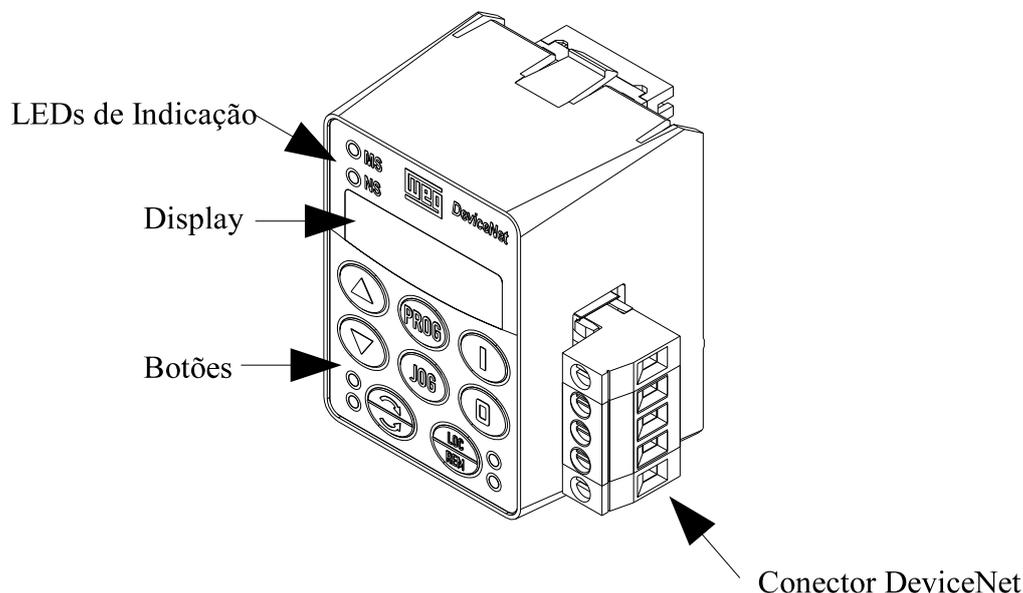


Figura 5: Detalhes do módulo de comunicação.

Já o LED bicolor **NS** fornece indicação do estado da rede DeviceNet:

<b><i>Estado</i></b>	<b><i>Descrição</i></b>
Apagado	Sem alimentação ou <i>not on-line</i> . Comunicação não pode ser estabelecida.
Intermitente verde	Dispositivo <i>on-line</i> mas não conectado. Escravo completou com sucesso o procedimento de verificação do Mac ID. Isto significa que a taxa de comunicação configurada está correta (ou foi detectada corretamente no caso da utilização do <i>auto-baud</i> ) e que não há outros nodos na rede com o mesmo endereço. Porém, neste estágio, ainda não há comunicação com o mestre.
Sólido verde	Dispositivo operacional e em condições normais. Mestre alocou um conjunto de conexões do tipo I/O com o escravo. Nesta etapa ocorre efetivamente a troca de dados através de conexões do tipo I/O.
Intermitente vermelho	Uma ou mais conexões do tipo I/O expiraram.
Sólido vermelho	Indica que o escravo não pode entrar na rede devido a problemas de endereçamento ou então devido a ocorrência de <i>bus-off</i> . Verifique se o endereço configurado já não está sendo utilizado por outro equipamento e se a taxa de comunicação escolhida está correta.
Intermitente vermelho/verde	Equipamento realizando auto-teste. Ocorre durante a inicialização.

Além dos LEDs **MS** e **NS**, mensagens mostradas na IHM fornecem uma outra fonte de informações a respeito do módulo e da rede. Estas mensagens são utilizadas para sinalizar erros do protocolo DeviceNet. Os quatro principais são:

- E33 :** Controlador CAN não alimentado. Verifique a fonte de alimentação da rede DeviceNet.
- E34 :** Sinaliza que o controlador CAN foi para o estado de bus-off. Certifique-se de que a taxa de comunicação ajustada no *drive* corresponde à do mestre.
- E36 :** Indica que o mestre da rede está no estado de *idle*.
- E37 :** Ocorre quando uma ou mais conexões do tipo I/O alocadas foram para o estado de *timeout*.

Por fim, existem ainda parâmetros do próprio equipamento para a indicação de erros e estados do *drive*. Maiores detalhes podem ser obtidos nas subseções [4.1.1](#) e [4.1.2](#).

## 3 Formato dos telegramas de I/O

### 3.1 Comando e Monitoramento

Cada uma das instâncias de I/O é composta por diferentes *bits* de controle e monitoramento possibilitando cobrir as operações mais comuns. São suportadas as instâncias 20/70 e 21/71 definidas pela ODVA. Além destas, suporta também as instâncias específicas WEG 100/150. A escolha de qual conjunto de instâncias utilizar é feita no parâmetro P710. Esta decisão, que cabe ao usuário, deverá levar em conta aspectos da aplicação em que o *drive* irá atuar.

#### 3.1.1 Opções de Monitoramento

As seguintes informações são disponibilizadas via rede<sup>6</sup> para o usuário:

- *Faulted* [*bit* 0, *byte* 0]: quando o inversor estiver em erro este *bit* será setado.
- *Warning* [*bit* 1, *byte* 0]: este *bit* não tem função no CFW-08 e portanto vale sempre zero.
- *Running1(fwd)* [*bit* 2, *byte* 0]: será setado quando o motor estiver girando no sentido horário.
- *Running2(rev)* [*bit* 3, *byte* 0]: quando o motor estiver girando no sentido anti-horário este *bit* será setado.
- *Ready* [*bit* 4, *byte* 0]: se o estado do conversor for *Ready*, *Enabled* ou *Stopping* este *bit* será setado.
- *Ctrl from Net* [*bit* 5, *byte* 0]: quando setado indica que o controle está sendo realizado via rede DeviceNet.
- *Ref from Net* [*bit* 6, *byte* 0]: indica que o envio da referência de velocidade está sendo feito via rede DeviceNet.
- *At Reference* [*bit* 7, *byte* 0]: indica que referência foi atingida, ou seja, quando houver uma diferença máxima de 1Hz entre a velocidade real e a referência enviada via rede.
- *Drive State* [*byte* 1]: indica o estado do conversor:
  - 0: Non-Existant
  - 1: Startup
  - 2: Not\_Ready
  - 3: Ready
  - 4: Enabled
  - 5: Stopping
  - 6: Fault\_Stop
  - 7: Faulted
- *Speed Actual (RPM)* [*bytes* 2 e 3]: palavra que retorna a velocidade real do motor em RPM.

---

<sup>6</sup>Válido para instâncias 20/70 e 21/71

### 3.1.2 Opções de Controle

Estão disponíveis as seguintes opções de controle do inversor via rede<sup>7</sup> DeviceNet:

- *Run Fwd* [bit 0, byte 0]: este *bit* faz com que o inversor gire no sentido horário.
- *Run Rev* [bit 1, byte 0]: este *bit* faz com que o inversor gire no sentido anti-horário.
- *Fault Reset* [bit 2, byte 0]: uma transição de 0 para 1 neste *bit* reseta o *drive* quando este estiver na condição de erro.
- *NetCtrl* [bit 5, byte 0]: faz com que o controle do CFW-08 seja feito através da rede (*bits* 0,1 e 2).
- *NetRef (RPM)* [bit 6, byte 0]: faz com que a referência de velocidade considerada seja a rede.
- *Speed Reference* [bytes 2 e 3]: valor da referência de velocidade para o motor em RPM.

### 3.2 Instâncias 20/70 (*Polled, Change of State e Cyclic*)

Chamada de *Basic Speed*, estas instâncias representam a mais simples interface de operação de um equipamento segundo o perfil *AC/DC Device Profile*. O mapeamento dos dados é mostrado na tabela 5.

Monitoramento ( <i>Input</i> )									
Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running1		Faulted
	1	-							
	2	Speed Actual(byte low)							
	3	Speed Actual(byte high)							

Controle ( <i>Output</i> )									
Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Fault Reset		Run Fwd
	1	-							
	2	Speed Reference(byte low)							
	3	Speed Reference(byte high)							

Tabela 5: Instâncias 20/70 definidas pela ODVA

<sup>7</sup>Válido para instâncias 20/70 e 21/71.

### 3.3 Instâncias 21/71 (*Polled, Change of State e Cyclic*)

Chamada de *Extended Speed*, estas instâncias representam uma interface um pouco mais aprimorada de operação do equipamento que segue o perfil *AC/DC Device Profile*. O mapeamento dos dados é mostrado na tabela 6.

#### Monitoramento (*Input*)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
71	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running <sup>2</sup> (Rev)	Running <sup>1</sup> (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual(byte low)							
	3	Speed Actual(byte high)							

#### Controle (*Output*)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1	-							
	2	Speed Reference(byte low)							
	3	Speed Reference(byte high)							

Tabela 6: Instâncias 21/71 definidas pela ODVA

### 3.4 Instâncias 100/150 (*Polled, Change of State e Cyclic*)

Estas instâncias específicas WEG foram criadas tendo como objetivo permitir ao usuário escrever comandos no inversor, ler estados do drive e possibilitar a leitura e escrita de qualquer parâmetro do CFW-08. O mapeamento dos dados é mostrado na tabela 7.

Monitoramento ( <i>Input</i> )									
Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
150	0	Estado Lógico (código de erro)							
	1	Erro		Subtensão	Local ou Remoto	JOG	Sentido de Giro	Habilitado Geral	Rampa Habilitada
	2	Velocidade Real(byte low)							
	3	Velocidade Real(byte high)							

**Rampa Habilitada** : 0 = rampa desabilitada e 1 = rampa habilitada.

**Habilitado Geral** : 0 = habilita geral inativo e 1 = habilita geral ativo.

**Sentido de giro** : 0 = anti-horário e 1 = horário.

**JOG** : 0 = Jog inativo e 1 = Jog ativo.

**Local ou Remoto** : 0 = modo local e 1 = modo remoto.

**Subtensão** : 0 = sem subtensão e 1 = com subtensão.

**Erro** : 0 = *drive* sem erro e 1 = *drive* com erro.

Controle (*Output*)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
100	0	Reset de Erros			Local ou Remoto	JOG	Sentido de Giro	Habilita Geral	Habilita Rampa
	1	-							
	2	Referência de Velocidade (byte low)							
	3	Referência de Velocidade (byte high)							

Tabela 7: Instâncias 100/150 definidas pela WEG

**Habilita Rampa** : 0 = desabilita rampa e 1 = habilita rampa.

**Habilita Geral** : 0 = desabilita *drive* e 1 = habilita *drive*.

**Sentido de giro** : 0 = seleciona sentido de rotação horário e 1 = sentido de rotação anti-horário.

**JOG** : 0 = desativa função Jog e 1 = ativa função Jog.

**Local ou Remoto** : 0 = coloca *drive* em modo local e 1 = *drive* em modo remoto.

**Reset de Erros** : se o *drive* estiver em condição de erro, uma transição de 0 para 1 neste bit faz com que o CFW-08 seja resetado.

**NOTA!**

Durante o *reset* o *drive* ficará *offline*.

## 4 Parâmetros relacionados ao DeviceNet

O CFW-08 possui um conjunto de parâmetros, descritos a seguir, para a configuração do dispositivo na rede, e também para diagnóstico e monitoração dos estados do inversor.

Os demais parâmetros não citados aqui não possuem relação direta com comunicação DeviceNet, porém são importantes para a operação do inversor CFW-08. Desta forma, deve-se saber como utilizar o inversor via parâmetros, pois os mesmos também poderão ser utilizados durante sua operação via rede DeviceNet. Para a lista completa de parâmetros consulte o manual do usuário do CFW-08.

### 4.1 Novos parâmetros

Os parâmetros apresentados nesta seção estão disponíveis apenas na versão **A4** do cartão de controle.

#### 4.1.1 P070 - Estado do controlador CAN

Fornece a informação do estado do dispositivo com relação ao barramento CAN. Ele irá indicar se o controlador está operando corretamente, ou então informará o tipo de erro que o inversor apresenta com relação à comunicação.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Comunicação não está habilitada	-	leitura
1 = Auto-baud Running		
2 = Error Active (Sem erro)		
3 = Warning		
4 = Error Passive		
5 = Bus-Off		
6 = Sem alimentação		

Estes erros são função do número de telegramas inválidos recebidos ou transmitidos para a rede. O estado *error passive*, por exemplo, ocorre quando somente um equipamento está conectado à rede, enviando telegramas sem que outro equipamento reconheça esses telegramas. O estado *bus-off* pode ocorrer, por exemplo, quando dispositivos com diferentes taxas de comunicação são conectados na mesma rede ou devido a problemas na instalação, como a falta de resistores de terminação.

#### 4.1.2 P081 - Estado do mestre da rede

Indica o estado do mestre da rede.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Run	-	leitura
1 = Idle		

### 4.1.3 P313 - Ação para erro de comunicação

Quando o CFW-08 estiver sendo operado via rede e o mesmo estiver habilitado, se um erro de comunicação ocorrer (cabo rompido, queda na tensão de alimentação da rede, etc.), não será possível enviar comandos para desabilitá-lo. Dependendo da aplicação, isto pode ser um problema, e para evitá-lo pode-se programar no P313 uma ação para ser executada automaticamente.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Desabilita por rampa 1 = Desabilita geral 2 = Sem ação 3 = Vai para modo local	2	leitura/escrita



#### NOTA!

Esta ação somente será executada se o respectivo comando estiver habilitado para DeviceNet.

### 4.1.4 P700 - Protocolo CAN

O P700 permite a seleção do protocolo da camada de aplicação desejado para o barramento CAN no CFW-08. É necessário selecionar a opção '2' para habilitar a comunicação DeviceNet no inversor.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Desabilitado 1 = CANopen 2 = DeviceNet	0	leitura/escrita



#### NOTA!

A alteração do protocolo de aplicação do equipamento somente será válida após um *reset* do CFW-08.

### 4.1.5 P701 - Endereço na rede CAN

Este parâmetro possibilita a seleção do endereço (Mac ID) do CFW-08 na rede DeviceNet.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0-63	63	leitura/escrita

Cada dispositivo da rede precisa ter um Mac ID distinto e portanto pode haver até 64 dispositivos em uma única rede. Não há endereços reservados, embora o valor '0' seja frequentemente utilizado para endereçar o mestre da rede (*scanner* DeviceNet) e o valor '63' utilizado para identificar novos dispositivos na rede.

**NOTA!**

A alteração do endereço do equipamento somente será válida após um *reset* do CFW-08.

**4.1.6 P702 - Taxa de comunicação**

Este parâmetro possibilita a seleção da taxa de comunicação (*baud rate*) utilizada pelo dispositivo.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Auto-baud	0	leitura/escrita
1 = Auto-baud		
2 = 500 kbit/s		
3 = 250 kbit/s		
4 = 125 kbit/s		
5 = Auto-baud		
6 = Auto-baud		
7 = Auto-baud		
8 = Auto-baud		

Para que os dispositivos da rede possam se comunicar, é necessário que todos possuam o mesmo *baud rate* configurado. Não esquecer também que existe uma limitação da taxa de comunicação de acordo com o comprimento do cabo usado na instalação (ver tabela 4). Lembre-se que é fundamental que exista comunicação acontecendo na rede, ou seja, o mestre deve estar trocando dados com pelo menos um dispositivo para que o mecanismo de detecção de auto-baud funcione.

**NOTA!**

A alteração da taxa de comunicação somente será válida após um *reset* do CFW-08.

**4.1.7 P703 - Reset de bus-off**

Quando o número de erros ocorridos na rede CAN é muito grande, o dispositivo pode entrar no estado de *bus-off*, onde ele deixa de acessar a rede. Caso ocorra este erro, o parâmetro P703 permite programar se o CFW-08 deve permanecer no estado de *bus-off*, ou fazer o *reset* do erro automaticamente e reiniciar a comunicação.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Permanecer em erro	0	leitura/escrita
1 = Reset automático		

**4.1.8 P710 - Instâncias de I/O**

Neste parâmetro é feita a escolha das instâncias de I/O de acordo com a aplicação do usuário. Todas as instâncias suportadas pelo CFW-08 tem tamanho de 2 *words* (4 *bytes*). Ou seja, deve-se reservar, no mínimo, 4 *bytes* de entrada e 4 *bytes* de saída na memória do

CLP, independente do modo de comunicação escolhido, *Polled*, *Change of State* ou *Cyclic*. Esta programação é feita, em geral, no *software* de configuração da rede. Para maiores informações, consulte a documentação do controlador utilizado.

A descrição de cada uma destas instâncias é feita na seção 3.

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Instância 20/70 (2 palavras <i>I/O</i> )	0	leitura/escrita
1 = Instância 21/71 (2 palavras <i>I/O</i> )		
2 = Instância 100/150 (2 palavras <i>I/O</i> )		



### NOTA!

A alteração do conteúdo do P710 somente será válida após um *reset* do CFW-08.

## 4.2 Parâmetros alterados

Os parâmetros mostrados a seguir sofreram alterações em suas funções. Também são válidos apenas se o cartão de controle em uso for o modelo **A4**.

### 4.2.1 P220 - Seleção da fonte local/remoto

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Sempre local	2	leitura/escrita
1 = Sempre remoto		
2 = Tecla HMI-CFW08-P ou HMI-CFW08-RP (default: local)		
3 = Tecla HMI-CFW08-P ou HMI-CFW08-RP (default: remoto)		
4 = DI2 a DI4		
5 = DeviceNet (default: local)		
6 = DeviceNet (default: remoto)		

### 4.2.2 P221 - Seleção da referência de velocidade - situação local

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Teclas ▲ e ▼ HMI	0	leitura/escrita
1 = AI1		
2,3 = AI2		
4 = E.P.		
5 = DeviceNet		
6 = Multispeed		
7 = Soma AI $\geq 0$		
8 = Soma AI		

## 4.2.3 P222 - Seleção da referência de velocidade - situação remoto

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Teclas ▲ e ▼ HMI 1 = AI1 2,3 = AI2 4 = E.P. 5 = DeviceNet 6 = Multispeed 7 = Soma AI $\geq 0$ 8 = Soma AI	1	leitura/escrita

## 4.2.4 P229 - Seleção de comandos - situação local

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Teclas HMI-CFW08-P ou HMI-CFW08-RP 1 = Bornes 2 = DeviceNet	0	leitura/escrita

## 4.2.5 P230 - Seleção de comandos - situação remoto

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Teclas HMI-CFW08-P ou HMI-CFW08-RP 1 = Bornes 2 = DeviceNet	1	leitura/escrita

## 4.2.6 P231 - Seleção do sentido de giro - situação local e remoto

<i>Faixa de valores</i>	<i>Padrão</i>	<i>Acesso</i>
0 = Horário 1 = Anti-horário 2 = Comandos	2	leitura/escrita

## 5 Classes de objetos suportadas

Todo dispositivo DeviceNet é modelado por um conjunto de objetos. São eles os responsáveis por definir que funções determinado equipamento terá. Ou seja, de acordo com os objetos implementados, este equipamento poderá ser um adaptador de comunicação, um *drive* AC/DC, um sensor fotoelétrico, etc.. Objetos obrigatórios e opcionais são definidos em cada um destes perfis de dispositivos (*Device Profile*).

O inversor de frequência CFW-08 suporta todas classes obrigatórias do perfil *AC/DC Device Profile*. Suporta também classes específicas WEG. Detalhes de cada uma delas são apresentadas nas seções a seguir.

### 5.1 Classe *Identity* (01h)

Fornece informações gerais sobre a identidade do dispositivo, tais como *VendorID*, *Product Name*, *Serial Number*, etc.. Estão implementados os seguintes atributos:

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>	<i>Padrão</i>	<i>Descrição</i>
1	Get	Vendor ID	355h	Identificador do fabricante
2	Get	Product Type	2h	Tipo do produto
3	Get	Product Code		Código do produto
4	Get	Vendor Revision		Revisão do <i>firmware</i>
5	Get	Status		Estado atual do dispositivo
6	Get	Serial Number		Número serial
7	Get	Product Name	CFW-08	Nome do produto

Tabela 8: Classe *Identity*

### 5.2 Classe *Message Router* (02h)

Fornece informações sobre o objeto roteador de mensagens do tipo *explicit*. No CFW-08, esta classe não possui qualquer atributo implementado.

### 5.3 Classe *DeviceNet* (03h)

Responsável por manter a configuração e o estado das conexões físicas do nodo DeviceNet. Estão implementados os seguintes atributos:

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>	<i>Mín./Max</i>	<i>Padrão</i>	<i>Descrição</i>
1	Get/Set	Mac ID	0-63	63	Endereço do nodo
2	Get/Set	Baud rate	0-2	0	Taxa de transmissão
4	Get/Set	Bus-Off Counter	0-255		Contador de bus-off
5	Get	Allocation Information			Informação sobre o <i>allocation byte</i>

Tabela 9: Classe *DeviceNet*

## 5.4 Classe *Assembly* (04h)

Classe cuja função é agrupar diversos atributos numa única conexão. No CFW-08 apenas o atributo *Data* (3) está implementado (tabela 10).

<i>Atributos</i>	<i>Método</i>	<i>Nome</i>	<i>Descrição</i>
3	Get/Set	Data	Dados da instância

Tabela 10: Atributos das instâncias da classe *Assembly*

No CFW-08, a classe *Assembly* contém as seguintes instâncias. Para maiores informações, consulte a seção 3.

<i>Instâncias</i>	<i>Tamanho</i>	<i>Descrição</i>
20	2 palavras	DeviceNet AC/DC Profile
21	2 palavras	DeviceNet AC/DC Profile
100	2 palavras	Específica WEG
70	2 palavras	DeviceNet AC/DC Profile
71	2 palavras	DeviceNet AC/DC Profile
150	2 palavras	Específica WEG

Tabela 11: Instâncias da classe *Assembly*

## 5.5 Classe *Connection* (05h)

Instancia conexões do tipo *I/O* e *explicit*. Estão implementados os seguintes atributos:

### 5.5.1 Instância 1: *Explicit Message*

<i>Atributo</i>	<i>Método</i>	<i>Name</i>	<i>Descrição</i>
1	Get	State	Estado do objeto
2	Get	Instance Type	<i>I/O</i> ou <i>explicit</i>
3	Get	Transport Class Trigger	Define o comportamento da conexão
4	Get	Produced Connection ID	Identificador CAN de transmissão
5	Get	Consumed Connection ID	Identificador CAN de recepção
6	Get	Initial Comm. Charac.	Define o grupo de mensagens associado a esta conexão
7	Get	Produced Connection Size	Tamanho em <i>bytes</i> desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em <i>bytes</i> desta conexão de recepção
9	Get/Set	Expected Packet Rate	Define valores de tempo utilizados internamente
12	Get	Watchdog Timeout Action	Define como tratar contador <i>Inactivity/Watchdog</i>
13	Get	Produced Connection Path Length	Número de <i>bytes</i> da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de <i>bytes</i> da conexão consumidora
16	Get	Consumed Connection Path	Caminho dos objetos consumidores de dados
17	Get/Set	Production Inhibit Time	Define o tempo mínimo para nova produção de dados

Tabela 12: Classe *Connection* - Instância 1:*Explicit Message*

5.5.2 Instância 2: *Polled*

<i>Atributo</i>	<i>Método</i>	<i>Name</i>	<i>Descrição</i>
1	Get	State	Estado do objeto
2	Get	Instance Type	<i>I/O</i> ou <i>explicit</i>
3	Get	Transport Class Trigger	Define o comportamento da conexão
4	Get	Produced Connection ID	Identificador CAN de transmissão
5	Get	Consumed Connection ID	Identificador CAN de recepção
6	Get	Initial Comm. Charac.	Define o grupo de mensagens associado a esta conexão
7	Get	Produced Connection Size	Tamanho em <i>bytes</i> desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em <i>bytes</i> desta conexão de recepção
9	Get/Set	Expected Packet Rate	Define valores de tempo utilizados internamente
12	Get	Watchdog Timeout Action	Define como tratar contador <i>Inactivity/Watchdog</i>
13	Get	Produced Connection Path Length	Número de <i>bytes</i> da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de <i>bytes</i> da conexão consumidora
16	Get	Consumed Connection Path	Caminho dos objetos consumidores de dados
17	Get/Set	Production Inhibit Time	Define o tempo mínimo para nova produção de dados

Tabela 13: Classe *Connection* - Instância 2:*Polled*

5.5.3 Instância 4: *Change of State/Cyclic*

<i>Atributo</i>	<i>Método</i>	<i>Name</i>	<i>Descrição</i>
1	Get	State	Estado do objeto
2	Get	Instance Type	<i>I/O</i> ou <i>explicit</i>
3	Get	Transport Class Trigger	Define o comportamento da conexão
4	Get	Produced Connection ID	Identificador CAN de transmissão
5	Get	Consumed Connection ID	Identificador CAN de recepção
6	Get	Initial Comm. Charac.	Define o grupo de mensagens associado a esta conexão
7	Get	Produced Connection Size	Tamanho em <i>bytes</i> desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em <i>bytes</i> desta conexão de recepção
9	Get/Set	Expected Packet Rate	Define valores de tempo utilizados internamente
12	Get	Watchdog Timeout Action	Define como tratar contador <i>Inactivity/Watchdog</i>
13	Get	Produced Connection Path Length	Número de <i>bytes</i> da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de <i>bytes</i> da conexão consumidora
16	Get	Consumed Connection Path	Caminho dos objetos consumidores de dados
17	Get/Set	Production Inhibit Time	Define o tempo mínimo para nova produção de dados

Tabela 14: Classe *Connection* - Instância 4:*Change of State/Cyclic*

## 5.6 Classe *Motor Data* (28h)

Classe que armazena dados de placa do motor conectado ao inversor de frequência. Estão implementados os seguintes atributos:

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>	<i>Mín./Max</i>	<i>Unidade</i>	<i>Padrão</i>	<i>Descrição</i>
3	Get/Set	Motor Type	0-10		7	0 = Non Standard Motor 1 = PM DC Motor 2 = FC DC Motor 3 = PM Synchronous Motor 4 = FC Synchronous Motor 5 = Switched Reluctance Motor 6 = Wound Rotor Induction Motor 7 = Squirrel Cage Induction Motor 8 = Stepper Motor 9 = Sinusoidal PM BL Motor 10 = Trapezoidal PM BL Motor
6	Get/Set	Rated Current	0-999.9	100mA		Corrente nominal
7	Get/Set	Rated Voltage	0-600	V	220	Tensão nominal

Tabela 15: Classe *Motor Data*

## 5.7 Classe *Control Supervisor* (29h)

Responsável por modelar funções de gerenciamento do *drive*. Estão implementados os seguintes atributos:

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>	<i>Mín./Max</i>	<i>Padrão</i>	<i>Descrição</i>
3	Set	Run1	0-1		Run Fwd
4	Set	Run2	0-1		Run Rev
5	Set	NetCtrl	0-1	0	0 = Controle local 1 = Controle via rede
6	Get	State	0-7		0 = Vendor specific 1 = Startup 2 = Not Ready 3 = Ready 4 = Enabled 5 = Stopping 6 = Fault Stop 7 = Fault
7	Get	Running1	0-1	0	0 = Outro estado 1 = (Enabled e Run1) ou (Stopping e Running1) ou (Fault Stop e Running1)
8	Get	Running2	0-1	0	0 = Outro estado 1 = (Enabled e Run2) ou (Stopping e Running2) ou (Fault Stop e Running2)
9	Get	Ready	0-1	0	0 = Outro estado 1 = Ready ou Enabled ou Stopping
10	Get	Faulted	0-1	0	0 = Sem falhas 1 = Falha ocorrida
11	Get	Warning	0	0	0 = Sem warnings
12	Set	Fault Reset	0-1	0	0 = Sem ação 0 → 1 = Reset de erros
15	Get	Ctrl from Net	0-1	0	0 = Controle é local 1 = Controle é via rede

Tabela 16: Classe *Control Supervisor*

## 5.8 Classe *AC/DC Drive* (2Ah)

Contém informações específicas de um *AC/DC Drive* tais como modo de operação e escalas de velocidade e torque. Estão implementados os seguintes atributos:

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>	<i>Mín./Max</i>	<i>Padrão</i>	<i>Descrição</i>
4	Get/Set	NetRef	0-1	0	0 = Referência local 1 = Referência via rede
6	Get/Set	DriveMode	1-2	2	1 = Speed control (open loop) 2 = Speed control (closed loop)
7	Get	Speed Actual	0-9999		Velocidade real (melhor aproximação)
8	Get/Set	Speed Ref	0-9999	0	Referência de velocidade

Tabela 17: Classe *AC/DC Drive*



### NOTA!

O CFW-08 irá operar em modo de velocidade (controle escalar ou vetorial) independente do conteúdo do atributo **DriveMode**.

## 5.9 Classe *Acknowledge Handler* (2Bh)

A função desta classe é gerenciar a recepção de mensagens de reconhecimento (*acknowledge*).

<i>Atributo</i>	<i>Método</i>	<i>Nome</i>
1	Get/Set	Acknowledge Timer
2	Get	Retry Limit
3	Get	COS Production Connection Instance

Tabela 18: Classe *Acknowledge Handler*

## 5.10 Classes WEG

As classes específicas WEG são utilizadas para mapear todos os parâmetros do CFW-08. Elas permitem que o usuário leia e escreva em qualquer parâmetro através da rede. Para isto mensagens DeviceNet do tipo *explicit* são usadas.

Existem faixas separadas para cada grupo de parâmetros, conforme mostrado na tabela 19:

<i>Faixa de valores</i>	<i>Classe</i>	<i>Nome</i>
Parâmetros 000-099	Classe 100 (64h)	WEG_CLASS_F1
Parâmetros 100-199	Classe 101 (65h)	WEG_CLASS_F2
Parâmetros 200-299	Classe 102 (66h)	WEG_CLASS_F3
Parâmetros 300-399	Classe 103 (67h)	WEG_CLASS_F4
Parâmetros 400-499	Classe 104 (68h)	WEG_CLASS_F5
Parâmetros 500-599	Classe 105 (69h)	WEG_CLASS_F6
Parâmetros 600-699	Classe 106 (6Ah)	WEG_CLASS_F7
Parâmetros 700-799	Classe 107 (6Bh)	WEG_CLASS_F8

Tabela 19: Classes WEG

<i>Parâmetro</i>	<i>Classe</i>	<i>Instância</i>	<i>Atributo</i>
P000	Classe 100 (64h)	1	100
P001	Classe 100 (64h)	1	101
P002	Classe 100 (64h)	1	102
⋮	⋮	⋮	⋮
P100	Classe 101 (65h)	1	100
P101	Classe 101 (65h)	1	101
P102	Classe 101 (65h)	1	102
⋮	⋮	⋮	⋮
P200	Classe 102 (66h)	1	100
P201	Classe 102 (66h)	1	101
P202	Classe 102 (66h)	1	102
⋮	⋮	⋮	⋮
P300	Classe 103 (67h)	1	100
P301	Classe 103 (67h)	1	101
P302	Classe 103 (67h)	1	102
⋮	⋮	⋮	⋮

Tabela 20: Parâmetros das classes WEG



### NOTA!

Para estas classes WEG, o CFW-08 utiliza apenas a instância 1.

**NOTA!**

Também para estas classes WEG, os parâmetros são acessados adicionando o valor decimal 100 aos dígitos da dezena de qualquer parâmetro. Este novo número é chamado de atributo.

Por exemplo:

**Parâmetro 23** : classe 64h, instância 1, atributo 123. Este caminho dá acesso ao P23.

**Parâmetro 100** : classe 65h, instância 1, atributo 100. Este caminho dá acesso ao P100.

**Parâmetro 202** : classe 66h, instância 1, atributo 102. Este caminho dá acesso ao P202.

**Parâmetro 432** : classe 68h, instância 1, atributo 132. Este caminho dá acesso ao P432.

## 6 Outras funções

### 6.1 Função *Automatic Device Replacement*

Este recurso, quando disponível no mestre da rede, permite que as configurações de cada escravo da rede sejam armazenadas em sua memória interna. Isto permite que um equipamento da rede seja substituído por outro idêntico sem que seja necessária uma nova reconfiguração. Isto porque o mestre verifica constantemente se há algum escravo cujo endereço valha '63'. Quando isto ocorrer, o mestre compara os atributos do objeto identidade deste novo nodo com algum que está em sua *scan list*. Se o mestre perder a comunicação com algum nodo que está em sua *scan list* e for encontrado um *drive* idêntico no endereço '63', as configurações previamente salvas no CLP serão automaticamente transferidas para este novo dispositivo (reparametrização automática).

A comunicação inicia-se logo após a transferência destes dados e tudo volta ao normal com o mínimo de tempo possível de parada. O inversor CFW-08 está pronto para utilizar esta função sem que seja necessária intervenção do usuário. A ativação deste recurso é feita em geral através do *software* de configuração e programação da rede. Verifique a documentação que acompanha o módulo mestre da DeviceNet.



#### **NOTA!**

Para que este recurso funcione, é necessário programar o protocolo de aplicação DeviceNet (P700 = 2).

## Glossário

***Predefined Master/Slave Connection Set:*** conjunto de conexões que facilitam a comunicação tipicamente encontrada em aplicações mestre/escravo. Muitos dos passos envolvidos na criação e configuração foram removidos da definição do *Master/Slave Connection Set*. Assim, a comunicação pode ser estabelecida com menos recursos de rede e do dispositivo.

***UCMM (Unconnected Message Manager):*** objeto presente em alguns dispositivos DeviceNet, o qual permite estabelecer comunicação ponto-a-ponto sem auxílio do mestre.

***Group 2 Only Server:*** escravo (servidor) que não possui objeto *UCMM* e deve utilizar o *Predefined Master/Slave Connection Set* para estabelecer comunicação. Um *Group 2 Only Server* pode transmitir e receber apenas aqueles identificadores definidos pelo *Predefined Master/Slave Connection Set*. O inversor CFW-08 atua como tal.

***Group 2 Server:*** escravo (servidor) que possui o objeto *UCMM* e portanto pode estabelecer conexões ponto-a-ponto com outros escravos.

***Scan List:*** lista de escravos do mestre da rede. Apenas escravos pertencentes a esta lista poderão comunicar-se com o mestre através de mensagens do tipo *I/O*.

***DeviceNet Master:*** dispositivo que coleta e distribui dados dos escravos da rede de acordo com sua *scan list*. A comunicação sempre tem início com o mestre. Esta função de mestre é, em geral, desempenhada por um módulo *scanner* num CLP.

***DeviceNet Slave:*** dispositivo que retorna dados para o mestre quando consultado. O CFW-08 atua como tal numa rede DeviceNet.

***Mac ID:*** atributo que representa o endereço do nodo na rede.

***Polled:*** tipo de comunicação *I/O* em que o mestre envia um telegrama à um escravo que responde imediatamente. Este processo é repetido enquanto houver escravos na *scan list* do mestre. Após todos terem sido consultados, a seqüência é repetida, indefinidamente.

***Change of State:*** outro método de comunicação, em que a troca de dados entre mestre e escravo, ocorre apenas quando houver mudanças dos valores monitorados e controlados; até um certo limite de tempo. Quando este limite for atingido, a transmissão e recepção ocorrerá mesmo que não tenha havido alterações. A configuração desta variável de tempo é feita no programa de configuração da rede.

***Cyclic:*** outro método de comunicação muito semelhante ao anterior. A única diferença fica por conta da produção e consumo de mensagens. Neste tipo de comunicação toda troca de dados ocorre em intervalos regulares de tempo, independente de terem sido alterados ou não. Este período também é ajustado no *software* de configuração da rede.

***Assembly:*** classe cuja função é agrupar atributos de múltiplos objetos numa única conexão.

***AC/DC Device Profile:*** definição do perfil de um dispositivo do tipo AC/DC. Fornece uma interface padronizada de apresentação dos dados ao usuário.