

DeviceNet

CFW700

Manual do Usuário

Idioma: Português





Manual do Usuário DeviceNet

Série: CFW700

Idioma: Português

N ° do Documento: 10000828098 / 00

Data da Publicação: 05/2010

SUMÁRIO

SUMÁRIO.....	3
SOBRE O MANUAL.....	5
ABREVIações E DEFINIções.....	5
REPRESENTAção NUMéRICA.....	5
DOCUMENTOS.....	5
1 INTRODUÇÃO À COMUNICAÇÃO DEVICENET	6
1.1 CAN.....	6
1.1.1 Frame de Dados.....	6
1.1.2 Frame Remoto.....	6
1.1.3 Acesso à Rede	6
1.1.4 Controle de Erros.....	6
1.1.5 CAN e DeviceNet.....	7
1.2 CARACTERÍSTICAS DA REDE DEVICENET	7
1.3 MEIO FÍSICO.....	7
1.3.1 Camada de Enlace de Dados	8
1.3.2 Camada de Transporte e Rede	8
1.3.3 Camada de Aplicação – Protocolo CIP	9
1.3.4 Arquivo EDS.....	9
1.3.5 Modos de Comunicação	10
1.3.6 Conjunto de Conexões Predefinidas Mestre/Escravo.....	10
2 ACESSÓRIOS PARA COMUNICAÇÃO DEVICENET	11
2.1 MÓDULO CAN-01.....	11
2.2 PINAGEM DO CONECTOR	11
2.3 FONTE DE ALIMENTAÇÃO.....	11
2.4 INDICAções.....	11
2.5 CONEXÃO COM A REDE.....	11
2.6 CONFIGURAÇÃO DO MÓDULO.....	12
2.7 ACESSO AOS PARÂMETROS	12
3 INSTALAÇÃO DA REDE DEVICENET	13
3.1 TAXA DE COMUNICAÇÃO	13
3.2 ENDEREÇO NA REDE DEVICENET	13
3.3 RESISTORES DE TERMINAÇÃO.....	13
3.4 CABO	13
3.5 LIGAÇÃO NA REDE.....	14
4 PARAMETRIZAÇÃO.....	15
4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES.....	15
P0105 – SELEÇÃO 1ª/2ª RAMPA.....	15
P0220 – SELEÇÃO FONTE LOCAL/REMOTO	15
P0221 – SELEÇÃO REFERÊNCIA LOCAL	15
P0222 – SELEÇÃO REFERÊNCIA REMOTA.....	15
P0223 – SELEÇÃO GIRO LOCAL	15
P0224 – SELEÇÃO GIRA/PARA LOCAL	15
P0225 – SELEÇÃO JOG LOCAL	15
P0226 – SELEÇÃO GIRO REMOTO.....	15
P0227 – SELEÇÃO GIRA/PARA REMOTO.....	15
P0228 – SELEÇÃO JOG REMOTO	15
P0313 – AÇÃO PARA ERRO DE COMUNICAÇÃO.....	15

P0680 – ESTADO LÓGICO	16
P0681 – VELOCIDADE DO MOTOR EM 13 BITS	17
P0684 – PALAVRA DE CONTROLE VIA DEVICENET	18
P0685 – REFERÊNCIA DE VELOCIDADE VIA DEVICENET	18
P0695 – VALOR PARA AS SAÍDAS DIGITAIS	19
P0696 – VALOR 1 PARA SAÍDAS ANALÓGICAS	20
P0697 – VALOR 2 PARA SAÍDAS ANALÓGICAS	20
P0700 – PROTOCOLO CAN	21
P0701 – ENDEREÇO CAN	21
P0702 – TAXA DE COMUNICAÇÃO CAN	21
P0703 – RESET DE BUS OFF	22
P0705 – ESTADO DO CONTROLADOR CAN	22
P0706 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS	23
P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS	23
P0708 – CONTADOR DE ERROS DE BUS OFF	23
P0709 – CONTADOR DE MENSAGENS CAN PERDIDAS	23
P0710 – INSTÂNCIAS DE I/O DEVICENET	24
P0711 – LEITURA #3 DEVICENET	27
P0712 – LEITURA #4 DEVICENET	27
P0713 – LEITURA #5 DEVICENET	27
P0714 – LEITURA #6 DEVICENET	27
P0715 – ESCRITA #3 DEVICENET	27
P0716 – ESCRITA #4 DEVICENET	27
P0717 – ESCRITA #5 DEVICENET	27
P0718 – ESCRITA #6 DEVICENET	27
P0719 – ESTADO DA REDE DEVICENET	28
P0720 – ESTADO DO MESTRE DEVICENET	28
5 CLASSES DE OBJETOS SUPORTADAS	29
5.1 CLASSE IDENTITY (01H)	29
5.2 CLASSE MESSAGE ROUTER (02H)	29
5.3 CLASSE DEVICENET (03H)	29
5.4 CLASSE ASSEMBLY (04H)	29
5.5 CLASSE CONNECTION (05H)	30
5.5.1 Instância 1: Explicit Message	30
5.5.2 Instância 2: Polled	30
5.5.3 Instância 4: Change of State/Cyclic	31
5.6 CLASSE MOTOR DATA (28H)	31
5.7 CLASSE CONTROL SUPERVISOR (29H)	31
5.8 CLASSE AC/DC DRIVE (2AH)	32
5.9 CLASSE ACKNOWLEDGE HANDLER (2BH)	32
5.10 CLASSES ESPECÍFICAS DO FABRICANTE	33
6 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO DEVICENET	34
A133/F233 – SEM ALIMENTAÇÃO NA INTERFACE CAN	34
A134/F234 – <i>BUS OFF</i>	34
A136/F236 – MESTRE EM <i>IDLE</i>	34
A137/F237 – TIMEOUT NA CONEXÃO DEVICENET	35

SOBRE O MANUAL

Este manual fornece a descrição necessária para a operação do inversor de frequência CFW700 utilizando o protocolo DeviceNet. Este manual deve ser utilizado em conjunto com manual do usuário do CFW700.

ABREVIações E DEFINIções

ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CiA	CAN in Automation
CIP	Common Industrial Protocol
PLC	Programmable Logic Controller
HMI	Human-Machine Interface
ODVA	Open DeviceNet Vendor Association
ro	Read only (somente leitura)
rw	Read/write (leitura e escrita)

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 foi desenvolvido baseado nas seguintes especificações e documentos:

Documento	Versão	Fonte
CAN Specification	2.0	CiA
Volume One Common Industrial Protocol (CIP) Specification	3.2	ODVA
Volume Three DeviceNet Adaptation of CIP	1.4	ODVA

Para obter esta documentação, deve-se consultar a ODVA, que atualmente é a organização que mantém, divulga e atualiza as informações relativas à rede DeviceNet.

1 INTRODUÇÃO À COMUNICAÇÃO DEVICENET

Para a operação de um equipamento em rede DeviceNet, é necessário conhecer a forma como a comunicação é feita. Este item traz uma descrição geral do funcionamento do protocolo DeviceNet, contendo as funções utilizadas pelo CFW700. Para uma descrição mais detalhada pode-se consultar a especificação do protocolo.

1.1 CAN

A rede DeviceNet é uma rede baseada em CAN, o que significa dizer que ela utiliza telegramas CAN para troca de dados na rede.

O protocolo CAN é um protocolo de comunicação serial que descreve os serviços da camada 2 do modelo ISO/OSI (camada de enlace de dados)¹. Nesta camada, são definidos os diferentes tipos de telegramas (frames), a forma de detecção de erros, validação e arbitragem de mensagens.

1.1.1 Frame de Dados

Os dados em uma rede CAN são transmitidos através de um frame de dados. Este tipo de frame é composto principalmente por um campo identificador de 11 bits² (arbitration field), e um campo de dados (data field), que pode conter até 8 bytes de dados.

Identificador	8 bytes de dados							
11 bits	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

1.1.2 Frame Remoto

Além do frame de dados, existe também o frame remoto (RTR frame). Este tipo de frame não possui campo de dados, apenas o identificador. Ele funciona como uma requisição para que outro dispositivo da rede transmita o frame de dados desejado. O protocolo DeviceNet não utiliza este tipo de frame.

1.1.3 Acesso à Rede

Em uma rede CAN, qualquer elemento da rede pode tentar transmitir um frame para a rede em um determinado instante. Caso dois elementos tentem acessar a rede ao mesmo tempo, conseguirá transmitir aquele que enviar a mensagem mais prioritária. A prioridade da mensagem é definida pelo identificador do frame CAN, quanto menor o valor deste identificador, maior a prioridade da mensagem. O telegrama com o identificador 0 (zero) corresponde ao telegrama mais prioritário.

1.1.4 Controle de Erros

A especificação CAN define diversos mecanismos para controle de erros, o que a torna uma rede muito confiável e com um índice muito baixo de erros de transmissão que não são detectados. Cada dispositivo da rede deve ser capaz de identificar a ocorrência destes erros, e informar os demais elementos que um erro foi detectado.

Um dispositivo da rede CAN possui contadores internos que são incrementados toda vez que um erro de transmissão ou recepção é detectado, e decrementado quando um telegrama é enviado ou recebido com sucesso. Cada dispositivo na rede CAN pode ser levado para os seguintes estados, de acordo com a quantidade de erros de transmissão ou recepção detectados:

- **Error Active:** os contadores internos de erro estão em um nível baixo e o dispositivo opera normalmente na rede CAN. Pode enviar e receber telegramas e atuar na rede CAN caso detecte algum erro na transmissão de telegramas.
- **Warning:** quando algum destes contadores passa de um determinado limite, o dispositivo entra no estado de *warning*, significando a ocorrência de uma elevada taxa de erros de comunicação.
- **Error Passive:** quando este valor ultrapassa um limite maior, ele entra no estado de *error passive*, onde ele para de atuar na rede ao detectar que outro dispositivo enviou um telegrama com erro.

¹ Na especificação do protocolo CAN, é referenciada a norma ISO 11898 como definição da camada 1 deste modelo (camada física).

² A especificação CAN 2.0 define dois tipos de frames de dados: *standard* (11 bits) e *extended* (29 bits). Para esta implementação do protocolo DeviceNet, somente frames *standard* são aceitos.

- **Bus Off:** por último, temos o estado de *bus off*, no qual o dispositivo não irá mais enviar ou receber telegramas. O dispositivo opera como se estivesse desconectado da rede.

1.1.5 CAN e DeviceNet

Somente a definição de como detectar erros, criar e transmitir um frame não são suficientes para definir um significado para os dados que são enviados via rede. É necessário que haja uma especificação que indique como o identificador e os dados devem ser montados e como as informações devem ser trocadas. Desta forma os elementos da rede podem interpretar corretamente os dados que são transmitidos. Neste sentido, a especificação DeviceNet define justamente como trocar dados entre os equipamentos e como cada dispositivo deve interpretar estes dados.

Existem diversos protocolos baseados em CAN, como DeviceNet, CANopen, J1939, etc., que utilizam frames CAN para a comunicação. Porém estes protocolos não podem operar em conjunto na mesma rede.

1.2 CARACTERÍSTICAS DA REDE DEVICENET

Apresentado em 1994, DeviceNet é uma implementação do protocolo *Common Industrial Protocol* (CIP) para redes de comunicação industrial. Desenvolvido originalmente pela Allen-Bradley, teve sua tecnologia transferida para a ODVA que, desde então, mantém, divulga e promove o DeviceNet e outras redes baseadas no protocolo CIP3. Além disso, utiliza o protocolo *Controller Area Network* (CAN) para enlace de dados e acesso ao meio, camadas 2 e 1 do modelo OSI/ISO, respectivamente.

Utilizado principalmente na interligação de controladores industriais e dispositivos de entrada/saída (I/O), o protocolo segue o modelo produtor-consumidor, suporta múltiplos modos de comunicação e possui prioridade entre mensagens.

É um sistema que pode ser configurado para operar tanto numa arquitetura mestre-escravo quanto numa arquitetura distribuída ponto a ponto. Além disso, define dois tipos de mensagens, I/O (dados de processo) e *explicit* (configuração e parametrização). Possui também mecanismos de detecção de endereços duplicados e isolamento dos nodos em caso de falhas críticas.

Uma rede DeviceNet pode conter até 64 dispositivos, endereçados de 0 a 63. Qualquer um destes pode ser utilizado. Não há qualquer restrição, embora se deva evitar o 63, pois este costuma ser utilizado para fins de comissionamento.

1.3 MEIO FÍSICO

DeviceNet usa uma topologia de rede do tipo tronco/derivação que permite que tanto a fiação de sinal quanto de alimentação estejam presentes no mesmo cabo. Esta alimentação, fornecida por uma fonte conectada diretamente na rede, supre os transceivers CAN dos nodos, e possui as seguintes características:

- 24Vdc;
- Saída DC isolada da entrada AC;
- Capacidade de corrente compatível com os equipamentos instalados.

O tamanho total da rede varia de acordo com a taxa de transmissão utilizada, conforme mostrado na tabela abaixo.

Tabela 1.1: Tamanho da rede x Taxa de transmissão

Taxa de transmissão	Tamanho da rede	Derivação	
		Máximo	Total
125kbps	500m	6m	156m
250kbps	250m		78m
500kbps	100m		39m

³ CIP representa, na realidade, uma família de redes. DeviceNet, EtherNet/IP e ControlNet utilizam CIP na camada de aplicação. A diferença entre eles está primordialmente nas camadas de enlace de dados e física.

Para evitar reflexões de sinal na linha, recomenda-se a instalação de resistores de terminação nas extremidades da rede, pois a falta destes pode provocar erros intermitentes. Este resistor deve possuir as seguintes características, conforme especificação do protocolo:

- 121Ω;
- 0,25W;
- 1% de tolerância.

Em DeviceNet, diversos tipos de conectores podem ser utilizados, tanto selados quanto abertos. A definição do tipo a ser utilizado dependerá da aplicação e do ambiente de operação do equipamento. O CFW700 utiliza um conector do tipo *plug-in* de 5 vias cuja pinagem está mostrada na seção 2. Para uma descrição completa dos conectores utilizados pelo DeviceNet consulte a especificação do protocolo.

1.3.1 Camada de Enlace de Dados

A camada de enlace de dados do DeviceNet é definida pela especificação do CAN, o qual define dois estados possíveis; dominante (nível lógico 0) e recessivo (nível lógico 1). Um nodo pode levar a rede ao estado dominante se transmitir alguma informação. Assim, o barramento somente estará no estado recessivo se não houver nodos transmissores no estado dominante.

CAN utiliza o CSMA/NBA para acessar o meio físico. Isto significa que um nodo, antes de transmitir, deve verificar se o barramento está livre. Caso esteja, então ele pode iniciar a transmissão do seu telegrama. Caso não esteja, deve aguardar. Se mais de um nodo acessar a rede simultaneamente, um mecanismo baseado em prioridade de mensagem entrará em ação para decidir qual deles terá prioridade sobre os outros. Este mecanismo é não destrutivo, ou seja, a mensagem é preservada mesmo que ocorra colisão entre dois ou mais telegramas.

CAN define quatro tipos de telegramas (*data*, *remote*, *overload*, *error*). Destes, DeviceNet utiliza apenas o frame de dados (*data frame*) e o frame de erros (*error frame*).

Dados são movimentados utilizando-se o frame de dados. A estrutura deste frame é mostrada na Figura 1.1.

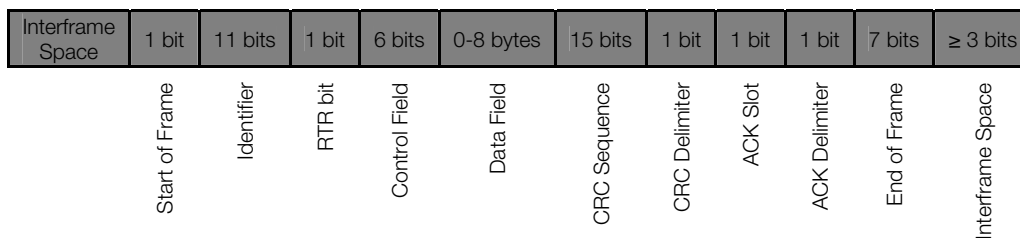


Figura 1.1: Frame de dados CAN

Já os erros são indicados através do frame de erros. CAN possui uma verificação e um confinamento de erros bastante robusto. Isto garante que um nodo com problemas não prejudique a comunicação na rede.

Para uma descrição completa dos erros, consulte a especificação do CAN.

1.3.2 Camada de Transporte e Rede

DeviceNet requer que uma conexão seja estabelecida antes de haver troca de dados com o dispositivo. Para estabelecer esta conexão, cada nodo DeviceNet deve implementar o *Unconnected Message Manager* (UCMM) ou o *Group 2 Unconnected Port*. Estes dois mecanismos de alocação utilizam mensagens do tipo explicit para estabelecer a conexão, que a seguir será utilizada para a troca de dados de processo entre um nodo e outro. Esta troca de dados utiliza mensagens do tipo I/O (ver item 1.3.5).

Os telegramas DeviceNet são classificados em grupos, o qual definem funções e prioridades específicas. Estes telegramas utilizam o campo identificador (11 bits) do frame de dados CAN para identificar unicamente cada uma das mensagens, garantindo assim o mecanismo de prioridades CAN.

Um nodo DeviceNet pode ser cliente, servidor ou ambos. Além disso, clientes e servidores podem ser produtores e/ou consumidores de mensagens. Num típico nodo cliente, por exemplo, sua conexão produzirá requisições e consumirá respostas. Outras conexões de clientes ou servidores apenas consumirão mensagens. Ou seja, o protocolo prevê diversas possibilidades de conexão entre os dispositivos.

O protocolo dispõe também de um recurso para detecção de nodos com endereços (Mac ID) duplicados. Evitar que endereços duplicados ocorram é, em geral, mais eficiente que tentar localizá-los depois.

1.3.3 Camada de Aplicação – Protocolo CIP

DeviceNet utiliza o *Common Industrial Protocol* (CIP) na camada de aplicação. Trata-se de um protocolo estritamente orientado a objetos utilizado também pelo ControlNet e pelo EtherNet/IP. Ou seja, ele é independente do meio físico e da camada de enlace de dados. A Figura 1.2 apresenta a estrutura deste protocolo.

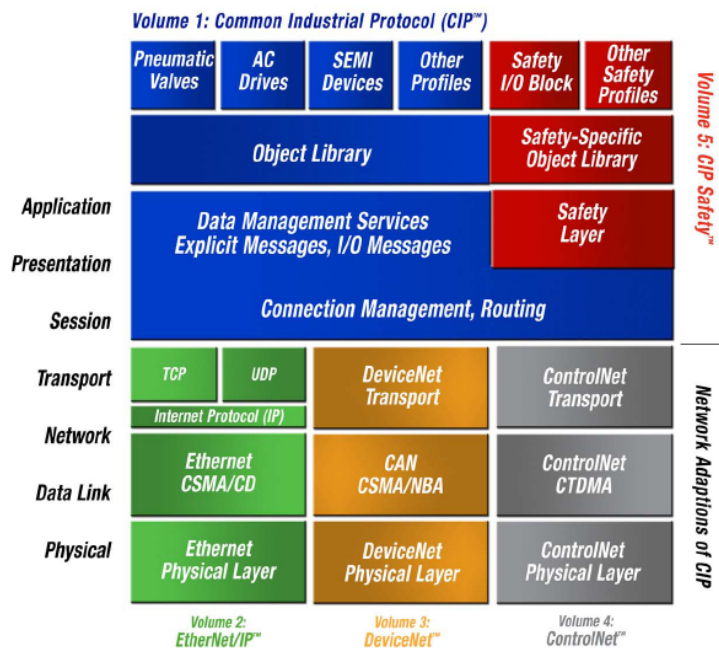


Figura 1.2: Estrutura em camadas do protocolo CIP

CIP tem dois objetivos principais:

- Transporte de dados de controle dos dispositivos de I/O.
- Transporte de informações de configuração e diagnóstico do sistema sendo controlado.

Um nodo (mestre ou escravo) DeviceNet é então modelado por um conjunto de objetos CIP, os quais encapsulam dados e serviços e determinam assim seu comportamento.

Existem objetos obrigatórios (todo dispositivo deve conter) e objetos opcionais. Objetos opcionais são aqueles que moldam o dispositivo conforme a categoria (chamado de perfil) a que pertencem, tais como: AC/DC Drive, leitor de código de barras ou válvula pneumática. Por serem diferentes, cada um destes conterá um conjunto também diferente de objetos.

Para mais informações, consulte a especificação do DeviceNet. Ela apresenta a lista completa dos perfis de dispositivos já padronizados pela ODVA, bem como os objetos que o compõem.

1.3.4 Arquivo EDS

Cada dispositivo em uma rede DeviceNet possui um arquivo de configuração EDS, que contém informações sobre o funcionamento do dispositivo e deve ser registrado no software de configuração, para programação dos dispositivos presentes na rede DeviceNet.

O arquivo de configuração EDS é fornecido em um CD juntamente com o produto, e também pode ser obtido através do site <http://www.weg.net>. É necessário observar a versão de software do equipamento, para utilizar um arquivo EDS que seja compatível com esta versão.

1.3.5 Modos de Comunicação

O protocolo DeviceNet possui dois tipos básicos de mensagens, I/O e explicit. Cada um deles é adequado a um determinado tipo de dado, conforme descrito abaixo:

- I/O: tipo de telegrama síncrono dedicado à movimentação de dados prioritários entre um produtor e um ou mais consumidores. Dividem-se de acordo com o método de troca de dados. Os principais são:
 - *Polled*: método de comunicação em que o mestre envia um telegrama a cada um dos escravos da sua lista (*scan list*). Assim que recebe a solicitação, o escravo responde prontamente a solicitação do mestre. Este processo é repetido até que todos sejam consultados, reiniciando o ciclo.
 - *Bit-strobe*: método de comunicação onde o mestre envia para a rede um telegrama contendo 8 bytes de dados. Cada bit destes 8 bytes representa um escravo que, se endereçado, responde de acordo com o programado.
 - *Change of State*: método de comunicação onde a troca de dados entre mestre e escravo ocorre apenas quando houver mudanças nos valores monitorados/controlados, até um certo limite de tempo. Quando este limite é atingido, a transmissão e recepção ocorrerão 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, 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 de rede.
- Explicit: tipo de telegrama de uso geral e não prioritário. Utilizado principalmente em tarefas assíncronas tais como parametrização e configuração do equipamento.



NOTA!

O inversor de frequência CFW700 não disponibiliza o método de comunicação *Bit-strobe*.

1.3.6 Conjunto de Conexões Predefinidas Mestre/Escravo

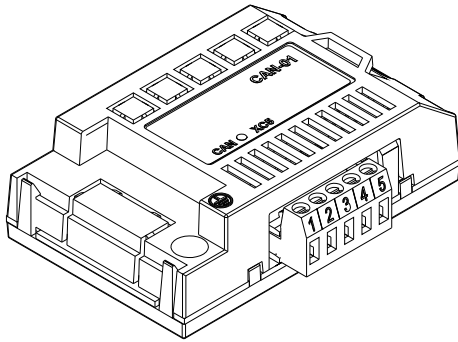
DeviceNet emprega fundamentalmente um modelo de mensagens ponto a ponto. Contudo, é bastante comum utilizar um esquema predefinido de comunicação baseado no mecanismo mestre/escravo.

Este esquema emprega um movimento simplificado de mensagens do tipo I/O muito comum em aplicações de controle. A vantagem deste método está nos requisitos necessários para rodá-lo, em geral menores se comparados ao UCMM. Até mesmo dispositivos simples com recursos limitados (memória, processador de 8 bits) são capazes de executar o protocolo.

2 ACESSÓRIOS PARA COMUNICAÇÃO DEVICENET

Para possibilitar a comunicação DeviceNet no produto, é necessário utilizar um dos acessórios para interface CAN descritos a seguir. Informações sobre a instalação destes módulos podem ser obtidas no guia que acompanha cada acessório.

2.1 MÓDULO CAN-01



- Item WEG: 10051961.
- Composto pelo módulo de comunicação CAN (figura ao lado) e um guia de montagem.
- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Alimentação externa de 24V.
- Permite a conexão de até 64 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada com o uso de repetidores⁴.
- Comprimento máximo do barramento de 1000 metros.

2.2 PINAGEM DO CONECTOR

O módulo para comunicação CAN possui um conector *plug-in* de 5 vias (XC5) com a seguinte pinagem:

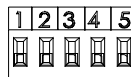


Tabela 2.1: Pinagem do conector XC5 para interface CAN

Pino	Nome	Função
1	V-	Polo negativo da fonte de alimentação
2	CAN_L	Sinal de comunicação CAN_L
3	Shield	Blindagem do cabo
4	CAN_H	Sinal de comunicação CAN_H
5	V+	Polo positivo da fonte de alimentação

2.3 FONTE DE ALIMENTAÇÃO

Os módulos de interface CAN necessitam de uma tensão de alimentação externa entre os pinos 1 e 5 do conector da rede. Os dados para consumo individual e tensão de entrada são apresentados na tabela a seguir.

Tabela 2.2: Características da alimentação para interface CAN

Tensão de alimentação (V _{cc})		
Mínimo	Máximo	Recomendado
11	30	24
Corrente (mA)		
Típico		Máximo
30		50

2.4 INDICAÇÕES

Os módulos de interface CAN possuem um LED na cor verde para indicar que a interface está alimentada. Demais indicações de alarmes, falhas e estados da comunicação são feitas através da HMI e dos parâmetros do produto.

2.5 CONEXÃO COM A REDE

Para a ligação do inversor utilizando a interface ativa DeviceNet, os seguintes pontos devem ser observados:

- Recomenda-se a utilização de cabos específicos para redes CAN/DeviceNet.

⁴ O número limite de equipamentos que podem ser conectados na rede também depende do protocolo utilizado.

- Aterramento da malha do cabo (blindagem) somente em um ponto, evitando assim loops de corrente. Este ponto costuma ser a própria fonte de alimentação da rede. Se houver mais de uma fonte de alimentação, somente uma delas deverá estar ligada ao terra de proteção.
- Instalação de resistores de terminação somente nos extremos do barramento principal, mesmo que existam derivações.
- A fonte de alimentação da rede deve ser capaz de suprir corrente para alimentar todos os transceivers dos equipamentos. O módulo DeviceNet do CFW700 consome em torno de 50mA.

2.6 CONFIGURAÇÃO DO MÓDULO

Para configurar o módulo DeviceNet siga os passos indicados abaixo:

- Com o inversor desligado instale o módulo no conector XC43 (slot 3 - verde).
- Certifique-se de que ele está corretamente encaixado e fixado pelos parafusos.
- Energize o inversor.
- Verifique o conteúdo do parâmetro P0028 e verifique se o acessório foi corretamente reconhecido. Consulte a bula de instalação se necessário.
- Ajuste o protocolo CAN para DeviceNet através do parâmetro P0700.
- Ajuste o endereço do inversor na rede através do parâmetro P0701.
 - Valores válidos: 0 a 63.
- Ajuste a taxa de comunicação no P0702. Valores válidos:
 - 0 = Auto
 - 1 = Auto
 - 2 = 500kbps
 - 3 = 250kbps
 - 4 = 125kbps
 - 5 = Auto
 - 6 = Auto
 - 7 = Auto
 - 8 = Auto
- No parâmetro P0710 configure a instância de I/O mais adequada à aplicação (esta escolha terá impacto na quantidade de palavras trocadas com o mestre da rede). Exatamente esta mesma quantidade de palavras deverá ser ajustada no mestre da rede. Por fim, programe um valor diferente de 0 para os parâmetros P0711 a P0718.
 - Valores válidos: 0 a 1199.
- Desligue e ligue novamente o CFW700 para que as mudanças tenham efeito.
- Conecte o cabo de rede no módulo.
- Registre o arquivo de configuração (arquivo EDS) no software de configuração da rede.
- Adicione o CFW700 na *scan list* do mestre.
- No *software* de configuração da rede escolha um método para troca de dados com mestre, ou seja, *polled*, *change of state* ou *cyclic*. O módulo DeviceNet do CFW700 suporta todos estes tipos de dados de I/O, além do *explicit* (dados acíclicos).
- Se tudo estiver corretamente configurado, o parâmetro P0719 indicará o estado '*Online, Conectado*'. Observe também o parâmetro que indica o estado do mestre da rede, P0720. Somente haverá troca efetiva de dados quando o estado do mestre for *Run*.

2.7 ACESSO AOS PARÂMETROS

Após o registro do arquivo EDS no software de configuração de rede, o usuário terá acesso à listagem completa dos parâmetros do equipamento os quais podem ser acessados via *explicit messages*. Isto significa que é possível fazer a parametrização e a configuração do drive através do software de configuração de rede.

Para detalhes de utilização deste recurso, consulte a documentação do software de programação do mestre da rede (PLC, PC, etc.).

3 INSTALAÇÃO DA REDE DEVICENET

A rede DeviceNet, como várias redes de comunicação industriais, pelo fato de ser aplicada muitas vezes em ambientes agressivos e com alta exposição à interferência eletromagnética, exige certos cuidados que devem ser tomados para garantir uma baixa taxa de erros de comunicação durante a sua operação. A seguir são apresentadas recomendações para realizar a instalação do produto na rede.

3.1 TAXA DE COMUNICAÇÃO

Equipamentos com interface DeviceNet em geral permitem configurar a taxa de comunicação desejada, podendo variar de 125Kbit/s até 500Kbit/s. A taxa de comunicação (*baud rate*) que pode ser utilizada por um equipamento também depende do comprimento do cabo utilizado na instalação. A tabela a seguir apresenta a relação entre as taxas de comunicação e o comprimento máximo de cabo que pode ser utilizado na instalação, de acordo com o recomendado pela ODVA.

Tabela 3.1: Taxas de comunicação suportadas e comprimento do cabo

Taxa de comunicação	Comprimento do cabo
500 Kbit/s	100 m
250 Kbit/s	250 m
125 Kbit/s	500 m

Todos os equipamentos da rede devem ser programados para utilizar a mesma taxa de comunicação.

Para o inversor de frequência CFW700, a taxa de comunicação é programada através do parâmetro P0702.

3.2 ENDEREÇO NA REDE DEVICENET

Todo dispositivo na rede DeviceNet deve possuir um endereço, ou MAC ID, entre 0 e 63. Este endereço precisa ser diferente para cada equipamento.

Para o inversor de frequência CFW700, o endereço do equipamento é programado através do parâmetro P0701.

3.3 RESISTORES DE TERMINAÇÃO

A utilização de resistores de terminação nas extremidades do barramento CAN é fundamental para evitar reflexão de linha, que pode prejudicar o sinal transmitido e ocasionar erros na comunicação. Resistores de terminação no valor de 120Ω / 0.25W devem ser conectados entre os sinais CAN_H e CAN_L nas extremidades do barramento principal.

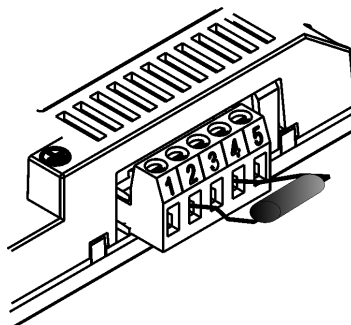


Figura 3.1: Exemplo de instalação do resistor de terminação

3.4 CABO

Para a ligação dos sinais CAN_L e CAN_H deve-se utilizar par trançado com blindagem. A tabela a seguir apresenta as características recomendadas para o cabo.

Tabela 3.2: Características do cabo para rede DeviceNet

Comprimento do cabo (m)	Resistência por metro (mOhm/m)	Área do condutor (mm ²)
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

A utilização de um par trançado adicional para levar a alimentação de 24Vcc para os equipamentos que necessitam deste sinal também é recomendada.

3.5 LIGAÇÃO NA REDE

Para interligar os diversos nós da rede, recomenda-se a conexão do equipamento diretamente a partir da linha principal, sem a utilização de derivações. Durante a instalação dos cabos, deve-se evitar sua passagem próxima a cabos de potência, pois isto facilita a ocorrência de erros durante a transmissão devido à interferência eletromagnética. Para evitar problemas de circulação de corrente por diferença de potencial entre diferentes aterramentos, é necessário que todos os dispositivos estejam conectados no mesmo ponto de terra.

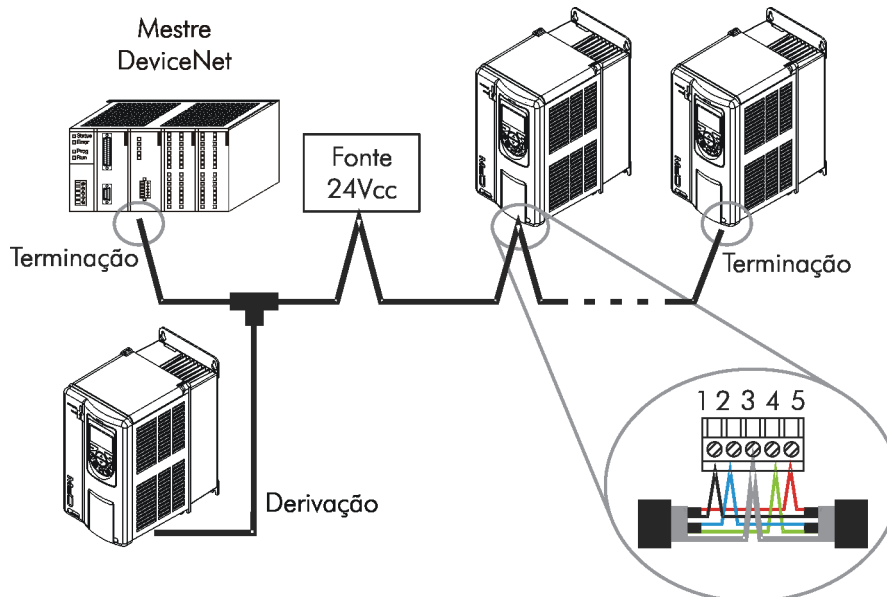


Figura 3.2: Exemplo de instalação em rede DeviceNet

Para evitar problemas de diferença de tensão na alimentação entre os dispositivos da rede, é recomendado que a rede seja alimentada em apenas um ponto, e o sinal de alimentação seja levado a todos os dispositivos através do cabo. Caso seja necessária mais de uma fonte de alimentação, estas devem estar referenciadas ao mesmo ponto.

O número máximo de dispositivos conectados em um único segmento da rede é limitado em 64. Repetidores podem ser utilizados para conectar um número maior de dispositivos.

4 PARAMETRIZAÇÃO

A seguir serão apresentados apenas os parâmetros do inversor de frequência CFW700 que possuem relação direta com a comunicação DeviceNet.

4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES

RO	Parâmetro somente de leitura
CFG	Parâmetro somente alterado com o motor parado
CAN	Parâmetro visível através da HMI se o produto possuir interface CAN instalada

P0105 – SELEÇÃO 1ª/2ª RAMPA

P0220 – SELEÇÃO FONTE LOCAL/REMOTO

P0221 – SELEÇÃO REFERÊNCIA LOCAL

P0222 – SELEÇÃO REFERÊNCIA REMOTA

P0223 – SELEÇÃO GIRO LOCAL

P0224 – SELEÇÃO GIRA/PARA LOCAL

P0225 – SELEÇÃO JOG LOCAL

P0226 – SELEÇÃO GIRO REMOTO

P0227 – SELEÇÃO GIRA/PARA REMOTO

P0228 – SELEÇÃO JOG REMOTO

Estes parâmetros são utilizados na configuração da fonte de comandos para os modos local e remoto do produto. Para que o equipamento seja controlado através da interface DeviceNet, deve-se selecionar uma das opções 'CANopen/DeviceNet/Profibus DP' disponíveis nos parâmetros.

A descrição detalhada destes parâmetros encontra-se no manual de programação do inversor de frequência CFW700.

P0313 – AÇÃO PARA ERRO DE COMUNICAÇÃO

Faixa de Valores:	0 = Inativo 1 = Para por Rampa 2 = Desabilita Geral 3 = Vai para Local 4 = Vai para Local e mantém comandos e referência 5 = Causa Falha	Padrão: 0
Propriedades:	CFG	
Grupo de acesso via HMI:	NET	

Descrição:

Este parâmetro permite selecionar qual a ação deve ser executada pelo inversor, caso ele seja controlado via rede e um erro de comunicação seja detectado.

Opções	Descrição
0 = Inativo	Nenhuma ação é tomada, inversor permanece no estado atual.
1 = Para por Rampa	O comando de parada por rampa é executado, e o motor para de acordo com a rampa de desaceleração programada.
2 = Desabilita Geral	O inversor é desabilitado geral, e o motor para por inércia.
3 = Vai para Local	O inversor é comandado para o modo local.
4 = Vai para Local e mantém comandos e referência	O inversor é comandado para o modo local, mas os comandos de habilitação e a referência de velocidade recebidos via rede são mantidos em modo local, desde que o inversor seja programado para utilizar, em modo local, comandos e referência de velocidade via HMI.
5 = Causa Falha	No lugar de alarme, um erro de comunicação causa uma falha no inversor, sendo necessário fazer o reset de falhas do inversor para o retorno da sua operação normal.

São considerados erros de comunicação os seguintes eventos:

Comunicação Serial (RS232/RS485):

- Alarme A128/Falha F228: *timeout* da interface serial.

Comunicação CANopen/DeviceNet

- Alarme A133/Falha F233: sem alimentação na interface CAN.
- Alarme A134/Falha F234: *bus off*.
- Alarme A135/Falha F235: erro de comunicação CANopen (*Node Guarding/Heartbeat*).
- Alarme A136/Falha F233: mestre da rede DeviceNet em modo *Idle*.
- Alarme A137/Falha F233: ocorreu *timeout* em uma ou mais conexões I/O DeviceNet.

Comunicação Profibus DP:

- Alarme A138/Falha F238: inversor recebeu comando para operar em modo *clear*.
- Alarme A139/Falha F239: inversor foi para o estado *offline*.
- Alarme A140/Falha F240: erro de acesso à interface Profibus.

As ações descritas neste parâmetro são executadas através da escrita automática dos respectivos bits no parâmetro de controle da interface de rede que corresponde à falha detectada. Desta forma, para que os comandos escritos neste parâmetro tenham efeito, é necessário que o inversor esteja programado para ser controlado via a interface de rede utilizada. Esta programação é feita através dos parâmetros P0220 até P0228.

P0680 – ESTADO LÓGICO

Faixa de Valores: 0000h a FFFFh

Padrão: -

Propriedades: RO

Grupo de acesso via HMI: NET

Descrição:

Permite a monitoração do estado do drive. Cada bit representa um estado:

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3 a 0
Função	Em Falha	Reservado	Subtensão	LOC/REM	JOG	Sentido de Giro	Habilitado Geral	Motor Girando	Em Alarme	Em modo de configuração	Segunda Rampa	Parada Rápida Ativa	Reservado

Tabela 4.1 – Funções dos bits para o parâmetro P0680

Bits	Valores
Bits 0 a 3	Reservado.
Bit 4 Parada Rápida Ativa	0: Inversor não possui comando de parada rápida ativo. 1: Inversor está executando o comando de parada rápida.
Bit 5 Segunda Rampa	0: Inversor está configurado para utilizar como rampa de aceleração e desaceleração para o motor a primeira rampa, programada nos parâmetros P0100 e P0101. 1: Inversor está configurado para utilizar como rampa de aceleração e desaceleração para o motor a segunda rampa, programada nos parâmetros P0102 e P0103.
Bit 6 Em Modo de Configuração	0: Inversor operando normalmente. 1: Inversor em modo de configuração. Indica uma condição especial na qual o inversor não pode ser habilitado: <ul style="list-style-type: none"> ▪ Executando rotina de auto-ajuste. ▪ Executando rotina de <i>start-up</i> orientado. ▪ Executando função <i>copy</i> da HMI. ▪ Executando rotina auto-guiada do cartão de memória flash. ▪ Possui incompatibilidade de parametrização. ▪ Sem alimentação no circuito de potência do inversor. Obs.: É possível obter a descrição exata do modo especial de operação no parâmetro P0692.
Bit 7 Em Alarme	0: Inversor não está no estado de alarme. 1: Inversor está no estado de alarme. Obs.: o número do alarme pode ser lido através do parâmetro P0048 – Alarme Atual.
Bit 8 Motor Girando (RUN)	0: Motor está parado. 1: Inversor está acionando o motor na velocidade de referência, ou executando rampa de aceleração ou desaceleração.
Bit 9 Habilitado Geral	0: Inversor está desabilitado geral. 1: Inversor está habilitado geral e pronto para girar o motor.
Bit 10 Sentido de Giro	0: Motor girando com velocidade negativa. 1: Motor girando com velocidade positiva.
Bit 11 JOG	0: Função JOG inativa. 1: Função JOG ativa.
Bit 12 LOC/REM	0: Inversor em modo local. 1: Inversor em modo remoto.
Bit 13 Subtensão	0: Sem subtensão. 1: Com subtensão.
Bit 14	Reservado.
Bit 15 Em Falha	0: Inversor não está no estado de falha. 1: Inversor está no estado de falha. Obs.: O número da falha pode ser lido através do parâmetro P0049 – Falha Atual.

P0681 – VELOCIDADE DO MOTOR EM 13 BITS
Faixa de - 32768 a 32767

Padrão: -

Valores:
Propriedades: RO

Grupo de acesso via HMI: NET

Descrição:

Permite monitorar a velocidade do motor. Esta palavra utiliza resolução de 13 bits com sinal para representar a rotação síncrona do motor:

- P0681 = 0000h (0 decimal) → velocidade do motor = 0 rpm
- P0681 = 2000h (8192 decimal) → velocidade do motor = rotação síncrona

Valores de velocidade em rpm intermediários ou superiores podem ser obtidos utilizando esta escala. Por exemplo, para um motor de 4 polos e 1800 rpm de rotação síncrona, caso o valor lido seja 2048 (0800h), para obter o valor em rpm deve-se calcular:

8192 => 1800 rpm
2048 => Velocidade em rpm

Velocidade em rpm = $\frac{1800 \times 2048}{8192}$

Velocidade em rpm = 450 rpm

Valores negativos para este parâmetro indicam motor girando no sentido reverso de rotação.

P0684 – PALAVRA DE CONTROLE VIA DEVICENET

Faixa de Valores:	0000h a FFFFh	Padrão: 0000h
Propriedades:	-	
Grupo de acesso via HMI:	NET	

Descrição:

Palavra de comando do inversor via interface DeviceNet. Este parâmetro somente pode ser alterado via interface CANopen/DeviceNet/Profibus DP. Para as demais fontes (HMI, etc.) ele se comporta como um parâmetro somente de leitura.

Para que os comandos escritos neste parâmetro sejam executados, é necessário que o inversor esteja programado para ser controlado via CANopen/DeviceNet/Profibus DP. Esta programação é feita através dos parâmetros P0105 e P0220 até P0228.

Cada bit desta palavra representa um comando que pode ser executado no produto.

Bits	15 a 8	7	6	5	4	3	2	1	0
Função	Reservado	Reset de Falhas	Parada Rápida	Utiliza Segunda Rampa	LOC/REM	JOG	Sentido de Giro	Habilita Geral	Gira/Para

Tabela 4.2 – Funções dos bits para o parâmetro P0684

Bits	Valores
Bit 0 Gira/Para	0: Para motor por rampa de desaceleração. 1: Gira motor de acordo com a rampa de aceleração até atingir o valor da referência de velocidade.
Bit 1 Habilita Geral	0: Desabilita geral o inversor, interrompendo a alimentação para o motor. 1: Habilita geral o inversor, permitindo a operação do motor.
Bit 2 Sentido de Giro	0: Sentido de giro do motor oposto ao da referência (sentido reverso). 1: Sentido de giro do motor igual ao da referência (sentido direto).
Bit 3 JOG	0: Desabilita a função JOG. 1: Habilita a função JOG.
Bit 4 LOC/REM	0: Inversor vai para o modo local. 1: Inversor vai para o modo remoto.
Bit 5 Utiliza Segunda Rampa	0: Inversor utiliza como rampa de aceleração e desaceleração do motor os tempos da primeira rampa, programada nos parâmetros P0100 e P0101. 1: Inversor utiliza como rampa de aceleração e desaceleração do motor os tempos da segunda rampa, programada nos parâmetros P0102 e P0103.
Bit 6 Parada Rápida	0: Não executa comando de parada rápida. 1: Executa comando de parada rápida. Obs.: quando o tipo de controle (P0202) for V/F ou VVV não se recomenda a utilização desta função.
Bit 7 Reset de Falhas	0: Sem função. 1: Se em estado de falha, executa o reset do inversor.
Bits 8 a 15	Reservado.

P0685 – REFERÊNCIA DE VELOCIDADE VIA DEVICENET

Faixa de Valores:	-32768 a 32767	Padrão: 0
Propriedades:	-	
Grupo de acesso via HMI:	NET	

Descrição:

Permite programar a referência de velocidade para o motor via interface DeviceNet. Este parâmetro somente pode ser alterado via CANopen/DeviceNet/Profibus DP. Para as demais fontes (HMI, etc.) ele se comporta como um parâmetro somente de leitura.

Para que a referência escrita neste parâmetro seja utilizada, é necessário que o produto esteja programado para utilizar a referência de velocidade via CANopen/DeviceNet/Profibus DP. Esta programação é feita através dos parâmetros P0221 e P0222.

Esta palavra utiliza resolução de 13 bits com sinal para representar a rotação síncrona do motor:

- P0685 = 0000h (0 decimal) → referência de velocidade = 0 rpm
- P0685 = 2000h (8192 decimal) → referência de velocidade = rotação síncrona

Valores de referência intermediários ou superiores podem ser programados utilizando esta escala. Por exemplo, para um motor de 4 polos e 1800 rpm de rotação síncrona, caso deseje-se uma referência de 900 rpm, deve-se calcular:

1800 rpm => 8192 900 rpm => Referência em 13 bits
Referência em 13 bits = $\frac{900 \times 8192}{1800}$
Referência em 13 bits = 4096 => Valor correspondente a 900 rpm na escala em 13 bits

Este parâmetro também aceita valores negativos para inverter o sentido de rotação do motor. O sentido de rotação da referência, no entanto, depende também do valor do bit 2 da palavra de controle – P0684:

- Bit 2 = 1 e P0685 > 0: referência para o sentido direto
- Bit 2 = 1 e P0685 < 0: referência para o sentido reverso
- Bit 2 = 0 e P0685 > 0: referência para o sentido reverso
- Bit 2 = 0 e P0685 < 0: referência para o sentido direto

P0695 – VALOR PARA AS SAÍDAS DIGITAIS

Faixa de 0000h a 001Fh **Padrão:** 0000h
Valores:
Propriedades: Net
Grupo de acesso via HMI: NET

Descrição:

Possibilita o controle das saídas digitais através das interfaces de rede (Serial, CAN, etc.). Este parâmetro não pode ser alterado através da HMI.

Cada bit deste parâmetro corresponde ao valor desejado para uma saída digital. Para que a saída digital correspondente possa ser controlada de acordo com este conteúdo, é necessário que sua função seja programada para “Conteúdo P0695”, nos parâmetros P0275 a P0279.

Bits	15 a 5	4	3	2	1	0
Função	Reservado	Valor para DO5	Valor para DO4	Valor para DO3	Valor para DO2	Valor para DO1

Tabela 4.3 – Funções dos bits para o parâmetro P0695

Bits	Valores
Bit 0 Valor para DO1	0: saída DO1 aberta. 1: saída DO1 fechada.
Bit 1 Valor para DO2	0: saída DO2 aberta. 1: saída DO2 fechada.
Bit 2 Valor para DO3)	0: saída DO3 aberta. 1: saída DO3 fechada.
Bit 3 Valor para DO4	0: saída DO4 aberta. 1: saída DO4 fechada.
Bit 4 Valor para DO5	0: saída DO5 aberta. 1: saída DO5 fechada.
Bits 5 a 15	Reservado.

P0696 – VALOR 1 PARA SAÍDAS ANALÓGICAS
P0697 – VALOR 2 PARA SAÍDAS ANALÓGICAS

Faixa de Valores:	-32768 a 32767	Padrão: 0
Propriedades:	Net	
Grupo de acesso via HMI:	NET	

Descrição:

Possibilita o controle das saídas analógicas através das interfaces de rede (Serial, CAN, etc.). Estes parâmetros não podem ser alterados através da HMI.

O valor escrito nestes parâmetros é utilizado como valor para a saída analógica, desde que a função da saída analógica desejada seja programada para “Conteúdo P0696 / P0697”, nos parâmetros P0251 ou P0254.

O valor deve ser escrito em uma escala de 15 bits (7FFFh = 32767)⁵ para representar 100% do valor desejado para a saída, ou seja:

- P0696 = 0000h (0 decimal) → valor para a saída analógica = 0 %
- P0696 = 7FFFh (32767 decimal) → valor para a saída analógica = 100 %

Neste exemplo foi mostrado o parâmetro P0696, mas a mesma escala é utilizada para o parâmetro P0697. Por exemplo, deseja-se controlar o valor da saída analógica 1 através da serial. Neste caso deve fazer a seguinte programação:

- Escolher um dos parâmetros P0696 ou P0697 para ser o valor utilizado pela saída analógica 1. Neste exemplo, vamos escolher o P0696.
- Programar, na função da saída analógica 1 (P0254), a opção “Conteúdo P0696”.
- Através da interface de rede, escrever no P0696 o valor desejado para a saída analógica 1, entre 0 e 100 %, de acordo com a escala do parâmetro.


NOTA!

Caso a saída analógica seja programada para operar de -10V até 10V, valores negativos para estes parâmetros devem ser utilizados para comandar as saídas com valores negativos de tensão, ou seja, -32768 até 32767 representa uma variação de -10V até 10V na saída analógica.

⁵ Para a resolução real da saída, consulte o manual do produto.

P0700 – PROTOCOLO CAN

Faixa de	1 = CANopen	Padrão: 2
Valores:	2 = DeviceNet	
Propriedades:	CFG	
Grupo de acesso via HMI:	NET	

Descrição:

Permite seleccionar o protocolo desejado para a interface CAN. Caso este parâmetro seja alterado, a alteração terá efeito somente se a interface CAN estiver sem alimentação, em *auto-baud* ou após o equipamento ser desligado e ligado novamente.

P0701 – ENDEREÇO CAN

Faixa de	0 a 127	Padrão: 63
Valores:		
Propriedades:	CFG	
Grupo de acesso via HMI:	NET	

Descrição:

Permite programar o endereço utilizado para comunicação CAN do dispositivo. É necessário que cada equipamento da rede possua um endereço diferente dos demais. Os endereços válidos para este parâmetro dependem do protocolo programado no P0700:

- P0700 = 1 (CANopen) → endereços válidos: 1 a 127.
- P0700 = 2 (DeviceNet) → endereços válidos: 0 a 63.

Caso este parâmetro seja alterado, a alteração terá efeito somente se a interface CAN estiver sem alimentação, em *autobaud* ou após o equipamento ser desligado e ligado novamente.

P0702 – TAXA DE COMUNICAÇÃO CAN

Faixa de	0 = 1 Mbit/s / <i>Autobaud</i>	Padrão: 0
Valores:	1 = 800 Kbit/s / <i>Autobaud</i>	
	2 = 500 Kbit/s	
	3 = 250 Kbit/s	
	4 = 125 Kbit/s	
	5 = 100 Kbit/s / <i>Autobaud</i>	
	6 = 50 Kbit/s / <i>Autobaud</i>	
	7 = 20 Kbit/s / <i>Autobaud</i>	
	8 = 10 Kbit/s / <i>Autobaud</i>	
Propriedades:	CFG	
Grupo de acesso via HMI:	NET	

Descrição:

Permite programar o valor desejado para a taxa de comunicação da interface CAN, em bits por segundo. Esta taxa deve ser a mesma para todos os equipamentos conectados na rede. As taxas de comunicação suportadas para o dispositivo dependem do protocolo programado no P0700:

- P0700 = 1 (CANopen): pode-se utilizar qualquer taxa indicada neste parâmetro, mas não possui a função de detecção automática da taxa – *autobaud*.
- P0700 = 2 (DeviceNet): somente as taxas de 500, 250 e 125 Kbit/s são suportadas. Demais opções habilitam a função de detecção automática da taxa – *autobaud*.

Caso este parâmetro seja alterado, a alteração terá efeito somente se a interface CAN estiver sem alimentação ou após o equipamento ser desligado e ligado novamente.

Para a função *autobaud*, após uma detecção com sucesso, o parâmetro da taxa de comunicação (P0702) altera-se automaticamente para a taxa detectada. Para executar novamente a função de *autobaud*, é necessário mudar o parâmetro P0702 para uma das opções '*Autobaud*'.

P0703 – RESET DE BUS OFF

Faixa de	0 = Manual	Padrão: 0
Valores:	1 = Automático	
Propriedades:	CFG	
Grupo de acesso via HMI:	NET	

Descrição:

Permite programar qual o comportamento do inversor ao detectar um erro de *bus off* na interface CAN.

Opções	Descrição
0 = Reset Manual	Caso ocorra <i>bus off</i> , será indicado na HMI o alarme A134/F234, a ação programada no parâmetro P0313 será executada e a comunicação será desabilitada. Para que o equipamento volte a se comunicar através da interface CAN, será necessário desligar e ligar novamente o produto.
1 = Reset Automático	Caso ocorra <i>bus off</i> , a comunicação será reiniciada automaticamente e o erro será ignorado. Neste caso, não será feita a indicação de alarme na HMI e o inversor não executará a ação descrita no P0313.

P0705 – ESTADO DO CONTROLADOR CAN

Faixa de	0 = Inativo	Padrão: -
Valores:	1 = <i>Autobaud</i> 2 = Interface CAN ativa 3 = <i>Warning</i> 4 = <i>Error Passive</i> 5 = <i>Bus Off</i> 6 = Sem alimentação	
Propriedades:	RO	
Grupo de acesso via HMI:	NET	

Descrição:

Permite identificar se o cartão de interface CAN está devidamente instalado, e se a comunicação apresenta erros.

Opções	Descrição
0 = Inativo	Interface CAN inativa. Ocorre quando o inversor não possui cartão de interface CAN instalado.
1 = <i>Autobaud</i>	Executando função para detecção automática da taxa de comunicação (apenas para o protocolo DeviceNet).
2 = Interface CAN ativa	Interface CAN ativa e sem erros.
3 = <i>Warning</i>	Controlador CAN atingiu o estado de <i>warning</i> .
4 = <i>Error Passive</i>	Controlador CAN atingiu o estado de <i>error passive</i> .
5 = <i>Bus Off</i>	Controlador CAN atingiu o estado de <i>bus off</i> .
6 = Sem alimentação	Interface CAN não possui alimentação entre os pinos 1 e 5 do conector.

P0706 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	
Grupo de acesso via HMI:	NET	

Descrição:

Este parâmetro funciona como um contador cíclico, que é incrementado toda vez que um telegrama CAN é recebido. Fornece um retorno para o operador se o dispositivo está conseguindo comunicar-se com a rede. Este contador é zerado sempre que o inversor for desligado, feito o reset ou ao atingir o limite máximo do parâmetro.

P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	
Grupo de acesso via HMI:	NET	

Descrição:

Este parâmetro funciona como um contador cíclico, que é incrementado toda vez que um telegrama CAN é transmitido. Fornece um retorno para o operador se o dispositivo está conseguindo comunicar-se com a rede. Este contador é zerado sempre que o inversor for desligado, feito o reset ou ao atingir o limite máximo do parâmetro.

P0708 – CONTADOR DE ERROS DE BUS OFF

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	
Grupo de acesso via HMI:	NET	

Descrição:

Contador cíclico que indica o número vezes que o inversor entrou em estado de *bus off* na rede CAN. Este contador é zerado sempre que o inversor for desligado, feito o reset ou ao atingir o limite máximo do parâmetro.

P0709 – CONTADOR DE MENSAGENS CAN PERDIDAS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	
Grupo de acesso via HMI:	NET	

Descrição:

Contador cíclico que indica o número de mensagens recebidas pela interface CAN, mas que não puderam ser processadas pelo equipamento. Caso o número de mensagens perdidas seja incrementado com frequência, recomenda-se diminuir a taxa de comunicação utilizada para a rede CAN. Este contador é zerado sempre que o inversor for desligado, feito o reset ou ao atingir o limite máximo do parâmetro.

P0710 – INSTÂNCIAS DE I/O DEVICENET

Faixa de Valores:	0 = ODVA Basic Speed (2 palavras) 1 = ODVA Extended Speed (2 palavras) 2 = Especific.Fab 2W (2 palavras) 3 = Especific.Fab 3W (3 palavras) 4 = Especific.Fab 4W (4 palavras) 5 = Especific.Fab 5W (5 palavras) 6 = Especific.Fab 6W (6 palavras)	Padrão: 0
Propriedades:	FG, CAN	
Grupos de acesso via HMI:	NET	

Descrição:

Permite selecionar qual a instância da classe *Assembly* utilizada durante a troca de dados de I/O com o mestre da rede.

O inversor de frequência CFW700 possui sete opções de ajustes. Duas delas seguem o padrão definido no perfil *AC/DC Drive Profile* da ODVA. As outras cinco, representam palavras específicas do inversor de frequência CFW700. As tabelas apresentadas a seguir detalham cada uma destas palavras de controle e monitoramento.


NOTA!

Caso este parâmetro seja alterado, ele somente será válido após o produto ser desligado e ligado novamente.

0 = Formato dos dados para as instâncias ODVA Basic Speed (2 palavras):

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 abaixo.

Monitoramento (Entrada)

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 (low byte)							
	3	Speed Actual (high byte)							

Controle (Saída)

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 (low byte)							
	3	Speed Reference (high byte)							

1 = Formato dos dados para as instâncias ODVA Extended Speed (2 palavras):

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 abaixo.

Monitoramento (Entrada)

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	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Controle (Saída)

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 (low byte)							
	3	Speed Reference (high byte)							

A tabela a seguir apresenta o significado dos dados para as instâncias 20/70 e 21/71.

Monitoramento:

Bits (Byte 0)	Valores
Bit 0 Faulted	0: Inversor não está em estado de falha. 1: Alguma falha registrada pelo inversor. Obs.: O número da falha pode ser lido através do parâmetro P0049 – Falha Atual.
Bit 1 Warning	0: Inversor não está em estado de alarme. 1: Algum alarme registrado pelo inversor. Obs.: O número do alarme pode ser lido através do parâmetro P0048 – Alarme Atual.
Bit 2 Running1 (Fwd)	0: Motor não está girando no sentido horário. 1: Motor girando no sentido horário.
Bit 3 Running2 (Rev)	0: Motor não está girando no sentido anti-horário. 1: Motor girando no sentido anti-horário.
Bit 4 Ready	0: Inversor não está pronto para operar. 1: Inversor pronto para operar (estados <i>Ready</i> , <i>Enabled</i> ou <i>Stopping</i>).
Bit 5 Ctrl from Net	0: Drive controlado localmente. 1: Drive controlado remotamente.
Bit 6 Ref. from Net	0: Referência de velocidade não está sendo enviada via rede DeviceNet. 1: Indica referência de velocidade sendo enviada via rede DeviceNet.
Bit 7 At Reference	0: Inversor ainda não atingiu velocidade programada. 1: Inversor atingiu velocidade programada.

- Byte 1 indica o estado do drive:
 - 0 = Non-Existent
 - 1 = Startup
 - 2 = Not_Ready
 - 3 = Ready
 - 4 = Enabled
 - 5 = Stopping
 - 6 = Fault_Stop
 - 7 = Faulted
- Bytes 2 (*low*) e 3 (*high*) representam a velocidade real do motor em RPM.

Controle:

Bits (Byte 0)	Valores
Bit 0 Run Fwd	0: Para motor. 1: Gira motor no sentido horário.
Bit 1 Run Rev	0: Para motor. 1: Gira motor no sentido anti-horário.
Bit 2 Fault Reset ⁶	0: Sem função. 1: Se em estado de falha, executa o reset do inversor.
Bits 3 e 4	Reservado.
Bit 5 NetCtrl ⁷	0: Seleciona o modo local. 1: Seleciona o modo remoto.
Bit 6 NetRef	0: Referência de velocidade não está sendo enviada via rede. 1: Envio da referência de velocidade seja feito via rede.
Bit 7	Reservado.

- Bytes 2 (*low*) e 3 (*high*) representam a referência de velocidade do motor em RPM.

2 = Formato dos dados para as instâncias *Manufacturer Specific 2W* (2 palavras):

3 = Formato dos dados para as instâncias *Manufacturer Specific 3W* (3 palavras):

4 = Formato dos dados para as instâncias *Manufacturer Specific 4W* (4 palavras):

5 = Formato dos dados para as instâncias *Manufacturer Specific 5W* (5 palavras):

6 = Formato dos dados para as instâncias *Manufacturer Specific 6W* (6 palavras):

Chamada de *Manufacturer Specific*, estas instâncias representam a mais simples interface de operação do equipamento segundo o perfil do inversor de frequência CFW700. O mapeamento dos dados é mostrado abaixo. Além das palavras de controle e estado, referência e valor atual da velocidade, é possível programar até 4 parâmetros do próprio equipamento para leitura e/ou escrita via rede, através dos parâmetros P0711 até P0718.

Monitoramento (Entrada)

Programável	Fixo	Instância	Palavras de 16 bits (word)	Função
		150	#1	Palavra de Estado (P0680)
			#2	Velocidade Atual (P0681)
			#3	Leitura #3 DeviceNet
			#4	Leitura #4 DeviceNet
			#5	Leitura #5 DeviceNet
			#6	Leitura #6 DeviceNet

Opção do P0710: 2, 3, 4, 5, 6

Controle (Saída)

Programável	Fixo	Instância	Palavras de 16 bits (word)	Função
		100	#1	Palavra de Controle (P0684)
			#2	Referência de Velocidade (P0685)
			#3	Escrita #3 DeviceNet
			#4	Escrita #4 DeviceNet
			#5	Escrita #5 DeviceNet
			#6	Escrita #6 DeviceNet

Opção do P0710: 2, 3, 4, 5, 6

⁶ Após um reset o drive irá para o estado offline.

⁷ Para que esta ação seja executada, os parâmetros P220 – P228 devem estar corretamente programados.

P0711 – LEITURA #3 DEVICENET
P0712 – LEITURA #4 DEVICENET
P0713 – LEITURA #5 DEVICENET
P0714 – LEITURA #6 DEVICENET

Faixa de	0 a 1199	Padrão: 0 (desabilitado)
Valores:		
Propriedades:	CFG, CAN	
Grupos de acesso via HMI:	NET	

Descrição:

Estes parâmetros permitem programar o conteúdo das palavras 3 a 6 de entrada (input: escravo envia para o mestre). Utilizando estes parâmetros, é possível programar o número de outro parâmetro cujo conteúdo deve ser disponibilizado na área de entrada do mestre da rede.

Por exemplo, caso se deseje ler do drive a corrente do motor em amperes, deve-se programar em algum dos parâmetros o valor 3, pois o parâmetro P0003 é o parâmetro que contém esta informação. Vale lembrar que o valor lido de qualquer parâmetro é representado com uma palavra de 16 bits. Mesmo que o parâmetro possua resolução decimal, o valor é transmitido sem a indicação das casas decimais. Por exemplo, se o parâmetro P0003 possuir o valor 4.7 A, o valor fornecido via rede será 47

Estes parâmetros são utilizados somente se o drive for programado no parâmetro P0710 para utilizar as opções 3 até 6. De acordo com a opção selecionada, são disponibilizadas até 6 palavras para leitura pelo mestre da rede.

As duas primeiras palavras de entrada são fixas.


NOTA!

O valor 0 (zero) desabilita a escrita na palavra. A quantidade de palavras de entrada, porém, permanece sempre igual ao que foi programado no parâmetro P0710.

P0715 – ESCRITA #3 DEVICENET
P0716 – ESCRITA #4 DEVICENET
P0717 – ESCRITA #5 DEVICENET
P0718 – ESCRITA #6 DEVICENET

Faixa de	0 a 1199	Padrão: 0 (desabilitado)
Valores:		
Propriedades:	CFG, CAN	
Grupos de acesso via HMI:	NET	

Descrição:

Estes parâmetros permitem programar o conteúdo das palavras 3 a 6 de saída (output: mestre envia para o escravo). Utilizando estes parâmetros, é possível programar o número de outro parâmetro cujo conteúdo deve ser disponibilizado na área de saída do mestre da rede.

Por exemplo, caso se deseje escrever no drive a rampa de aceleração, deve-se programar em algum dos parâmetros o valor 100, pois o parâmetro P0100 é o parâmetro onde esta informação é programada. Vale lembrar que o valor escrito de qualquer parâmetro é representado com uma palavra de 16 bits. Mesmo que o parâmetro possua resolução decimal, o valor é transmitido sem a indicação das casas decimais. Por exemplo, caso deseje-se programar o parâmetro P0100 com o valor 5,0s, o valor escrito via rede deverá ser 50.

Estes parâmetros são utilizados somente se o drive for programado no parâmetro P0710 para utilizar as opções 3 até 6. De acordo com a opção selecionada, são disponibilizadas até 6 palavras para escrita pelo mestre da rede.

As duas primeiras palavras de saída são fixas.


NOTA!

O valor 0 (zero) desabilita a escrita na palavra. A quantidade de palavras de saída, porém, permanece sempre igual ao que foi programado no parâmetro P0710.

P0719 – ESTADO DA REDE DEVICENET

Faixa de Valores:	0 = Offline 1 = Online, Não Conectado 2 = Online, Conectado 3 = Conexão expirou 4 = Falha na Conexão 5 = Auto-baud	Padrão: -
Propriedades:	RO, CAN	
Grupos de acesso via HMI:	NET	

Descrição:

Indica o estado da rede DeviceNet. A tabela a seguir apresenta uma breve descrição destes estados.

Estado	Descrição
<i>Offline</i>	Sem alimentação ou não <i>online</i> . Comunicação não pode ser estabelecida.
<i>Online, Não Conectado</i>	Dispositivo <i>online</i> , mas não conectado. Escravo completou com sucesso o procedimento de verificação do MacID. Isto significa que a taxa de comunicação configurada está correta (ou foi detectada corretamente no caso da utilização do autobaud) e que não há outros nodos na rede com o mesmo endereço. Porém, neste estágio, ainda não há comunicação com o mestre..
<i>Online, Conectado</i>	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.
Conexão Expirou	Uma ou mais conexões do tipo I/O expiraram.
Falha na Conexão	Indica que o escravo não pode entrar na rede devido a problemas de endereçamento ou então devido à ocorrência de <i>bus off</i> . Verifique se o endereço configurado já não está sendo utilizado por outro equipamento, se a taxa de comunicação escolhida está correta ou se existem problemas na instalação.
Autobaud	Equipamento executando rotina do mecanismo de autobaud.

P0720 – ESTADO DO MESTRE DEVICENET

Faixa de Valores:	0 = Run 1 = Idle	Padrão: -
Propriedades:	RO, CAN	
Grupos de acesso via HMI:	NET	

Descrição:

Indica o estado do mestre da rede DeviceNet. Este pode estar em modo de operação (*Run*) ou modo de configuração (*Idle*).

Quando em *Run*, telegramas de leitura e escrita são processados e atualizados normalmente pelo mestre. Quando em *Idle*, apenas telegramas de leitura dos escravos são atualizados pelo mestre. A escrita, neste caso, fica desabilitada.

Quando a comunicação esta desabilitada este parâmetro não representa o estado real do mestre.

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 CFW700 suporta todas as classes obrigatórias do perfil AC/DC Device Profile. Suporta também classes específicas do fabricante. Detalhes de cada uma delas são apresentados nas seções a seguir.

5.1 CLASSE IDENTITY (01H)

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

Tabela 5.1: Atributos da Instancia da Classe Identity

Atributo	Método	Nome	Padrão	Descrição
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 firmware
5	Get	Status		Estado atual do dispositivo
6	Get	Serial Number		Número serial
7	Get	Product Name	CFW700	Nome do produto

5.2 CLASSE MESSAGE ROUTER (02H)

Fornecer informações sobre o objeto roteador de mensagens do tipo explicit. No CFW700, 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:

Tabela 5.2: Atributos da Classe DeviceNet

Atributo	Método	Nome	Mín./Max	Padrão	Descrição
1	Get	Revision	1-65535		Revisão da definição do Objeto de Classe DeviceNet sobre qual a implementação foi baseada.

Tabela 5.3: Atributos da Instancia da Classe DeviceNet

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

5.4 CLASSE ASSEMBLY (04H)

Classe cuja função é agrupar diversos atributos numa única conexão. No CFW700 apenas o atributo Data (3) está implementado (Tabela 5.4).

Tabela 5.4: Atributos das instâncias da classe Assembly

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

No CFW700, a classe Assembly contém as seguintes instâncias:

Tabela 5.5: Instâncias da classe Assembly

Instâncias	Tamanho	Descrição
20	2 palavras	ODVA Basic Speed Control Output
70	2 palavras	ODVA Basic Speed Control Input
21	2 palavras	ODVA Extended Speed Control Output
71	2 palavras	ODVA Extended Speed Control Input
100	2 – 6 palavras	Específica do Fabricante Saída
150	2 – 6 palavras	Específica do Fabricante Entrada

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

Tabela 5.6: Classe Connection – Instância 1: Explicit Message

Atributo	Método	Nome	Descrição
1	Get	State	Estado do objeto
2	Get	Instance Type	I/O ou explicit
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 bytes desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em bytes 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 Inactivity/Watchdog
13	Get	Produced Connection Path Length	Número de bytes da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de bytes 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

5.5.2 Instância 2: Polled

Tabela 5.7: Classe Connection – Instância 2: Polled

Atributo	Método	Nome	Descrição
1	Get	State	Estado do objeto
2	Get	Instance Type	I/O ou explicit
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 bytes desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em bytes 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 Inactivity/Watchdog
13	Get	Produced Connection Path Length	Número de bytes da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de bytes 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

5.5.3 Instância 4: Change of State/Cyclic
Tabela 5.8: Classe Connection – Instância 4: Change of State/Cyclic

Atributo	Método	Nome	Descrição
1	Get	State	Estado do objeto
2	Get	Instance Type	I/O ou explicit
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 bytes desta conexão de transmissão
8	Get	Consumed Connection Size	Tamanho em bytes 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 Inactivity/Watchdog
13	Get	Produced Connection Path Length	Número de bytes da conexão produtora
14	Get	Produced Connection Path	Caminho dos objetos produtores de dados
15	Get	Consumed Connection Path Length	Número de bytes 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

5.6 CLASSE MOTOR DATA (28H)

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

Tabela 5.9: Atributos da Classe Motor Data

Atributo	Método	Nome	Mín./Max	Descrição
1	Get	Revision	1-65535	Revisão da definição do Objeto de Classe Motor Data sobre qual a implementação foi baseada.
2	Get	Max Instance		Número máximo de instancias

Tabela 5.10: Atributos da Instancia da Classe Motor Data

Atributo	Método	Nome	Mín./Max	Unidade	Padrão	Descrição
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		Tensão nominal

5.7 CLASSE CONTROL SUPERVISOR (29H)

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

Tabela 5.11: Atributos da Classe Control Supervisor

Atributo	Método	Nome	Mín./Max	Descrição
1	Get	Revision	1-65535	Revisão da definição do Objeto de Classe Control Supervisor sobre qual a implementação foi baseada.
2	Get	Max Instance		Número máximo de instancias

Tabela 5.12: Atributos da Instancia da Classe Control Supervisor

Atributo	Método	Nome	Mín./Max	Padrão	Descrição
3	Get/Set	Run1	0-1		Run Fwd
4	Get/Set	Run2	0-1		Run Rev
5	Get/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	Get/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

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:

Tabela 5.13: Atributos da Classe AC/DC Drive

Atributo	Método	Nome	Mín./Max	Descrição
1	Get	Revision	1-65535	Revisão da definição do Objeto de Classe AC/DC Drive sobre qual a implementação foi baseada.
2	Get	Max Instance		Número máximo de instancias

Tabela 5.14: Atributos da Instancia da Classe AC/DC Drive

Atributo	Método	Nome	Mín./Max	Padrão	Descrição
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


NOTA!

O CFW700 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).

Tabela 5.15: Atributos da Instancia da Classe Acknowledge Handler

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

5.10 CLASSES ESPECÍFICAS DO FABRICANTE

As classes específicas do fabricante são utilizadas para mapear todos os parâmetros do CFW700. 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 5.16:

Tabela 5.16: Classes Específicas do Fabricante

Classe	Nome	Faixa de valores
Classe 100 (64h)	VENDOR CLASS F1	Parâmetros 000-099
Classe 101 (65h)	VENDOR CLASS F2	Parâmetros 100-199
Classe 102 (66h)	VENDOR CLASS F3	Parâmetros 200-299
Classe 103 (67h)	VENDOR CLASS F4	Parâmetros 300-399
Classe 104 (68h)	VENDOR CLASS F5	Parâmetros 400-499
Classe 105 (69h)	VENDOR CLASS F6	Parâmetros 500-599
Classe 106 (6Ah)	VENDOR CLASS F7	Parâmetros 600-699
Classe 107 (6Bh)	VENDOR CLASS F8	Parâmetros 700-799
Classe 108 (6Ch)	VENDOR CLASS F9	Parâmetros 800-899
Classe 109 (6Dh)	VENDOR CLASS F10	Parâmetros 900-999
Classe 110 (6Eh)	VENDOR CLASS F11	Parâmetros 1000-1099

Tabela 5.17: Parâmetros das classes específicas do fabricante

Parâmetro	Classe	Instância	Atributo
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
...



NOTA!

- Para estas classes, o CFW700 utiliza apenas a instância 1.
- Para estas classes, 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 P0023.

Parâmetro 100: classe 65h, instância 1, atributo 100. Este caminho dá acesso ao P0100.

Parâmetro 202: classe 66h, instância 1, atributo 102. Este caminho dá acesso ao P0202.

6 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO DEVICENET

A133/F233 – SEM ALIMENTAÇÃO NA INTERFACE CAN

Descrição:

Indica que a interface CAN não possui alimentação entre os pinos 1 e 5 do conector.

Atuação:

Para que seja possível enviar e receber telegramas através da interface CAN, é necessário fornecer alimentação externa para o circuito de interface.

Se a interface CAN estiver alimentada e for detectada a falta de alimentação na interface CAN, será sinalizada através da HMI a mensagem de alarme A133 – ou falha F233, dependendo da programação feita no P0313. Se a alimentação do circuito for restabelecida, a comunicação CAN será reiniciada. Para alarmes, a indicação do alarme também será retirada da HMI caso a alimentação seja restabelecida.

Possíveis Causas/Correção:

- Medir se existe tensão dentro da faixa permitida entre os pinos 1 e 5 do conector da interface CAN.
- Verificar se os cabos de alimentação não estão trocados ou invertidos.
- Verificar problemas de contato no cabo ou no conector da interface CAN.

A134/F234 – BUS OFF

Descrição:

Detectado erro de *bus off* na interface CAN.

Atuação:

Caso o número de erros de recepção ou transmissão detectados pela interface CAN seja muito elevado⁸, o controlador CAN pode ser levado ao estado de *bus off*, onde ele interrompe a comunicação e desabilita a interface CAN.

Neste caso será sinalizada através da HMI a mensagem de alarme A134 – ou falha F234, dependendo da programação feita no P0313. Para que a comunicação seja restabelecida, é necessário desligar e ligar novamente o produto, ou retirar e ligar novamente a alimentação da interface CAN, para que a comunicação seja reiniciada.

Possíveis Causas/Correção:

- Verificar curto-circuito nos cabos de transmissão do circuito CAN.
- Verificar se os cabos não estão trocados ou invertidos.
- Verificar se todos os dispositivos da rede utilizam a mesma taxa de comunicação.
- Verificar se resistores de terminação com valores corretos foram colocados somente nos extremos do barramento principal.
- Verificar se a instalação da rede CAN foi feita de maneira adequada.

A136/F236 – MESTRE EM IDLE

Descrição:

Alarme que indica que o mestre da rede DeviceNet está em modo *idle*.

Atuação:

Atua quando o CFW700 detectar que o mestre da rede foi para o modo *idle*. Neste modo, apenas as variáveis lidas do escravo continuam sendo atualizadas na memória do mestre. Nenhum dos comandos enviados ao escravo é processado.

Neste caso será sinalizada através da HMI a mensagem de alarme A136 – ou falha F236 dependendo da programação feita no P0313. Para alarmes, caso o mestre seja novamente colocado em modo *Run* (estado normal de operação do equipamento), a indicação de alarme será retirada da HMI.

⁸ Para mais informações sobre detecção de erros, consultar especificação CAN.

Possíveis Causas/Correção:

- Ajuste a chave que comanda o modo de operação do mestre para execução (*Run*) ou então o bit correspondente na palavra de configuração do software do mestre. Em caso de dúvidas, consulte a documentação do mestre em uso.

A137/F237 – TIMEOUT NA CONEXÃO DEVICENET

Descrição:

Alarme que indica que uma ou mais conexões I/O DeviceNet expiraram.

Atuação:

Ocorre quando, por algum motivo, após iniciada a comunicação cíclica do mestre com o produto, esta comunicação é interrompida.

Neste caso será sinalizada através da HMI a mensagem de alarme A137 – ou falha F237 dependendo da programação feita no P0313. Para alarmes, caso a conexão com o mestre seja restabelecida, a indicação de alarme será retirada da HMI.

Possíveis Causas/Correção:

- Verificar o estado do mestre da rede.
- Verificar instalação da rede, cabo rompido ou falha/mal contato nas conexões com a rede.



WEG Equipamentos Elétricos S.A.
Jaraguá do Sul – SC – Brasil
Fone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020
São Paulo – SP – Brasil
Fone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212
automacao@weg.net
www.weg.net