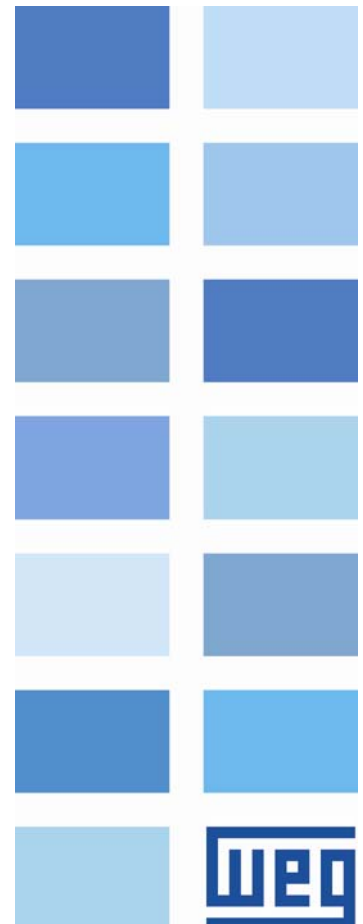


CANopen

CFW-11

Manual da Comunicação

Idioma: Português





Manual da Comunicação CANopen

Série: CFW-11

Idioma: Português

Número do Documento: 0899.5746 / 04

Data de Publicação: 02/2012

Sumário

SOBRE O MANUAL	7
ABREVIações E DEFINIções.....	7
REPRESENTAÇÃO NUMÉRICA.....	7
DOCUMENTOS.....	7
1 INTRODUÇÃO À COMUNICAÇÃO CANOPEN	8
1.1 CAN.....	8
1.1.1 <i>Frame de Dados</i>	8
1.1.2 <i>Frame Remoto</i>	8
1.1.3 <i>Acesso à Rede</i>	8
1.1.4 <i>Controle de Erros</i>	8
1.1.5 <i>CAN e CANopen</i>	9
1.2 CARACTERÍSTICAS DA REDE CANOPEN.....	9
1.3 MEIO FÍSICO.....	9
1.4 ENDEREÇO NA REDE CANOPEN.....	9
1.5 ACESSO AOS DADOS.....	9
1.6 TRANSMISSÃO DE DADOS.....	10
1.7 OBJETOS RESPONSÁVEIS PELA COMUNICAÇÃO – COBS.....	10
1.8 COB-ID.....	11
1.9 ARQUIVO EDS.....	11
2 KITS OPCIONAIS	12
2.1 INTERFACES CAN.....	12
2.1.1 <i>Kit CAN-01</i>	12
2.1.2 <i>Kit CAN/RS485-01</i>	12
2.1.3 <i>Pinagem do Conector</i>	12
2.1.4 <i>Fonte de Alimentação</i>	12
2.1.5 <i>Resistor de Terminação</i>	13
2.1.6 <i>Taxa de Comunicação</i>	13
2.1.7 <i>Ligação do Inversor na Rede CAN</i>	13
3 PARAMETRIZAÇÃO DO INVERSOR	15
3.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES.....	15
P0105 – SELEÇÃO 1º/2º RAMPAS.....	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 CANOPEN/DEVICENET.....	18
P0685 – REFERÊNCIA DE VELOCIDADE VIA CANOPEN/DEVICENET.....	19
P0695 – VALOR PARA AS SAÍDAS DIGITAIS.....	19
P0696 – VALOR 1 PARA SAÍDAS ANALÓGICAS.....	20
P0697 – VALOR 2 PARA SAÍDAS ANALÓGICAS.....	20
P0698 – VALOR 3 PARA SAÍDAS ANALÓGICAS.....	20
P0699 – VALOR 4 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	24
P0721 – ESTADO DA COMUNICAÇÃO CANOPEN	24
P0722 – ESTADO DO NÓ CANOPEN	24
4 DICIONÁRIO DE OBJETOS	26
4.1 ESTRUTURA DO DICIONÁRIO	26
4.2 TIPOS DE DADOS	26
4.2.1 Tipos Básicos	26
4.2.2 Tipos Compostos	27
4.2.3 Tipos Estendidos	28
4.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICAÇÃO	28
4.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DO CFW-11	28
4.5 DEVICE PROFILE – OBJETOS COMUNS PARA DRIVES	29
5 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO	30
5.1 OBJETOS DE IDENTIFICAÇÃO	30
5.1.1 Objeto 1000h – Device Type	30
5.1.2 Objeto 1001h – Error Register	30
5.1.3 Objeto 1018h – Identity Object	31
5.2 SERVICE DATA OBJECTS – SDOs	31
5.2.1 Objeto 1200h – Servidor SDO	32
5.2.2 Funcionamento dos SDOs	32
5.3 PROCESS DATA OBJECTS – PDOs	33
5.3.1 Objetos Mapeáveis para os PDOs	34
5.3.2 PDOs de Recepção	35
5.3.3 PDOs de Transmissão	38
5.4 EMERGENCY OBJECT – EMCY	41
5.5 SYNCHRONIZATION OBJECT – SYNC	42
5.6 NETWORK MANAGEMENT – NMT	42
5.6.1 Controle dos Estados do Escravo	42
5.6.2 Controle de Erros – Node Guarding	44
5.6.3 Controle de Erros – Heartbeat	45
5.7 PROCEDIMENTO DE INICIALIZAÇÃO	47
6 DESCRIÇÃO DOS OBJETOS PARA DRIVES	48
6.1 DEVICE CONTROL – OBJETOS PARA CONTROLE DO DRIVE	49
6.1.1 Objeto 6040h – Controlword	50
6.1.2 Objeto 6041h – Statusword	51
6.1.3 Objeto 6060h – Modes of Operation	52
6.1.4 Objeto 6061h – Modes of Operation Display	52
6.2 VELOCITY MODE – OBJETOS PARA CONTROLE DO DRIVE	53
6.2.1 Bits de Controle e Estado	53
6.2.2 Objeto 6042h – vl Target Velocity	53
6.2.3 Objeto 6043h – vl Velocity Demand	53
6.2.4 Objeto 6044h – vl Control Effort	54
6.2.5 Objeto 6046h – vl Velocity Min Max Amount	54
6.2.6 Objeto 6048h – vl Velocity Acceleration	54
6.2.7 Objeto 6049h – vl Velocity Deceleration	55
POSITION CONTROL FUNCTION – OBJETOS PARA CONTROLE DE POSIÇÃO	56
Objeto 6063h – Position actual value*	56
7 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO CANOPEN	57
A133/F233 – SEM ALIMENTAÇÃO NA INTERFACE CAN	57
A134/F234 – BUS OFF	57
A135/F235 – NODE GUARDING/HEARTBEAT	57

Sobre o Manual

Este manual fornece a descrição necessária para a operação do inversor de frequência CFW-11 utilizando o protocolo CANopen. Este manual deve ser utilizado em conjunto com manual do usuário do CFW-11.

Abreviações e Definições

CAN	Controllor Area Network
CiA	CAN in Automation
COB	Communication Object
COB-ID	Communication Object Identifier
SDO	Service Data Object
PDO	Process Data Object
RPDO	Receive PDO
TPDO	Transmit PDO
NMT	Network Management Object
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 CANopen para CFW-11 foi desenvolvido baseado nas seguintes especificações e documentos:

Documento	Versão	Fonte
CAN Specification	2.0	CiA
CiA DS 301 CANopen Application Layer and Communication Profile	4.02	CiA
CiA DRP 303-1 Cabling and Connector Pin Assignment	1.1.1	CiA
CiA DSP 306 Electronic Data Sheet Specification for CANopen	1.1	CiA
CiA DSP 402 Device Profile Drives and Motion Control	2.0	CiA

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

1 Introdução à Comunicação CANopen

Para a operação do inversor de frequência CFW-11 em rede CANopen, é necessário conhecer a forma como a comunicação é feita. Para isto, este item traz uma descrição geral do funcionamento do protocolo CANopen, contendo as funções utilizadas pelo CFW-11. Para uma descrição detalhada do protocolo, consulte a documentação CANopen indicada no item anterior.

1.1 CAN

A rede CANopen 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.

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. Caso ocorra uma quantidade considerável de erros, o dispositivo pode ser levado para os seguintes estados:

- Warning:** quando esse contador passa de um determinado limite, o dispositivo entra no estado de *warning*, significando a ocorrência de uma elevada taxa de erros.
- 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 um outro dispositivo enviou um telegrama com erro.
- Bus Off:** por último, temos o estado de *bus off*, no qual o dispositivo não irá mais enviar ou receber telegramas.

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 o protocolo CANopen do CFW-11, somente frames *standard* são aceitos.

1.1.5 CAN e CANopen

Somente a definição de como detectar erros, criar e transmitir um frame não é suficiente 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 CANopen define justamente como trocar dados entre os equipamentos e como cada dispositivo deve interpretar estes dados.

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

1.2 Características da Rede CANopen

Por utilizar um barramento CAN como forma de transmissão de telegramas, todos os dispositivos da rede CANopen têm os mesmos direitos de acesso à rede, onde a prioridade do identificador é responsável por resolver problemas de conflito quando acessos simultâneos ocorrem. Isto traz o benefício de possibilitar a comunicação direta entre escravos da rede, além do fato de que os dados podem ser disponibilizados de maneira mais otimizada, sem a necessidade de um mestre que controle toda a comunicação fazendo acesso cíclico a todos os dispositivos da rede para atualização dos dados.

Outra característica importante é a utilização do modelo produtor / consumidor para a transmissão de dados. Isto significa dizer que uma mensagem que trafega na rede não possui um endereço fixo na rede como destino. Esta mensagem possui um identificador que indica qual o dado que ela está transportando. Qualquer elemento da rede que necessite utilizar desta informação para a sua lógica de operação, poderá consumi-la e, portanto, uma mesma mensagem pode ser utilizada por vários elementos da rede ao mesmo tempo.

1.3 Meio Físico

O meio físico para a transmissão de sinais em uma rede CANopen é especificado pela norma ISO 11898. Ela define como barramento de transmissão um par trançado com sinal elétrico diferencial.

O inversor de frequência CFW-11 utiliza ainda um circuito de interface com a rede isolado. A fonte de alimentação para a interface CANopen é compartilhada com as entradas e saídas digitais e analógicas presente no cartão de controle do CFW-11. O componente responsável pela transmissão e recepção de sinais é denominado transceiver, que obedece ao especificado pela ISO 11898.

1.4 Endereço na Rede CANopen

Toda a rede CANopen deve possuir um mestre, responsável por serviços de gerenciamento da rede, e também pode possuir um conjunto de até 127 escravos. Cada dispositivo da rede também pode ser chamado de nó. Todo escravo em uma rede CANopen é identificado na rede através de seu endereço, ou Node-ID, que deve ser único para cada escravo da rede, e pode variar de 1 até 127.

O CFW-11 não possui funções que implementam os serviços de gerenciamento de rede, e portanto ele deve ser utilizado em conjunto com algum equipamento que possua tais serviços, em geral um mestre da rede CANopen.

1.5 Acesso aos Dados

Cada escravo da rede CANopen possui uma lista, denominada dicionário de objetos, que contém todos os dados que são acessíveis via rede. Cada objeto desta lista é identificado através de um índice, e durante a configuração do equipamento e troca de mensagens, este índice é utilizado para identificar o que está sendo transmitido.

Uma descrição mais detalhada de como o dicionário de objetos está estruturado é fornecida no item 6.

1.6 Transmissão de Dados

A transmissão de dados numéricos através de telegramas CANopen é feita utilizando a representação hexadecimal do número, e enviando o byte menos significativo do dado primeiro.

Exemplo: transmissão de um inteiro com sinal de 32 bits (12345678h = 305419896 decimal), mais um inteiro com sinal de 16 bits (FF00h = -256 decimal), em um frame CAN.

Identificador	6 bytes de dados					
11 bits	Inteiro 32 bits				Inteiro 16 bits	
	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	78h	56h	34h	12h	00h	FFh

1.7 Objetos Responsáveis pela Comunicação – COBs

Existe um determinado conjunto de objetos que são responsáveis pela comunicação entre os dispositivos da rede. Estes objetos estão divididos de acordo com os tipos de dados e a forma como são enviados ou recebidos por um dispositivo. O CFW-11 suporta os seguintes objetos de comunicação (COBs):

Tabela 1.1 - Tipos de Objetos de Comunicação (COBs)

Tipo de Objeto	Descrição
Service Data Object (SDO)	Os SDOs são objetos responsáveis pelo acesso direto ao dicionário de objetos de um dispositivo. Através de mensagens utilizando os SDOs, é possível indicar explicitamente (através do índice do objeto), qual o dado que está sendo manipulado. Existem dois tipos de SDOs: Cliente SDO, responsável por fazer uma requisição leitura ou escrita para um dispositivo da rede, e o Servidor SDO, responsável por atender esta requisição. Como os SDOs são utilizados geralmente para configuração de um nó da rede, são menos prioritários que outros tipos de mensagens. Somente um SDO do tipo servidor está disponível para o CFW-11.
Process Data Object (PDO)	Os PDOs são utilizados para acessar dados do equipamento sem a necessidade de indicar explicitamente qual o objeto do dicionário está sendo acessado. Para isso, é necessário configurar previamente quais os dados que o PDO estará transmitindo (mapeamento dos dados). Também existem dois tipos de PDOs: PDO de recepção e PDO de transmissão. PDOs usualmente são utilizados para transmissão e recepção de dados utilizados durante a operação do dispositivo, e por isso são mais prioritários que os SDOs.
Emergency Object (EMCY)	Este objeto é responsável pelo envio de mensagens para indicar a ocorrência de erros no dispositivo. Quando um erro ocorre em um determinado dispositivo (Produtor EMCY), este pode enviar uma mensagem para a rede. Caso algum dispositivo da rede esteja monitorando esta mensagem (Consumidor EMCY), é possível programar para que uma ação seja tomada (desabilitar demais dispositivos da rede, reset de erros, etc.). O CFW-11 possui apenas a funcionalidade de produtor EMCY.
Synchronization Object (SYNC)	Na rede CANopen é possível programar um dispositivo (Produtor SYNC) para enviar, periodicamente, uma mensagem de sincronização para todos os dispositivos da rede. Estes dispositivos (Consumidores SYNC) podem então, por exemplo, enviar um determinado dado que necessita ser disponibilizado periodicamente. O CFW-11 possui a função de consumidor SYNC.
Network Management (NMT)	Toda a rede CANopen precisa ter um mestre que controle os demais dispositivos da rede (escravos). Este mestre será responsável por um conjunto de serviços que controlam a comunicação dos escravos e seu estado na rede CANopen. Os escravos são responsáveis por receber os comandos enviados pelo mestre e executar as ações solicitadas. O CFW-11 opera como um escravo da rede CANopen, e disponibiliza dois tipos de serviços que o mestre pode utilizar: serviços de controle do dispositivo, onde o mestre controla o estado de cada escravo na rede, e serviços de controle de erros (<i>Node Guarding</i>), onde o escravo envia mensagens periódicas para o mestre para informar que a conexão está ativa.

Toda a comunicação do inversor com a rede é feita utilizando-se estes objetos, e os dados que podem ser acessados são os existentes no dicionário de objetos do dispositivo. A descrição do funcionamento de cada COB é feita no item 5.

1.8 COB-ID

Um telegrama da rede CANopen sempre é transmitido por um objeto de comunicação (COB). Todo COB possui um identificador que indica o tipo de dado que está sendo transportado. Este identificador, chamado de COB-ID, possui um tamanho de 11 bits, e é transmitido no campo identificador de um telegrama CAN. Ele pode ser subdividido em duas partes:

Código da Função				Endereço do nó						
bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- Código da função: indica o tipo de objeto que está sendo transmitido;
- Endereço do nó: indica com qual dispositivo da rede o telegrama está vinculado.

A seguir é apresentada uma tabela com os valores padrão para os diferentes objetos de comunicação disponíveis no CFW-11. É necessário observar que o valor padrão do objeto depende do endereço do escravo, com exceção dos COB-IDs para NMT e SYNC, que são comuns para todos os elementos da rede. Estes valores também podem ser alterados durante a etapa de configuração do dispositivo.

Tabela 1.2 - COB-ID para os diferentes objetos

COB	Código da Função (bits 10 – 7)	COB-ID Resultante (função + endereço)
NMT	0000	0
SYNC	0001	128 (80h)
EMCY	0001	129 – 255 (81h – FFh)
PDO1 (tx)	0011	385 – 511 (181h – 1FFh)
PDO1 (rx)	0100	513 – 639 (201h – 27Fh)
PDO2 (tx)	0101	641 – 767 (281h – 2FFh)
PDO2 (rx)	0110	769 – 895 (301h – 37Fh)
PDO3 (tx)	0111	897 – 1023 (381h – 3FFh)
PDO3 (rx)	1000	1025 – 1151 (401h – 47Fh)
PDO4 (tx)	1001	1153 – 1279 (481h – 4FFh)
PDO4 (rx)	1010	1281 – 1407 (501h – 57Fh)
SDO (tx)	1011	1409 – 1535 (581h – 5FFh)
SDO (rx)	1100	1537 – 1663 (601h – 67Fh)
Node Guarding/ Heartbeat	1110	1793 – 1919 (701h – 77Fh)

1.9 Arquivo EDS

Cada dispositivo em uma rede CANopen possui um arquivo de configuração EDS, que contém informações sobre o funcionamento do dispositivo na rede CANopen, bem como a descrição de todos os objetos existentes para comunicação. Em geral este arquivo é utilizado por um mestre ou software de configuração, para programação dos dispositivos presentes na rede CANopen.

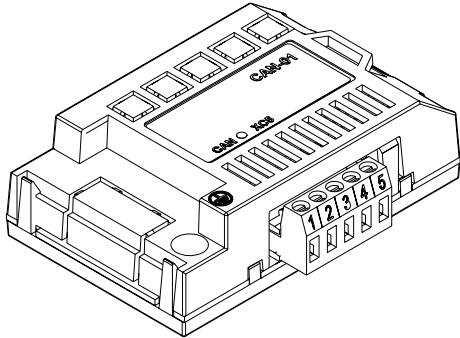
O arquivo de configuração EDS para o CFW-11 é fornecido juntamente com o produto, e também pode ser obtido através do site <http://www.weg.net>. É necessário observar a versão de software do inversor, para utilizar um arquivo EDS que seja compatível com esta versão.

2 Kits Opcionais

Para possibilitar a comunicação CANopen no inversor de frequência CFW-11, é necessário utilizar um dos kits para comunicação CAN descritos a seguir. Informações sobre a instalação destes módulos no inversor podem ser obtidas na bula que acompanha o kit.

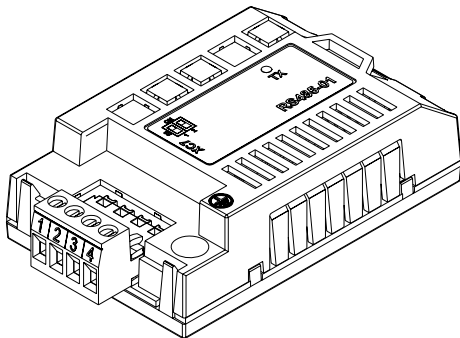
2.1 Interfaces CAN

2.1.1 Kit CAN-01



- ☑ Item WEG: 10051961.
- ☑ Composto pelo módulo de comunicação CAN (figura ao lado) mais uma bula de montagem.
- ☑ Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- ☑ Alimentação externa de 24 V.
- ☑ Permite a conexão de até 32 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada e com o uso de repetidores³.
- ☑ Comprimento máximo do barramento de 1000 metros.

2.1.2 Kit CAN/RS485-01



- ☑ Item WEG: 10051960.
- ☑ Composto pelo módulo de comunicação CAN/RS485 (figura ao lado) mais uma bula de montagem.
- ☑ Possuem as mesmas características da interface CAN-01, mais uma interface RS485, para aplicações onde seja necessária a operação em conjunto de ambas as interfaces.

2.1.3 Pinagem do Conector

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

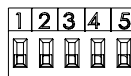


Tabela 2.1 - Pinagem do conector XC5 para interface CAN

Pino	Nome	Função
1	V-	Pólo negativo da fonte de alimentação
2	CAN L	Sinal de comunicação CAN L
3	Shield	Blindagem o cabo
4	CAN H	Sinal de comunicação CAN H
5	V+	Pólo positivo da fonte de alimentação

2.1.4 Fonte de Alimentação

A interface CAN para o CFW-11 necessita de uma tensão de alimentação externa entre os pinos 1 e 5 do conector da rede. Para evitar problemas de diferença de tensão 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. Os dados para consumo individual e tensão de entrada são apresentados na tabela a seguir.

³ O número limite de equipamentos que podem ser conectados na rede também depende do protocolo utilizado.

Tabela 2.2 - Características da alimentação para interface CAN

Tensão de alimentação (V_{CC})		
Mínimo	Máximo	Recomendado
11	30	24
Corrente (mA)		
Mínimo	Máximo	Médio
20	50	30

Os módulos de interface CAN possuem um LED na cor verde para indicar que a interface está alimentada.

2.1.5 Resistor de Terminação

As extremidades do barramento CAN devem possuir um resistor de terminação no valor de $120 \Omega / 0.25 W$, conectando os sinais CAN_H e CAN_L caso o drive seja o primeiro ou último elemento do segmento.

2.1.6 Taxa de Comunicação

A taxa de comunicação (*baud rate*) que pode ser utilizada por um equipamento na rede CANopen depende do comprimento do cabo utilizado na instalação. A tabela a seguir mostra as taxas de comunicação disponíveis para o CFW-11, e o comprimento máximo de cabo que pode ser utilizado na instalação, de acordo com o recomendado pela CiA.

Tabela 2.3 - Taxas de comunicação suportadas e tamanho da instalação

Taxa de comunicação	Comprimento do cabo
1 Mbit/s	40 m
500 Kbit/s	100 m
250 Kbit/s	250 m
125 Kbit/s	500 m
100 Kbit/s	600 m
50 Kbit/s	1000 m
20 Kbit/s	1000 m
10 Kbit/s	1000 m

2.1.7 Ligação do Inversor na Rede CAN

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 devido à interferência eletromagnética, isto facilita 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 no mesmo ponto de terra.

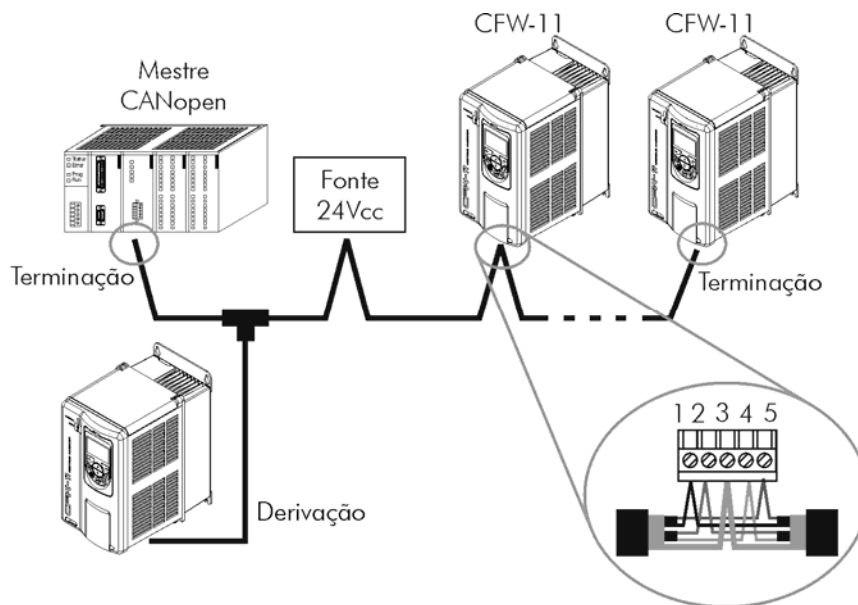


Figura 2-1 - CFW-11 em rede CANopen

O cabo para a ligação dos sinais CAN_L e CAN_H deve ter impedância característica de aproximadamente 120 Ohm, e um atraso máximo de propagação do sinal de 5ns/m. Outras características dependem do comprimento do cabo, que deve estar de acordo com a tabela a seguir.

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

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.

3 Parametrização do Inversor

A seguir serão apresentados apenas os parâmetros do inversor de frequência CFW-11 que possuem relação com a comunicação CANopen.

3.1 Símbolos para Descrição das Propriedades

RO	Parâmetro somente de leitura
CFG	Parâmetro somente alterado com o motor parado
Net	Parâmetro visível através da HMI se o inversor possuir interface de rede instalada - RS232, RS485, CAN, Anybus-CC, Profibus - ou se a interface USB for conectada
Serial	Parâmetro visível através da HMI se o inversor possuir interface RS232 ou RS485 instalada
CAN	Parâmetro visível através da HMI se o inversor 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 inversor CFW-11. Para que o inversor seja controlado através da interface CANopen, deve-se selecionar uma das opções 'CANopen/DNet' disponíveis nos parâmetros.

A descrição detalhada deste parâmetros encontra-se no Manual de Programação do CFW-11.

P0313 – Ação para Erro de Comunicação

Faixa de	0 = Inativo	Padrão: 0
Valores:	1 = Para por Rampa	
	2 = Desabilita Geral	
	3 = Vai para Local	
	4 = Vai para Local e mantém comandos e referência	
	5 = Causa Falha	

Propriedades: CFG, Net

Grupos de acesso via HMI:

01	GRUPOS PARÂMETROS	
L	49	Comunicação
L	111	Estados / Comandos

Descrição:

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

Tabela 3.1 - Valores para o parâmetro P0313

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, caso o inversor seja programado para utilizar em modo local comandos via HMI ou 3 wire start stop, e a referência de velocidade via HMI ou potenciômetro eletrônico.
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.

Para a interface CAN utilizando o protocolo CANopen, são considerados erros de comunicação os seguintes eventos:

- 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*)

A descrição destes alarmes/falhas é feita no item 7.

As ações descritas neste parâmetro são executadas através da escrita automática dos respectivos bits no parâmetro de controle via CAN – P0684. Para que a ação executada tenha efeito, é necessário que o inversor esteja programado para ser controlado via interface CAN. Esta programação é feita através dos parâmetros P0220 até P0228.

P0680 – Estado Lógico

Faixa de 0000h – FFFFh

Padrão: -

Valores:

Propriedades: RO

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

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

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

Tabela 3.2 - 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: <input checked="" type="checkbox"/> Executando rotina de auto-ajuste; <input checked="" type="checkbox"/> Executando rotina de <i>start-up</i> orientado; <input checked="" type="checkbox"/> Executando função <i>copy</i> da HMI; <input checked="" type="checkbox"/> Executando rotina auto-guiada do cartão de memória flash; <input checked="" type="checkbox"/> Possui incompatibilidade de parametrização; <input checked="" type="checkbox"/> 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 Rampa Habilitada (RUN)	0: Motor está parado 1: Inversor está girando o motor à 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 motor
Bit 10 Sentido de Giro	0: Motor girando no sentido reverso. 1: Motor girando no sentido direto.
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 Manual/ Automático	0: Em modo manual (função PID) 1: Em modo automático (função PID)
Bit 15 Em Falha	0: Inversor não está no estado de falha 1: Alguma falha registrada pelo inversor Obs.: O número da falha pode ser lido através do parâmetro P0049 – Falha Atual

P0681 – Velocidade do Motor em 13 bits

Faixa de -32768 – +32767

Padrão: -

Valores:

Propriedades: RO

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Permite monitorar a velocidade do motor. Esta palavra utiliza resolução de 13 bits 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 pólos e 1800 rpm de rotação síncrona, caso o valor lido seja 2048 (0800h), para obter o valor em rpm deve-se calcular:

$$\frac{8192 - 1800 \text{ rpm}}{2048 - \text{velocidade em rpm}} = \frac{1800 \times 2048}{8192}$$

Velocidade em rpm = 450 rpm

Valores negativos para este parâmetro indicam motor girando no sentido anti-horário.

P0684 – Palavra de Controle via CANopen/DeviceNet

Faixa de 0000h – FFFFh

Padrão: 0000h

Valores:

Propriedades: CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Palavra de comando do inversor via interface serial. Este parâmetro somente pode ser alterado via interface CAN (protocolos CANopen ou DeviceNet). Para as demais fontes (HMI, USB, Serial, 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 CAN. Esta programação é feita através dos parâmetros P0105 e P0220 até P0228.

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

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 3.3 - 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: Girar motor no sentido oposto ao da referência 1: Girar motor no sentido indicado na referência
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 VVW 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 CANopen/DeviceNet

Faixa de -32768 – 32767
Valores:

Padrão: 0

Propriedades: CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Permite programar a referência de velocidade para o inversor via interface serial. Este parâmetro somente pode ser alterado via interface CAN (protocolos CANopen ou DeviceNet). Para as demais fontes (HMI, USB, Serial, etc.) ele se comporta como um parâmetro somente de leitura.

Para que a referência escrita neste parâmetro seja utilizada, é necessário que o inversor esteja programado para utilizar a referência de velocidade via serial. Esta programação é feita nos parâmetros P0221 e P0222.

Esta palavra utiliza resolução de 13 bits 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 pólos e 1800 rpm de rotação síncrona, caso deseje-se uma referência de 900 rpm, deve-se calcular:

$$\begin{array}{rcl} 1800 \text{ rpm} & - & 8192 \\ 900 \text{ rpm} & - & \text{referência em 13 bit} \end{array} \quad \text{referência em 13 bit} = \frac{900 \times 8192}{1800}$$

Referência em 13 bit = 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 – FFFFh
Valores:

Padrão: 0000h

Propriedades: Net

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Possibilita o controle das saídas digitais através das interfaces de rede (Serial, USB, 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 P0280.

Bits	15 a 5	4	3	2	1	0
Função	Reservado	Valor para DO5	Valor para DO4	Valor para DO3 (RL3)	Valor para DO2 (RL2)	Valor para DO1 (RL1)

Tabela 3.4 - Funções dos bits para o parâmetro P0695

Bits	Values
Bit 0 Valor para DO1 (RL1)	0: saída DO1 aberta 1: saída DO1 fechada
Bit 1 Valor para DO2 (RL2)	0: saída DO2 aberta 1: saída DO2 fechada
Bit 2 Valor para DO3 (RL3)	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

P0698 – Valor 3 para Saídas Analógicas

P0699 – Valor 4 para Saídas Analógicas

Faixa de -32768 – +32767

Padrão: 0

Valores:

Propriedades: Net

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Possibilita o controle das saídas analógicas através das interfaces de rede (Serial, USB, CAN, etc.). Este parâmetro não pode ser alterado 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 P0696/7/8/9”, nos parâmetros P0251, P0254, P0257 ou P0260.

O valor deve ser escrito em uma escala de 15 bits (7FFFh = 32767)⁴ 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 os parâmetros P0697/8/9. Por exemplo, deseja-se controlar o valor da saída analógica 1 através da interface CAN. Neste caso deve fazer a seguinte programação:

- Escolher um dos parâmetros P0696 a P0699 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”.

⁴ Para a resolução real da saída, consulte o manual do CFW-11.

- Através da interface CAN, 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 -10 V até 10 V, valores negativos para estes parâmetros devem ser utilizados para comandar a saída com valores negativos de tensão, ou seja, -32768 até 32767 representa uma variação de -10 V até 10 V na saída analógica.

P0700 – Protocolo CAN

Faixa de 1 = CANopen Padrão: 2
Valores: 2 = DeviceNet

Propriedades: CFG, CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS
L 49 Comunicação
L 112 CANopen / DeviceNet

Descrição:

Permite selecionar o protocolo desejado para a interface CAN. Para habilitar o protocolo CANopen, é necessário programar este parâmetro com a opção '1 = CANopen'.

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

P0701 – Endereço CAN

Faixa de 0 a 127 Padrão: 63
Valores:

Propriedades: CFG, CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS
L 49 Comunicação
L 112 CANopen / DeviceNet

Descrição:

Permite programar o endereço utilizado para comunicação CAN do inversor. 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, ele somente é válido após o inversor ser desligado e ligado novamente.

P0702 – Taxa de Comunicação CAN

Faixa de 0 = 1 Mbit/s Padrão: 0
Valores: 1 = 800 Kbit/s
2 = 500 Kbit/s
3 = 250 Kbit/s
4 = 125 Kbit/s
5 = 100 Kbit/s
6 = 50 Kbit/s
7 = 20 Kbit/s
8 = 10 Kbit/s

Propriedades: CFG, CAN

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

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.

P0703 – Reset de Bus Off

Faixa de 0 = Manual Padrão: 1
Valores: 1 = Automático

Propriedades: CFG, CAN

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

Descrição:

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

Tabela 3.5 - Valores para o parâmetro P0703

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 inversor volte a se comunicar através da interface CAN, será necessário desligar e ligar novamente o inversor.
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 = Desabilitado Padrão: -
Valores: 1 = Reservado
2 = Interface CAN ativa
3 = Warning
4 = Error Passive
5 = Bus Off
6 = Sem alimentação

Propriedades: RO

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

Descrição:

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

Tabela 3.6 - Valores do parâmetro P0705

Opções	Descrição
0 = Inativo	Interface CAN inativa. Ocorre quando o inversor não possui cartão de interface CAN instalado.
1 = Reservado	
2 = Interface CAN ativa	Interface CAN ativa e sem erros.
3 = <i>Warning</i>	Controlador CAN atingiu o estado de <i>warning</i> ⁵ .
4 = <i>Error Passive</i>	Controlador CAN atingiu o estado de <i>error passive</i> ⁴ .
5 = <i>Bus Off</i>	Controlador CAN atingiu o estado de <i>bus off</i> ⁴ .
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, CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 112 CANopen / DeviceNet

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, CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 112 CANopen / DeviceNet

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, CAN

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 112 CANopen / DeviceNet

⁵ Ver descrição no item 1.1.4.

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, CAN

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

Descrição:

Contador cíclico que indica o número de mensagens recebidas pela interface CAN, mas que não puderam ser processadas pelo inversor. 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.

P0721 – Estado da Comunicação CANopen

Faixa de Valores: 0 = Desabilitado

Padrão: -

- 1 = Reservado
- 2 = Comunicação Habilitada
- 3 = Controle de Erros Habilitado
- 4 = Erro de *Guarding*
- 5 = Erro de *Heartbeat*

Propriedades: RO, CAN

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

Descrição:

Indica o estado do cartão com relação à rede CANopen, informando se o protocolo foi habilitado e se o serviço de controle de erros está ativo (*Node Guarding* ou *Heartbeat*).

P0722 – Estado do Nó CANopen

Faixa de Valores: 0 = Desabilitado

Padrão: -

- 1 = Inicialização
- 2 = Parado
- 3 = Operacional
- 4 = Pré-Operacional

Propriedades: RO, CAN

Grupos de acesso via HMI:

- 01 GRUPOS PARÂMETROS
 - L 49 Comunicação
 - L 112 CANopen / DeviceNet

Descrição:

O inversor CFW-11 opera como escravo da rede CANopen, e como tal possui uma máquina de estados que controla o seu comportamento com relação à comunicação. Este parâmetro indica em qual estado encontra-se o dispositivo.

P0799 – Atraso Atualização I/O

Faixa de 0.0 a 999.0
Valores:

Padrão: 0.0

Propriedades: RW

Grupos de acesso via HMI:

01 GRUPOS PARÂMETROS

L 49 Comunicação

L 111 Estados / Comandos

Descrição:

Permite programar o tempo de atraso para atualização dos dados mapeados nas palavras de escrita (informações recebidas pelo equipamento) via rede de comunicação Profibus DP, Devicenet, CANopen e interface Anybus. O tempo de atraso é acionado na transição do estado do equipamento na rede de offline para online⁶, conforme figura 3.1.

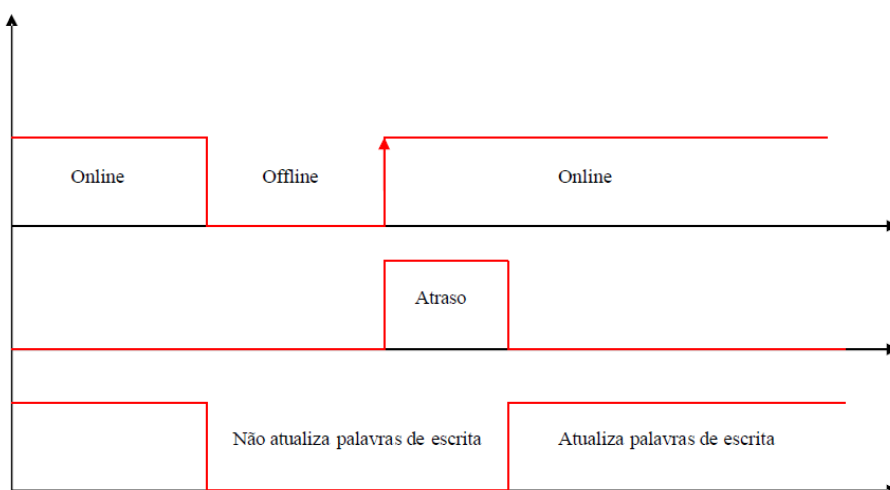


Figura 3.1- Atraso na atualização das palavras de I/Os

⁶ Para esta função, online representa o estado onde ocorre a troca de dados de I/O cíclicos de operação entre os equipamentos da rede.

4 Dicionário de Objetos

O dicionário de objetos é uma lista com os diversos dados do equipamento que são acessíveis através rede CANopen. Um objeto desta lista é identificado através de um índice de 16 bits, e é baseado nesta lista que toda a troca de dados entre os dispositivos é efetuada.

O documento CiA DS 301 define um conjunto mínimo de objetos que todo o escravo da rede CANopen deve possuir. Os objetos disponíveis nesta lista são agrupados de acordo com o tipo de função que ele executa. Os objetos são dispostos no dicionário da seguinte maneira:

Tabela 4.1 - Agrupamentos do dicionário de objetos

Índice	Objetos	Descrição
0001h – 0360h	Definição dos tipos de dados	Utilizado como referência para os tipos de dados suportados pelo sistema
1000h – 1FFFh	Objetos de comunicação	São objetos comuns a todos os dispositivos CANopen. Contém informações gerais sobre o equipamento e também dados para a configuração da comunicação
2000h – 5FFFh	Objetos específicos do fabricante	Nesta faixa, cada fabricante de equipamentos CANopen é livre para definir quais dados estes objetos irão representar
6000h – 9FFFh	Objetos padronizados para dispositivos	Esta faixa é reservada para objetos que descrevem o comportamento de equipamentos similares, independente do fabricante

Demais índices não referenciados nesta lista são reservados para uso futuro.

4.1 Estrutura do Dicionário

A estrutura geral do dicionário de objetos possui o seguinte formato:

Índice	Objeto	Nome	Tipo	Acesso
--------	--------	------	------	--------

- Índice:** indica diretamente o índice do objeto no dicionário.
- Objeto:** descreve que informação o índice armazena (variável simples, *array*, *record*, etc.)
- Nome:** contém o nome do objeto para facilitar sua identificação.
- Tipo:** indica diretamente o tipo de dado armazenado. Para variáveis simples, este tipo pode ser um inteiro, um *float*, etc. Para *arrays*, ele indica o tipo do dado contido no *array*. Para *records*, ele indica o formato do *record*, de acordo com os tipos descritos na primeira parte do dicionário de objetos (índices 0001h – 0360h).
- Acesso:** informa se o objeto em questão está acessível somente para leitura (*ro*), para leitura e escrita (*rw*), ou é uma constante (*const*).

Para objetos do tipo *array* ou *records*, ainda é necessário um sub-índice, que não é descrito na estrutura do dicionário.

4.2 Tipos de Dados

A primeira parte do dicionário de objetos (índices 0001h – 0360h) descreve os tipos de dados que podem ser acessados em um dispositivo na rede CANopen. Estes podem ser tipos básicos, como inteiros e *floats*, ou tipos compostos, formados por um conjunto de entradas, como *records* e *arrays*.

4.2.1 Tipos Básicos

Os seguintes tipos básicos de dados são suportados:

- Inteiros com sinal:** existem três tipos de inteiros com sinal suportados pelo CFW-11, INTEGER8, INTEGER16 e INTEGER32, que representam, respectivamente, inteiros com 8, 16 e 32 bits de dados. Inteiros com sinal são calculados utilizando complemento de dois, e durante a transmissão, sempre o byte menos significativo é transmitido primeiro em um telegrama CAN.
- Inteiros sem sinal:** existem três tipos de inteiros sem sinal suportados pelo CFW-11, UNSIGNED8, UNSIGNED16 e UNSIGNED32, que representam, respectivamente, inteiros com 8, 16 e 32 bits de dados. Também durante a transmissão, sempre o byte menos significativo é transmitido primeiro.

4.2.2 Tipos Compostos

É possível formar novos tipos de dados através do agrupamento de tipos básicos em listas (*arrays* – formados por um único tipo de dado) e estruturas (*records* – formado por diversos tipos de dados). Neste caso, cada item deste tipo é identificado através de um sub-índice. Os tipos compostos utilizados pelo inversor são listados abaixo.

PDO COMM PARAMETER: este *record* define as informações necessárias para configurar um PDO para a comunicação CANopen. O conteúdo e forma como cada campo é utilizado são detalhados no item 5.3.

Tabela 4.2 - Record para configuração dos PDOs

Sub-índice	Descrição da entrada	Tipo
00h	Número de entradas suportadas neste record	UNSIGNED8
01h	COB-ID	UNSIGNED32
02h	<i>Transmission type</i>	UNSIGNED8
03h	<i>Inhibit time</i>	UNSIGNED16
04h	Reservado	UNSIGNED8
05h	<i>Event timer</i>	UNSIGNED16

PDO MAPPING: este *record* define como mapear os dados que serão transmitidos por um PDO durante a comunicação CANopen. O conteúdo e forma como cada campo é utilizado são detalhados no item 5.3.

Tabela 4.3 - Record para mapeamento dos dados de um PDO

Sub-índice	Descrição da entrada	Tipo
00h	Número de objetos mapeados no PDO	UNSIGNED8
01h	1o objeto mapeado	UNSIGNED32
02h	2o objeto mapeado	UNSIGNED32
	...	
	...	
	...	
40h	64o objeto mapeado	UNSIGNED32

SDO PARAMETER: este *record* define as informações necessárias para configurar um SDO para a comunicação CANopen. O conteúdo e forma como cada campo é utilizado são detalhados no item 5.2.

Tabela 4.4 - Record para configuração dos SDOs

Sub-índice	Descrição da entrada	Tipo
00h	Número de entradas suportadas neste record	UNSIGNED8
01h	COB-ID cliente → servidor	UNSIGNED32
02h	COB-ID servidor → cliente	UNSIGNED32
03h	Node-ID do cliente/servidor	UNSIGNED8

IDENTITY: este *record* é utilizado para descrever o tipo de dispositivo presente na rede.

Tabela 4.5 - Record para identificação do dispositivo

Sub-índice	Descrição da entrada	Tipo
00h	Número de entradas suportadas neste record	UNSIGNED8
01h	<i>Vendor-ID</i>	UNSIGNED32
02h	<i>Product Code</i>	UNSIGNED32
03h	<i>Revision Number</i>	UNSIGNED32
04h	<i>Serial Number</i>	UNSIGNED32

4.2.3 Tipos Estendidos

O CFW-11 não possui tipos estendidos.

4.3 Communication Profile – Objetos para Comunicação

Os índices de 1000h até 1FFFh correspondem, no dicionário de objetos, à parte responsável pelas configurações da comunicação na rede CANopen. Estes objetos são comuns a todos os dispositivos, porém somente alguns são obrigatórios. A seguir é apresentada uma lista com os objetos desta faixa suportados pelo inversor.

Tabela 4.6 - Lista de objetos do CFW-11 – Communication Profile

Índice	Objeto	Nome	Tipo	Acesso
1000h	VAR	device type	UNSIGNED32	ro
1001h	VAR	error register	UNSIGNED8	ro
1005h	VAR	COB-ID SYNC	UNSIGNED32	rw
100Ch	VAR	guard time	UNSIGNED16	rw
100Dh	VAR	life time factor	UNSIGNED8	rw
1014h	VAR	COB-ID EMCY	UNSIGNED32	ro
1016h	ARRAY	Consumer heartbeat time	UNSIGNED32	rw
1017h	VAR	Producer heartbeat time	UNSIGNED16	rw
1018h	RECORD	Identity Object	Identity	ro
Server SDO Parameter				
1200h	RECORD	1st Server SDO parameter	SDO Parameter	ro
Receive PDO Communication Parameter				
1400h	RECORD	1st receive PDO Parameter	PDO CommPar	rw
1401h	RECORD	2nd receive PDO Parameter	PDO CommPar	rw
1402h	RECORD	3rd receive PDO Parameter	PDO CommPar	rw
1403h	RECORD	4th receive PDO Parameter	PDO CommPar	rw
Receive PDO Mapping Parameter				
1600h	RECORD	1st receive PDO mapping	PDO Mapping	rw
1601h	RECORD	2nd receive PDO mapping	PDO Mapping	rw
1602h	RECORD	3rd receive PDO mapping	PDO Mapping	rw
1603h	RECORD	4th receive PDO mapping	PDO Mapping	rw
Transmit PDO Communication Parameter				
1800h	RECORD	1st transmit PDO Parameter	PDO CommPar	rw
1801h	RECORD	2nd transmit PDO Parameter	PDO CommPar	rw
1802h	RECORD	3rd transmit PDO Parameter	PDO CommPar	rw
1803h	RECORD	4th transmit PDO Parameter	PDO CommPar	rw
Transmit PDO Mapping Parameter				
1A00h	RECORD	1st transmit PDO mapping	PDO Mapping	rw
1A01h	RECORD	2nd transmit PDO mapping	PDO Mapping	rw
1A02h	RECORD	3rd transmit PDO mapping	PDO Mapping	rw
1A03h	RECORD	4th transmit PDO mapping	PDO Mapping	rw

Demais objetos não mostrados nesta lista não são utilizados pelo CFW-11, ou então estão em faixas reservadas do dicionário.

4.4 Manufacturer Specific – Objetos Específicos do CFW-11

Nos índices de 2000h até 5FFFh, cada fabricante é livre para definir quais objetos estarão presentes, o tipo e a função de cada objeto. No caso do CFW-11, nesta faixa de objetos foi disponibilizada toda a lista de parâmetros. Através destes parâmetros é possível operar o CFW-11, executando qualquer função que o inversor possa realizar. Os parâmetros foram disponibilizados a partir do índice 2000h, e com o número do parâmetro somado a este índice para obter sua posição no dicionário. A tabela a seguir ilustra como estão distribuídos os parâmetros no dicionário de objetos.

Tabela 4.7 - Lista de objetos do CFW-11 – Manufacturer Specific

Índice	Objeto	Nome	Tipo	Acesso
2000h	VAR	P0000 – Parâmetro de acesso	INTEGER16	rw
2001h	VAR	P0001 – Referência de Velocidade	INTEGER16	ro
2002h	VAR	P0002 – Velocidade do motor	INTEGER16	ro
2003h	VAR	P0003 – Corrente do motor	INTEGER16	ro
2004h	VAR	P0004 – Tensão CC	INTEGER16	ro
...
2064h	VAR	P0100 – Tempo de aceleração	INTEGER16	rw
2065h	VAR	P0101 – Tempo de desaceleração	INTEGER16	rw
...

Para a lista completa e uma descrição detalhada dos parâmetros, consulte o manual do CFW-11. É necessário conhecer a operação do inversor através dos parâmetros para poder programar corretamente sua operação via rede CANopen.

4.5 Device Profile – Objetos Comuns para Drives

A documentação CANopen também inclui propostas para padronização de determinados tipos de dispositivos. No caso do CFW-11, ele segue o descrito pela *CiA DPS 402 – Device Profile Drives and Motion Control*. Este documento descreve um conjunto de objetos que devem ser comuns para drives, independente do fabricante. Isto facilita a interoperabilidade entre dispositivos com a mesma função (como inversores de frequência), pois tanto os dados quanto o comportamento do dispositivo são disponibilizados de uma forma padronizada.

Para estes objetos foram reservados os índices de 6000h até 9FFFh. É possível operar o inversor através da rede CANopen, tanto através dos parâmetros (localizados a partir do índice 2000h) quanto através destes objetos padronizados. Para uma descrição detalhada de quais objetos estão disponíveis nesta faixa do dicionário de objetos, consulte a seção 6.

5 Descrição dos Objetos de Comunicação

Nesta seção são descritos detalhadamente cada um dos objetos citados na Tabela 1.1, além de descrever também o funcionamento dos objetos de comunicação (COBs) referenciados no item 1.7. É necessário conhecer como estes objetos são operados para utilizar as funções disponíveis para a comunicação do CFW-11.

5.1 Objetos de Identificação

Existe um conjunto de objetos no dicionário utilizados para identificação do equipamento, porém não possuem influência no seu comportamento na rede CANopen.

5.1.1 Objeto 1000h – Device Type

Este objeto fornece um código em 32 bits que descreve o tipo de objeto e sua funcionalidade.

Índice	1000h
Nome	Device type
Objeto	VAR
Tipo	UNSIGNED32

Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0001.0192h

Este código pode ser dividido em duas partes: 16 bits inferiores, descrevendo o tipo de perfil (*profile*) que o dispositivo utiliza, e 16 bits superiores, indicando uma função específica, de acordo com o perfil especificado. Para o CFW-11, estes valores são respectivamente 0192h (segue o especificado pelo documento *CiA 402 - Device Profile Drives and Motion Control*), e 0001h (possui funcionalidades de inversor de frequência).

5.1.2 Objeto 1001h – Error Register

Este objeto indica a ocorrência ou não de erro no dispositivo. O tipo de erro registrado para o CFW-11 segue o descrito pela descrição da tabela 5.1.

Índice	1001h
Nome	Error register
Objeto	VAR
Tipo	UNSIGNED8

Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED8
Valor Padrão	0

Tabela 5.1 - Estrutura do objeto Error Register

Bit	Significado
0	Erro genérico
1	Corrente
2	Tensão
3	Temperatura
4	Comunicação
5	Reservado (sempre 0)
6	Reservado (sempre 0)
7	Específico do fabricante

Caso o dispositivo apresente algum erro, o bit equivalente deve ser ativado. O primeiro bit (erro genérico) deverá ser ativado em qualquer situação de erro.

5.1.3 Objeto 1018h – Identity Object

Traz informações gerais sobre o dispositivo.

Índice	1018h
Nome	Identity object
Objeto	Record
Tipo	Identity

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	4

Sub-índice	1
Descrição	Vendor ID
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0000.0123h

Sub-índice	2
Descrição	Código do produto
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0000.0400h

Sub-índice	3
Descrição	Número da revisão
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	De acordo com a versão de firmware do equipamento

Sub-índice	4
Descrição	Número serial
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	Diferente para cada CFW-11

O Vendor ID é um número que identifica o fabricante junto à CiA. O código do produto é definido pelo fabricante de acordo com o tipo de produto. O número da revisão representa a versão de firmware do equipamento. O sub-índice 4 é um número serial único para cada inversor de frequência CFW-11 em rede CANopen.

5.2 Service Data Objects – SDOs

Os SDOs são responsáveis pelo acesso direto ao dicionário de objetos de um determinado dispositivo na rede. Eles são utilizados para a configuração e, portanto, possuem baixa prioridade, já que não devem ser utilizados para comunicar dados necessários para a operação do dispositivo.

Existem dois tipos de SDOs: cliente e servidor. Basicamente, a comunicação inicia com o cliente (usualmente o mestre da rede) fazendo uma requisição de leitura (*upload*) ou escrita (*download*) para um servidor, e este responde ao que foi requisitado.

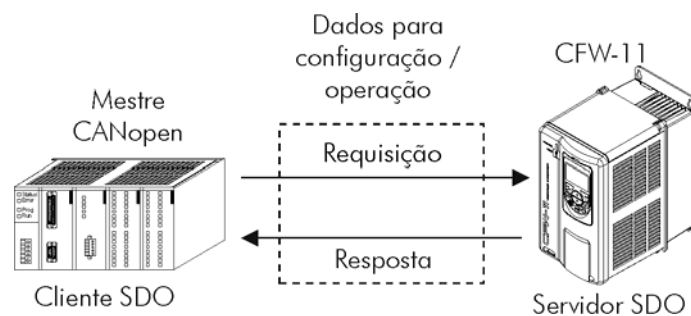


Figura 5-1 - Comunicação entre Cliente e Servidor SDO

5.2.1 Objeto 1200h – Servidor SDO

O CFW-11 possui um único SDO do tipo servidor, que possibilita o acesso a todo o seu dicionário de objetos. Através dele, um cliente SDO pode configurar a comunicação, parâmetros e modos de operação do inversor. Todo o servidor SDO possui um objeto, do tipo SDO_PARAMETER (ver item 4.2.2), para a sua configuração, possuindo a seguinte estrutura:

Índice	1200h
Nome	Server SDO Parameter
Objeto	Record
Tipo	SDO Parameter

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	2

Sub-índice	1
Descrição	COB-ID Cliente - Servidor (rx)
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	600h + Node-ID

Sub-índice	2
Descrição	COB-ID Servidor - Cliente (tx)
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	580h + Node-ID

5.2.2 Funcionamento dos SDOs

Um telegrama enviado por um SDO possui 8 bytes de tamanho, com a seguinte estrutura:

Identificador	8 bytes de dados							
	Comando	Índice		Sub-índice	Dados do objeto			
11 bits	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

O identificador depende do sentido da transmissão (rx ou tx) e do endereço (ou Node-ID) do servidor destino. Por exemplo, um cliente que faz uma requisição para um servidor cujo Node-ID é 1, deve enviar uma mensagem com o identificador igual a 601h. O servidor irá receber esta mensagem e responder com um telegrama cujo COB-ID é igual a 581h.

O código do comando depende do tipo de função utilizada. Para as transmissões de um cliente para um servidor, podem ser utilizados os seguintes comandos:

Tabela 5.2 - Código dos comandos para cliente SDO

Comando	Função	Descrição	Dados do objeto
22h	Download	Escrita em objeto	Indefinido
23h	Download	Escrita em objeto	4 bytes
2Bh	Download	Escrita em objeto	2 bytes
2Fh	Download	Escrita em objeto	1 byte
40h	Upload	Leitura de objeto	Não utilizado
60h ou 70h	Upload segment	Leitura segmentada	Não utilizado

Ao fazer a requisição, o cliente indicará através de seu COB-ID, qual o endereço do escravo para o qual esta requisição se destina. Somente um escravo (usando seu respectivo servidor SDO) poderá responder para o cliente o telegrama recebido. O telegrama de resposta possuirá também a mesma estrutura do telegrama de requisição, porém os comandos serão diferentes:

Tabela 5.3 - Código dos comandos para servidor SDO

Comando	Função	Descrição	Dados do objeto
60h	Download	Resposta para escrita em objeto	Não utilizado
43h	Upload	Resposta para leitura de objeto	4 bytes
4Bh	Upload	Resposta para leitura de objeto	2 bytes
4Fh	Upload	Resposta para leitura de objeto	1 byte
41h	Upload segment	Inicia resposta segmentada para leitura	4 bytes
01h ... 0Dh	Upload segment	Último segmento de dados para leitura	8 ... 2 bytes

Para leituras que envolvem até quatro bytes de dados, uma única mensagem pode ser transmitida pelo servidor; para leitura de uma quantidade maior de bytes, é necessário que cliente e servidor troquem múltiplos telegramas.

Um telegrama somente é completo após a confirmação do servidor para a requisição feita pelo cliente. Caso algum erro seja detectado durante a troca de telegramas (por exemplo, não há resposta do servidor), o cliente poderá abortar o processo com uma mensagem de aviso com o código do comando igual a 80h.



NOTA!

Quando o SDO é utilizado para escrita nos objetos que representam os parâmetros do CFW-11 (objetos a partir do índice 2000h), este valor é salvo na memória não volátil do inversor de frequência. Desta forma, depois de desligado ou feito o reset do equipamento, os valores configurados não são perdidos. Para os demais objetos, estes valores não são salvos automaticamente, de maneira que é necessário reescrever os valores desejados.

Exemplo: um cliente SDO solicita para um CFW-11 no endereço 1, a leitura do objeto identificado pelo índice 2000h, sub-índice 0 (zero), que representa um inteiro de 16 bits. O telegrama do mestre possui a seguinte forma:

Identificador	Comando	Índice		Sub-índice	Dados			
601h	40h	00h	20h	00h	00h	00h	00h	00h

O CFW-11 responde à requisição, indicando que o valor para o referido objeto é igual a 999⁷:

Identificador	Comando	Índice		Sub-índice	Dados			
581h	4Bh	00h	20h	00h	E7	03h	00h	00h

5.3 Process Data Objects – PDOs

Os PDOs são utilizados para enviar e receber dados utilizados durante a operação do dispositivo, que muitas vezes precisam ser transmitidos de forma rápida e eficiente. Por isso, eles possuem uma prioridade maior do que os SDOs.

⁷ Não esquecer que qualquer dado do tipo inteiro, a ordem de transferência dos bytes vai do menos significativo até o mais significativo.

Nos PDOs, apenas os dados são transmitidos no telegrama (índices e sub-índices são omitidos), e desta forma é possível fazer uma transmissão mais eficiente, com maior volume de dados em um único telegrama. Porém é necessário configurar previamente o que está sendo transmitido pelo PDO, de forma que, mesmo sem a indicação do índice e sub-índice, seja possível saber o conteúdo do telegrama.

Existem dois tipos de PDOs, os PDOs de recepção e os PDOs de transmissão. Os PDOs de transmissão são responsáveis por enviar dados para a rede, enquanto que os PDOs de recepção ficam responsáveis por receber e tratar estes dados. Desta forma é possível que haja comunicação entre escravos da rede CANopen, basta configurar um escravo para transmitir uma informação, e um ou mais escravos para receber esta informação.

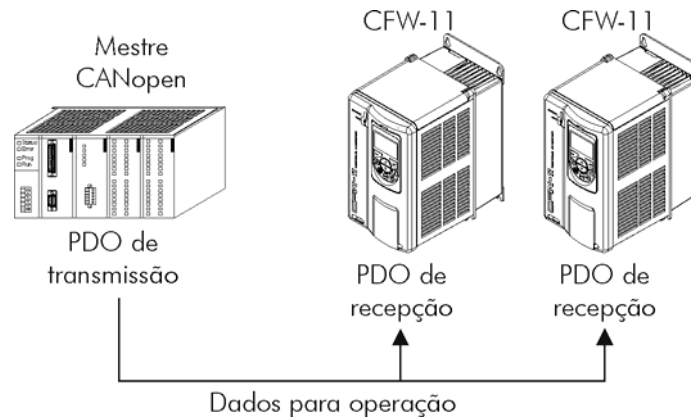


Figura 5-2 - Comunicação utilizando PDOs



NOTA!

PDOs somente podem ser transmitidos ou recebidos quando o dispositivo está no estado operacional. A figura 5.5 ilustra os estados disponíveis para um nó da rede CANopen.

5.3.1 Objetos Mapeáveis para os PDOs

Para um objeto poder ser transmitido através de um PDO, é necessário que ele seja mapeável para o conteúdo do PDO. Na descrição dos objetos de comunicação (1000h – 1FFFh), o campo "Mapeável" informa esta condição. Usualmente, apenas informações necessárias para a operação do dispositivo são mapeáveis, como comandos para habilitação, status do dispositivo, referências, etc. Informações para configuração do dispositivo não são acessíveis através de PDOs, e caso seja necessário acessá-las via rede deve-se utilizar os SDOs.

Para os objetos específicos do CFW-11 (2000h – 5FFFh), a tabela a seguir apresenta alguns objetos mapeáveis para os PDOs. Parâmetros com acesso apenas para leitura (ro) podem ser utilizados apenas por PDOs de transmissão, enquanto que os demais parâmetros podem ser utilizados apenas por PDOs de recepção. O arquivo EDS do CFW-11 traz a lista de todos os objetos disponíveis para o inversor, informando se o objeto é mapeável ou não.

Tabela 5.4 - Exemplos de parâmetros mapeáveis para PDOs

Índice	Objeto	Nome	Tipo	Acesso
2002h	VAR	P0002 – Velocidade do Motor	UNSIGNED16	ro
2003h	VAR	P0003 – Corrente do Motor	UNSIGNED16	ro
2004h	VAR	P0004 – Tensão Barram.CC (Ud)	UNSIGNED16	ro
2005h	VAR	P0005 – Freqüência do Motor	UNSIGNED16	ro
2006h	VAR	P0006 – Estado do Inversor	UNSIGNED16	ro
2007h	VAR	P0007 – Tensão de Saída	UNSIGNED16	ro
2009h	VAR	P0009 – Torque no Motor	INTEGER16	ro
200Ah	VAR	P0010 – Potência de Saída	UNSIGNED16	ro
200Ch	VAR	P0012 – Estado DI1 a DI8	UNSIGNED16	ro
200Dh	VAR	P0013 – Estado DO1 a RL3	UNSIGNED16	ro
2012h	VAR	P0018 – Valor de AI1	INTEGER16	ro
2013h	VAR	P0019 – Valor de AI2	INTEGER16	ro
2014h	VAR	P0020 – Valor de AI3	INTEGER16	ro
2015h	VAR	P0021 – Valor de AI4	INTEGER16	ro
2064h	VAR	P0100 – Tempo Aceleração	UNSIGNED16	rw
2065h	VAR	P0101 – Tempo Desaceleração	UNSIGNED16	rw
22A8h	VAR	P0680 – Estado Lógico	UNSIGNED16	ro
22A9h	VAR	P0681 – Velocidade 13 bits	UNSIGNED16	ro
22ACh	VAR	P0684 – Controle CANopen/DNet	UNSIGNED16	rw
22ADh	VAR	P0685 – Ref. Vel. CANop./DNet	UNSIGNED16	rw

5.3.2 PDOs de Recepção

Os PDOs de recepção, ou RPDOs, são responsáveis por receber dados que outros dispositivos enviam para a rede CANopen. O CFW-11 possui 4 PDOs de recepção, cada um podendo receber até 8 bytes de dados. Cada RPDO possui dois parâmetros para sua configuração, um PDO_COMM_PARAMETER e um PDO_MAPPING, conforme descrito a seguir.

PDO_COMM_PARAMETER

Índice	1400h
Nome	Receive PDO communication parameter
Objeto	Record
Tipo	PDO COMM PARAMETER

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	2

Sub-índice	1
Descrição	COB-ID usado pelo PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1400h: 200h + Node-ID 1401h: 300h + Node-ID 1402h: 400h + Node-ID 1403h: 500h + Node-ID

Sub-índice	2
Descrição	Tipo de transmissão
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	254

O sub-índice 1 contém o COB-ID do PDO de recepção. Sempre que uma mensagem for enviada para a rede, este objeto irá ler qual o COB-ID desta mensagem, e caso ele seja igual ao valor deste campo, a mensagem será recebida pelo dispositivo. Este campo é formado por um UNSIGNED32 com a seguinte estrutura:

Tabela 5.5 - Descrição do COB-ID

Bit	Valor	Descrição
31 (MSB)	0	PDO está habilitado
	1	PDO está desabilitado
30	0	RTR permitido
29	0	Tamanho do identificador = 11 bits
28 – 11	0	Não utilizado, sempre 0
10 – 0 (LSB)	X	COB-ID de 11 bits

O bit 31 permite habilitar ou desabilitar o PDO. Os bits 30 e 29, que devem ser mantidos em 0 (zero), indicam respectivamente que o PDO aceita frames remotos (RTR frames) e que utiliza identificador de 11 bits. Como o inversor de frequência CFW-11 não utiliza identificadores de 29 bits, os bits de 28 até 11 devem ser mantidos em 0 (zero), enquanto que os bits de 10 até 0 (zero) são usados para configurar o COB-ID para o PDO.

O sub-índice 2 indica o tipo de transmissão deste objeto, de acordo com a tabela a seguir.

Tabela 5.6 - Descrição do tipo de transmissão

Tipo de transmissão	Transmissão de PDOs				
	Cíclico	Acíclico	Síncrono	Assíncrono	RTR
0		•	•		
1 – 240	•		•		
241 – 251	Reservado				
252			•		•
253				•	•
254				•	
255				•	

- Valores 0 – 240:** qualquer RPDOs programado nesta faixa possui o mesmo funcionamento. Ao detectar uma mensagem, ele irá receber os dados, porém não atualizará os valores recebidos até detectar o próximo telegrama SYNC.
- Valores 252 e 253:** não permitido para PDOs de recepção.
- Valores 254 e 255:** indica que não possui relação com o objeto de sincronização. Ao receber uma mensagem, seus valores serão atualizados imediatamente.

PDO_MAPPING

Índice	1600h
Nome	Receive PDO mapping
Objeto	Record
Tipo	PDO MAPPING

Sub-índice	0
Descrição	Número de objetos mapeados
Acesso	ro
Mapeável	Não
Faixa	0 = desabilitado 1 ... 4 = número de objetos mapeados
Valor Padrão	1600h: 1 1601h: 2 1602h: 2 1603h: 2

Sub-índice	1
Descrição	1º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1600h: 6040.0010h 1601h: 6040.0010h 1602h: 6040.0010h 1603h: 22AC.0010h

Sub-índice	2
Descrição	2º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1600h: 0 1601h: 6060.0008h 1602h: 6042.0010h 1603h: 22AD.0010h

Sub-índice	3
Descrição	3º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1600h: 0 1601h: 0 1602h: 0 1603h: 0

Sub-índice	4
Descrição	4º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1600h: 0 1601h: 0 1602h: 0 1603h: 0

Este parâmetro indica os objetos mapeados nos PDOs de recepção do CFW-11. Para cada RPDO, é possível mapear até quatro objetos diferentes, desde que o tamanho total não ultrapasse oito bytes. O mapeamento de um objeto é feito indicando o seu índice, sub-índice⁸ e tamanho (em bits) em um campo UNSIGNED32, com o seguinte formato:

UNSIGNED32		
Índice (16 bits)	Sub-índice (8 bits)	Tamanho do objeto (8 bits)

Por exemplo, analisando o mapeamento padrão do PDO de recepção, temos:

- Sub-índice 0 = 2:** o RPDO possui dois objetos mapeados.
- Sub-índice 1 = 22AC.0010h:** o primeiro objeto mapeado possui índice igual a 22ACh, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0684 do inversor, que representa a palavra de controle via CANopen.
- Sub-índice 2 = 22AD.0010h:** o segundo objeto mapeado possui índice igual a 22ADh, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0685 do inversor, que representa a referência de velocidade.

⁸ Caso o objeto seja do tipo VAR e não possua sub-índice, deve ser indicado o valor 0 (zero) para o sub-índice.

Desta forma, sempre que este PDO receber um telegrama, ele vai saber que o telegrama deverá conter quatro bytes de dados, com o conteúdo para a palavra de controle e referência de velocidade para o CFW-11. É possível modificar este mapeamento, alterando a quantidade ou o número dos objetos mapeados. Lembrar que no máximo podem ser mapeados 4 objetos ou 8 bytes.



NOTA!

- ☑ Para poder alterar os objetos mapeados em um PDO, primeiro é necessário escrever o valor 0 (zero) no sub-índice 0 (zero). Desta forma, os valores dos sub-índices 1 até 4 podem ser alterados. Depois de feito o mapeamento desejado, deve-se escrever novamente no sub-índice 0 (zero) o número de objetos que foram mapeados, habilitando novamente o PDO.
- ☑ Para agilizar a atualização dos dados via PDO, os valores recebidos através destes objetos não são salvos na memória não volátil do inversor. Desta forma, após um desligamento ou reset do equipamento, os objetos modificados por um RPDO voltam para o seu valor padrão.
- ☑ Não esquecer que os PDOs somente podem ser recebidos caso o CFW-11 esteja no estado operacional.

5.3.3 PDOs de Transmissão

Os PDOs de transmissão, ou TPDOs, como o nome diz, são responsáveis por transmitir dados para a rede CANopen. O CFW-11 possui 4 PDOs de transmissão, cada um podendo transmitir até 8 bytes de dados. De forma semelhante aos RPDOs, cada TPDO possui dois parâmetros para sua configuração, um PDO_COMM_PARAMETER e um PDO_MAPPING, conforme descrito a seguir.

PDO_COMM_PARAMETER

Índice	1800h
Nome	Transmit PDO Parameter
Objeto	Record
Tipo	PDO COMM PARAMETER

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	5

Sub-índice	1
Descrição	COB-ID usado pelo PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1800h: 180h + Node-ID 1801h: 280h + Node-ID 1802h: 380h + Node-ID 1803h: 480h + Node-ID

Sub-índice	2
Descrição	Tipo de transmissão
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	254

Sub-índice	3
Descrição	Tempo entre transmissões
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED16
Valor Padrão	-

Sub-índice	4
Descrição	Reservado
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	-

Sub-índice	5
Descrição	Temporizador de eventos
Acesso	rw
Mapeável	Não
Faixa	0 = desabilitado UNSIGNED16
Valor Padrão	0

O sub-índice 1 contém o COB-ID do PDO de transmissão. Sempre que este PDO enviar uma mensagem para a rede, o identificador desta mensagem será este COB-ID. A estrutura deste campo é descrita na tabela 5.5.

O sub-índice 2 indica o tipo de transmissão deste objeto, que segue o descrito pela tabela 5.6. Porém seu funcionamento é diferente para PDOs de transmissão:

- ☑ **Valor 0:** indica que a transmissão deve ocorrer imediatamente após a recepção de um telegrama SYNC, mas não periodicamente.
- ☑ **Valores 1 – 240:** o PDO deve ser transmitido a cada telegrama SYNC detectado (ou ocorrências múltiplas de SYNC, de acordo com o número escolhido entre 1 e 240).
- ☑ **Valor 252:** indica que o conteúdo da mensagem deve ser atualizado (mas não enviado), após a recepção de um telegrama SYNC. O envio da mensagem deve ser feito após a recepção de um frame remoto (RTR frame).
- ☑ **Valor 253:** o PDO deve atualizar e enviar uma mensagem assim que receber um frame remoto.
- ☑ **Valores 254:** o objeto deve ser transmitido de acordo com o timer programado no sub-índice 5.
- ☑ **Valores 255:** o objeto é transmitido automaticamente quando o valor de algum dos objetos mapeados neste PDO for alterado. Funciona por alteração de estado (*Change Of State*). Este tipo também permite que o PDO seja transmitido de acordo com o timer programado no sub-índice 5.

No sub-índice 3 é possível programar um tempo mínimo (em múltiplos de 100 us) que deve transcorrer para que, depois de transmitido um telegrama, um novo telegrama possa ser enviado por este PDO. O valor 0 (zero) desabilita esta função.

O sub-índice 5 contém um valor para habilitar um temporizador para o envio automático de um PDO. Desta forma, sempre que um PDO for configurado para o tipo assíncrono, é possível programar o valor deste temporizador (em múltiplos de 1ms), para que o PDO seja transmitido periodicamente no tempo programado.



NOTA!

- ☑ Deve-se observar o tempo programado neste temporizador, de acordo com a taxa de transmissão utilizada. Tempos muito pequenos (próximos ao tempo de transmissão do telegrama) podem monopolizar o barramento, causando a retransmissão indefinida do PDO e impedindo que outros objetos menos prioritários possam transmitir seus dados.
- ☑ O tempo mínimo permitido para esta função no CFW-11 é 2ms.
- ☑ É importante observar o tempo entre transmissões programado no sub-índice 3 principalmente quando o PDO for programado com o valor 255 no sub-índice 2 (*Change Of State*).
- ☑ Não esquecer que os PDOs somente podem ser transmitidos caso o escravo esteja no estado operacional.

PDO_MAPPING

Índice	1A00h
Nome	Transmit PDO mapping
Objeto	Record
Tipo	PDO MAPPING

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	0 = desabilitado 1 ... 4 = número de objetos mapeados
Valor Padrão	1A00h: 1 1A01h: 2 1A02h: 2 1A03h: 2

Sub-índice	1
Descrição	1º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1A00h: 6041.0010h 1A01h: 6041.0010h 1A02h: 6041.0010h 1A03h: 22A8.0010h

Sub-índice	2
Descrição	2º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1A00h: 0 1A01h: 6061.0008h 1A02h: 6044.0010h 1A03h: 22A9.0010h

Sub-índice	3
Descrição	3º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1A00h: 0 1A01h: 0 1A02h: 0 1A03h: 0

Sub-índice	4
Descrição	4º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1A00h: 0 1A01h: 0 1A02h: 0 1A03h: 0

O PDO MAPPING para a transmissão funciona de forma semelhante que para a recepção, porém neste caso são definidos os dados a serem transmitidos pelo PDO. Cada objeto mapeado deve ser colocado na lista de acordo com o descrito a seguir:

UNSIGNED32		
Índice (16 bits)	Sub-índice (8 bits)	Tamanho do objeto (8 bits)

Por exemplo, analisando o mapeamento padrão do quarto PDO de transmissão, temos:

- ☑ **Sub-índice 0 = 2:** este PDO possui dois objetos mapeados.
- ☑ **Sub-índice 1 = 22A8.0010h:** o primeiro objeto mapeado possui índice igual a 22A8h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0680 do inversor, que representa o estado do inversor.
- ☑ **Sub-índice 2 = 22A9.0010h:** o segundo objeto mapeado possui índice igual a 22A9h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0681 do inversor, que representa a velocidade do motor.

Desta forma, sempre que este PDO for transmitir seus dados, ele vai elaborar o seu telegrama contendo quatro bytes de dados, com os valores dos parâmetros P0680 e P0681. É possível modificar este mapeamento, alterando a quantidade ou o número dos parâmetros mapeados. Lembrar que no máximo podem ser mapeados 4 objetos ou 8 bytes.



NOTA!

Para poder alterar os objetos mapeados em um PDO, primeiro é necessário escrever o valor 0 (zero) no sub-índice 0 (zero). Desta forma, os valores dos sub-índices 1 até 4 podem ser alterados. Depois de feito o mapeamento desejado, deve-se escrever novamente no sub-índice 0 (zero) o número de objetos que foram mapeados, habilitando novamente o PDO.

5.4 Emergency Object – EMCY

O objeto de emergência (EMCY) é utilizado para sinalizar a ocorrência de um erro no dispositivo. Sempre que um erro ocorrer no CFW-11 (curto-circuito, sobretensão, falha na comunicação, etc.), este objeto irá enviar uma mensagem de emergência para a rede. Esta mensagem poderá ser interpretada por um consumidor EMCY (usualmente o mestre da rede), que poderá tomar uma ação de acordo com o programado para a aplicação, como fazer o reset de erro ou desabilitar os demais dispositivos da rede.

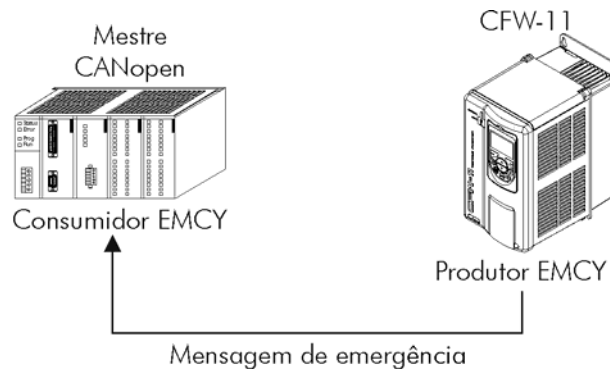


Figura 5-3 - EMCY

Ao transmitir uma mensagem, três informações são enviadas nos oitos bytes de dados do telegrama: o código de erro CiA, o objeto 1001h (error register) e o código da falha do CFW-11. O telegrama possui então a seguinte estrutura:

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
Código do erro CiA		Objeto 1001h Error Register	Código do Erro CFW-11	Reservado			

No CFW-11, este objeto está sempre ativo e, portanto, sempre reportará para a rede a ocorrência de erros no inversor. Existe um único parâmetro que possibilita ler qual o COB-ID do objeto, ou seja, qual o identificador do telegrama para as mensagens de erro enviadas.

Índice	1014h
Nome	COB-ID emergency message
Objeto	VAR
Tipo	UNSIGNED32

Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	80h + Node-ID

5.5 Synchronization Object – SYNC

Este objeto é transmitido com o objetivo de permitir a sincronização de eventos entre os dispositivos da rede CANopen. Ele é transmitido por um produtor SYNC, e os dispositivos que detectam a sua transmissão são denominados consumidores SYNC.

O CFW-11 possui a função de consumidor SYNC e, portanto, pode programar seus PDOs para serem síncronos. Como descrito na tabela 5.6, PDOs síncronos são aqueles relacionados com o objeto de sincronização, e, portanto podem ser programados para serem transmitidos ou atualizados com base neste objeto.

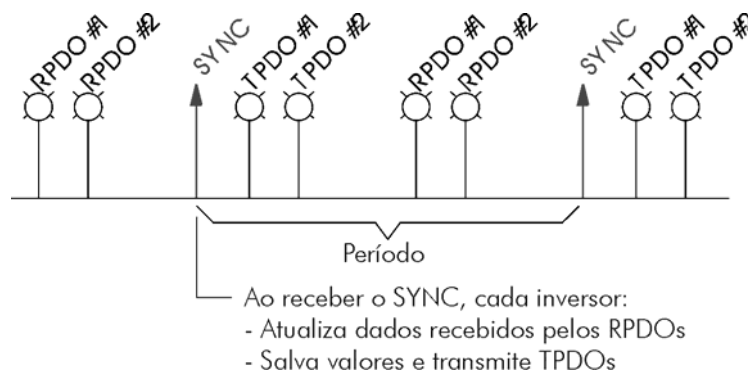


Figura 5-4 - SYNC

A mensagem SYNC transmitida pelo produtor não possui dado algum em seu campo de dados, pois seu objetivo é fornecer uma base de tempo para os demais objetos. No CFW-11, existe um objeto para a configuração do COB-ID do consumidor SYNC.

Índice	1015h
Nome	COB-ID SYNC
Objeto	VAR
Tipo	UNSIGNED32

Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	80h



NOTA!

Deve-se observar o tempo programado no produtor para o período dos telegramas SYNC, de acordo com a taxa de transmissão utilizada e o número de PDOs síncronos a serem transmitidos. É necessário que haja tempo suficiente para a transmissão destes objetos, e também é recomendado que haja folga para possibilitar o envio de mensagens assíncronas, como EMCY, PDOs assíncronos e SDOs.

5.6 Network Management – NMT

O objeto de gerenciamento da rede é responsável por um conjunto de serviços que controlam a comunicação do dispositivo na rede CANopen. Para o CFW-11 estão disponíveis os serviços de controle do nó e de controle de erros (utilizando *Node Guarding* ou *Heartbeat*).

5.6.1 Controle dos Estados do Escravo

Com relação à comunicação, um dispositivo da rede CANopen pode ser descrito pela seguinte máquina de estados:

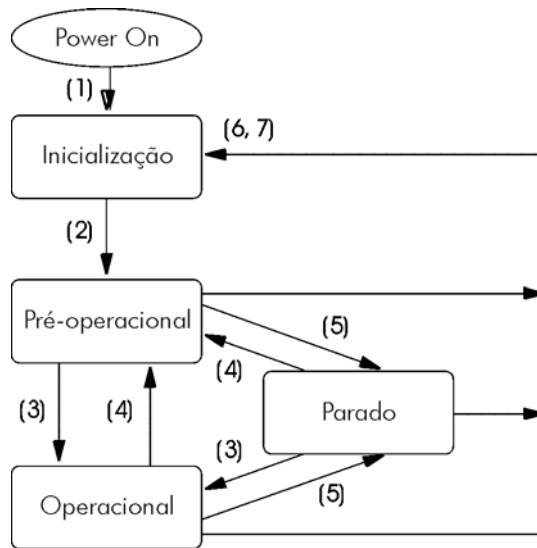


Figura 5-5 - Diagrama de estados do nó CANopen

Tabela 5.7 - Descrição das transições

Transição	Descrição
1	Dispositivo é ligado e começa a inicialização (automático)
2	Inicialização concluída, vai para o estado pré-operacional (automático)
3	Recebe comando Start Node para entrar no estado operacional
4	Recebe comando Enter Pre-Operational, e vai para o estado pré-operacional
5	Recebe comando Stop Node para entrar no estado parado
6	Recebe comando Reset Node, onde executa o reset completo do dispositivo
7	Recebe comando Reset Communication, onde reinicializa o valor dos objetos e a comunicação CANopen do dispositivo

Durante a inicialização, é definido o Node-ID, criados os objetos e configurada a interface com a rede CAN. Não é possível comunicar-se com o dispositivo nesta etapa, que é concluída automaticamente. No final desta etapa, o escravo envia para rede um telegrama do objeto Boot-up, utilizado apenas para indicar que a inicialização foi concluída e que o escravo entrou no estado pré-operacional. Este telegrama possui identificador 700h + Node-ID, e apenas um byte de dados com valor igual a 0 (zero).

No estado pré-operacional, já é possível comunicar-se com o escravo, porém os PDOs ainda não estão disponíveis para operação. No estado operacional, todos os objetos estão disponíveis, enquanto que no estado parado, apenas o objeto NMT pode receber ou transmitir telegramas para a rede. A tabela a seguir mostra os objetos disponíveis para cada estado.

Tabela 5.8 - Objetos acessíveis em cada estado

	Inicialização	Pré-operacional	Operacional	Parado
PDO			•	
SDO		•	•	
SYNC		•	•	
EMCY		•	•	
Boot-up	•			
NMT		•	•	•

Esta máquina de estados é controlada pelo mestre da rede, que envia para cada escravo, comandos para que seja executada a transição de estados desejada. Estes telegramas não possuem confirmação, o que significa que o escravo apenas recebe o telegrama sem retornar resposta para o mestre. Os telegramas recebidos possuem a seguinte estrutura:

Identificador	byte 1	byte 2
00h	Código do comando	Node-ID destino

Tabela 5.9 - Comandos para a transição de estados

Código do comando	Node-ID destino
1 = <i>START node</i> (transição 3)	0 = Todos os escravos
2 = <i>STOP node</i> (transição 4)	1 ... 127 = Escravo específico
128 = <i>Enter pre-operational</i> (transição 5)	
129 = <i>Reset node</i> (transição 6)	
130 = <i>Reset communication</i> (transição 7)	

As transições indicadas no código do comando equivalem às transições de estado executadas pelo nó após receber o comando (conforme figura 5.5). O comando *Reset node* faz com que o CFW-11 execute um reset completo do dispositivo, enquanto que o comando *Reset communication* faz com que o dispositivo reinicialize apenas os objetos relativos à comunicação CANopen.

5.6.2 Controle de Erros – Node Guarding

Este serviço é utilizado para possibilitar a monitoração da comunicação com a rede CANopen, tanto pelo mestre quanto pelo escravo. Neste tipo de serviço, o mestre envia telegramas periódicos para o escravo, que responde o telegrama recebido. Caso ocorra algum erro que interrompa a comunicação, será possível identificar este erro, pois tanto o mestre quanto o escravo serão notificados pelo *timeout* na execução deste serviço. Os eventos de erro são chamados de *Node Guarding* para o mestre, e de *Life Guarding* para o escravo.

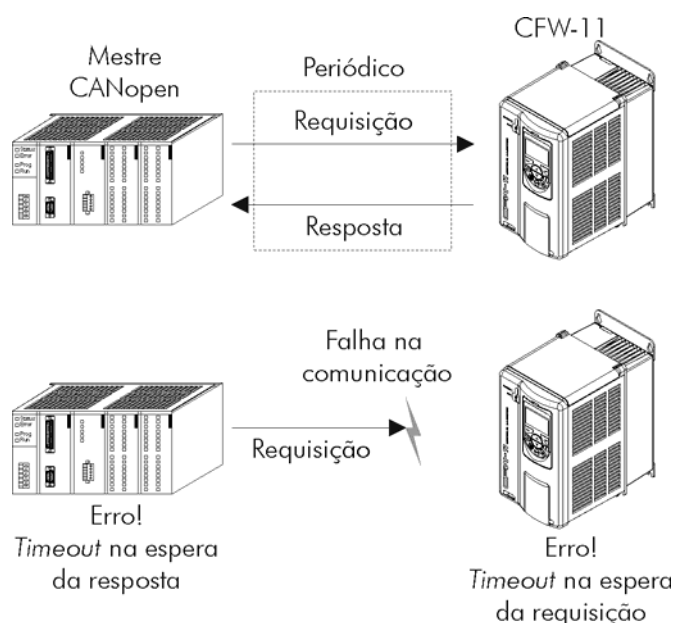


Figura 5-6 - Serviço de controle de erros – Node Guarding

Para o serviço de *Node Guarding*, existem dois objetos do dicionário para configuração dos tempos para detecção de erros de comunicação:

Índice	100Ch
Nome	Guard Time
Objeto	VAR
Tipo	UNSIGNED16

Acesso	rw
Mapeável	Não
Faixa	UNSIGNED16
Valor Padrão	0

Índice	100Dh
Nome	Life Time Factor
Objeto	VAR
Tipo	UNSIGNED8

Acesso	rw
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	0

O objeto 100Ch permite programar o tempo necessário (em milissegundos) para que uma ocorrência de falha seja detectada, caso o CFW-11 não receba nenhum telegrama do mestre. O objeto 100Dh indica quantas falhas em sequência são necessárias até que se considere que houve realmente erro de comunicação. Portanto, a multiplicação destes dois valores fornecerá o tempo total necessário para detecção de erros de comunicação utilizando este objeto. O valor 0 (zero) desabilita esta função.

Uma vez configurado, o CFW-11 começa a contar estes tempos a partir do primeiro telegrama *Node Guarding* recebido do mestre da rede. O telegrama do mestre é do tipo remoto, não possuindo bytes de dados. O identificador é igual a 700h + Node-ID do escravo destino. Já o telegrama de resposta do escravo possui 1 byte de dados com a seguinte estrutura:

Identificador	byte 1	
	bit 7	bit 6 ... bit 0
700h + Node-ID	Toggle	Estado do escravo

Este telegrama possui um único byte dados. Este byte contém, nos sete bits menos significativos, um valor para indicar o estado do escravo (4 = Parado, 5 = Operacional e 127 = Pré-operacional), e no oitavo bit, um valor que deve ser alterado a cada telegrama de enviado pelo escravo (*toggle bit*).

Caso o CFW-11 detecte um erro utilizando este mecanismo, ele irá automaticamente para o estado pré-operacional e indicará alarme A135 na sua HMI. A ocorrência deste erro também pode ser observada através do parâmetro P0721. É possível também programar o inversor para tomar uma ação quando este erro ocorrer, através do parâmetro P0313. Consulte o item 3 para a descrição detalhada dos parâmetros.



NOTA!

- Este objeto está ativo mesmo no estado parado (consulte a tabela 5.8).
- O valor 0 (zero) em um dos dois objetos desabilita esta função.
- Depois de detectado o erro, caso o serviço seja habilitado mais uma vez, a indicação do erro é retirada da HMI.
- O valor mínimo aceito para o CFW-11 é de 2 ms, mas levando-se em conta a taxa de transmissão e o número de pontos na rede, os tempos programados para essa função devem ser coerentes, de maneira que haja tempo suficiente para transmissão dos telegramas e também para que o resto da comunicação possa ser processada.
- Para cada escravo, somente um dos serviços – *Heartbeat* ou *Node Guarding* – pode ser habilitado.

5.6.3 Controle de Erros – *Heartbeat*

A detecção de erros através do mecanismo de *heartbeat* é feita utilizando dois tipos de objetos: o produtor *heartbeat* e o consumidor *heartbeat*. O produtor é responsável por enviar telegramas periódicos para a rede, simulando uma batida do coração, indicando que a comunicação está ativa e sem erros. Um ou mais consumidores podem monitorar estes telegramas periódicos e, caso estes telegramas deixem de ocorrer, significa que algum problema de comunicação ocorreu.

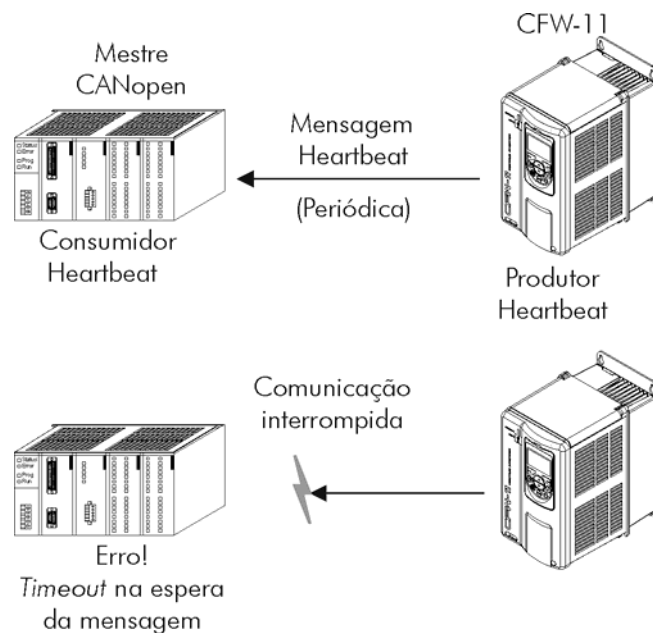


Figura 5-7 - Serviço de controle de erros – Heartbeat

Um mesmo dispositivo da rede pode ser produtor e consumidor de mensagens *heartbeat*. Por exemplo, o mestre da rede pode consumir mensagens enviadas por um escravo, permitindo detectar problemas de comunicação com o escravo, e ao mesmo tempo o escravo pode consumir mensagens *heartbeat* enviadas pelo mestre, também possibilitando ao escravo detectar falhas na comunicação com o mestre.

O CFW-11 possui os serviços de produtor e consumidor *heartbeat*. Como consumidor, é possível programar até 4 diferentes produtores para serem monitorados pelo inversor:

Índice	1016h
Nome	Consumer Heartbeat Time
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	-
Valor Padrão	4

Sub-índices	1 – 4
Descrição	Consumer Heartbeat Time 1 – 4
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0

Nos sub-índices de 1 até quatro, é possível programar o consumidor escrevendo um valor no seguinte formato:

UNSIGNED32		
Reservado (8 bits)	Node-ID (8 bits)	Heartbeat time (16 bits)

- Node-ID:** permite programar o Node-ID do produtor *heartbeat* o qual se deseja monitorar.
- Heartbeat time:** permite programar o tempo, em múltiplos de 1 milissegundo, até a detecção de erro, caso nenhuma mensagem do produtor seja recebida. O valor 0 (zero) neste campo desabilita o consumidor.

Depois de configurado, o consumidor *heartbeat* inicia a monitoração após o primeiro telegrama enviado pelo produtor. Caso seja detectado erro pelo fato do consumidor deixar de receber mensagens dos produtos *heartbeat*, o inversor irá automaticamente para o estado pré-operacional e indicará alarme A135 na sua IHM. A ocorrência deste erro também pode ser observada através do parâmetro P0721.

Como produtor, o CFW-11 possui um objeto para configuração deste serviço:

Índice	1017h
Nome	Producer Heartbeat Time
Objeto	VAR
Tipo	UNSIGNED16

Acesso	rw
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	0

O objeto 1017h permite programar o tempo em milissegundos no qual o produtor envie um telegrama *heartbeat* para a rede. Uma vez programado, o inversor inicia a transmissão de mensagens com o seguinte formato:

Identificador	byte 1	
	bit 7	bit 6 ... bit 0
700h + Node-ID	Sempre 0	Estado do escravo



NOTA!

- ☑ Este objeto está ativo mesmo no estado parado (ver tabela 5.8).
- ☑ O valor 0 (zero) em um dos objetos desabilita esta função.
- ☑ Depois de detectado o erro, caso o serviço seja habilitado mais uma vez, a indicação do erro é retirada da IHM.
- ☑ O valor de tempo programado para o consumidor deve ser maior do que o programado para o respectivo produtor. Recomenda-se inclusive programar o consumidor com valores múltiplos do utilizado para o produtor.
- ☑ Para cada escravo, somente um dos serviços – *Heartbeat* ou *Node Guarding* – pode ser habilitado.

5.7 Procedimento de Inicialização

Uma vez conhecido o funcionamento dos objetos disponíveis para o inversor de frequência CFW-11, é necessário agora programar os diferentes objetos para operarem em conjunto na rede. De forma geral, o procedimento para inicialização dos objetos em uma rede CANopen segue o descrito pelo fluxograma a seguir:

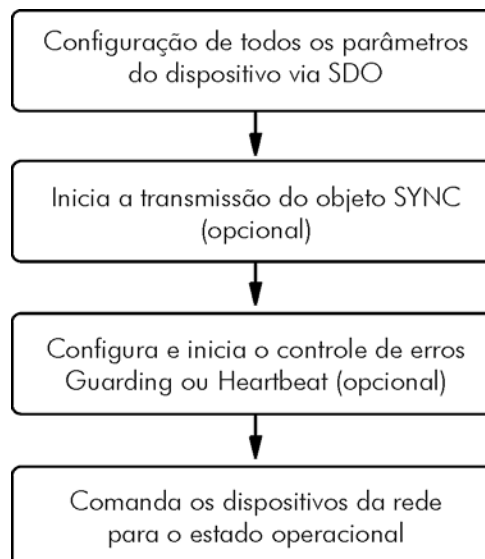


Figura 5-8 - Fluxograma do processo de inicialização

É necessário observar que os objetos de comunicação do CFW-11 (1000h até 1FFFh) não são armazenados na memória não volátil. Desta forma, sempre que for feito o reset ou desligado o equipamento, é necessário refazer a parametrização dos objetos de comunicação. Para os objetos específicos do fabricante (a partir de 2000h, que representam os parâmetros), estes são armazenados na memória não volátil e, portanto, podem ser parametrizados uma única vez.

6 Descrição dos Objetos para Drives

Neste item serão descritos os objetos comuns para drives, definidos pela especificação CANopen, no documento CiA DSP 402. Os objetos citados aqui possuem descrição e operação semelhantes, independente do fabricante do drive. Isto facilita a interoperabilidade e intercambiabilidade entre diferentes dispositivos.

A figura 6.7 mostra um diagrama com a arquitetura lógica de funcionamento de um drive através da rede CANopen, com os diferentes modos de operação definidos nesta especificação. Cada modo de operação possui um conjunto de objetos que permite a configuração e operação do drive na rede.

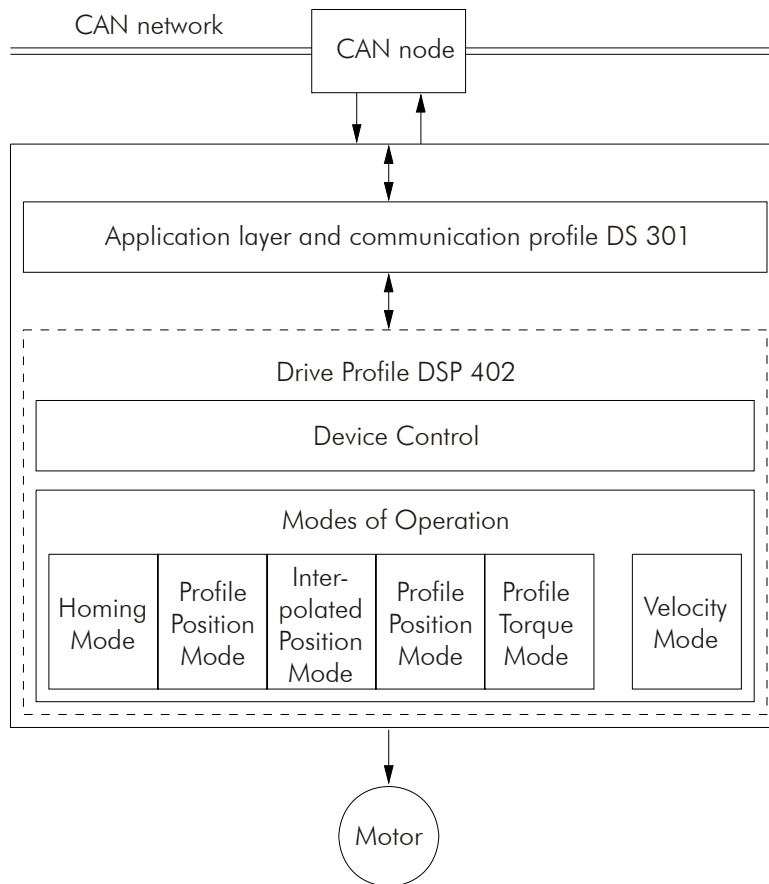


Figura 6-1 - Arquitetura de comunicação para um drive na rede CANopen

Para o CFW-11, somente o *Velocity Mode* é suportado. A Tabela 6.1 mostra a lista dos objetos disponíveis para o CFW-11, divididos de acordo com os diferentes modos de operação do inversor.

Tabela 6.1 - Lista de objetos do CFW-11 – Drive Profile

Índice	Objeto	Nome	Tipo	Acesso	Mapeável
Device Control					
6040h	VAR	Controlword	UNSIGNED16	rw	Sim
6041h	VAR	Statusword	UNSIGNED16	ro	Sim
6060h	VAR	Modes of Operation	INTEGER8	rw	Sim
6061h	VAR	Modes of Operation Display	INTEGER8	ro	Sim
Velocity Mode					
6042h	VAR	vl target velocity	INTEGER16	rw	Sim
6043h	VAR	vl velocity demand	INTEGER16	ro	Sim
6044h	VAR	vl control effort	INTEGER16	ro	Sim
6046h	ARRAY	vl velocity min max amount	UNSIGNED32	rw	Sim
6048h	RECORD	vl velocity acceleration	vl vel. accel. decl. record	rw	Sim
6049h	RECORD	vl velocity deceleration	vl vel. accel. decl. record	rw	Sim
Position Control Function					
6063h	VAR	Position actual value*	INTEGER32	Ro	Sim

Sempre que um objeto desta lista for lido ou escrito, o CFW-11 irá mapear suas funções nos parâmetros do inversor. Desta forma, ao operar o sistema através destes objetos, o valor dos parâmetros pode ser alterado de acordo com a função utilizada. Nos itens seguintes, é feita uma descrição detalhada de cada um destes objetos, onde são indicados os parâmetros do inversor que são utilizados para executar as funções destes objetos.

6.1 Device Control – Objetos para Controle do Drive

Todo o drive que opera em uma rede CANopen seguindo a especificação DSP 402 deve estar de acordo com o descrito pela seguinte máquina de estados:

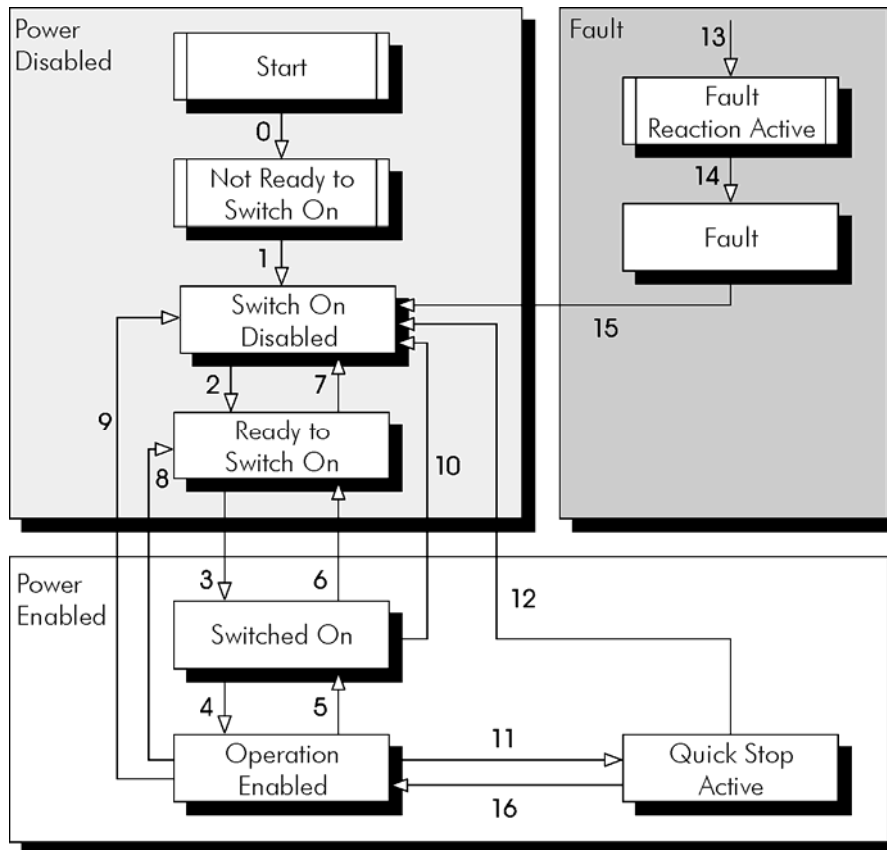


Figura 6-2 - Máquina de estado para drives

Descrição dos estados:

- ☑ **Not ready to switch on:** o inversor está inicializando, não pode ser comandado.
- ☑ **Switch on disabled:** inicialização completa, inversor pode receber comandos.
- ☑ **Ready to switch on:** comando para permitir a alimentação do drive foi recebido.
- ☑ **Switched on:** comando para energizar a potência do drive foi recebido.
- ☑ **Operation enabled:** o drive está habilitado, sendo controlado de acordo com o modo de operação programado. Potência está sendo aplicada ao motor.
- ☑ **Quick stop active:** durante a operação, o comando de *quick stop* foi recebido. Potência está sendo aplicada ao motor.
- ☑ **Fault reaction active:** uma falha ocorreu e o drive está executando a ação relativa ao tipo de erro.
- ☑ **Fault:** drive com erro. Função desabilitada, sem potência sendo aplicada no motor.



NOTA!

O inversor de frequência CFW-11 não possui chave para bloqueio / habilitação da alimentação de potência do equipamento. Desta forma, os estados descritos no grupo *Power disabled* foram implementados por questões de compatibilidade com a máquina de estados descrita, porém a alimentação de potência do dispositivo permanece ativa mesmo nestes estados.

Descrição das transições:

- ☑ **Transição 0:** O drive é ligado e inicia o procedimento de inicialização. A alimentação de potência do drive está ativa.
- ☑ **Transição 1:** Inicialização completa (automático).
- ☑ **Transição 2:** Comando *Shutdown* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo CFW-11.
- ☑ **Transição 3:** Comando *Switch on* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo CFW-11.
- ☑ **Transição 4:** Comando *Enable operation* recebido. O drive é habilitado. Equivale a ativar o bit 1 da palavra de controle via CAN do inversor – P0684.
- ☑ **Transição 5:** Comando *Disable operation* recebido. O drive é desabilitado. Equivale a zerar o bit 1 da palavra de controle via CAN do inversor – P0684.
- ☑ **Transição 6:** Comando *Shutdown* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo CFW-11.
- ☑ **Transição 7:** Comandos *Quick stop* e *Disable voltage* recebidos. É feita a transição de estados, mas nenhuma ação é tomada pelo CFW-11.
- ☑ **Transição 8:** Comando *Shutdown* recebido. Durante a operação do drive este é desabilitado, bloqueando a alimentação para o motor. Equivale a zerar o bit 1 da palavra de controle via CAN do inversor – P0684.
- ☑ **Transição 9:** Comando *Shutdown* recebido. Durante a operação do drive este é desabilitado, bloqueando a alimentação para o motor. Equivale a zerar o bit 1 da palavra de controle via CAN do inversor – P0684.
- ☑ **Transição 10:** Comando *Quick stop* ou *Disable voltage* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo CFW-11.
- ☑ **Transição 11:** Comando *Quick stop* recebido. Inversor executa a função de parada por rampa. Equivale a zerar o bit 0 da palavra de controle via CAN do inversor – P0684.
- ☑ **Transição 12:** Comando *Disable voltage* recebido. O drive é desabilitado. Equivale a zerar o bit 1 da palavra de controle via CAN do inversor – P0684..
- ☑ **Transição 13:** Erro é detectado e o drive é desabilitado.
- ☑ **Transição 14:** Depois de desabilitar o drive, ele vai para o estado de erro (automático).
- ☑ **Transição 15:** Comando *Fault reset* recebido. O inversor executa o reset do erro e retorna para o estado desabilitado e sem falha.
- ☑ **Transição 16:** Comando *Enable operation* recebido. Inversor executa a função de partida por rampa. Equivale a ativar o bit 0 da palavra de controle via CAN do inversor – P0684.

Esta máquina de estado é controlada pelo objeto 6040h, e os estados podem ser monitorados pelo objeto 6041h. Ambos os objetos são descritos a seguir.

6.1.1 Objeto 6040h – Controlword

Controla o estado do inversor.

Índice	6040h
Nome	Controlword
Objeto	VAR
Tipo	UNSIGNED16
Parâmetros utilizados	P0684

Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED16
Valor Padrão	-

Os bits desta palavra possuem as seguintes funções:

15 – 9	8	7	6 – 4	3	2	1	0
Reservado	Halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch on

Os bits 0, 1, 2, 3 e 7 permitem controlar a máquina de estados do drive. Os comandos para transição de estados são dados através das combinações de bits indicadas na Tabela 6.2. Os bits marcados com 'x' são irrelevantes para a execução do comando.

Tabela 6.2 - Comandos da palavra de controle

Comando	Bits da palavra de controle					Transições
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	x	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Disable voltage	0	x	x	0	x	7, 9, 10, 12
Quick stop	0	x	0	1	x	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0 → 1	x	x	x	x	15

Os bits 4, 5, 6 e 8 possuem diferentes funções de acordo com o modo de operação utilizado. A descrição detalhada da função destes bits para o modo velocidade (velocity mode) está descrita no item 6.2.1.

6.1.2 Objeto 6041h – Statusword

Indica o estado atual do CFW-11.

Índice	6041h
Nome	Statusword
Objeto	VAR
Tipo	UNSIGNED16
Parâmetros utilizados	PO680
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16
Valor Padrão	-

Os bits desta palavra possuem as seguintes funções:

Tabela 6.3 - Função dos bits da palavra de estado (Statusword)

Bit	Descrição
0	Ready to switch on
1	Switched on
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7	Warning
8	Reservado
9	Remote
10	Target reached
11	Internal limit active
12 – 13	Operation mode specific
14 – 15	Reservado

Nesta palavra, os bits 0, 1, 2, 3, 5 e 6 indicam o estado do dispositivo de acordo com a máquina de estados descrita na figura 6.2. A Tabela 6.4 descreve as combinações destes bits para indicação dos estados. Os bits marcados com 'x' são irrelevantes para a indicação do estado.

Tabela 6.4 - Estados do drive indicados através da palavra de estado

Valor (binário)	Estado
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Demais bits indicam uma condição específica para o drive.

- ☑ **Bit 4 – Voltage enabled:** indica que a potência do drive está sendo alimentada.
- ☑ **Bit 7 – Warning:** Não utilizado para o CFW-11.
- ☑ **Bit 9 – Remote:** indica quando o drive está no modo remoto e aceita comandos via rede CANopen⁹. Representa o valor do bit 4 da palavra de estado – P0680.
- ☑ **Bit 10 – Target reached:** indica quando o drive está operando no valor da referência, que depende do modo de operação utilizado. Também é colocado em 1 quando as funções *quick stop* ou *halt* são acionadas.
- ☑ **Bit 11 – Internal limit active:** indica quando o drive está operando em um valor limite interno, como limite de corrente.
- ☑ **Bits 12 e 13 – Operation mode specific:** dependem do modo de operação do drive.

6.1.3 Objeto 6060h – Modes of Operation

Permite programar o modo de operação do CFW-11.

Índice	6060h
Nome	Modes of operation
Objeto	VAR
Tipo	INTEGER8
Parâmetros utilizados	-

Acesso	rw
Mapeável	Sim
Faixa	INTEGER8
Valor Padrão	-

O único modo suportado pelo inversor CFW-11 é o *Velocity Mode*, representado pelo valor 2.

6.1.4 Objeto 6061h – Modes of Operation Display

Indica o modo de operação do CFW-11.

Índice	6061h
Nome	Modes of operation display
Objeto	VAR
Tipo	INTEGER8
Parâmetros utilizados	-

Acesso	rw
Mapeável	Sim
Faixa	INTEGER8
Valor Padrão	-

O único modo suportado pelo inversor CFW-11 é o *Velocity Mode*, representado pelo valor 2.

⁹ Depende da programação do inversor.

6.2 Velocity Mode – Objetos para Controle do Drive

Este modo de operação permite o controle do inversor de forma simples, disponibilizando funções do tipo:

- Cálculo do valor de referência.
- Captura e monitoração da velocidade.
- Limitação de velocidade.
- Rampas de velocidade, dentre outras funções.

Estas funções são executadas com base em um conjunto de objetos para configuração deste modo de operação.

6.2.1 Bits de Controle e Estado

Os bits 4, 5, 6 e 8 da palavra de controle (objeto 6040h – Controlword) possuem as seguintes funções no modo velocidade:

Bit	Nome	Valor	Descrição
4	<i>rfg enable</i> (habilita rampa)	0	Desabilita Rampa – bit 0 do P0684 = 0
		1	Habilita Rampa – bit 0 do P0684 = 1
5	Reservado		
6	Reservado		
8	<i>Halt</i>	0	Habilita Rampa – bit 0 do P0684 = 1
		1	Desabilita Rampa – bit 0 do P0684 = 0

Para que o motor gire de acordo com a rampa de aceleração, é necessário que o bit 4 seja ativado.

Para a palavra de status, os bits específicos do modo de operação (bits 12 e 13) são reservados para uso futuro.

6.2.2 Objeto 6042h – *vl Target Velocity*

Permite programar o valor da referência de velocidade para o inversor, em rpm:

Índice	6042h
Nome	<i>vl target velocity</i>
Objeto	VAR
Tipo	INTEGER16
Parâmetros utilizados	P0684, P0685

Acesso	rw
Mapeável	Sim
Faixa	INTEGER16
Valor Padrão	0

O objeto *vl target velocity* permite a escrita de valores negativos para indicar referência de velocidade para o motor no sentido de giro anti-horário. Esta alteração no sentido de giro é executada através da escrita do bit 2 na palavra de comando via CAN – P0684.

6.2.3 Objeto 6043h – *vl Velocity Demand*

Indica o valor da referência de velocidade após a rampa, em rpm:

Índice	6043h
Nome	<i>vl velocity demand</i>
Objeto	VAR
Tipo	INTEGER16
Parâmetros utilizados	-

Acesso	ro
Mapeável	Sim
Faixa	INTEGER16
Valor Padrão	-

6.2.4 Objeto 6044h – vl Control Effort

Indica o valor da velocidade de acordo com o medido para o motor, em rpm. Para os modos de controle sem realimentação, este objeto possui o mesmo valor que o objeto 6043h.

Índice	6044h
Nome	vl control effort
Objeto	VAR
Tipo	INTEGER16
Parâmetros utilizados	P0681

Acesso	ro
Mapeável	Sim
Faixa	INTEGER16
Valor Padrão	-

6.2.5 Objeto 6046h – vl Velocity Min Max Amount

Permite programar o valor da velocidade mínima e máxima para o inversor. Somente valores positivos são aceitos, porém os valores programados também são válidos para o sentido de giro anti-horário. Os valores são escritos em rpm.

Índice	6046h
Nome	vl velocity min max amount
Objeto	ARRAY
Tipo	UNSIGNED32
Parâmetros utilizados	P0133, P0134

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	2
Valor Padrão	2

Sub-índice	1
Descrição	vl velocity min amount
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

Sub-índice	2
Descrição	vl velocity max amount
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

6.2.6 Objeto 6048h – vl Velocity Acceleration

Permite programar a rampa de aceleração do inversor.

Índice	6048h
Nome	vl velocity acceleration
Objeto	RECORD
Tipo	VI velocity acceleration deceleration record
Parâmetros utilizados	P100

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	2
Valor Padrão	2

Sub-índice	1
Descrição	Delta speed
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

Sub-índice	2
Descrição	Delta time
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

O valor da aceleração é calculado dividindo-se a velocidade programada no sub-índice 1 pelo tempo programado no sub-índice 2. Os valores programados devem respeitar a faixa de valores para o parâmetro P100.

6.2.7 Objeto 6049h – vl Velocity Deceleration

Permite programar a rampa de desaceleração do inversor.

Índice	6049h
Nome	vl velocity deceleration
Objeto	RECORD
Tipo	Vl velocity acceleration deceleration record
Parâmetros utilizados	P100

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	2
Valor Padrão	2

Sub-índice	1
Descrição	Delta speed
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

Sub-índice	2
Descrição	Delta time
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	-

O valor da desaceleração é calculado dividindo-se a velocidade programada no sub-índice 1 pelo tempo programado no sub-índice 2. Os valores programados devem respeitar a faixa de valores para o parâmetro P101.

Position control function – Objetos para controle de posição

Apesar de não suportar o modo de operação para controle de posição, foram disponibilizados objetos relativos a esta função no inversor CFW-11.

Objeto 6063h – Position actual value*

Informa o valor do contador de pulsos do encoder. Este contador só é válido caso o inversor possua acessório para interface com encoder. O valor deste objeto é incrementado até atingir o número de pulsos do encoder, retornando para zero ao atingir este valor.

Índice	6063h
Nome	Position actual value
Objeto	VAR
Tipo	INTEGER32
Parâmetros utilizados	P0681

Acesso	ro
Mapeável	Sim
Faixa	0 até o número de pulsos do encoder menos 1
Valor Padrão	-

Para mais informações sobre a operação desta função, consulte os parâmetros P0039, P0191 e P0405 no manual de programação do inversor CFW-11.

7 Falhas e Alarmes Relacionados com a Comunicação CANopen

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 for detectada a falta de alimentação na interface CAN, será sinalizada através da HMI a mensagem de alarme A133 – ou falha F229, 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.

Possíveis Causas/Correção:

- Medir se existe tensão entre os pinos 1 e 5 do conector da interface CAN, conforme indicado na tabela 2.2.
- 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 elevado¹⁰, 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 inversor, 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.

A135/F235 –Node Guarding/Heartbeat

Descrição:

Controle de erros da comunicação CANopen detectou erro de comunicação utilizando o mecanismo de *guarding*.

Atuação:

Utilizando os mecanismos de controle de erro – *Node Guarding* ou *Heratbeat* – o mestre e o escravo podem trocar telegramas periódicos, em um período pré-determinado. Caso a comunicação seja interrompida por algum motivo, tanto mestre quanto escravo poderão detectar erro na comunicação pelo *timeout* na troca destas mensagens.

Neste caso será sinalizada através da HMI a mensagem de alarme A135 – ou falha F235, dependendo da programação feita no P0313. Para alarmes, caso este controle de erros seja habilitado novamente, a indicação de alarme será retirada da HMI.

¹⁰ Para mais informações sobre detecção de erros, consultar especificação CAN.

Possíveis Causas/Correção:

- Verificar os tempos programados no mestre e no escravo para troca de mensagens. Para evitar problemas devido a atrasos na transmissão e diferenças na contagem dos tempos, recomenda-se que os valores programados para troca de mensagens no mestre seja um pouco menor que os tempos programados para detecção de erros pelo escravo.
- Verificar se o mestre está enviando os telegramas de *guarding* no tempo programado.
- Verificar problemas na comunicação que possam ocasionar perda de telegramas ou atrasos na transmissão.