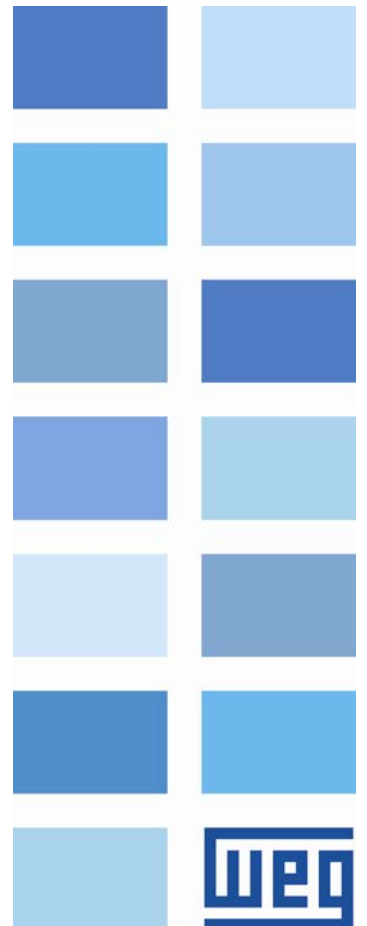


# CANopen

MW500

**Manual do Usuário**





# **Manual do Usuário CANopen**

Série: MW500

Idioma: Português

N ° do Documento: 10002252901 / 00

Data da Publicação: 06/2013

# SUMÁRIO

<b>SUMÁRIO.....</b>	<b>3</b>
<b>SOBRE O MANUAL.....</b>	<b>6</b>
<b>ABREVIações E DEFINIções.....</b>	<b>6</b>
<b>REPRESENTAção NUMÉRICA.....</b>	<b>6</b>
<b>DOCUMENTOS.....</b>	<b>6</b>
<b>1 INTRODUção À COMUNICAção CANOPEN.....</b>	<b>7</b>
<b>1.1 CAN.....</b>	<b>7</b>
1.1.1 Frame de Dados.....	7
1.1.2 Frame Remoto.....	7
1.1.3 Acesso à Rede.....	7
1.1.4 Controle de Erros.....	7
1.1.5 CAN e CANopen.....	8
<b>1.2 CARACTERÍSTICAS DA REDE CANOPEN.....</b>	<b>8</b>
<b>1.3 MEIO FÍSICO.....</b>	<b>8</b>
<b>1.4 ENDEREçO NA REDE CANOPEN.....</b>	<b>8</b>
<b>1.5 ACESSO AOS DADOS.....</b>	<b>8</b>
<b>1.6 TRANSMISSÃO DE DADOS.....</b>	<b>8</b>
<b>1.7 OBJETOS RESPONSÁVEIS PELA COMUNICAção - COBS.....</b>	<b>9</b>
<b>1.8 COB-ID.....</b>	<b>9</b>
<b>1.9 ARQUIVO EDS.....</b>	<b>10</b>
<b>2 ACESSÓRIOS PARA COMUNICAção CANOPEN.....</b>	<b>11</b>
<b>2.1 MÓDULO PLUG-IN CFW500-CCAN.....</b>	<b>11</b>
<b>2.2 PINAGEM DO CONECTOR.....</b>	<b>11</b>
<b>2.3 FONTE DE ALIMENTAção.....</b>	<b>11</b>
<b>2.4 INDICAções.....</b>	<b>11</b>
<b>3 INSTALAção DA REDE CANOPEN.....</b>	<b>12</b>
<b>3.1 TAXA DE COMUNICAção.....</b>	<b>12</b>
<b>3.2 ENDEREçO NA REDE CANOPEN.....</b>	<b>12</b>
<b>3.3 RESISTORES DE TERMINAção.....</b>	<b>12</b>
<b>3.4 CABO.....</b>	<b>12</b>
<b>3.5 LIGAção NA REDE.....</b>	<b>13</b>
<b>4 PARAMETRIZAção.....</b>	<b>14</b>
<b>4.1 SÍMBOLOS PARA DESCRIção DAS PROPRIEDADES.....</b>	<b>14</b>
P0105 – SELEção 1ª/2ª RAMPA.....	14
P0220 – SELEção FONTE LOCAL/REMOTO.....	14
P0221 – SELEção REFERência LOCAL.....	14
P0222 – SELEção REFERência REMOTA.....	14
P0223 – SELEção GIRO LOCAL.....	14
P0224 – SELEção GIRA/PARA LOCAL.....	14
P0225 – SELEção JOG LOCAL.....	14
P0226 – SELEção GIRO REMOTO.....	14
P0227 – SELEção GIRA/PARA REMOTO.....	14
P0228 – SELEção JOG REMOTO.....	14
P0313 – AÇÃO PARA ERRO DE COMUNICAção.....	14
P0680 – ESTADO LÓGICO.....	15
P0681 – VELOCIDADE DO MOTOR EM 13 BITS.....	16
P0684 – PALAVRA DE CONTROLE VIA CANOPEN.....	17

P0685 – REFERÊNCIA DE VELOCIDADE VIA CANOPEN .....	18
P0695 – VALOR PARA AS SAÍDAS DIGITAIS.....	18
P0696 – VALOR 1 PARA SAÍDAS ANALÓGICAS .....	19
P0697 – VALOR 2 PARA SAÍDAS ANALÓGICAS .....	19
P0698 – VALOR 3 PARA SAÍDAS ANALÓGICAS .....	19
P0700 – PROTOCOLO CAN.....	20
P0701 – ENDEREÇO CAN.....	20
P0702 – TAXA DE COMUNICAÇÃO CAN .....	20
P0703 – RESET DE BUS OFF .....	21
P0705 – ESTADO DO CONTROLADOR CAN .....	21
P0706 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS.....	22
P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS .....	22
P0708 – CONTADOR DE ERROS DE BUS OFF.....	22
P0709 – CONTADOR DE MENSAGENS CAN PERDIDAS .....	22
P0721 – ESTADO DA COMUNICAÇÃO CANOPEN .....	23
P0722 – ESTADO DO NÓ CANOPEN .....	23
<b>5 DICIONÁRIO DE OBJETOS .....</b>	<b>24</b>
5.1 ESTRUTURA DO DICIONÁRIO .....	24
5.2 TIPOS DE DADOS.....	24
5.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICAÇÃO .....	24
5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DO FABRICANTE.....	25
5.5 DEVICE PROFILE – OBJETOS COMUNS PARA DRIVES.....	25
<b>6 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO.....</b>	<b>27</b>
6.1 OBJETOS DE IDENTIFICAÇÃO .....	27
6.1.1 Objeto 1000h – Device Type .....	27
6.1.2 Objeto 1001h – Error Register .....	27
6.1.3 Objeto 1018h – Identity Object .....	27
6.2 SERVICE DATA OBJECTS – SDOS.....	28
6.2.1 Objeto 1200h – Servidor SDO .....	29
6.2.2 Funcionamento dos SDOs .....	29
6.3 PROCESS DATA OBJECTS – PDOS.....	30
6.3.1 Objetos Mapeáveis para os PDOs .....	31
6.3.2 PDOs de Recepção.....	31
6.3.3 PDOs de Transmissão.....	33
6.4 SYNCHRONIZATION OBJECT – SYNC .....	36
6.5 NETWORK MANAGEMENT – NMT .....	36
6.5.1 Controle dos Estados do Escravo .....	37
6.5.2 Controle de Erros – Node Guarding.....	38
6.5.3 Controle de Erros – Heartbeat.....	39
6.6 PROCEDIMENTO DE INICIALIZAÇÃO .....	41
<b>7 DESCRIÇÃO DOS OBJETOS PARA DRIVES.....</b>	<b>43</b>
7.1 DEVICE CONTROL – OBJETOS PARA CONTROLE DO DRIVE .....	44
7.1.1 Objeto 6040h – Controlword .....	45
7.1.2 Objeto 6041h – Statusword.....	46
7.1.3 Objeto 6060h – Modes of Operation .....	47
7.1.4 Objeto 6061h – Modes of Operation Display.....	47
7.2 VELOCITY MODE – OBJETOS PARA CONTROLE DO DRIVE.....	47
7.2.1 Bits de Controle e Estado .....	48
7.2.2 Objeto 6042h – vl Target Velocity .....	48
7.2.3 Objeto 6043h – vl Velocity Demand.....	48
7.2.4 Objeto 6044h – vl Control Effort .....	48
7.2.5 Objeto 6046h – vl Velocity Min Max Amount.....	49
7.2.6 Objeto 6048h – vl Velocity Acceleration .....	49
7.2.7 Objeto 6049h – vl Velocity Deceleration .....	50

<b>8 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO CANOPEN .....</b>	<b>51</b>
A133/F233 – SEM ALIMENTAÇÃO NA INTERFACE CAN .....	51
A134/ F234 – BUS OFF .....	51
A135/F235 – NODE GUARDING/HEARTBEAT .....	51

## **SOBRE O MANUAL**

Este manual fornece a descrição necessária para a operação do inversor de frequência MW500 utilizando o protocolo CANopen.

### **ABREVIações E DEFINIções**

<b>CAN</b>	Controller Area Network
<b>CiA</b>	CAN in Automation
<b>COB</b>	Communication Object
<b>COB-ID</b>	Communication Object Identifier
<b>SDO</b>	Service Data Object
<b>PDO</b>	Process Data Object
<b>RPDO</b>	Receive PDO
<b>TPDO</b>	Transmit PDO
<b>NMT</b>	Network Management Object
<b>ro</b>	Read only (somente leitura)
<b>rw</b>	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 foi desenvolvido baseado nas seguintes especificações e documentos:

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 de um equipamento em rede CANopen, é necessário conhecer a forma como a comunicação é feita. Este item traz uma descrição geral do funcionamento do protocolo CANopen, contendo as funções utilizadas pelo MW500. Para uma descrição mais detalhada pode-se consultar a especificação do protocolo.

## 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)<sup>1</sup>. 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 bits<sup>2</sup> (arbitration field), e um campo de dados (data field), que pode conter até 8 bytes de dados.

<b>Identificador</b>	<b>8 bytes de dados</b>							
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 aos demais elementos que um erro foi detectado.

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

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

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

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

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

### 1.1.5 CAN e CANopen

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

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

## 1.2 CARACTERÍSTICAS DA REDE 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.

### 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.

Para o inversor de frequência MW500, o endereço do escravo é programado através do parâmetro P0701.

### 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.

### 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					
	Inteiro 32 bits				Inteiro 16 bits	
	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
11 bits	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. Os seguintes objetos de comunicação (COBs) são descritos pela especificação:

*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.
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.).
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.
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. Dentre os serviços descritos pelo protocolo estão: serviços de controle do dispositivo, onde o mestre controla o estado de cada escravo na rede, e serviços de controle de erros (Node Guarding e Heartbeat), onde o dispositivo envia mensagens periódicas para informar que a conexão está ativa.

Toda a comunicação do escravo com a rede é feita utilizando estes objetos, e os dados que podem ser acessados são os existentes no dicionário de objetos do dispositivo.

## 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 descritos no protocolo. 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

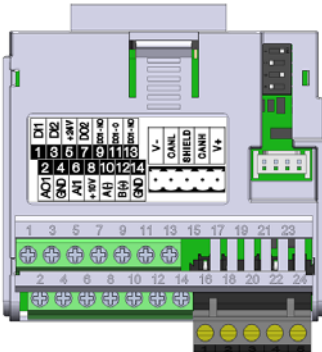
Cada dispositivo em uma rede CANopen possui um arquivo de configuração EDS, que contém informações sobre o funcionamento do dispositivo na rede CANopen, bem como a descrição de todos os objetos existentes para comunicação. Em geral este arquivo é utilizado por um mestre ou software de configuração, para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

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

## 2 ACESSÓRIOS PARA COMUNICAÇÃO CANOPEN

Para possibilitar a comunicação CANopen no produto, é necessário utilizar o módulo plug-in com a interface CAN descrito a seguir. Informações sobre a instalação destes módulos podem ser obtidas no guia que acompanha cada acessório.

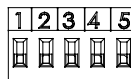
### 2.1 MÓDULO PLUG-IN CFW500-CCAN



- Item WEG: 11593087
- Composto pelo módulo plug-in de comunicação CAN (figura ao lado) e um guia de montagem.
- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Alimentação externa de 24V.
- Permite a conexão de até 64 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada com o uso de repetidores<sup>3</sup>.
- Comprimento máximo do barramento de 1000 metros.

### 2.2 PINAGEM DO CONECTOR

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



**Tabela 2.1:** Pinagem do conector 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 do cabo
4	CAN_H	Sinal de comunicação CAN_H
5	V+	Pólo positivo da fonte de alimentação



#### NOTA!

Recomenda-se o aterramento do pino de GND do módulo CFW500-CCAN. Este aterramento é necessário para que a blindagem do cabo possua uma ligação com o terra.

### 2.3 FONTE DE ALIMENTAÇÃO

O módulo plug-in CFW500-CCAN necessita de uma tensão de alimentação externa entre os pinos 1 e 5 do conector da rede. Os dados para consumo individual e tensão de entrada são apresentados na tabela a seguir.

**Tabela 2.2:** Características da alimentação para interface CAN

Tensão de alimentação (VCC)		
Mínimo	Máximo	Recomendado
11	30	24
Corrente (mA)		
Típico	Máximo	
30	50	

### 2.4 INDICAÇÕES

As indicações de alarmes, falhas e estados da comunicação são feitas através da HMI e dos parâmetros do produto.

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

### 3 INSTALAÇÃO DA REDE CANOPEN

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

#### 3.1 TAXA DE COMUNICAÇÃO

Equipamentos com interface CANopen em geral permitem configurar a taxa de comunicação desejada, podendo variar de 10Kbit/s até 1Mbit/s. A taxa de comunicação (*baud rate*) que pode ser utilizada por um equipamento também depende do comprimento do cabo utilizado na instalação. A tabela a seguir apresenta a relação entre as taxas de comunicação e o comprimento máximo de cabo que pode ser utilizado na instalação, de acordo com o recomendado pela CiA<sup>4</sup>.

*Tabela 3.1: Taxas de comunicação suportadas e comprimento do cabo*

Taxa de comunicação	Comprimento do cabo
1 Mbit/s	25 m
800 Kbit/s	50 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

Todos os equipamentos da rede devem ser programados para utilizar a mesma taxa de comunicação. Para o inversor de frequência MW500, a taxa de comunicação é programada através do parâmetro P0702.

#### 3.2 ENDEREÇO NA REDE CANOPEN

Todo dispositivo na rede CANopen deve possuir um endereço, ou Node ID, entre 1 e 127. Este endereço precisa ser diferente para cada equipamento. Para o inversor de frequência MW500, o endereço do equipamento é programado através do parâmetro P0701.

#### 3.3 RESISTORES DE TERMINAÇÃO

A utilização de resistores de terminação nas extremidades do barramento CAN é fundamental para evitar reflexão de linha, que pode prejudicar o sinal transmitido e ocasionar erros na comunicação. Resistores de terminação no valor de 121Ω / 0.25W devem ser conectados entre os sinais CAN\_H e CAN\_L nas extremidades do barramento principal.

#### 3.4 CABO

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

*Tabela 3.2: Características do cabo para rede CANopen*

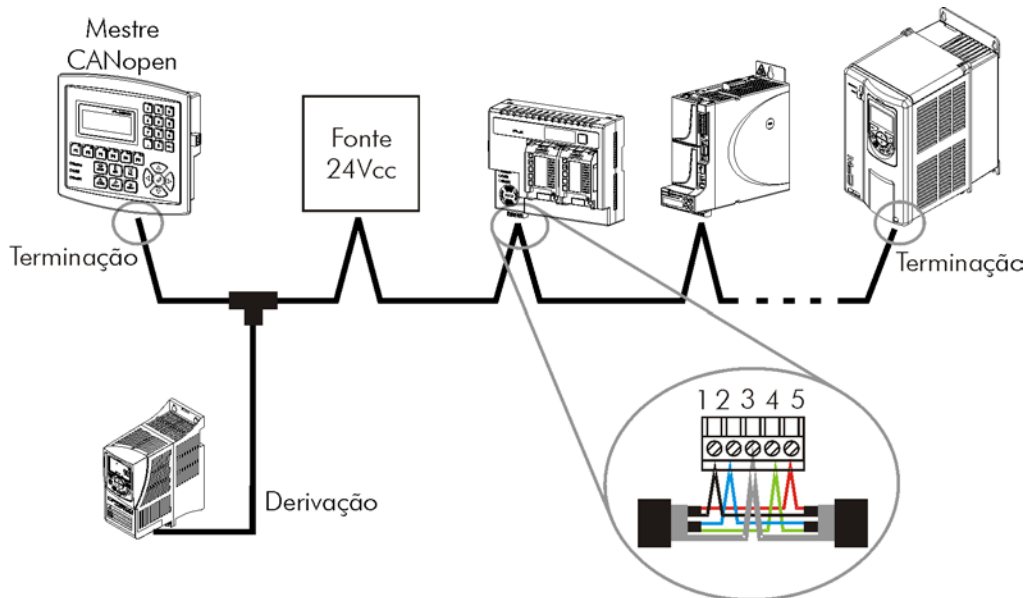
Comprimento do cabo (m)	Resistência por metro (mOhm/m)	Área do condutor (mm <sup>2</sup> )
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

Também é necessária a utilização de um par trançado adicional para levar a alimentação de 24Vcc para os equipamentos que necessitam deste sinal. Recomenda-se utilizar um cabo certificado para rede DeviceNet.

<sup>4</sup> Diferentes produtos podem apresentar variações no comprimento máximo do cabo suportado para a instalação.

## 3.5 LIGAÇÃO NA REDE

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



*Figura 3.1: Exemplo de instalação em rede CANopen*

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

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

## 4 PARAMETRIZAÇÃO

A seguir serão apresentados apenas os parâmetros do inversor de frequência MW500 que possuem relação direta com a comunicação CANopen.

### 4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES

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

#### P0105 – SELEÇÃO 1ª/2ª RAMPA

#### P0220 – SELEÇÃO FONTE LOCAL/REMOTO

#### P0221 – SELEÇÃO REFERÊNCIA LOCAL

#### P0222 – SELEÇÃO REFERÊNCIA REMOTA

#### P0223 – SELEÇÃO GIRO LOCAL

#### P0224 – SELEÇÃO GIRA/PARA LOCAL

#### P0225 – SELEÇÃO JOG LOCAL

#### P0226 – SELEÇÃO GIRO REMOTO

#### P0227 – SELEÇÃO GIRA/PARA REMOTO

#### P0228 – SELEÇÃO JOG REMOTO

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

A descrição detalhada destes parâmetros encontra-se no manual de programação do inversor de frequência MW500.

#### P0313 – AÇÃO PARA ERRO DE COMUNICAÇÃO

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

#### Descrição:

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

**Tabela 4.1: Opções para o parâmetro P0313**

Opção	Descrição
0 = Inativo	Nenhuma ação é tomada, equipamento 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 equipamento é desabilitado geral, e o motor para por inércia.
3 = Vai para Local	O equipamento é comandado para o modo local.
4 = Vai para Local e mantém comandos e referência	O equipamento é comandado para o modo local, mas os comandos de habilitação e a referência de velocidade recebidos via rede são mantidos em modo local, desde que o equipamento seja programado para utilizar, em modo local, comandos via HMI ou Start/Stop a 3 fios, 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 equipamento, sendo necessário fazer o reset de falhas do equipamento para o retorno da sua operação normal.

São considerados erros de comunicação os seguintes eventos:

Comunicação CANopen/DeviceNet

- Alarme A133/Falha F233: sem alimentação na interface CAN.
- Alarme A134/Falha F234: *bus off*.
- Alarme A135/Falha F235: erro de comunicação CANopen (*Node Guarding/Heartbeat*).
- Alarme A136/Falha F236: mestre da rede DeviceNet em modo *Idle*.
- Alarme A137/Falha F237: ocorreu *timeout* em uma ou mais conexões I/O DeviceNet.

As ações descritas neste parâmetro são executadas através da escrita automática dos respectivos bits no parâmetro de controle da interface de rede que corresponde à falha detectada. Desta forma, para que os comandos escritos neste parâmetro tenham efeito, é necessário que o equipamento esteja programado para ser controlado pela interface de rede utilizada (com exceção da opção “Causa Falha”, que bloqueia o equipamento mesmo que ele não seja controlado via rede). Esta programação é feita através dos parâmetros P0220 até P0228.

### P0680 – ESTADO LÓGICO

**Faixa de** 0000h a FFFFh

**Padrão:** -

**Valores:**

**Propriedades:** RO

**Grupo de acesso via HMI:** NET

**Descrição:**

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

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

**Tabela 4.2: Funções dos bits para o parâmetro P0680**

Bits	Valores
Bits 0 a 3	Reservado.
Bit 4 Parada Rápida Ativa	0: Drive não possui comando de parada rápida ativo. 1: Drive está executando o comando de parada rápida.
Bit 5 Segunda Rampa	0: Drive está configurado para utilizar como rampa de aceleração e desaceleração para o motor a primeira rampa, programada nos parâmetros P0100 e P0101. 1: Drive 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: Drive operando normalmente. 1: Drive em modo de configuração. Indica uma condição especial na qual o drive não pode ser habilitado: Executando rotina de auto-ajuste. Executando rotina de start-up orientado. Executando função copy da HMI. Executando rotina auto-guiada do cartão de memória flash. Possui incompatibilidade de parametrização. Sem alimentação no circuito de potência do drive.
Bit 7 Em Alarme	0: Drive não está no estado de alarme. 1: Drive está no estado de alarme. Obs.: o número do alarme pode ser lido através do parâmetro P0048 – Alarme Atual.
Bit 8 Motor Girando	0: Motor está parado. 1: Drive está girando o motor à velocidade de referência, ou executando rampa de aceleração ou desaceleração.
Bit 9 Habilitado Geral	0: Drive está desabilitado geral. 1: Drive 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: Drive em modo local. 1: Drive em modo remoto.
Bit 13 Subtensão	0: Sem subtensão. 1: Com subtensão.
Bit 14 Automático (PID)	0: Em modo manual (função PID). 1: Em modo automático (função PID).
Bit 15 Em Falha	0: Drive não está no estado de falha. 1: Alguma falha registrada pelo drive. Obs.: O número da falha pode ser lido através do parâmetro P0049 – Falha Atual.

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

**Padrão:** -

**Valores:**
**Propriedades:** RO

**Grupo de acesso via HMI:** NET

**Descrição:**

Permite monitorar a velocidade do motor. Esta palavra utiliza resolução de 13 bits com sinal para representar a frequência nominal (P0403) do motor:

- P0681 = 0000h (0 decimal) → velocidade do motor = 0
- P0681 = 2000h (8192 decimal) → velocidade do motor = frequência nominal

Valores de velocidade intermediários ou superiores podem ser obtidos utilizando esta escala. Por exemplo, 60 Hz de frequência nominal, caso o valor lido seja 2048 (0800h), para obter o valor em Hz deve-se calcular:

8192 => 60 Hz 2048 => Frequência em Hz
---

Frequência em Hz = $\frac{60 \times 2048}{8192}$
--

Frequência em Hz = 30 Hz
--------------------------

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




**NOTA!**

Os valores transmitidos via rede apresentam uma limitação na escala utilizada, permitindo que no máximo seja indicada uma velocidade de 4 vezes a velocidade síncrona do motor, saturando em 32767 (ou -32768).

**P0684 – PALAVRA DE CONTROLE VIA CANOPEN**

<b>Faixa de</b>	0000h a FFFFh	<b>Padrão:</b> 0000h
<b>Valores:</b>		
<b>Propiedades:</b>	-	
<b>Grupo de acesso via HMI:</b>	NET	

**Descrição:**

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

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

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

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

*Tabela 4.3: 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 drive, interrompendo a alimentação para o motor. 1: Habilita geral o drive, permitindo a operação do motor.
Bit 2 Sentido de Giro	0: Sentido de giro do motor oposto ao da referência (sentido reverso). 1: Sentido de giro do motor igual ao da referência (sentido direto).
Bit 3 JOG	0: Desabilita a função JOG. 1: Habilita a função JOG.
Bit 4 LOC/REM	0: Drive vai para o modo local. 1: Drive vai para o modo remoto.
Bit 5 Utiliza Segunda Rampa	0: Drive utiliza como rampa de aceleração e desaceleração do motor os tempos da primeira rampa, programada nos parâmetros P0100 e P0101. 1: Drive 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 drive.
Bits 8 a 15	Reservado.

**P0685 – REFERÊNCIA DE VELOCIDADE VIA CANOPEN**

<b>Faixa de</b>	-32768 a 32767	<b>Padrão:</b> 0
<b>Valores:</b>		
<b>Propiedades:</b>	-	
<b>Grupo de acesso via HMI:</b>	NET	

**Descrição:**

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

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

Esta palavra utiliza resolução de 13 bits com sinal para representar a frequência nominal (P0403) do motor:

- P0685 = 0000h (0 decimal) → referência de velocidade = 0
- P0685 = 2000h (8192 decimal) → referência de velocidade = frequência nominal (P0403)

Valores de velocidade intermediários ou superiores podem ser obtidos utilizando esta escala. Por exemplo, 60 Hz de frequência nominal, caso deseje-se uma referência de 30 Hz, deve-se calcular:

60 Hz => 8192
30 Hz => Referência em 13 bits

Referência em 13 bits = $\frac{30 \times 8192}{60}$
---

Referência em 13 bits = 4096	=> Valor correspondente a 30 Hz 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


**NOTA!**

Os valores transmitidos via rede apresentam uma limitação devido à escala utilizada, permitindo que no máximo seja programado uma referência de velocidade de 4 vezes a velocidade síncrona do motor.

**P0695 – VALOR PARA AS SAÍDAS DIGITAIS**

<b>Faixa de</b>	0000h a 001Fh	<b>Padrão:</b> 0000h
<b>Valores:</b>		
<b>Propiedades:</b>	-	
<b>Grupo de acesso via HMI:</b>	NET	

**Descrição:**

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

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

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

**Tabela 4.4:** Funções dos bits para o parâmetro P0695

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



**NOTA!**

Algumas saídas digitais podem não estar disponíveis dependendo do módulo plug-in utilizado.

**P0696 – VALOR 1 PARA SAÍDAS ANALÓGICAS**

**P0697 – VALOR 2 PARA SAÍDAS ANALÓGICAS**

**P0698 – VALOR 3 PARA SAÍDAS ANALÓGICAS**

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

**Descrição:**

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

O valor escrito nestes parâmetros é utilizado como valor para a saída analógica, desde que a função da saída analógica desejada seja programada para “Conteúdo P0696 / P0697 / P0698”, nos parâmetros P0251, P0254, P0257.

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

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

Neste exemplo foi mostrado o parâmetro P0696, mas a mesma escala é utilizada para o parâmetro P0697 / P0698. Por exemplo, deseja-se controlar o valor da saída analógica 1 através da serial. Neste caso deve fazer a seguinte programação:

- Escolher um dos parâmetros P0696, P0697, P0698 para ser o valor utilizado pela saída analógica 1. Neste exemplo, vamos escolher o P0696.
- Programar, na função da saída analógica 1 (P0254), a opção “Conteúdo P0696”.
- Através da interface de rede, escrever no P0696 o valor desejado para a saída analógica 1, entre 0 e 100 %, de acordo com a escala do parâmetro.

<sup>5</sup> Para a resolução real da saída, consulte o manual do produto.


**NOTA!**

Para o inversor de frequência MW500, a saída analógica 3 representa a saída em frequência (FO).

**P0700 – PROTOCOLO CAN**

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

**Descrição:**

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

**P0701 – ENDEREÇO CAN**

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

**Descrição:**

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

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

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

**P0702 – TAXA DE COMUNICAÇÃO CAN**

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

**Descrição:**

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

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

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

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

**P0703 – RESET DE BUS OFF**

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

**Descrição:**

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

*Tabela 4.5: Opções para o parâmetro P0703*

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

**P0705 – ESTADO DO CONTROLADOR CAN**

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

**Descrição:**

Permite identificar se a interface CAN está devidamente instalada, e se a comunicação apresenta erros.

*Tabela 4.6: Valores para o parâmetro P0705*

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

**P0706 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS**

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

**Descrição:**

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

**P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS**

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

**Descrição:**

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

**P0708 – CONTADOR DE ERROS DE BUS OFF**

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

**Descrição:**

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

**P0709 – CONTADOR DE MENSAGENS CAN PERDIDAS**

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

**Descrição:**

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

**P0721 – ESTADO DA COMUNICAÇÃO CANOPEN**

<b>Faixa de</b>	0 = Desabilitado	<b>Padrão:</b> -
<b>Valores:</b>	1 = Reservado	
	2 = Comunicação Habilitada	
	3 = Controle de Erros Habilitado	
	4 = Erro de Guarding	
	5 = Erro de Heartbeat	
<b>Grupo de acesso via HMI:</b>	NET	

**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**

<b>Faixa de</b>	0 = Desabilitado	<b>Padrão:</b> -
<b>Valores:</b>	1 = Inicialização	
	2 = Parado	
	3 = Operacional	
	4 = Pré-Operacional	
<b>Grupo de acesso via HMI:</b>	NET	

**Descrição:**

Cada escravo da rede CANopen possui uma máquina de estados que controla o seu comportamento com relação à comunicação. Este parâmetro indica em qual estado encontra-se o dispositivo, conforme a especificação do protocolo.

## 5 DICIONÁRIO DE OBJETOS

O dicionário de objetos é uma lista com os diversos dados do equipamento que são acessíveis através da 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 5.1: Agrupamentos do dicionário de objetos*

Índice	Objetos	Descrição
0001h – 025Fh	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 representarão.
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.

### 5.1 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 – 025Fh).
- **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.

### 5.2 TIPOS DE DADOS

A primeira parte do dicionário de objetos (índices 0001h – 025Fh) 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.

### 5.3 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, mas somente alguns são obrigatórios. A seguir é apresentada uma lista com alguns dos objetos desta faixa suportados pelo inversor de frequência MW500.



**Tabela 5.2:** Lista de objetos – 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
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
Receive PDO Mapping Parameter				
1600h	RECORD	1st receive PDO mapping	PDO Mapping	rw
1601h	RECORD	2nd 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
Transmit PDO Mapping Parameter				
1A00h	RECORD	1st transmit PDO mapping	PDO Mapping	rw
1A01h	RECORD	2nd transmit PDO mapping	PDO Mapping	rw

Estes objetos somente podem ser lidos e escritos através da rede CANopen, não estão disponíveis via HMI ou outra interface de rede. O mestre da rede, em geral, é o equipamento responsável pela configuração do equipamento antes de iniciar a operação. O arquivo de configuração EDS traz a lista de todos os objetos de comunicação suportados.

Para uma descrição detalhada de quais objetos estão disponíveis nesta faixa do dicionário de objetos, consulte o item 6.

#### 5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DO FABRICANTE

Nos índices de 2000h até 5FFFh, cada fabricante é livre para definir quais objetos estarão presentes, o tipo e a função de cada objeto. Para o drive, nesta faixa de objetos foi disponibilizada toda a lista de parâmetros. Através destes parâmetros é possível operar o equipamento, executando qualquer função que o drive 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 5.3:** Lista de objetos – Manufacturer Specific

Índice	Objeto	Nome	Tipo	Acesso
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
...	...	...	...	...

É necessário conhecer a operação do drive através dos parâmetros para poder programar corretamente sua operação via rede CANopen.

Para a lista completa e uma descrição detalhada dos parâmetros, consulte o manual de programação do inversor de frequência MW500.

#### 5.5 DEVICE PROFILE – OBJETOS COMUNS PARA DRIVES

A documentação CANopen também inclui propostas para padronização de determinados tipos de dispositivos. O inversor de frequência MW500 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 o item 7.

## 6 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO

Neste item são descritos detalhadamente cada um dos objetos de comunicação disponíveis para o inversor de frequência MW500. É necessário conhecer como estes objetos são operados para utilizar as funções disponíveis para a comunicação do drive.

### 6.1 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.

#### 6.1.1 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.

#### 6.1.2 Objeto 1001h – Error Register

Este objeto indica a ocorrência ou não de erro no dispositivo. O tipo de erro registrado para o equipamento segue o descrito pela tabela a seguir.

Índice	1001h
Nome	Error register
Objeto	VAR
Tipo	UNSIGNED8

Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED8
Valor Padrão	0

**Tabela 6.1:** 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.

#### 6.1.3 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	0000.0A00h

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

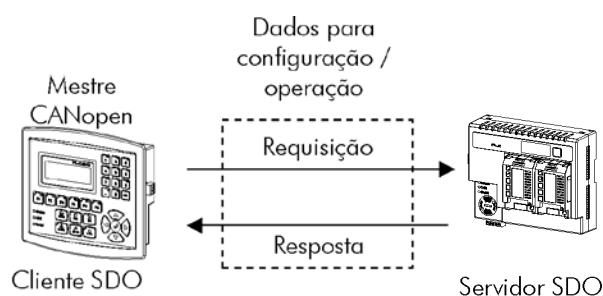
Sub-índice	4
Descrição	Número serial
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	Diferente para cada MW500

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 MW500 em rede CANopen.

## 6.2 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 6.1:** Comunicação entre cliente e servidor SDO

### 6.2.1 Objeto 1200h – Servidor SDO

O inversor de frequência MW500 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 drive. Todo o servidor SDO possui um objeto, do tipo SDO\_PARAMETER, 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

### 6.2.2 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 6.2:** 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, mas os comandos serão diferentes:

**Tabela 6.3:** 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 drive (objetos a partir do índice 2000h), este valor é salvo na memória não volátil do produto. 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 escravo 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 escravo responde à requisição, indicando que o valor para o referido objeto é igual a 999<sup>6</sup>:

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

### 6.3 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.

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. É necessário, porém, 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.

<sup>6</sup> Não esquecer que qualquer dado do tipo inteiro, a ordem de transferência dos bytes vai do menos significativo até o mais significativo.

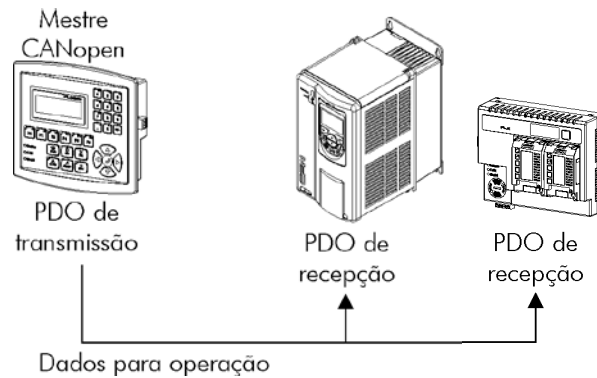


Figura 6.2: Comunicação utilizando PDOs



**NOTA!**

PDOs somente podem ser transmitidos ou recebidos quando o dispositivo está no estado operacional.

### 6.3.1 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 fabricante (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 equipamento traz a lista de todos os objetos disponíveis, informando se o objeto é mapeável ou não.

Tabela 6.4: 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
2005h	VAR	P0005 – Frequência 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
2012h	VAR	P0018 – Valor de AI1	INTEGER16	ro
2013h	VAR	P0019 – Valor de AI2	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	INTEGER16	ro
22ACh	VAR	P0684 – Controle CANopen/DNet	UNSIGNED16	rw
22ADh	VAR	P0685 – Ref. Vel. CANop./DNet	INTEGER16	rw

O arquivo EDS do equipamento traz a lista de todos os objetos disponíveis, informando se o objeto é mapeável ou não.

### 6.3.2 PDOs de Recepção

Os PDOs de recepção, ou RPDOs, são responsáveis por receber dados que outros dispositivos enviam para a rede CANopen. O inversor de frequência MW500 possui 2 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 até 1401h
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

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 6.5:** 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 drive 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 6.6:** 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 até 1601h
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	0

Sub-índice	1 até 2
Descrição	1° até 4° objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	Indicado no arquivo EDS

Este parâmetro indica os objetos mapeados nos PDOs de recepção do inversor de frequência MW500. O valor padrão destes objetos é indicado no arquivo EDS do produto. Para cada RPDO, é possível mapear até 4 objetos diferentes, desde que o tamanho total não ultrapasse oito bytes. O mapeamento de um objeto é feito indicando o seu índice, sub-índice<sup>7</sup> 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, supondo um mapeamento para um PDO de recepção com os seguintes valores configurados, 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 drive, 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 drive, que representa a referência de velocidade.

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.
- Os valores recebidos através destes objetos não são salvos na memória não volátil. 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 dispositivo esteja no estado operacional.

### 6.3.3 PDOs de Transmissão

Os PDOs de transmissão, ou TPDOs, como o nome diz, são responsáveis por transmitir dados para a rede CANopen. O inversor de frequência MW500 possui 2 PDOs de transmissão, cada um podendo transmitir até 8

<sup>7</sup> Caso o objeto seja do tipo VAR e não possua sub-índice, deve ser indicado o valor 0 (zero) para o sub-índice.

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 até 1801h
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

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 6.5.

O sub-índice 2 indica o tipo de transmissão deste objeto, que segue o descrito pela tabela 6.6. 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 100us) 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 inversor de frequência MW500 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 até 1A01h
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	0

Sub-índice	1 até 4
Descrição	1º até 4º objeto mapeado no PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0

O PDO MAPPING para a transmissão funciona de forma semelhante ao de 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, supondo um mapeamento para um PDO de recepção com os seguintes valores configurados, 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 drive, 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 drive, que representa a velocidade do motor.

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.

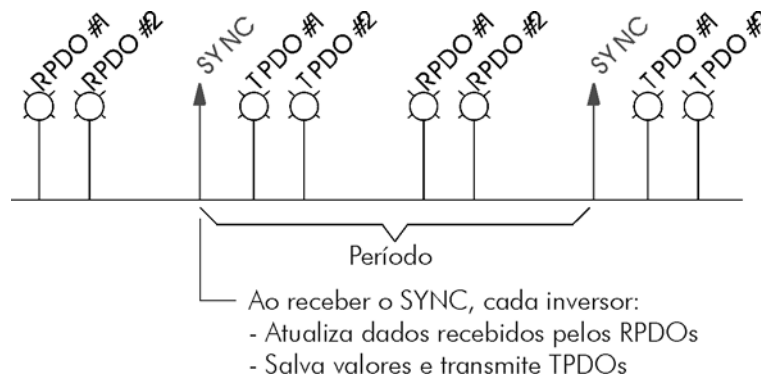

**NOTA!**

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.

**6.4 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 inversor de frequência MW500 possui a função de consumidor SYNC e, portanto, pode programar seus PDOs para serem síncronos. 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 6.3:** SYNC

A mensagem SYNC transmitida pelo produtor não possui dado algum em seu campo de dados, pois seu objetivo é fornecer um evento sincronizado entre os dispositivos da rede. O seguinte objeto está disponível para configuração 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.

**6.5 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 este objeto estão disponíveis os serviços de controle do nó e de controle de erros (utilizando *Node Guarding* ou *Heartbeat*).

### 6.5.1 Controle dos Estados do Escravo

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

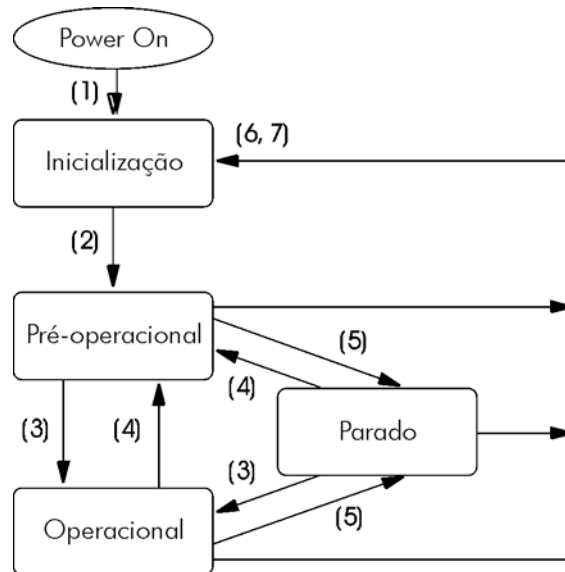


Figura 6.4: Diagrama de estados do nó CANopen

Tabela 6.7: 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	Recebo 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. Os PDOs, porém, 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 6.8: 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

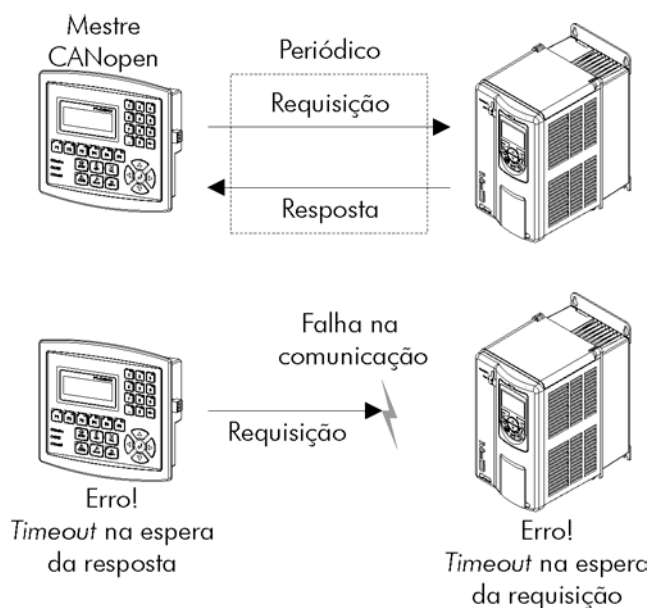
**Tabela 6.9:** Comandos para a transição de estados

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

As transições indicadas no código do comando equivalem às transições de estado executadas pelo nó após receber o comando (conforme figura 6.4). O comando *Reset node* faz com que o escravo execute um reset completo do dispositivo, enquanto que o comando *Reset communication* faz com que o escravo reinicialize apenas os objetos relativos à comunicação CANopen.

## 6.5.2 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 6.5:** 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 escravo não receba nenhum telegrama do mestre. O objeto 100Dh indica quantas falhas em sequência são necessárias até que se considere que houve realmente perda da 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 escravo 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 enviado pelo escravo (*toggle bit*).

Caso o inversor de frequência MW500 detecte um erro utilizando este mecanismo, ele irá automaticamente para o estado pré-operacional e indicará A135 na sua HMI.

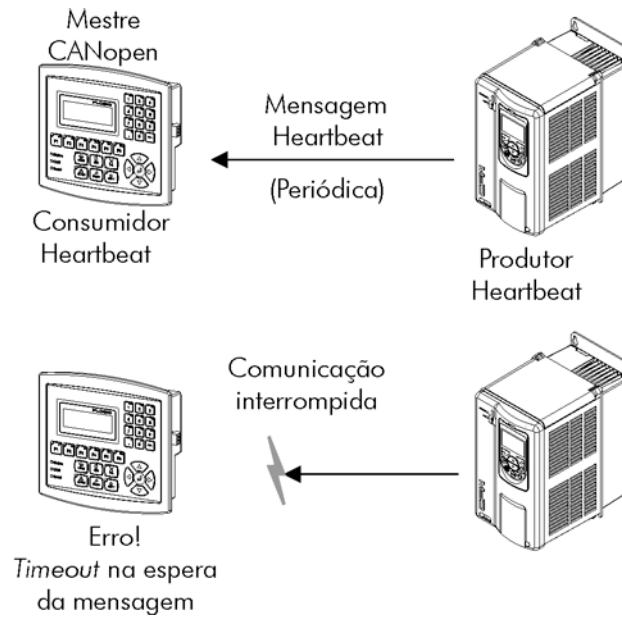


### NOTA!

- Este objeto está ativo mesmo no estado parado (consulte a tabela 6.8).
- 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 inversor de frequência MW500 é de 2ms. 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.

### 6.5.3 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 6.6:** 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 inversor de frequência MW500 possui os serviços de produtor e consumidor *heartbeat*. Como consumidor, é possível programar diferentes produtores para serem monitorados pelo equipamento:

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 4, 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 do produtor *heartbeat*, este irá automaticamente para o estado pré-operacional e indicará A135 na sua HMI.



Como produtor, o inversor de freqüência MW500 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 dispositivo 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

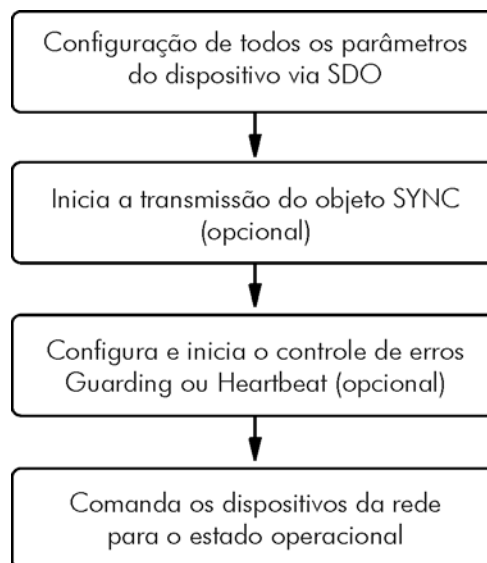


**NOTA!**

- Este objeto está ativo mesmo no estado parado (consulte a tabela 6.8).
- 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 HMI.
- O valor de tempo programado para o consumidor deve ser maior do que o programado para o respectivo produtor. Recomenda-se 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.

## 6.6 PROCEDIMENTO DE INICIALIZAÇÃO

Uma vez conhecido o funcionamento dos objetos disponíveis para o inversor de freqüência MW500 operando no modo escravo, 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 6.7:** Fluxograma do processo de inicialização

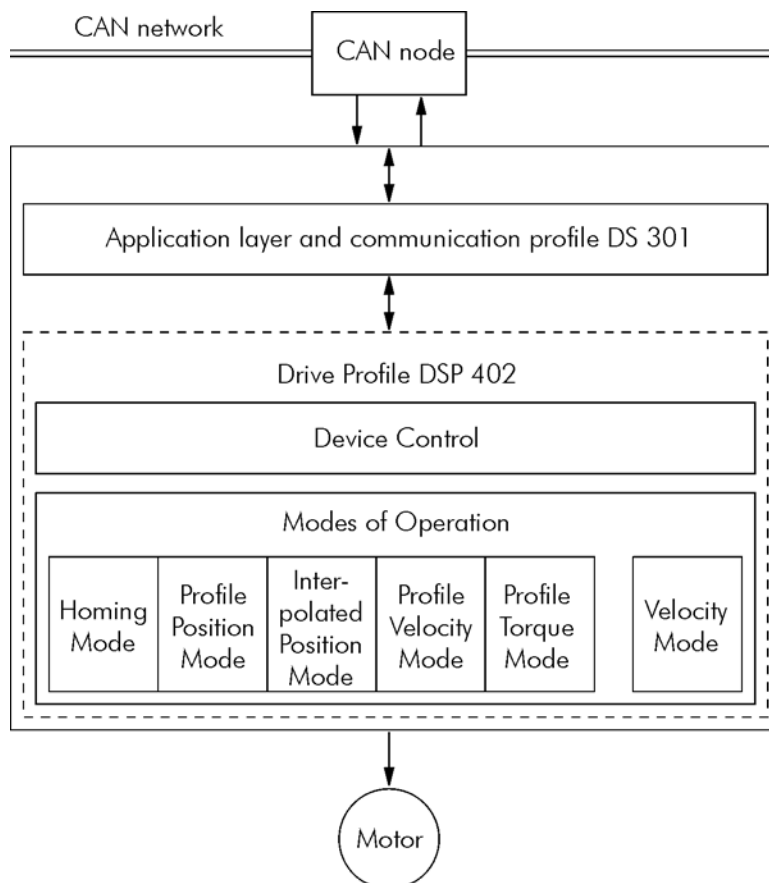
É necessário observar que os objetos de comunicação do inversor de freqüência MW500 (1000h até 1FFFh) não são armazenados na memória não volátil. Desta forma, sempre que for feito o reset ou desligado o equipamento, é necessário refazer a parametrização dos objetos de comunicação. Para os objetos específicos

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.

## 7 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 7.1 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 7.1:** Arquitetura de comunicação para um drive na rede CANopen

A tabela a seguir mostra a lista dos objetos disponíveis para o inversor de frequência MW500, divididos de acordo com os diferentes modos de operação do equipamento.

**Tabela 7.1:** Lista de objetos do MW500 – 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
6502h	VAR	Supported drive modes	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

Sempre que um objeto desta lista for lido ou escrito, o drive irá mapear suas funções nos parâmetros do usuário. Desta forma, ao operar o sistema através destes objetos, os valores dos parâmetros podem ser

alterados de acordo com a função utilizada. Nos itens seguintes, é feita uma descrição detalhada de cada um destes objetos, onde são indicados os parâmetros que são utilizados para executar as funções destes objetos.

## 7.1 DEVICE CONTROL – OBJETOS PARA CONTROLE DO DRIVE

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

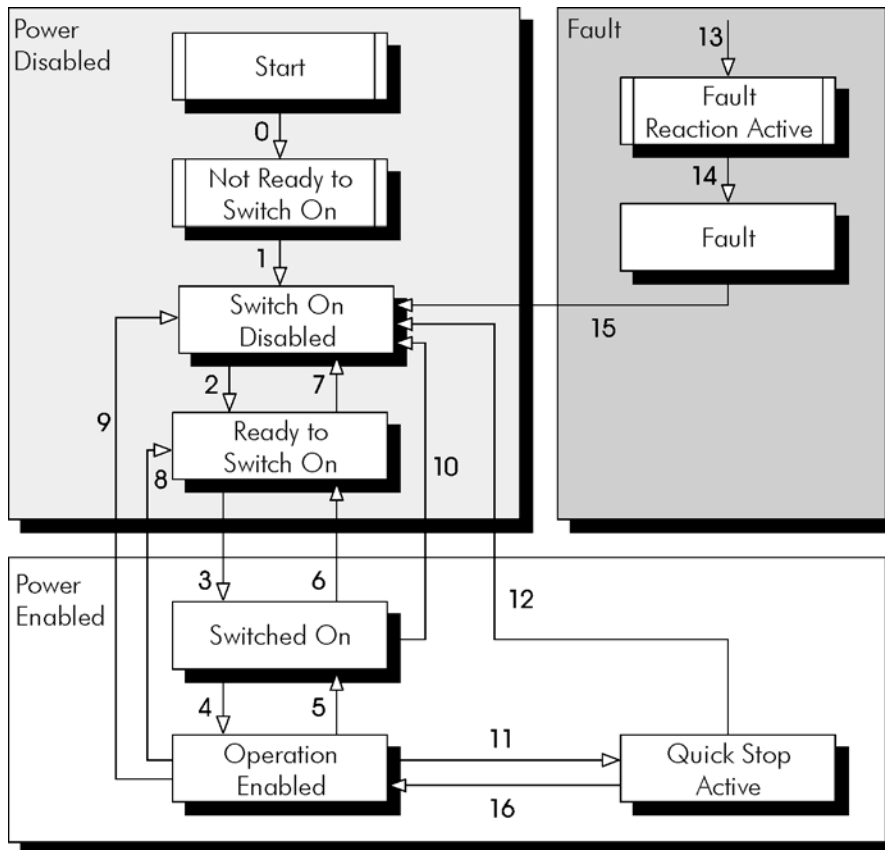


Figura 7.2: Máquina de estado para drives

Descrição dos estados:

- **Not ready to switch on:** o drive está inicializando, não pode ser comandado.
- **Switch on disabled:** inicialização completa, drive 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!**

Dependendo do equipamento e configuração, é possível que o drive não possua chave para bloqueio / habilitação da alimentação de potência. Desta forma, os estados descritos no grupo *Power disabled* são implementados por questões de compatibilidade com a máquina de estados descrita, mas 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.
- **Transição 1:** Inicialização completa (automático).

- **Transição 2:** Comando *Shutdown* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo drive.
- **Transição 3:** Comando *Switch on* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo drive.
- **Transição 4:** Comando *Enable operation* recebido. O drive é habilitado. Equivale a ativar o bit 1 da palavra de controle via CAN – P0684.
- **Transição 5:** Comando *Disable operation* recebido. O drive é desabilitado. Equivale a zerar o bit 1 da palavra de controle via CAN – P0684.
- **Transição 6:** Comando *Shutdown* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo drive.
- **Transição 7:** Comandos *Quick stop* e *Disable voltage* recebidos. É feita a transição de estados, mas nenhuma ação é tomada pelo drive.
- **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 – P0684.
- **Transição 9:** Comando *Disable voltage* 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 – P0684.
- **Transição 10:** Comando *Quick stop* ou *Disable voltage* recebido. É feita a transição de estados, mas nenhuma ação é tomada pelo drive.
- **Transição 11:** Comando *Quick stop* recebido. Drive executa a função de parada rápida. Equivale a ativar o bit 6 da palavra de controle via CAN – P0684.
- **Transição 12:** Comando *Disable voltage* recebido. O drive é desabilitado. Equivale a zerar o bit 1 da palavra de controle via CAN – 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. Driver executa o reset do erro e retorna para o estado desabilitado e sem falha.
- **Transição 16:** Comando *Enable operation* recebido. Drive executa a função de partida por rampa. Equivale a ativar o bit 0 da palavra de controle via CAN – 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.

### 7.1.1 Objeto 6040h – Controlword

Controla o estado do drive.

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 7.2. Os bits marcados com 'x' são irrelevantes para a execução do comando.

**Tabela 7.2:** 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 7.2.1.

Os bits 11, 12, 13, 14 e 15 são reservados para uso específico do fabricante. Para o inversor de frequência MW500 o bit 11 é utilizado para selecionar o modo de operação Local (0) ou remoto (1) do inversor de frequência.

### 7.1.2 Objeto 6041h – Statusword

Indica o estado atual do drive.

Índice	6041h
Nome	Statusword
Objeto	VAR
Tipo	UNSIGNED16
Parâmetros utilizados	P0680

Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16
Valor Padrão	-

Os bits desta palavra possuem as seguintes funções:

**Tabela 7.3:** 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 7.2. A tabela 7.4 descreve as combinações destes bits para indicação dos estados. Os bits marcados com 'x' são irrelevantes para a indicação do estado.

**Tabela 7.4:** 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:** Indica que o inversor de frequência MW500 possui algum alarme ativo.
- **Bit 9 – Remote:** indica quando o drive está no modo remoto e aceita comandos via rede CANopen<sup>8</sup>. 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:** não utilizado para o inversor de frequência MW500.
- **Bits 12 e 13 – Operation mode specific:** dependem do modo de operação do drive.

### 7.1.3 Objeto 6060h – Modes of Operation

Permite programar o modo de operação do drive.

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 de frequência MW500 é o *Velocity Mode*, representado pelo valor 2.

### 7.1.4 Objeto 6061h – Modes of Operation Display

Indica o modo de operação do drive.

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

Acesso	rw
Mapeável	Sim
Faixa	INTEGER8
Valor Padrão	-

O valor mostrado neste objeto está de acordo com o utilizado no objeto 6060h.

## 7.2 VELOCITY MODE – OBJETOS PARA CONTROLE DO DRIVE

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

<sup>8</sup> Depende da programação do equipamento.

- 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.

### 7.2.1 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:

**Tabela 7.5:** Modo Velocidade – definição dos bits 4, 5, 6 e 8

Bit	Nome	Valor	Descrição
4			Reservado
5			Reservado
6	Reference ramp (habilita rampa)	0	Desabilita Rampa – bit 0 do P0684 = 0
		1	Habilita Rampa – bit 0 do P0684 = 1
8	Halt	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 os bits 6 e 8 sejam ativados.

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

### 7.2.2 Objeto 6042h – vl Target Velocity

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

Índice	6042h
Nome	vl target velocity
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.

### 7.2.3 Objeto 6043h – vl Velocity Demand

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

Índice	6043h
Nome	vl velocity demand
Objeto	VAR
Tipo	INTEGER16
Parâmetros utilizados	-

Acesso	ro
Mapeável	Sim
Faixa	INTEGER16
Valor Padrão	-

### 7.2.4 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	-

### 7.2.5 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	-

### 7.2.6 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 (em rpm) pelo tempo programado no sub-índice 2 (em segundos). Os valores programados devem respeitar a faixa de valores para o parâmetro P0100.

### 7.2.7 Objeto 6049h – VI Velocity Deceleration

Permite programar a rampa de desaceleração do inversor.

Índice	6049h
Nome	VI velocity deceleration
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 desaceleração é calculado dividindo-se a velocidade programada no sub-índice 1 (em rpm) pelo tempo programado no sub-índice 2 (em segundos). Os valores programados devem respeitar a faixa de valores para o parâmetro P0101.

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

**Possíveis Causas/Correção:**

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

### A134/ F234 – BUS OFF

**Descrição:**

Detectado erro de *bus off* na interface CAN.

**Atuação:**

Caso o número de erros de recepção ou transmissão detectados pela interface CAN seja muito elevado<sup>9</sup>, o controlador CAN pode ser levado ao estado de *bus off*, onde ele interrompe a comunicação e desabilita a interface CAN.

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

**Possíveis Causas/Correção:**

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

### 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 *Heartbeat* – 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.

<sup>9</sup> Para mais informações sobre detecção de erros, consultar especificação CAN.

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.

### 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 detecção de erros pelo escravo sejam múltiplos dos tempos programados para a troca de mensagens no mestre.
- 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.



WEG Drives & Controls - Automação LTDA.  
Jaraguá do Sul – SC – Brasil  
Fone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020  
São Paulo – SP – Brasil  
Fone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212  
automacao@weg.net  
[www.weg.net](http://www.weg.net)