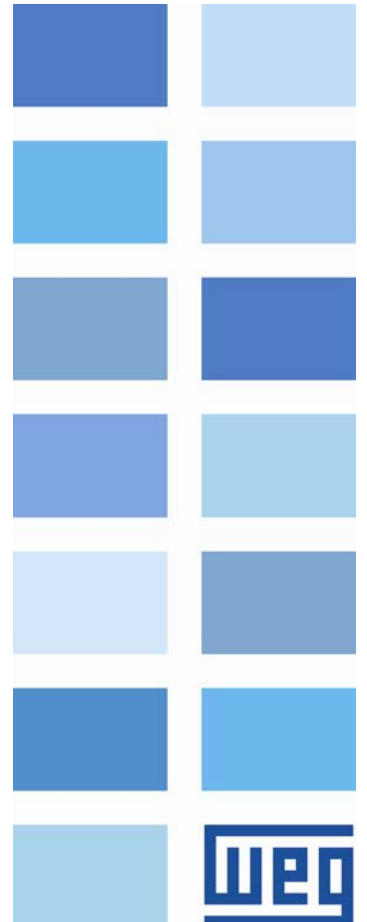


CANopen

SCA06

Manual do Usuário





Manual do Usuário CANopen

Série: SCA06

Idioma: Português

N ° do Documento: 10000649265 / 05

Data da Publicação: 07/2014

SUMÁRIO

SUMÁRIO.....	3
SOBRE O MANUAL.....	6
ABREVIações E DEFINIções.....	6
REPRESENTAção NUMÉRICA.....	6
DOCUMENTOS.....	6
1 INTRODUÇÃO À COMUNICAÇÃO CANOPEN.....	7
1.1 CAN.....	7
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
1.2 CARACTERÍSTICAS DA REDE CANOPEN.....	8
1.3 MEIO FÍSICO.....	8
1.4 ENDEREÇO NA REDE CANOPEN.....	8
1.5 ACESSO AOS DADOS.....	8
1.6 TRANSMISSÃO DE DADOS.....	8
1.7 OBJETOS RESPONSÁVEIS PELA COMUNICAÇÃO - COBS.....	9
1.8 COB-ID.....	9
1.9 ARQUIVO EDS.....	10
2 INTERFACE DE COMUNICAÇÃO CANOPEN.....	11
2.1 CARACTERÍSTICAS DA INTERFACE CAN.....	11
2.2 PINAGEM DO CONECTOR.....	11
2.3 FONTE DE ALIMENTAÇÃO.....	11
2.4 INDICAções.....	12
3 INSTALAÇÃO DA REDE CANOPEN.....	13
3.1 TAXA DE COMUNICAÇÃO.....	13
3.2 ENDEREÇO NA REDE CANOPEN.....	13
3.3 RESISTORES DE TERMINAÇÃO.....	13
3.4 CABO.....	13
3.5 LIGAÇÃO NA REDE.....	14
4 PARAMETRIZAÇÃO.....	15
4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES.....	15
P0070 – ESTADO DO CONTROLADOR CAN.....	15
P0071 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS.....	15
P0072 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS.....	15
P0073 – CONTADOR DE ERROS DE BUS OFF.....	16
P0074 – CONTADOR DE MENSAGENS CAN PERDIDAS.....	16
P0075 – ESTADO DA COMUNICAÇÃO CANOPEN.....	16
P0076 – ESTADO DO NÓ CANOPEN.....	16
P0202 – MODO DE OPERAÇÃO.....	17
P0662 – AÇÃO PARA ERRO DE COMUNICAÇÃO.....	17
P0700 – PROTOCOLO CAN.....	18
P0701 – ENDEREÇO CAN.....	18
P0702 – TAXA DE COMUNICAÇÃO CAN.....	18
P0703 – RESET DE BUS OFF.....	19
P0704 – FOLLOW.....	19

P0705 – COB ID FOLLOW	20
P0706 – PERÍODO FOLLOW	20
4.2 FUNÇÃO FOLLOW VIA CANOPEN.....	20
4.2.1 Follow programado por parâmetros.....	20
4.2.2 Follow programado pelo software WSCAN	21
5 DICIONÁRIO DE OBJETOS	22
5.1 ESTRUTURA DO DICIONÁRIO	22
5.2 TIPOS DE DADOS.....	22
5.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICAÇÃO	22
5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DO FABRICANTE.....	23
5.4.1 Objeto 3000h – Entradas digitais.....	24
5.4.2 Objeto 3001h – Saídas digitais	24
5.4.3 Objetos 3002h a 3009h – Follow	25
5.5 DEVICE PROFILE – OBJETOS COMUNS PARA DRIVES.....	25
6 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO.....	26
6.1 OBJETOS DE IDENTIFICAÇÃO	26
6.1.1 Objeto 1000h – Device Type	26
6.1.2 Objeto 1001h – Error Register	26
6.1.3 Objeto 1018h – Identity Object	27
6.2 SERVICE DATA OBJECTS – SDOS	27
6.2.1 Objeto 1200h – Servidor SDO	28
6.2.2 Funcionamento dos SDOs	28
6.3 PROCESS DATA OBJECTS – PDOS	29
6.3.1 Objetos Mapeáveis para os PDOs	30
6.3.2 PDOs de Recepção.....	30
6.3.3 PDOs de Transmissão	33
6.4 SYNCHRONIZATION OBJECT – SYNC	35
6.5 NETWORK MANAGEMENT – NMT	36
6.5.1 Controle dos Estados do Escravo	36
6.5.2 Controle de Erros – Node Guarding.....	37
6.5.3 Controle de Erros – Heartbeat.....	38
6.6 PROCEDIMENTO DE INICIALIZAÇÃO	40
7 DESCRIÇÃO DOS OBJETOS PARA DRIVES.....	42
7.1 DEVICE CONTROL – OBJETOS PARA CONTROLE DO DRIVE	43
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
7.1.5 Objeto 6502h – Supported drives modes	47
7.2 FACTOR GROUP – OBJETOS PARA CONVERSÃO DE UNIDADES	48
7.2.1 Objeto 608Fh – Position Encoder Resolution	48
7.2.2 Objeto 6091h – Gear Ratio	48
7.2.3 Objeto 6092h – Feed constant.....	49
7.3 POSITION CONTROL FUNCTION – CONTROLADOR DE POSIÇÃO	50
7.3.1 Objeto 6063h – Position actual value.....	50
7.3.2 Objeto 6064h – Position Actual Value in User Units.....	50
7.4 PROFILE POSITION MODE – OBJETOS PARA CONTROLE DO DRIVE.....	50
7.4.1 Bits de Controle e Estado	53
7.4.2 Objeto 607Ah – Target Position.....	54
7.4.3 Objeto 6081h – Profile Velocity.....	54
7.4.4 Objeto 6083h – Profile Acceleration.....	54
7.4.5 Objeto 6084h – Profile Deceleration	55
7.4.6 Objeto 6086h – Motion Profile Type.....	55
7.5 PROFILE VELOCITY MODE – OBJETOS PARA CONTROLE DO DRIVE	55

7.5.1	Bits de Controle e Estado	55
7.5.2	Objeto 6069h – Velocity Sensor Actual Value	56
7.5.3	Objeto 606Bh – Velocity Demand Value	56
7.5.4	Objeto 606Ch – Velocity Actual Value.....	56
7.5.5	Objeto 60FFh – Target Velocity	57
7.6	PROFILE TORQUE MODE – OBJETOS PARA CONTROLE DO DRIVE.....	57
7.6.1	Bits de Controle e Estado	57
7.6.2	Objeto 6071h – Target Torque	58
7.6.3	Objeto 6077h – Torque Actual Value.....	58
7.6.4	Objeto 6087h – Torque Slope.....	58
7.6.5	Objeto 6088h – Torque Profile Type.....	58
8	OPERAÇÃO NA REDE CANOPEN – MODO MESTRE	60
8.1	HABILITAÇÃO DA FUNÇÃO CANOPEN MESTRE.....	60
8.2	CARACTERÍSTICAS DO MESTRE CANOPEN	60
8.3	OPERAÇÃO DO MESTRE	60
8.4	BLOCOS PARA O MESTRE CANOPEN	61
8.4.1	CANopen SDO – Leitura/Escrita de Dados via SDO	61
9	MARCADORES DE SISTEMA PARA CAN/CANOPEN	63
9.1	WORDS DE LEITURA DE ESTADO	63
9.2	WORDS DE ESCRITA DE COMANDO	63
10	FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO CANOPEN ...	64
	A133/F33 – SEM ALIMENTAÇÃO NA INTERFACE CAN	64
	A134/ F34 – BUS OFF	64
	A135/F35 – NODE GUARDING/HEARTBEAT	64

SOBRE O MANUAL

Este manual fornece a descrição necessária para a operação do servoconversor SCA06 utilizando o protocolo CANopen. Este manual deve ser utilizado em conjunto com manual do usuário do SCA06.

ABREVIações E DEFINIções

CAN	Controller Area Network
CiA	CAN in Automation
COB	Communication Object
COB-ID	Communication Object Identifier
SDO	Service Data Object
PDO	Process Data Object
RPDO	Receive PDO
TPDO	Transmit PDO
NMT	Network Management Object
ro	Read only (somente leitura)
rw	Read/write (leitura e escrita)

REPRESENTAÇÃO NUMÉRICA

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

DOCUMENTOS

O protocolo CANopen foi desenvolvido baseado nas seguintes especificações e documentos:

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

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

1 INTRODUÇÃO À COMUNICAÇÃO CANOPEN

Para a operação 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 SCA06. 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)¹. Nesta camada, são definidos os diferentes tipos de telegramas (frames), a forma de detecção de erros, validação e arbitragem de mensagens.

1.1.1 Frame de Dados

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

Identificador	8 bytes de dados							
11 Bits	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

1.1.2 Frame Remoto

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

1.1.3 Acesso à Rede

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

1.1.4 Controle de Erros

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

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

² A especificação CAN 2.0 define dois tipos de frames de dados: *standard* (11 bits) e *extended* (29 bits). Para esta implementação, 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 servoconversor SCA06, 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 INTERFACE DE COMUNICAÇÃO CANOPEN

O servoconversor SCA06 possui por padrão no produto uma interface CAN. Ela pode ser utilizada para comunicação no protocolo CANopen como mestre ou escravo da rede. Características desta interface são descritas a seguir.

2.1 CARACTERÍSTICAS DA INTERFACE CAN



Figura 2.1: Detalhe do conector CAN na parte inferior do produto

- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Alimentação externa de 24 V.
- Permite a conexão de até 64 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada e com o uso de repetidores³.
- Comprimento máximo do barramento de 1000 metros.

2.2 PINAGEM DO CONECTOR

A interface CAN possui um conector *plug-in* de 5 vias (X4) com a seguinte pinagem:

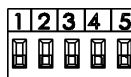


Tabela 2.1: Pinagem do conector X4 para interface CAN

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

2.3 FONTE DE ALIMENTAÇÃO

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

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

Tabela 2.2: Características da alimentação para interface CAN

Tensão de alimentação (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 CANopen para o servoconversor SCA06 são feitas através da HMI e dos parâmetros do produto.

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 CiA⁴.

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 servoconversor SCA06, 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 servoconversor SCA06, 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 ²)
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

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.

⁴ 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.

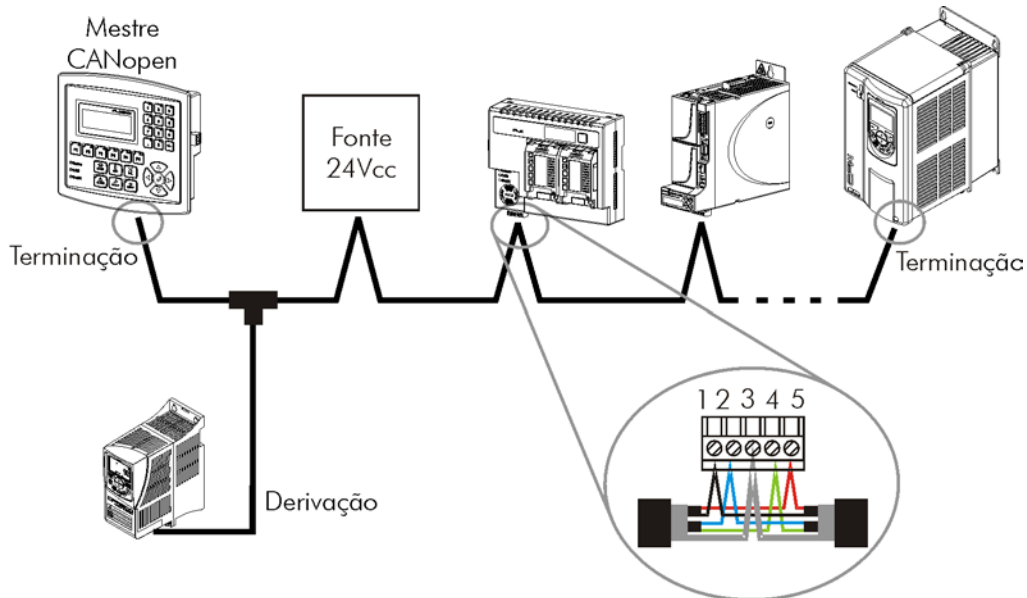


Figura 3.1: Exemplo de instalação em rede CANopen

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 servoconversor SCA06 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

P0070 – ESTADO DO CONTROLADOR CAN

Faixa de Valores:	0 = Inativo 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	Padrão: -
Propriedades:	RO	

Descrição:

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

Tabela 4.1: Valores para o parâmetro P0070

Valor	Descrição
0 = Inativo	Interface CAN inativa. Ocorre quando equipamento não possui protocolo CAN programado no P0700.
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.

P0071 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	

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.

P0072 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	

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.

P0073 – CONTADOR DE ERROS DE BUS OFF

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	

Descrição:

Contador cíclico que indica o número de 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.

P0074 – CONTADOR DE MENSAGENS CAN PERDIDAS

Faixa de Valores:	0 a 65535	Padrão: -
Propriedades:	RO	

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.

P0075 – ESTADO DA COMUNICAÇÃO CANOPEN

Faixa de Valores:	0 = Desabilitado 1 = Reservado 2 = Comunicação Habilitada 3 = Controle de Erros Habilitado 4 = Erro de Guarding 5 = Erro de Heartbeat	Padrão: -
Propriedades:	RO, CAN	

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*).

P0076 – ESTADO DO NÓ CANOPEN

Faixa de Valores:	0 = Desabilitado 1 = Inicialização 2 = Parado 3 = Operacional 4 = Pré-Operacional	Padrão: -
Propriedades:	RO, CAN	

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.

P0202 – MODO DE OPERAÇÃO

Faixa de Valores:	1 = Modo Torque 2 = Modo Velocidade 3 = Reservado 4 = Modo Ladder 5 = CANopen 6 = Profibus DP	Padrão: 2
Propriedades:	CFG	

Descrição:

Este parâmetro define o modo de operação do servoconversor SCA06. Para que o equipamento seja controlado através da rede CANopen, é necessário utilizar o modo 5 = CANopen. Caso este modo esteja programado, comandos e referências para operação do produto serão dados via rede CANopen, utilizando os objetos definidos no dicionário de objetos.

Dentre os principais objetos utilizados para controle e monitoração do equipamento, pode-se citar:

- 6040h: ControlWord
- 6041h: StatusWord
- 6060h: Mode of operation
- 6063h: Position actual value
- 607Ah: Target position
- 60FFh: Target velocity
- 6071h: Target Torque

A descrição detalhada destes e demais objetos é feita no item 7. Para detalhes sobre os modos de operação de 1 até 4, consulte o manual do usuário do servoconversor SCA06.


NOTA!

- O controle o equipamento através dos objetos para drives somente é possível para o modo de operação 5, mas a comunicação CANopen pode ser utilizada em qualquer modo de operação.
- Para o SCA06 operando como escravo da função Follow, deve-se programar o modo de operação 4 (Ladder), e utilizar o bloco MC_GearInPos.

P0662 – AÇÃO PARA ERRO DE COMUNICAÇÃO

Faixa de Valores:	0 = Causa Alarme 1 = Causa Falha 2 = Executa STOP 3 = Desabilita drive	Padrão: 0
Propriedades:	CFG	

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 4.2: Opções para o parâmetro P0662

Opção	Descrição
0 = Causa Alarme	Apenas indica alarme.
1 = Causa Falha	No lugar de alarme, um erro de comunicação causa uma falha no equipamento, sendo necessário fazer o reset de falhas para o retorno da sua operação normal.
2 = Executa STOP	Será feita a indicação de alarme juntamente com a execução do comando STOP. Para que o servo saia desta condição, será necessário realizar o reset de falhas ou desabilitar o drive.
3 = Desabilita Drive	Será feita a indicação de alarme juntamente com a execução do comando desabilita.

São considerados erros de comunicação os seguintes eventos:

Comunicação CANopen:

- 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*).

P0700 – PROTOCOLO CAN

Faixa de	0 = Desabilitado	Padrão: 0
Valores:	1 = CANopen 2 = Reservado 3 = CANespecial 1	
Propriedades:		

Descrição:

Permite selecionar 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

Faixa de	0 a 127	Padrão: 63
Valores:		
Propriedades:		

Descrição:

Permite programar o endereço utilizado para a 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 = 3 (CANespecial 1) → endereços válidos: 1 a 8.

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

P0702 – TAXA DE COMUNICAÇÃO CAN

Faixa de	0 = 1 Mbit/s / <i>Autobaud</i>	Padrão: 0
Valores:	1 = 800 Kbit/s / <i>Autobaud</i> 2 = 500 Kbit/s 3 = 250 Kbit/s 4 = 125 Kbit/s 5 = 100 Kbit/s / <i>Autobaud</i> 6 = 50 Kbit/s / <i>Autobaud</i>	
Propriedades:		

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 = 3 (CANespecial 1): somente as taxas de 1Mbit/s, 800Kbit/s e 500Kbit/s são suportadas. Não possui 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.

P0703 – RESET DE BUS OFF

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

Descrição:

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

Tabela 4.3: Opções para o parâmetro P0703

Opção	Descrição
0 = Reset Manual	Caso ocorra <i>bus off</i> , será indicado na HMI o alarme A134/F34, a ação programada no parâmetro P0662 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 P0662.

P0704 – FOLLOW

Faixa de	0 = Desabilitado	Padrão: 0
Valores:	1 = Mestre Follow Real 2 = Mestre Follow Virtual 3 = Escravo Follow	
Propriedades:	CFG	

Descrição:

Permite habilitar a função Follow via CANopen, além de definir se o equipamento deverá ser mestre (produtor) ou escravo (consumidor) Follow.

Tabela 4.4: Opções para o parâmetro P0704

Opção	Descrição
0 = Desabilitado	Não envia mensagem Follow
1 = Mestre Follow Real	Envia telegramas Follow contendo posição e velocidade do eixo real.
2 = Mestre Follow Virtual	Envia telegramas Follow contendo posição e velocidade do eixo virtual.
3 = Escravo Follow	Recebe telegramas Follow. Necessita bloco MC_GearInPos para executar função Follow

Uma vez programado como mestre ou escravo, o servoconversor SCA06 deverá entrar automaticamente no modo operacional na rede CANopen, de forma a possibilitar a troca de PDOs entre os equipamentos da rede.

Para detalhes sobre o funcionamento da função Follow consulte o item 4.2.1.

P0705 – COB ID FOLLOW

Faixa de	385 – 511	Padrão: 0
Valores:		
Propriedades:	CFG	

Descrição:

Define o COB ID (Communication Object Identifier) do PDO Follow. A faixa de valores 385 (181h) a 511 (1FFh) é definida pela especificação CANopen como faixa padrão para o TPDO1. Tanto o mestre quanto os escravos devem utilizar o mesmo COB ID.

A função do TPDO1 (mestre) e do RPDO1 (escravos) é dedicada para esta função e, portanto, estes PDOs não devem ser configurados para comunicação de outros dados.

Para detalhes sobre o funcionamento da função Follow consulte o item 4.2.1.

P0706 – PERÍODO FOLLOW

Faixa de	0,2 a 5,0 ms	Padrão: 1,0
Valores:		
Propriedades:	CFG	

Descrição:

Permite programar o período de transmissão do telegrama Follow pelo mestre da rede. Não é utilizado pelos escravos.

Quanto menor o período, mais rapidamente as referências são transmitidas e mais preciso será o sincronismo, porém o tempo de ocupação do barramento também será maior, o que pode dificultar a comunicação caso existam outros dados a serem comunicados pela rede CANopen. Este período também deve ser programado em função da taxa de comunicação. A 1 Mbit/s, um telegrama Follow leva aproximadamente 100 us para ser transmitido. À medida que a taxa de comunicação fica mais lenta, este tempo de transmissão aumenta proporcionalmente.

Se apenas a função Follow for utilizada na rede, pode-se programar o menor período possível, uma vez que não existirão outros telegramas transmitidos na rede. Mas se a função de mestre CANopen for utilizada em paralelo, é importante que haja tempo disponível para transmissão dos demais telegramas CANopen. A título de recomendação, quando o mestre CANopen for utilizado, os telegramas Follow devem ocupar cerca de 10 a 20 % do tempo no barramento.

Para detalhes sobre o funcionamento da função Follow consulte o item 4.2.1.

4.2 FUNÇÃO FOLLOW VIA CANOPEN

A função Follow permite realizar o sincronismo de posição entre dois ou mais servomotores. O sincronismo é estabelecido através do envio de telegramas do tipo PDO, onde o mestre Follow envia valores de posição e velocidade do motor, que serão utilizados como referência por um ou mais escravos Follow.

A função Follow pode ser programada por duas fontes diferentes: por parâmetros ou pelo software WSCAN.

4.2.1 Follow programado por parâmetros

Caso não se deseje utilizar a função de mestre CANopen disponível para o servoconversor SCA06, pode-se programar a função Follow apenas utilizando parâmetros. Neste caso, os parâmetros P0704, P0705 e P0706 são utilizados, e é necessário fazer a programação/configuração dos seguintes elementos:

- **Rede:** para utilização da função Follow, é necessário primeiramente configurar a interface CANopen, definindo protocolo, endereço e taxa de comunicação, bem como realizar a instalação necessária para a comunicação – cabos, alimentação, resistores de terminação, etc..
- **Mestre:** para o mestre (ou produtor), é necessário habilitar a função Follow, selecionar o eixo de referência, e programar o COB ID e o período de transmissão.

- **Escravos:** nos escravos (ou consumidores), deve-se habilitar a função Follow no modo escravo e programar o COB ID, igual ao programado para o mestre. Para que os escravos recebam e utilizem estes valores de velocidade e posição, é necessário também que eles sejam programados para o modo Ladder (P0204 = 4), e deve-se utilizar o bloco MC_GearInPos. Mais detalhes sobre a configuração do bloco MC_GearInPos no Help do software WLP.

4.2.2 Follow programado pelo software WSCAN

Outra opção para programação da função Follow é através do software WSCAN. Neste caso, os parâmetros P0704, P0705 e P0706 não devem ser programados, e a habilitação da função é feita através da janela de configuração do servoconversor SCA06 no software WSCAN.

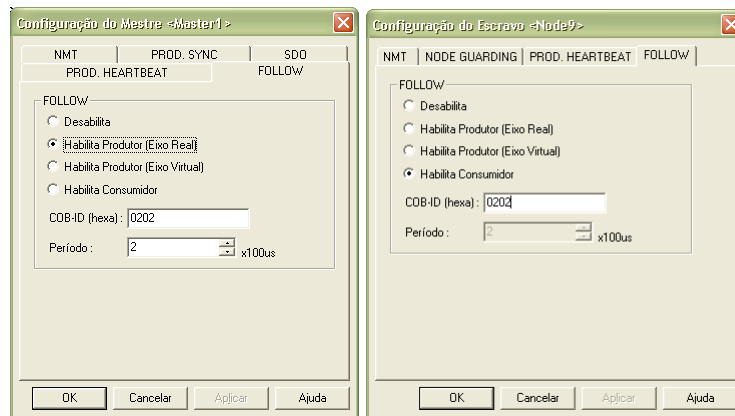


Figura 4.1: Janelas de configuração para mestre e escravo Follow

Uma vez programado, durante a inicialização dos escravos com a função Follow, o mestre da rede CANopen fará a programação da função utilizando objetos internos do equipamento. A programação é feita online, e não é necessário reiniciar o equipamento para que os novos valores sejam utilizados.

Juntamente com a programação da função Follow, também é possível programar outros serviços para a rede CANopen, como PDOs, SDOs, Node Guarding, etc., observando sempre que o tempo de ocupação do barramento CAN pelos telegramas Follow deve permitir a utilização destes outros serviços.

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 servoconversor SCA06, operando no modo escravo.

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
...				
1407h	RECORD	8th 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
...				
1607h	RECORD	8th 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
...				
1807h	RECORD	8th 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
...				
1A07h	RECORD	8th 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 servoconversor SCA06, nesta faixa de objetos foram disponibilizados os parâmetros do equipamento. 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
2000h	VAR	P0000 – Parâmetro de acesso	INTEGER16	rw
2002h	VAR	P0002 – Velocidade do motor	INTEGER16	ro
2003h	VAR	P0003 – Corrente do motor	INTEGER16	ro
2004h	VAR	P0004 – Tensão CC	INTEGER16	ro
...
2077h	VAR	P0119 – Referência de Corrente	INTEGER16	rw
2079h	VAR	P0121 – Referência de Velocidade	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 servoconversor SCA06.

Além dos parâmetros, o SCA06 apresenta os seguintes objetos:

- 0x3000h – entradas digitais;
- 0x3001h – saídas digitais;
- 0x3002h – Follow Position Actual Value;
- 0x3003h – Follow Velocity Actual Value;
- 0x3004h – Follow Target Position;
- 0x3005h – Follow Target Velocity.
- 0x3008h – Follow Type.
- 0x3009h – Follow Period.

5.4.1 Objeto 3000h – Entradas digitais

Permite acesso ao estado das entradas digitais do servoconversor SCA06.

Índice	3000h
Nome	Entradas Digitais
Objeto	Array
Tipo	UNSIGNED16

Sub-índice	0
Descrição	Number of Entries
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED16

Sub-índice	1
Descrição	Entrada digital padrão
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16

Sub-índice	2
Descrição	Entrada digital Slot 1
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16

Sub-índice	3
Descrição	Entrada digital Slot 2
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16

Sub-índice	4
Descrição	Entrada digital Slot 3
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED16

5.4.2 Objeto 3001h – Saídas digitais

Permite controlar ao estado das saídas digitais do servoconversor SCA06.

Índice	3001h
Nome	Saídas Digitais
Objeto	Array
Tipo	UNSIGNED8

Sub-índice	0
Descrição	Number of Entries
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8

Sub-índice	1
Descrição	Saída digital padrão
Acesso	ro
Mapeável	sim
Faixa	UNSIGNED8

Sub-índice	2
Descrição	Saída digital Slot 1
Acesso	ro
Mapeável	sim
Faixa	UNSIGNED8

Sub-índice	3
Descrição	Saída digital Slot 2
Acesso	ro
Mapeável	sim
Faixa	UNSIGNED8

Sub-índice	4
Descrição	Saída digital Slot 3
Acesso	ro
Mapeável	sim
Faixa	UNSIGNED8

5.4.3 Objetos 3002h a 3009h – Follow

Os objetos 3002h a 3009h são objetos de uso exclusivo da função Follow.

5.5 DEVICE PROFILE – OBJETOS COMUNS PARA DRIVES

A documentação CANopen também inclui propostas para padronização de determinados tipos de dispositivos. O servoconversor SCA06 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.

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 servoconversor SCA06 operando no modo escravo. É necessário conhecer como estes objetos são operados para utilizar as funções disponíveis para a comunicação do drive.


NOTA!

O servoconversor SCA06 pode operar como mestre ou escravo da rede CANopen. Os objetos descritos a seguir descrevem a operação do equipamento como escravo da rede CANopen. Para a descrição das características do produto operando como mestre da rede CANopen, consulte o item 8 juntamente com o software de configuração da rede CANopen WSCAN.

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	0002.0192h

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	0000.0700h

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 SCA06

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 servoconversor SCA06 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.

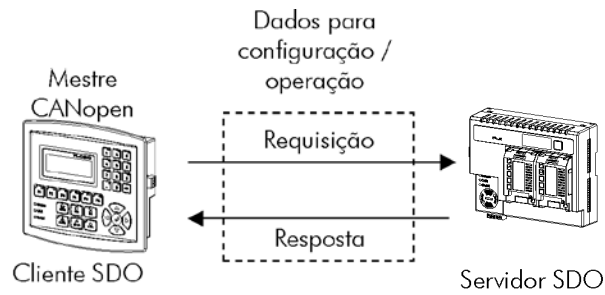


Figura 6.1: Comunicação entre cliente e servidor SDO

6.2.1 Objeto 1200h – Servidor SDO

O servoconversor SCA06 operando no modo escravo 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 999⁵:

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

⁵ Não esquecer que qualquer dado do tipo inteiro, a ordem de transferência dos bytes vai do menos significativo até o mais significativo.

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.

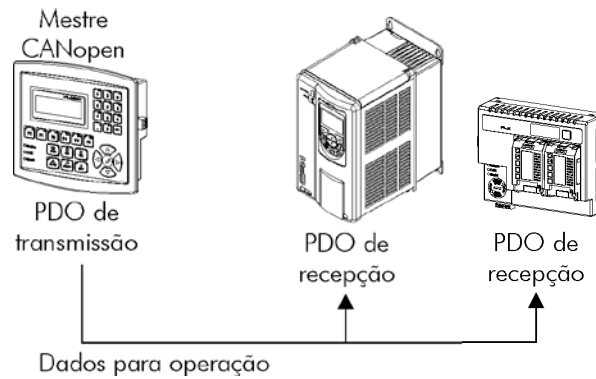


Figura 6.2: Comunicação utilizando PDOs



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	INTEGER16	ro
2003h	VAR	P0003 – Corrente do Motor	INTEGER16	ro
2004h	VAR	P0004 – Tensão Barram.CC (Ud)	UNSIGNED16	ro
2008h	VAR	P0008 – Estado DI1 a DI3	UNSIGNED16	ro
2063h	VAR	P0099 – Habilidade	UNSIGNED16	rw
2077h	VAR	P0119 – Referência de Corrente	INTEGER16	rw
2079h	VAR	P0121 – Referência de Velocidade	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 servoconversor SCA06 operando no modo escravo possui 8 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 1407h
Nome	Receive PDO communication parameter
Objeto	Record
Tipo	PDO COMM PARAMETER

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

Sub-índice	1
Descrição	COB-ID usado pelo PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1400h: 200h + Node-ID 1401h: 300h + Node-ID 1402h: 400h + Node-ID 1403h: 500h + Node-ID 1404h – 1407h: 0

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 1607h
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 4
Descrição	1º até 4º 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 servoconversor SCA06. 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-índice⁶ 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 = 2063.0010h:** o primeiro objeto mapeado possui índice igual a 2063h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0099 do drive, que representa o comando de habilitação do SCA06.
- **Sub-índice 2 = 2079.0010h:** o segundo objeto mapeado possui índice igual a 2079h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0121 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 Caso o objeto seja do tipo VAR e não possua sub-índice, deve ser indicado o valor 0 (zero) para o sub-índice.

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 servoconversor SCA06 possui 8 PDOs de transmissão, cada um podendo transmitir até 8 bytes de dados. De forma semelhante aos RPDOs, cada TPDO possui dois parâmetros para sua configuração, um PDO_COMM_PARAMETER e um PDO_MAPPING, conforme descrito a seguir.

PDO_COMM_PARAMETER

Índice	1800h até 1807h
Nome	Transmit PDO Parameter
Objeto	Record
Tipo	PDO COMM PARAMETER

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

Sub-índice	1
Descrição	COB-ID usado pelo PDO
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	1800h: 180h + Node-ID 1801h: 280h + Node-ID 1802h: 380h + Node-ID 1803h: 480h + Node-ID 1804h – 1807h: 0

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 servoconversor SCA06 é 1ms.
- 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 1A07h
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 = 2002.0010h:** o primeiro objeto mapeado possui índice igual a 2002h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0002 do drive, que representa a velocidade do motor.
- **Sub-índice 2 = 2023.0010h:** o segundo objeto mapeado possui índice igual a 2023h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0035 do drive, que representa a falha atual.

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 servoconversor SCA06 operando no modo escravo 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.

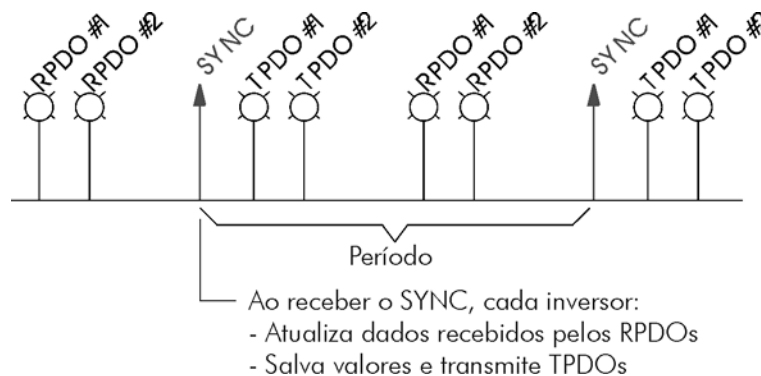


Figura 6.3: SYNC

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:

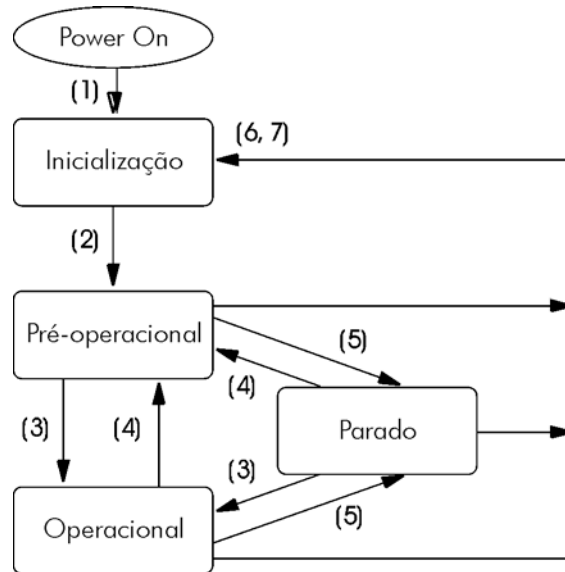


Figura 6.4: Diagrama de estados do nó CANopen

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	Recebe comando Reset Communication, onde reinicializa o valor dos objetos e a comunicação CANopen do dispositivo

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

No estado pré-operacional, já é possível comunicar-se com o escravo. 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.

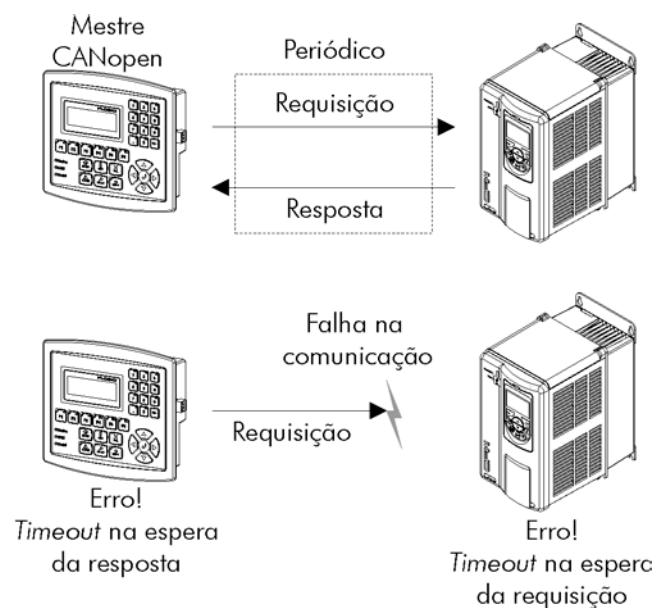


Figura 6.5: *Serviço de controle de erros – Node Guarding*

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 servoconversor SCA06 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 servoconversor SCA06 é de 1ms. 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.

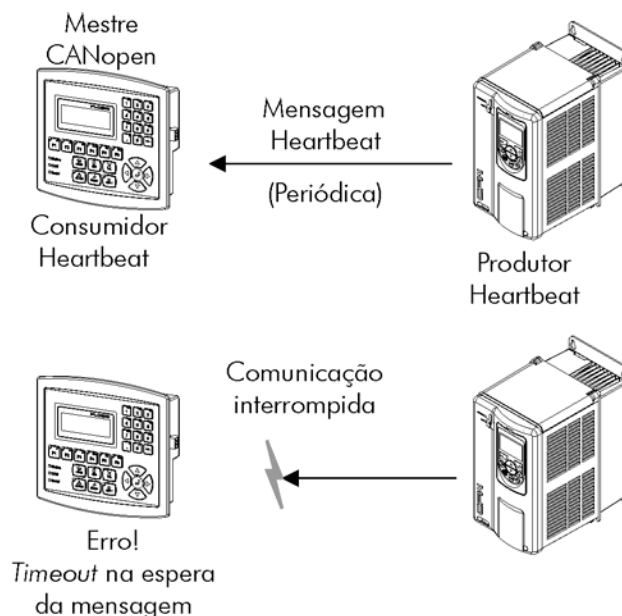


Figura 6.6: Serviço de controle de erros – Heartbeat

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 servoconversor SCA06 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	8

Sub-índices	1 – 8
Descrição	Consumer Heartbeat Time 1 – 8
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	0

Nos sub-índices de 1 até 8, 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 servoconversor SCA06 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 servoconversor SCA06 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:

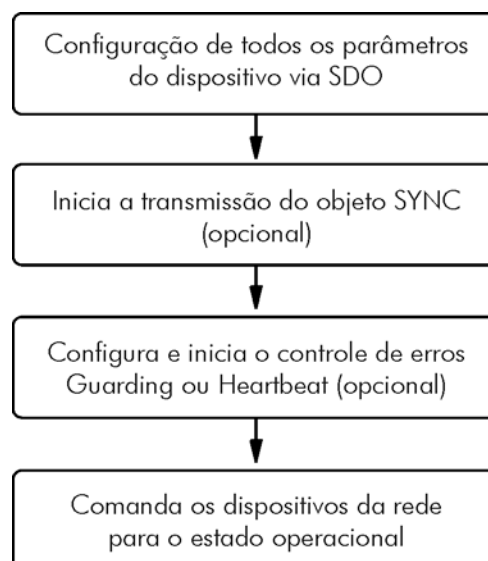


Figura 6.7: Fluxograma do processo de inicialização

É necessário observar que os objetos de comunicação do servoconversor SCA06 (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.

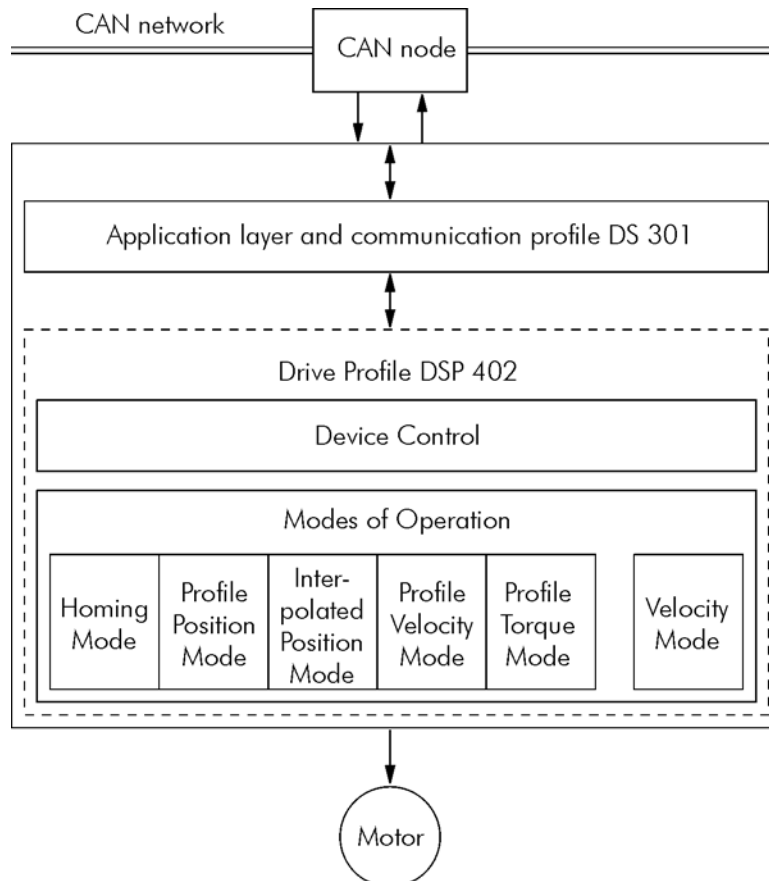


Figura 7.1: Arquitetura de comunicação para um drive na rede CANopen

A tabela a seguir mostra a lista dos objetos disponíveis para o servoconversor SCA06, divididos de acordo com os diferentes modos de operação do equipamento.

Tabela 7.1: Lista de objetos do SCA06 – Drive Profile

Índice	Objeto	Nome	Tipo	Acesso	Mapeável
Control Device					
6040h	VAR	ControlWord	Unsigned16	rw	Sim
6041h	VAR	StatusWord	Unsigned16	ro	Sim
6060h	VAR	Mode of operation	Integer8	rw	Sim
6061h	VAR	Modes of operation display	Integer8	ro	Sim
6502h	VAR	Supported drives modes	Unsigned32	ro	Sim
Factor Group					
608Fh	VAR	Position encoder resolution	Unsigned32	rw	Não
6091h	VAR	Gear ration	Unsigned32	rw	Não
6092h	VAR	Feed constant	Unsigned32	rw	Não
Position Control Function					
6063h	VAR	Position actual value	Integer32	ro	Sim
6064h	VAR	Position actual value in user units	Integer32	ro	Sim
Profile Position Mode					
607Ah	VAR	Target position	Integer32	rw	Sim
6081h	VAR	Profile velocity	Unsigned32	rw	Sim
6083h	VAR	Profile acceleration	Unsigned32	rw	Sim
6084h	VAR	Profile deceleration	Unsigned32	rw	Sim
6086h	VAR	Motion profile type	Integer16	rw	Sim
Profile Velocity Profile					
6069h	VAR	Velocity sensor actual value	Integer32	ro	Sim
606Bh	VAR	Velocity demand value	Integer32	ro	Sim
606Ch	VAR	Velocity actual value	Integer32	ro	Sim
60FFh	VAR	Target velocity	Integer32	rw	Sim
Profile Torque Mode					
6071h	VAR	Target torque	Integer16	rw	Sim
6077h	VAR	Torque actual value	Integer16	ro	Sim
6087h	VAR	Torque slope	Unsigned32	rw	Sim
6088h	VAR	Torque profile type	Integer16	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:

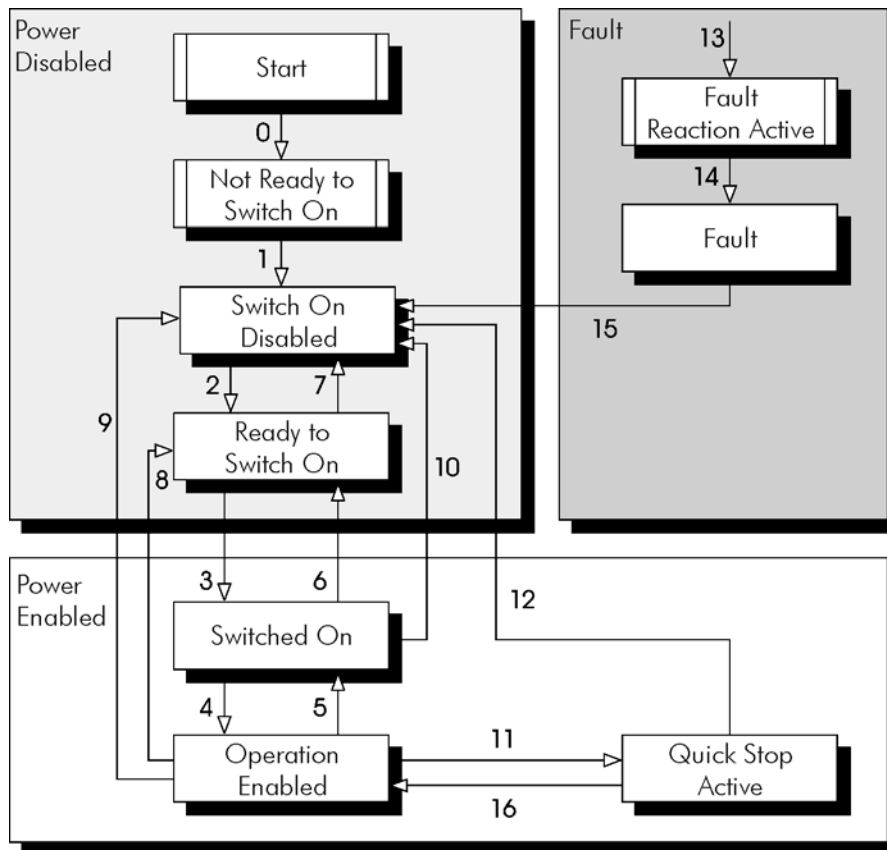


Figura 7.2: Máquina de estado para drives

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.
- **Transição 5:** Comando *Disable operation* recebido. O drive é desabilitado.
- **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.
- **Transição 9:** Comando *Disable voltage* recebido. Durante a operação do drive este é desabilitado, bloqueando a alimentação para o motor.
- **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.
- **Transição 12:** Comando *Disable voltage* recebido. O drive é desabilitado.
- **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.

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.



NOTA!

Para que os comandos enviados pela palavra de controle sejam executados pelo servoconversor SCA06, é necessário que o drive seja programado para o modo de operação "CANopen". Esta programação é feita no parâmetro P0202.

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 servoconversor SCA06 possui algum alarme ativo.
- **Bit 9 – Remote:** indica quando o drive está no modo remoto e aceita comandos via rede CANopen⁷.
- **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 servoconversor SCA06.
- **Bits 12 e 13 – Operation mode specific:** dependem do modo de operação do drive.

⁷ Depende da programação do equipamento.

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	-

Os valores aceitos por este objeto são descritos na tabela 7.5. Demais valores são reservados.

Tabela 7.5: Modos de operação para o servoconversor SCA06

Valor	Modo de operação
1	Profile Position Mode
3	Profile Velocity Mode
4	Profile Torque Mode

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.

7.1.5 Objeto 6502h – Supported drives modes

Indica os modos de operação suportados pelo drive. Cada bit representa um modo de operação, e o valor 1 no bit indica que o modo de operação é suportado.

31 – 15	15 – 7	9	8	7	6	5	4	3	2	1	0
Manufacturer specific	reserved	cst	csv	csp	ip	hm	reserved	tq	pv	vl	PP

O servoconversor SCA06 apresenta 3 modos de operação:

- pp: Profile Position mode;
- pv: Profile Velocity mode;
- tq: Torque mode.

Conhecendo os modos suportados no SCA06, defini-se o valor 0Dh para este objeto.

Índice	6502h
Nome	Supported drives modes
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acesso	ro
Mapeável	Sim
Faixa	UNSIGNED8
Valor Padrão	0Dh

7.2 FACTOR GROUP – OBJETOS PARA CONVERSÃO DE UNIDADES

Este grupo de objetos permite fazer a conversão de unidades para objetos que representam valores de posição. Estes valores terão sua escala e dimensão definida de acordo com os valores de notação e dimensão programados, conforme descritos a seguir:

7.2.1 Objeto 608Fh – Position Encoder Resolution

Este objeto define o incremento do encoder de acordo com a rotação do motor.

Position encoder resolution = encoder increments / motor revolutions

Índice	608Fh
Nome	Position encoder resolution
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	02h

Sub-índice	1
Descrição	Encoder increments
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	FFh

Sub-índice	2
Descrição	Motor revolutions
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	01h

Valores possíveis para o sub-índice 1 (Encoder increments):

Tabela 7.6: Valores para o Sub-índice Encoder Increments

Valor	Modo de operação
41	Graus
42	Minutos
43	Segundos
FF	Unidade interna – 65536 incrementos por volta

O sub-índice 2 (Motor revolutions) aceita somente valor igual a 1.

7.2.2 Objeto 6091h – Gear Ratio

Este objeto indica a configuração do número de rotações do eixo do motor e o número de rotações do eixo motriz, ou seja, define a relação de transmissão. A relação de transmissão é definida pela seguinte fórmula:

Gear ratio = motor shaft revolutions / driving shaft revolutions

Índice	6091h
Nome	Gear ratio
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	02h

Sub-índice	1
Descrição	Motor revolutions
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	01h

Sub-índice	2
Descrição	Shaft revolutions
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	01h

O único valor possível para o sub-índice 1 e sub-índice 2 é 1.

7.2.3 Objeto 6092h – Feed constant

Este objeto indica a distância por 1 volta do eixo do motor.

Índice	6092h
Nome	Feed constant
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descrição	Número do último sub-índice
Acesso	ro
Mapeável	Não
Faixa	UNSIGNED8
Valor Padrão	02h

Sub-índice	1
Descrição	Feed
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	FFh

Sub-índice	2
Descrição	Shaft revolutions
Acesso	rw
Mapeável	Não
Faixa	UNSIGNED32
Valor Padrão	01h

Valores possíveis para o sub-índice 1 (Feed):

Tabela 7.7: Valores para o Sub-índice Feed

Valor	Modo de operação
41	Graus
42	Minutos
43	Segundos
FF	Unidade interna – 65536 incrementos por volta

O sub-índice 2 (Shaft revolutions) aceita somente valor igual a 1.

7.3 POSITION CONTROL FUNCTION – CONTROLADOR DE POSIÇÃO

Este grupo de objetos é utilizado para descrever o funcionamento do controlador de posição em malha fechada.

7.3.1 Objeto 6063h – Position actual value

Representa a posição atual do eixo do motor em incrementos. Uma volta completa representa 65536 incrementos.

Índice	6063h
Nome	Position actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	ro
Mapeável	Sim
Faixa	INTEGER32
Valor Padrão	-

O valor deste objeto representa sempre a posição de eixo em uma volta apenas. O número de voltas não é controlado por este objeto.

7.3.2 Objeto 6064h – Position Actual Value in User Units

Representa a posição atual do eixo do motor. O valor deste objeto pode ser transformado de unidades internas para valores definidos pelo usuário, de acordo com o programado nos objetos 608Fh, 6091h e 6092h, conforme Tabela 7.8.

Índice	6064h
Nome	Position actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	ro
Mapeável	Sim
Faixa	INTEGER32
Valor Padrão	-

O valor deste objeto representa sempre a posição de eixo em uma volta apenas. O número de voltas não é controlado por este objeto.

Tabela 7.8: Programação dos objetos Factor Group

Objeto Unidade	608Fh sub-índice 1	608Fh sub-índice 2	6091h sub-índice 1	6091h sub-índice 2	6092h sub-índice 1	6092h sub-índice 2
Graus	41h	1	1	1	41h	1
Minutos	42h	1	1	1	42h	1
Segundos	43h	1	1	1	43h	1
Unidade interna	FFh	1	1	1	FFh	1

7.4 PROFILE POSITION MODE – OBJETOS PARA CONTROLE DO DRIVE

Este modo de operação permite o controle do servoconversor SCA06 através do ajuste de set-point de posição, que podem ser executados seguindo dois métodos:

- single set-point;
- set of set-points.

Independente do método utilizado, os seguintes objetos devem ser configurados:

- 0x6081 – Profile Velocity;
- 0x6083 – Profile Acceleration;
- 0x6084 – Profile Deceleration;
- 0x6086 – Motion Profile Type;
- 0x607A – Target Position;

Após o ajuste da velocidade, da aceleração e do set-point deve-se realizar o seguinte procedimento:

- Habilitar o drive escrevendo 15 no objeto 0x6041 - ControlWord;
- Escrever nos bits 9, 8, 6, 5 e 4 do objeto 0x6041 – ControlWord, conforme Tabela 7.9 e Tabela 7.10;

O estado da execução do posicionamento pode ser verificado no objeto 0x6040 – StatusWord conforme Tabela 7.11. O bit SET-POINT ACKNOWLEDGE no objeto de status (StatusWord – 6040h) será setado indicando que um novo set-point foi recebido. Se o set-point for aceito o bit é resetado. Quando o set-point for alcançado, o bit TRAGET REACHED no objeto de status será setado. A figura 7.3 ilustra um exemplo de escrita de set-point.

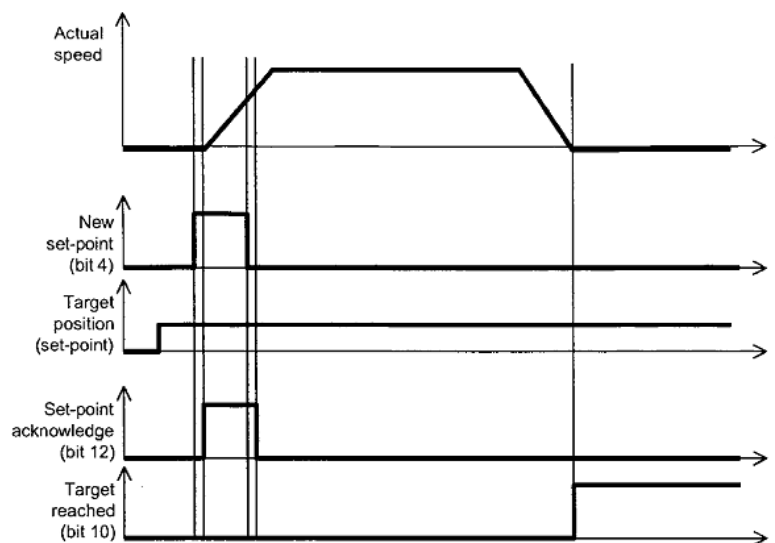


Figura 7.3: Ajuste de set-point de posição (Fonte: IEC 61800-7-201)

Single set-point

O método set-point único é utilizado quando se deseja executar um novo set-point imediatamente. A figura 7.4 ilustra o funcionamento do método.

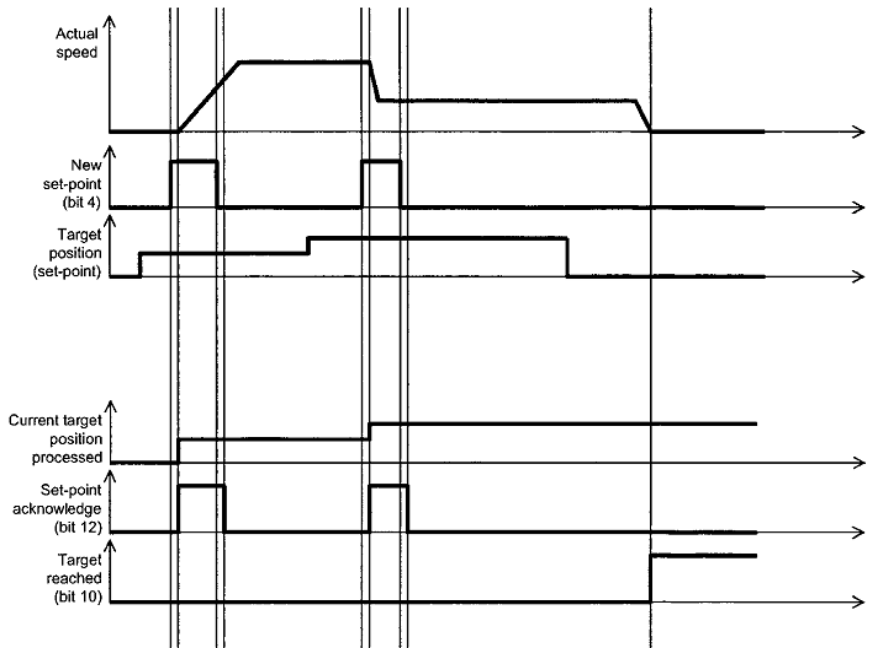


Figura 7.4: Método single set-point (Fonte: IEC 61800-7-201)

Set of set-point

O método conjunto de set-point é utilizado quando se deseja executar um novo set-point somente após a finalização do anterior. A figura 7.5 ilustra o funcionamento do método.

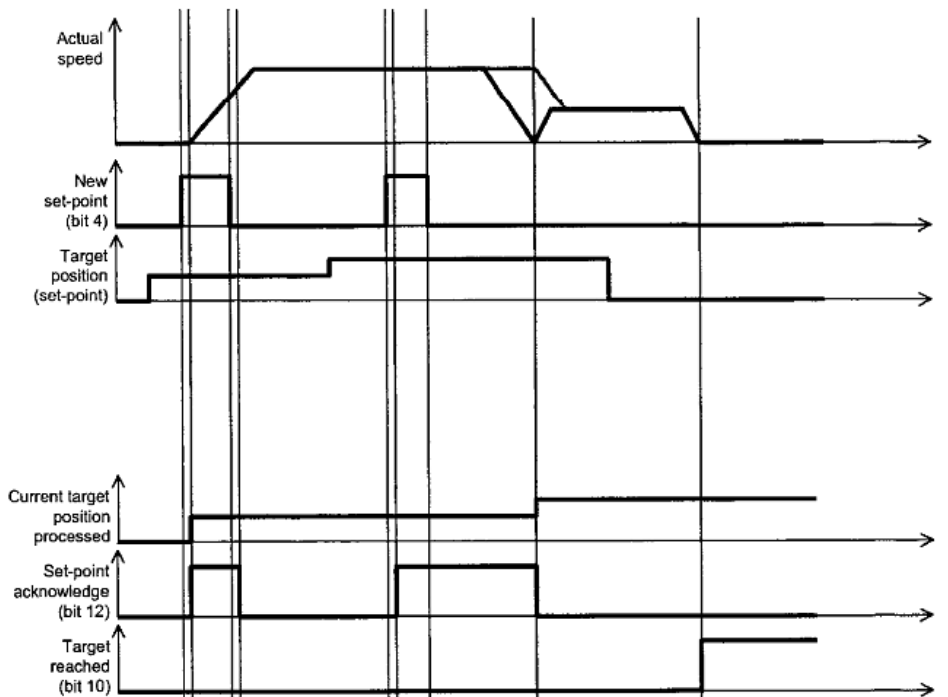


Figura 7.5: Método set of set-point (Fonte: IEC 61800-7-201)

O servoconversor SCA06 pode armazenar dois set-points, o que está em execução e o que será executado, como ilustra a figura 7.6.

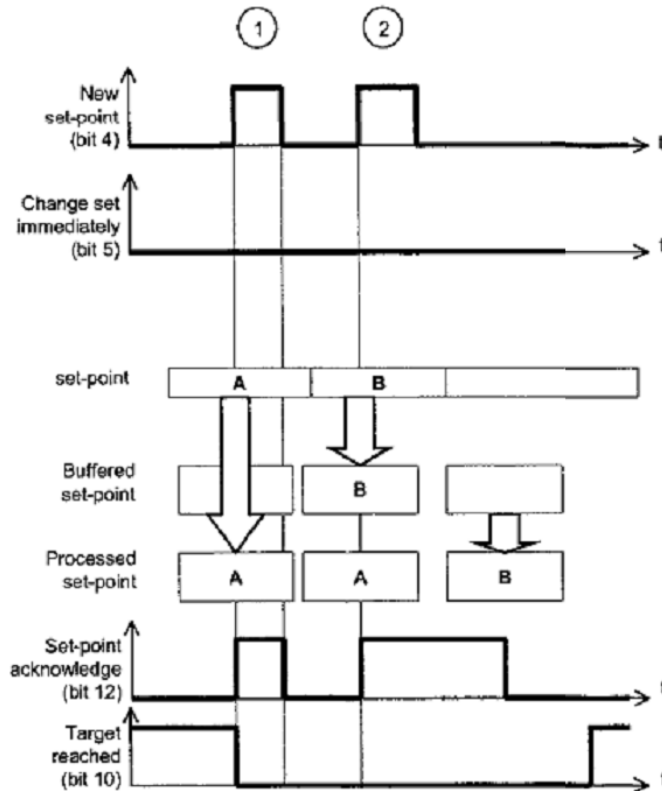


Figura 7.6: Armazenamento de set-point (Fonte: IEC 61800-7-201)

7.4.1 Bits de Controle e Estado

O profile mode position utiliza alguns bits dos objetos ControlWord e StatusWord para controlar e monitorar seu funcionamento. Para o objeto ControlWord são utilizados os seguintes bits:

- Bit 4 – New set-point;
- Bit 5 – Change set immediately;
- Bit 6 – absolute (0) / relative (1);
- Bit 8 – Halt (não implementado no SCA06);
- Bit 9 – Change on set-point.

As tabela 7.9 e tabela 7.10 informam a definição dos bits de controle.

Tabela 7.9: Modo Posicionamento – definição dos bits 4, 5 e 9

Bit 9	Bit 5	Bit 4	Definição
0	0	0 → 1	Posição deve ser concluída antes de a próxima iniciar.
X	1	0 → 1	Próxima posição deve ser iniciada imediatamente.
1	0	0 → 1	Opção não implementada no SCA06.

Tabela 7.10: Modo Posicionamento – definição dos bits 6 e 8

Bit	Valor	Definição
6	0	Referência de posição deve ser um valor absoluto.
	1	Referência de posição deve ser um valor relativo.
8	0	Posicionamento deve ser executado ou continuado.
	1	Eixo deve ser parado conforme objeto 605Dh.

Para o objeto StatusWord são utilizados os seguintes bits:

- Bit 10 – Target reached;
- Bit 12 – Set-point acknowledge;
- Bit 13 – Following error.

A Tabela 7.11 informa a definição dos bits de status.

Tabela 7.11: Modo Posicionamento – definição dos bits 10,12 e13

Bit	Valor	Definição
10	0	Referência de posição não alcançada.
	1	Referência de posição alcançada.
12	0	Referência de posição anterior já processada, aguardando nova referência de posição.
	1	Referência de posição anterior em processamento, substituição de referência de posição será aceita.
13	0	Sem erro de Following
	1	Erro de Following

7.4.2 Objeto 607Ah – Target Position

Permite programar a referência de posição para o servoconversor SCA06 no modo posicionamento. Os 16 bits mais significativos informam o número de volta e os 16 bits menos significativos informam a fração de volta. A escala utilizada neste objeto é 65536 para número de voltas e 65536 incrementos para uma volta do eixo. O valor deste objeto deve ser interpretado como absoluto ou relativo, conforme estado do Bit 6 do objeto ControlWord.

Número de volta 16MSB	Fração de volta 16LSB
--------------------------	--------------------------

Índice	607Ah
Nome	Target position
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTEGER32
Valor Padrão	0000 0000h

7.4.3 Objeto 6081h – Profile Velocity

Permite programar a velocidade normalmente atingida no final da rampa de aceleração durante um perfil de movimento. O valor a ser programado neste objeto deve estar entre 0 a 9999 rpm.

Índice	6081h
Nome	Profile Velocity
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	0000 0000h

7.4.4 Objeto 6083h – Profile Acceleration

Permite programar a rampa de aceleração até que o eixo do motor atinja a velocidade programada. A escala utilizada é a escala ms/krpm e os valores devem estar entre 1 a 32767

Índice	6083h
Nome	Profile Acceleration
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	0000 0001h

7.4.5 Objeto 6084h – Profile Deceleration

Permite programar a rampa de desaceleração até que o eixo do motor atinja a velocidade zero. A escala utilizada neste objeto é a mesma do objeto 6083h.

Índice	6084h
Nome	Profile deceleration
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	0000 0001h

7.4.6 Objeto 6086h – Motion Profile Type

Permite programar o perfil da rampa de aceleração e desaceleração para o drive.

Índice	6086h
Nome	Motion profile type
Objeto	VAR
Tipo	INTERGER32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTERGER16
Valor Padrão	0h

Valores possíveis para este objeto:

Tabela 7.12: Valores para o Sub-índice Motion Profile Type

Valor	Definição
0000h	Rampa linear
FFFFh	Sem rampa

7.5 PROFILE VELOCITY MODE – OBJETOS PARA CONTROLE DO DRIVE

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

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

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

7.5.1 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 7.13: Modo Velocidade – definição dos bits 4, 5, 6 e 8

Bit	Nome	Valor	Descrição
4			Reservado
5			Reservado
6			Reservado
8	Halt	0	Executa movimento
		1	Pára eixo

Para o objeto StatusWord são utilizados os seguintes bits:

- Bit 10 – Target reached;
- Bit 12 – Speed ;
- Bit 13 – Max slippage error (não implementado).

Tabela 7.14: Modo Velocidade – definição dos bits 10,12 e13

Bit	Valor	Definição
10	0	Halt = 0 - referência de velocidade não alcançada. Halt = 1 - velocidade diferente de 0 (zero)
	1	Halt = 0 - referência de velocidade alcançada. Halt = 1 - velocidade igual a 0 (zero).
12	0	Velocidade diferente de 0(zero).
	1	Velocidade igual a 0 (zero).
13	0	Não implementado
	1	

7.5.2 Objeto 6069h – Velocity Sensor Actual Value

Permite a leitura do sensor utilizado para medir a velocidade do motor. O servoconversor SCA06 utiliza um resolver como posição (a velocidade angular é obtida derivando este valor no tempo), logo o sensor fornece um valor proporcional à posição angular. O sensor possui resolução de 14 bits, e uma volta completa fornece 16384 valores diferentes de posição.

Índice	6069h
Nome	Velocity sensor actual valor
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	RO
Mapeável	Sim
Faixa	INTERGER32
Valor Padrão	-

7.5.3 Objeto 606Bh – Velocity Demand Value

Indica a velocidade fornecida pelo gerador de trajetória do servoconversor, utilizada pelo regulador de velocidade para controle do motor. O valor fornecido por este objeto é dado na escala interna do SCA06 onde, 0x7FFF FFFF → 10.000 rpm.

Índice	606Bh
Nome	Velocity demand value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	ro
Mapeável	Sim
Faixa	INTERGER32
Valor Padrão	-
Valor mínimo	0x8000 0001
Valor máximo	0x7FFF FFFF

7.5.4 Objeto 606Ch – Velocity Actual Value

Indica a velocidade do motor. O valor fornecido por este objeto é dado na escala interna do SCA06 onde, 0x7FFF FFFF → 10.000 rpm.

Índice	606Ch
Nome	Velocity actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acesso	ro
Mapeável	Sim
Faixa	INTERGER32
Valor Padrão	-

7.5.5 Objeto 60FFh – Target Velocity

Permite programar a referência de velocidade para o servoconversor SCA06 no modo velocidade. O valor a ser programado neste objeto deve respeita a escala interna do SCA06 onde 0x7FFF FFFF → 10.000 rpm e 0x8000 0000 → -10.000 rpm

Índice	60FFh
Nome	Target Velocity
Objeto	VAR
Tipo	INTERGER32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTERGER32
Valor Padrão	0000 0000h

7.6 PROFILE TORQUE MODE – OBJETOS PARA CONTROLE DO DRIVE

Este modo possibilita o controle do drive via uma referência de torque recebida pela rede CANopen.

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

7.6.1 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 7.15: Modo Torque – definição dos bits 4, 5, 6 e 8

Bit	Nome	Valor	Descrição
4			Reservado
5			Reservado
6			Reservado
8	Halt	0	Executa movimento
		1	Para eixo

Para o objeto StatusWord são utilizados os seguintes bits:

- Bit 10 – Target reached;
- Bit 12 – Reservado;
- Bit 13 – Reservado.

Tabela 7.16: Modo Torque – definição dos bits 10, 12 e 13

Bit	Valor	Definição
10	0	Referência de torque não alcançada.
	1	Referência de torque alcançada.
12	0	Reservado
	1	
13	0	Reservado
	1	

7.6.2 Objeto 6071h – Target Torque

Permite programar a referência de torque para o servoconversor SCA06 no modo torque. A escala utilizada para escrita neste objeto é fornecida em partes por mil do torque nominal do motor.

Índice	6071h
Nome	Target Torque
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTERGER16
Valor Padrão	0000h

7.6.3 Objeto 6077h – Torque Actual Value

Indica o torque atual do motor. O valor é fornecido em parte por mil do torque nominal do motor.

Índice	6077h
Nome	Torque actual value
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTERGER16
Valor Padrão	0

7.6.4 Objeto 6087h – Torque Slope

Permite programar a taxa de variação do torque no tempo (rampa de torque) para o servoconversor SCA06. A escala utilizada é de partes por mil do torque nominal do motor por segundo.

Para obter o torque nominal do motor deve-se multiplicar a constante de torque do motor (P0417) pela corrente nominal do motor (P0401):

$$\text{Torque nominal do motor} = P0417 \times P0401$$

Portanto, o valor máximo aceito pelo objeto 0x6087 depende do motor que está sendo acionado.

Índice	6087h
Nome	Torque slope
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	UNSIGNED32
Valor Padrão	0

7.6.5 Objeto 6088h – Torque Profile Type

Indica o torque atual do motor. O valor é fornecido em parte por mil do torque nominal do motor.

Índice	6077h
Nome	Torque Profile type
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acesso	rw
Mapeável	Sim
Faixa	INTERGER16
Valor Padrão	0h

Valores possíveis para este objeto:

Tabela 7.17: Valores para o Sub-índice Torque Profile Type

Valor	Definição
0000h	Rampa linear de torque
FFFFh	Sem rampa

8 OPERAÇÃO NA REDE CANOPEN – MODO MESTRE

Além da operação como escravo, o servoconversor SCA06 também permite a operação como mestre da rede CANopen. A seguir serão descritas características e funções do SCA06 como mestre da rede CANopen.

8.1 HABILITAÇÃO DA FUNÇÃO CANOPEN MESTRE

Como padrão, o servoconversor SCA06 está programado para operar como escravo da rede CANopen. A programação do equipamento como mestre da rede deve ser feita utilizando o software WSCAN, que permite também a configuração de toda a rede CANopen. A descrição detalhada das janelas e funções do software WSCAN deve ser obtida no menu “Ajuda” do próprio software.

Depois de elaborada a configuração do mestre, é necessário fazer o download⁸ das configurações, utilizando uma das interfaces de programação do produto – consulte o manual do usuário para maiores informações. Uma vez programado como mestre da rede, caso seja necessário apagar estas configurações, a função para apagar o programa do usuário – através do P00204 – também apaga as configurações do mestre CANopen.



NOTA!

A rede CANopen é uma rede flexível e que permite várias formas de configuração e operação. No entanto, esta flexibilidade exige que o usuário tenha bom conhecimento das funções e objetos de comunicação utilizados para configuração da rede, bem como o conhecimento do software de programação WSCAN.

8.2 CARACTERÍSTICAS DO MESTRE CANOPEN

O servoconversor SCA06 permite controlar um conjunto de até 8 escravos, utilizando os seguintes serviços e recursos de comunicação:

- Serviço de gerenciador da rede (NMT)
- 8 PDOs de transmissão
- 8 PDOs de recepção
- 8 Consumidores Heartbeat
- Produtor Heartbeat
- Cliente SDO
- Produtor/consumidor SYNC
- Mapeamento nos PDOs feito utilizando parâmetros do usuário

As características físicas – instalação, conector, cabo, etc. – são as mesmas, tanto para o SCA06 operando como mestre quanto como escravo. As configurações de endereço e taxa de comunicação também são necessárias para a operação como mestre, mas estas configurações são programadas pelo software WSCAN de acordo com as propriedades definidas para o mestre no próprio software.

8.3 OPERAÇÃO DO MESTRE

Uma vez programado para operar como mestre, o servoconversor SCA06 executará as seguintes etapas para realizar a inicialização, em seqüência, para cada um dos escravos:

- 1ª: enviado o comando de reset da comunicação para toda a rede, para que os escravos iniciem com valores conhecidos para os objetos de comunicação.
- 2ª: Identificação do equipamento na rede, através da leitura via SDO do objeto 1000h/00h – Object Identification.
- 3ª: Escrita via SDO de todos os objetos programados para o escravo, que usualmente inclui a configuração e mapeamento dos TPDOs e RPDOs, node guarding, heartbeat, além dos objetos específicos do fabricante, caso sejam programados.
- 4ª: Iniciado serviço de controle de erros – node guarding ou heartbeat – caso sejam programados.
- 5ª: envio do escravo para modo operacional.

⁸ Durante o download das configurações, a comunicação CANopen será desabilitada, sendo reiniciada ao término da operação.

Se uma destas etapas falhar, será indicado erro de comunicação com o escravo. Dependendo das configurações, a inicialização do escravo será abortada, e o mestre fará a inicialização do escravo seguinte, retornando para o escravo com erro após tentar inicializar todos os demais escravos da rede.

De forma semelhante, se, durante a operação de um escravo, for identificado erro no serviço de controle de erros, dependendo das configurações feitas para o mestre, o escravo será automaticamente resetado e o procedimento de inicialização será executado novamente.



NOTA!

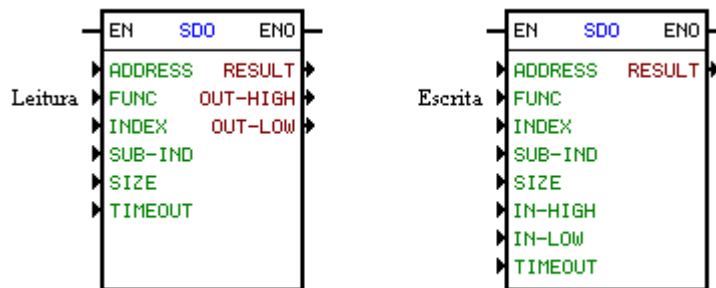
O estado da comunicação e o estado de cada escravo podem ser observados em marcadores de sistema.

8.4 BLOCOS PARA O MESTRE CANOPEN

Além dos objetos de comunicação e das configurações feitas no software WSCAN, também estão disponíveis blocos para monitoração e envio de comandos, que podem ser utilizados durante a elaboração do aplicativo em ladder para o servoconversor SCA06. Não é necessário utilizar estes blocos durante a operação do equipamento, mas seu uso confere maior flexibilidade e facilita o diagnóstico de problemas de comunicação durante a operação do servoconversor SCA06.

8.4.1 CANopen SDO – Leitura/Escrita de Dados via SDO

Bloco para leitura ou escrita de dados via SDO de um escravo remoto. Permite realizar a leitura ou escrita de objetos na rede com tamanho de até 4 bytes.



DESCRIÇÃO:

É composto por 1 entrada EN, 1 saída ENO e 9 argumentos, sendo eles:

- ADDRESS : Endereço do nó da rede CANopen
- FUNC : Função (leitura ou escrita)
- INDEX : Índice do objeto que deseja-se ler ou escrever (decimal)
- SUB-IND : Sub-índice do objeto que deseja-se ler ou escrever (decimal)
- SIZE : Tamanho do objeto que deseja-se ler ou escrever (bytes)
- TIMEOUT : Tempo em ms de espera para leitura ou escrita do valor
- RESULT : Resultado da execução do bloco
 - 0 = executado com sucesso
 - 1 = cartão não pode executar a função (exemplo : mestre não habilitado)
 - 2 = timeout na espera da resposta pelo mestre
 - 3 = escravo retornou erro
- OUT-HIGH : Valor mais significativo do objeto lido (word)
- OUT-LOW : Valor menos significativo do objeto lido (word)
- IN-HIGH : Valor mais significativo a ser escrito no objeto (word)
- IN-LOW : Valor menos significativo a ser escrito no objeto (word)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO vai para 1 após executar o bloco

FUNCIONAMENTO:

Se a entrada EN for zero, o bloco não é executado.

Se a entrada EN sofrer uma transição de 0 para 1, o cartão envia uma mensagem via rede CANopen para um escravo da rede, de acordo com os argumentos programados. Se o bloco estiver programado para leitura, o cartão fará a requisição para o escravo, e o valor informado pelo escravo será salvo nos argumentos de saída. Se o bloco estiver programado para escrita, os argumentos de entrada serão escritos no objeto correspondente do escravo. Após a execução do bloco a saída ENO vai para 1 e só retorna a zero após a entrada EN for para zero.

9 MARCADORES DE SISTEMA PARA CAN/CANOPEN

Para interface CAN e comunicação CANopen, foram disponibilizados os seguintes marcadores de leitura (%RS) e marcadores de escrita (%WC), para controle e monitoração desta interface:

9.1 WORDS DE LEITURA DE ESTADO

Estado do Mestre e Escravos CANopen: conjunto de marcadores de leitura para indicar informações sobre o estado geral do mestre CANopen e o estado da comunicação entre o mestre e cada um dos escravos.	
Marcador	Descrição
%RS4000	Estado do mestre CANopen: Bit 0: todos os escravos forma contatados. Bit 1: download das configurações dos escravos realizada. Bit 2: controle de erros dos escravos iniciado. Bit 3: fim da inicialização dos escravos. Bit 4: detectado erro na inicialização de pelo menos um escravo. Bit 5: detectado erro no serviço de controle de erros de pelo menos um escravo. Bits 6 e 7: reservado Bit 8: assume o valor do toggle bit (ver %CD3200) após o mestre enviar comando NMT. Bits 9 ... 12: reservado Bit 13: interface CAN no estado de bus off. Bit 14: sem alimentação na interface CAN. Bit 15: comunicação desabilitada.
%RS4001 ... %RS4127	Estado dos escravos CANopen. São 127 marcadores de Word, onde cada marcador está associado a um endereço na rede CANopen, e indica o estado do escravo no endereço: Bit 0: mestre contactou escravo com sucesso. Bit 1: download das configurações do escravo realizada com sucesso. Bit 2: controle de erros do escravo iniciado. Bit 3: fim da inicialização do escravo. Bit 4: detectado erro na inicialização do escravo. Bit 5: detectado erro no serviço de controle de erros do escravo. Bits 6 ... 15: reservado

Último Erro no Cliente SDO: conjunto de marcadores de leitura para informar dados sobre erros no cliente SDO. Caso alguma requisição seja feita ao cliente SDO e o escravo não responder, ou responder com erro, os dados relativos ao último erro detectado pelo cliente SDO são salvos nestes marcadores.	
Marcador	Descrição
%RS4128	Endereço do escravo destino, para o qual a requisição SDO foi enviada.
%RS4129	Índice do objeto acesso via SDO.
%RS4130	Sub-índice do objeto acessado.
%RS4131	Tipo de acesso realizado: 1 = leitura, 2 = escrita.
%RS4132 ... %RS4133	Para acessos de escrita, indica o valor escrito.
%RS4134 ... %RS4135	Indica o código do erro recebido, de acordo com os erros de comunicação via SDO da especificação do protocolo CANopen.

Último EMCY detectado: conjunto de marcadores de leitura para informar dados sobre erros reportados por produtores EMCY. O mestre CANopen não possui consumidor EMCY. Telegramas EMCY enviados por escravos da rede, no entanto, são capturados pelo mestre, e as informações do último EMCY detectado são salvas nestes marcadores.	
Marcador	Descrição
%RS4136	Endereço do escravo que reportou o EMCY.
%RS4137... % RS4140	Oito bytes de dados do telegrama EMCY, com informações sobre o código do erro reportado pelo escravo.

9.2 WORDS DE ESCRITA DE COMANDO

Controle do Mestre CANopen: conjunto de marcadores de escrita para controlar o mestre CANopen.	
Marcador	Descrição
%WC4142	Comando para controle do mestre CANopen e envio de telegrama NMT. Bits 0 ... 7: código do comando NMT: 1 = START 2 = STOP 128 = ENTER PRE-OPERATIONAL 129 = RESET NODE 130 = RESET COMMUNICATION Bit 8: toggle bit, sempre que o valor deste bit for alterado envia o comando programado. Bits 9 ... 14: reservado Bit 15: desabilita comunicação CANopen
%WC4143	Bits 16 ... 23: endereço do escravo destino para envio do comando NMT.

10 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO CANOPEN

A133/F33 – 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 F33, dependendo da programação feita no P0662. 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/ F34 – 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 elevado⁹, 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 F34, dependendo da programação feita no P0662. 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/F35 – 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.

⁹ Para mais informações sobre detecção de erros, consultar especificação CAN.

Neste caso será sinalizada através da HMI a mensagem de alarme A135 – ou falha F35, dependendo da programação feita no P0662. 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