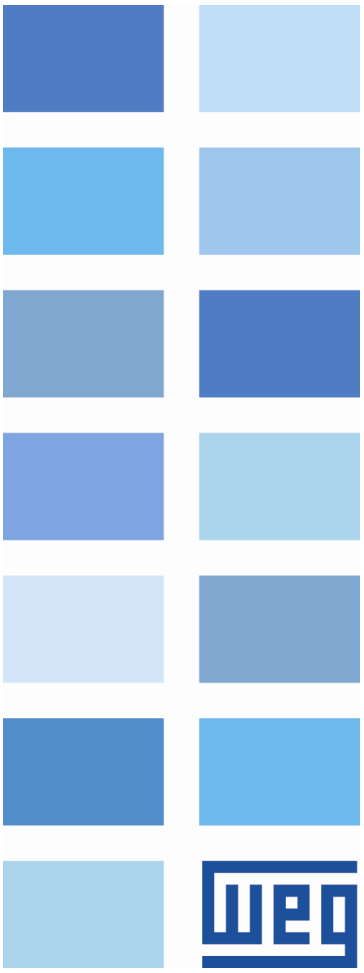
DeviceNet

SCA06

Manual do Usuário



Manual do Usuário - DeviceNet

Série: SCA06

Idioma: Português

Nº do Documento: 10003341880 / 00

Build 332

Data de publicação: 03/2015

Sumário

SOBRE O MANUAL

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

ABREVIAÇÕES E DEFINIÇÕES

ASCII American Standard Code for Information Interchange
CRC Cycling Redundancy Check
CiA CAN in Automation
CIP Common Industrial Protocol
PLC Programmable Logic Controller
HMI Human-Machine Interface
ODVA Open DeviceNet Vendor Association
ro Read only (solamente de lectura)
rw Read/write (lectura y escrita)

REPRESENTAÇÃO NUMÉRICA

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

DOCUMENTOS

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

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

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

1 INTRODUÇÃO À COMUNICAÇÃO DEVICENET

Para a operação de um equipamento em rede DeviceNet, é necessário conhecer a forma como a comunicação é feita. Este item traz uma descrição geral do funcionamento do protocolo DeviceNet, contendo as funções utilizadas pelo SCA06. Para uma descrição mais detalhada pode-se consultar a especificação do protocolo.

1.1 CAN

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

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

1.1.1 Frame de Dados

Os dados em uma rede CAN são transmitidos através de um frame de dados. Este tipo de frame é composto principalmente por um campo identificador de 11 bits[2] (arbitration field), e um campo de dados (data field), que pode conter até 8 bytes de dados.

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

1.1.2 Frame Remoto

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

1.1.3 Acesso à Rede

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

1.1.4 Controle de Erros

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

[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

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

1.1.5 CAN e DeviceNet

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

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

1.2 CARACTERÍSTICAS DA REDE DEVICENET

Apresentado em 1994, DeviceNet é uma implementação do protocolo Common Industrial Protocol (CIP) para redes de comunicação industrial. Desenvolvido originalmente pela Allen-Bradley, teve sua tecnologia transferida para a ODVA que, desde então, mantém, divulga e promove o DeviceNet e outras redes baseadas no protocolo CIP[3]. Além disso, utiliza o protocolo Controller Area Network (CAN) para enlace de dados e acesso ao meio, camadas 2 e 1 do modelo OSI/ISO, respectivamente.

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

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

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

1.3 MEIO FÍSICO

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

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

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

O tamanho total da rede varia de acordo com a taxa de transmissão utilizada, conforme mostrado na tabela 1.1

Tabela 1.1: *Tamanho da rede x Taxa de transmissão*

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

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

- 121Ω;
- 0,25 W;
- 1% de tolerância.

Em DeviceNet, diversos tipos de conectores podem ser utilizados, tanto selados quanto abertos. A definição do tipo a ser utilizado dependerá da aplicação e do ambiente de operação do equipamento. O SCA06 utiliza um conector de 5 vias cuja pinagem está mostrada na seção 3. Para uma descrição completa dos conectores utilizados pelo DeviceNet consulte a especificação do protocolo.

1.3.1 Camada de Enlace de Dados

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

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

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

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

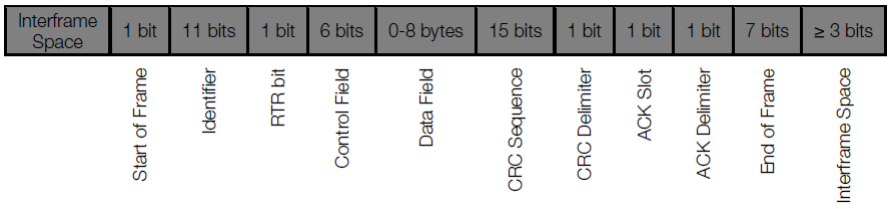


Figura 1.1: *Frame de dados CAN*

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

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

1.3.2 Camada de Transporte e Rede

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

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

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

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

1.3.3 Camada de Aplicação – Protocolo CIP

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

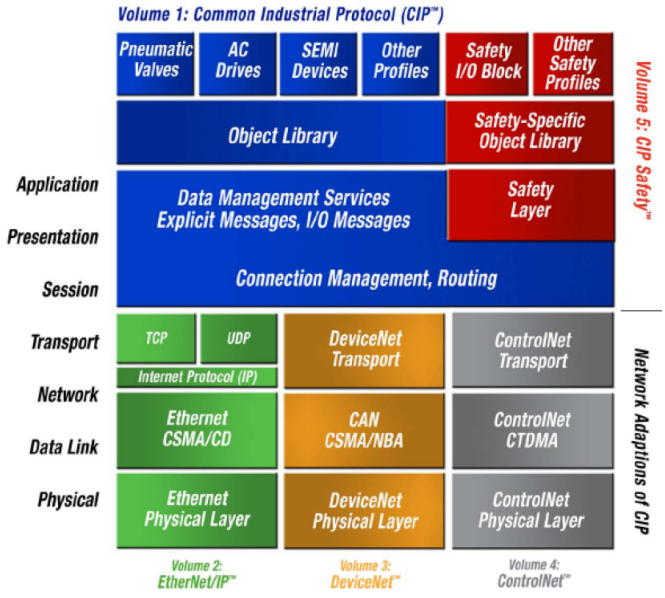


Figura 1.2: *Estrutura em camadas do protocolo CIP*

CIP tem dois objetivos principais:

- Transporte de dados de controle dos dispositivos de I/O;

- Transporte de informações de configuração e diagnóstico do sistema sendo controlado.

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

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

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

1.3.4 Arquivo EDS

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

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

1.3.5 Modos de Comunicação

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

- I/O: tipo de telegrama síncrono dedicado à movimentação de dados prioritários entre um produtor e um ou mais consumidores. Dividem-se de acordo com o método de troca de dados. Os principais são:
  - Polled: método de comunicação em que o mestre envia um telegrama a cada um dos escravos da sua lista (scan list). Assim que recebe a solicitação, o escravo responde prontamente a solicitação do mestre. Este processo é repetido até que todos sejam consultados, reiniciando o ciclo;
  - Bit-strobe: método de comunicação onde o mestre envia para a rede um telegrama contendo 8 bytes de dados. Cada bit destes 8 bytes representa um escravo que, se endereçado, responde de acordo com o programado;
  - Change of State: método de comunicação onde a troca de dados entre mestre e escravo ocorre apenas quando houver mudanças nos valores monitorados/controlados, até um certo limite de tempo. Quando este limite é atingido, a transmissão e recepção ocorrerão mesmo que não tenha havido alterações. A configuração desta variável de tempo é feita no programa de configuração da rede;
  - Cyclic: outro método de comunicação muito semelhante ao anterior. A única diferença fica por conta da produção e consumo de mensagens. Neste tipo, toda troca de dados ocorre em intervalos regulares de tempo, independente de terem sido alterados ou não. Este período também é ajustado no software de configuração de rede.
- Explicit: tipo de telegrama de uso geral e não prioritário. Utilizado principalmente em tarefas assíncronas tais como parametrização e configuração do equipamento.

NOTA!
o servoconversor SCA06 não disponibiliza o método de comunicação Bit-strobe.

1.3.6 Conjunto de Conexões Predefinidas Mestre/Escravo

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

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

2 INTERFACE DE COMUNICAÇÃO DEVICENET

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

2.1 CARACTERÍSTICAS DA INTERFACE CAN

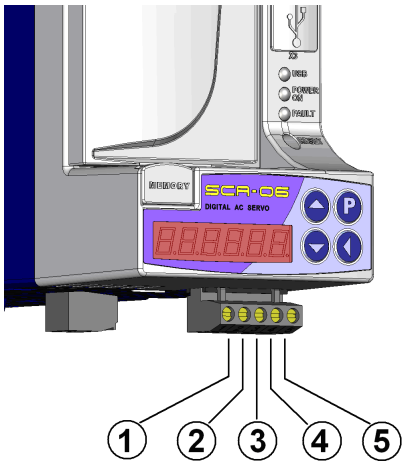


Figura 2.1: *Detalhe do conector CAN na parte inferior do produto*

- 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 repetidores[4].
- 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:

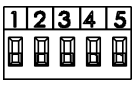


Tabela 2.1: *Pinagem do conector X4 para interface CAN*

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 co munição CAN_H
5	V+	Pólo positivo da fonte de alimentação

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

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.

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 DeviceNet para o servoconversor SCA06 são feitas através da HMI e dos parâmetros do produto.

3 INSTALAÇÃO EM REDE DEVICENET

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

3.1 TAXA DE COMUNICAÇÃO

Equipamentos com interface DeviceNet em geral permitem configurar a taxa de comunicação desejada, podendo variar de 125Kbit/s até 500Kbit/s. A taxa de comunicação (baud rate) que pode ser utilizada por um equipamento também depende do comprimento do cabo utilizado na instalação. A tabela a seguir apresenta a relação entre as taxas de comunicação e o comprimento máximo de cabo que pode ser utilizado na instalação, de acordo com o recomendado pela especificação do protocolo [5].

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

Taxa de comunicação	Comprimento do cabo
125 Kbps	500 m
250 Kbps	250 m
500 Kbps	100 m

Todos os equipamentos da rede devem ser programados para utilizar a mesma taxa de comunicação. Para o servoconversor SCA06, a taxa de comunicação CAN é programada através do parâmetro P0702.

3.2 ENDEREÇO NA REDE DEVICENET

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

Comprimento do Cabo (m)	Resistência por metro m (Ω|m)	Área do Condutor (mm2)
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

[5] Diferentes produtos podem apresentar variações no comprimento máximo do cabo suportado para a instalação.

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.

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.

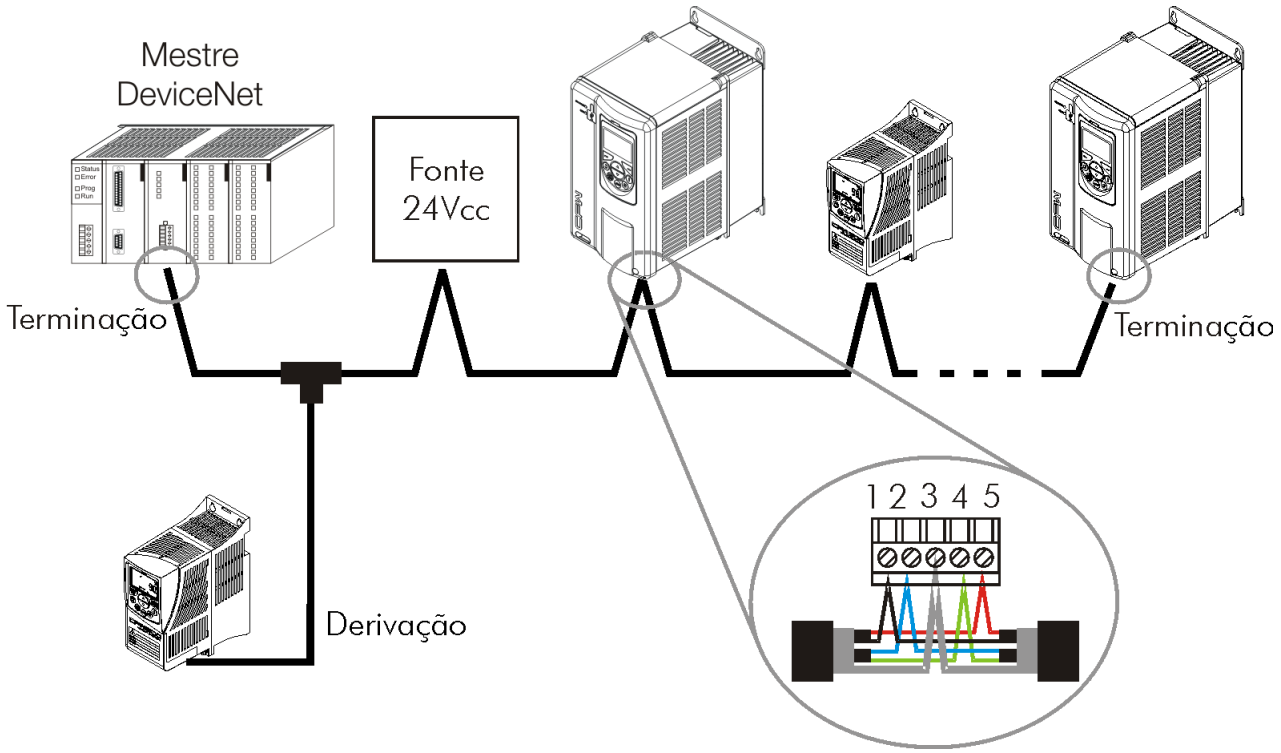


Figura 3.1: *Exemplo de instalação em rede DeviceNet*

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

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

4 PARAMETRIZAÇÃO

A seguir serão descritos os parâmetros do servoconversor SCA06 que possuem relação direta com a comunicação DeviceNet.

4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES

- **RO** Parâmetro somente de leitura
- **RW** Parâmetro de leitura e escrita
- **CFG** Parâmetro somente alterado com o motor parado

P0070 – ESTADO DO CONTROLADOR CAN

Faixa de Valores:	0 = Inativo 1 = Autobaud 2 = Interface CAN Ativa 3 = Warning 4 = Error Passive 5 = Bus Off 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 P0070.
1 = Autobaud	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 = Warning	Controlador CAN atingiu o estado de warning.
4 = Error PAssive	Controlador CAN atingiu o estado de error passive.
5 = Bus Off	Controlador CAN atingiu o estado de bus off.
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.

P0077 – ESTADO DA REDE DEVICENET

Faixa de Valores:	0 = Offline 1 = Online, Não Conectado 2 = Online, Conectado 3 = Conexão expirou 4 = Falha na Conexão 5 = Autobaud	**Padrão:** -
Propriedades:	RO	

Descrição:

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

Tabela 4.2: Valores para o parâmetro P0077

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

P0078 – ESTADO DO MESTRE DEVICENET

Faixa de Valores:	0 = RUN 1 = IDLE	**Padrão:** -
Propriedades:	RO	

Descrição:

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

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

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

P0202 – MODO DE OPERAÇÃO

Faixa de Valores:	1 = Modo Torque 2 = Modo Velocidade 3 = Reservado 4 = Modo Ladder 5 = CANopen/DeviceNet/EtherCAT 6 = Profibus DP	**Padrão:** 2
Propriedades:	RW	

Descrição:

Este parâmetro define o modo de operação do servoconversor SCA06, permitindo programar qual variável deseja-se controlar no motor e a fonte de comandos para execução das funções.

Para que o equipamento seja controlado através da rede DeviceNet, é necessário utilizar o modo 5 = DeviceNet. Caso este modo esteja programado, comandos e referências para operação do produto serão dados via rede DeviceNet.

NOTA!

- O controle do equipamento através dos objetos para drives somente é possível selecionando-se a opção desejada neste parâmetro, mas a comunicação DeviceNet pode ser utilizada em qualquer modo de operação.
- A interface DeviceNet permite o controle de velocidade e torque do servoconversor SCA06. Para realizar funções de posicionamento, deve-se utilizar o modo de operação Ladder, elaborando um programa aplicativo em ladder e utilizando parâmetros do usuário como interface com o mestre da rede para controle e monitoração do equipamento.

P0662 – AÇÃO PARA ERRO DE COMUNICAÇÃO

Faixa de Valores:	0 = Mostra Alarme 1 = Gera Falha 2 = Executa função STOP 3 = Desabilita drive	**Padrão:** 0
Propriedades:	RW	

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 4.3: *Opções para o parâmetro P0662*

Opção	Descrição
0 = Mostra Alarme	Apenas indica alarme na HMI em caso de erro de comunicação. Se a comunicação for restabelecida, a indicação de alarme é retirada automaticamente.
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 função STOP	Será feita a indicação de alarme juntamente com a execução do comando STOP. Para que o drive 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 DeviceNet:

- Alarme A133/Falha F233: Sem alimentação na interface CAN.
- Alarme A134/Falha F234: Bus off.
- Alarme A136/Falha F236: Mestre DeviceNet em Idle.
- Alarme A137/Falha F237: Timeout na conexão DeviceNet

P0700 – PROTOCOLO CAN

Faixa de Valores:	0 = Desabilitado 1 = CANopen 2 = DeviceNet 3 = CANespecial 1	**Padrão:** 0
Propriedades:	RW	

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 Valores:	0 a 127	**Padrão:** 63
Propriedades:	RW	

Descrição:

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

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

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

P0702 – TAXA DE COMUNICAÇÃO CAN

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

Descrição:

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

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

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

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

P0703 – RESET DE BUS OFF

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

Descrição:

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

***Tabela 4.4:** : Opções para o parâmetro P0703*

Opção	Descrição
0 = Reset Manual	Caso ocorra bus off, 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 bus off, 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.

P0710 – INSTÂNCIAS DE I/O DEVICENET

Faixa de Valores:	0 = ODVA Basic Speed (2 palavras) 1 = ODVA Extended Speed (2 palavras) 2 = ODVA Extended Speed/Torque (3 palavras) 3 = Especif.Fab 2W (2 palavras) 4 = Especif.Fab 3W (3 palavras) 5 = Especif.Fab 4W (4 palavras) 6 = Especif.Fab 5W (5 palavras) 7 = Especif.Fab 6W (6 palavras) 8 = Especif.Fab 7W (7 palavras) 9 = Especif.Fab 8W (8 palavras) 10 = Especif.Fab 9W (9 palavras) 11 = Especif.Fab 10W (10 palavras) 12 = Especif.Fab 11W (11 palavras) 13 = Especif.Fab 12W (12 palavras) 14 = Especif.Fab 13W (13 palavras) 15 = Especif.Fab 14W (14 palavras)	**Padrão:** 0
Propriedades:	RW	

Descrição:

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

O servoconversor SCA06 possui quinze opções de ajustes. Três delas seguem o padrão definido no perfil AC/DC Drive Profile da ODVA. As outras representam palavras específicas do servoconversor SCA06. As tabelas apresentadas a seguir detalham cada uma destas palavras de controle e monitoramento.

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

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

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

Monitoramento (Entrada)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running1		Faulted
	1	-							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Controle (Saída)

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

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

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

Monitoramento (Entrada)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
71	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Controle (Saída)

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

2 = Formato dos dados para as instâncias ODVA Extended Torque (3 palavras):

Chamada de Extended Torque, estas instâncias representam uma interface mutio semelhante à Extended Speed, tendo como única diferença a possibilidade de envio do limite de torque. O mapeamento dos dados é mostrado abaixo.

Monitoramento (Entrada)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
73	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							
	4	Torque Actual (low byte)							
	5	Torque Actual (high byte)							

Controle (Saída)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1	-							
	2	Speed Reference (low byte)							
	3	Speed Reference (high byte)							
	4	Torque Limit (low byte)							
	5	Torque Limit (low byte)							

A tabela a seguir apresenta o significado dos dados para as instâncias 20/70, 21/71 e 23/73.

Monitoramento:

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

- Byte 1 indica o estado do drive:
  - 0 = Non Existent
  - 1 = Startup
  - 2 = Not Ready
  - 3 = Ready
  - 4 = Enabled
  - 5 = Stopping
  - 6 = Fault Stop
  - 7 = Faulted
- Bytes 2 (low) e 3 (high) representam a velocidade real do motor em RPM.
- Bytes 4 (low) e 5 (high) representam o valor real da corrente do motor, proporcional ao torque.

Controle:

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

- Bytes 2 (low) e 3 (high) representam a referência de velocidade do motor em RPM.
- Bytes 4 (low) e 5 (high) representam a limitação da corrente real do motor, proporcional ao torque. Esse valor é escrito nos parâmetros P0131 e P0132, limite de corrrente negativo e positivo respectivamente.

3 = Formato dos dados para as instâncias Manufacturer Specific 2W (2 palavras):
4 = Formato dos dados para as instâncias Manufacturer Specific 3W (3 palavras):
5 = Formato dos dados para as instâncias Manufacturer Specific 4W (4 palavras):
6 = Formato dos dados para as instâncias Manufacturer Specific 5W (5 palavras):
7 = Formato dos dados para as instâncias Manufacturer Specific 6W (6 palavras):
8 = Formato dos dados para as instâncias Manufacturer Specific 7W (7 palavras):
9 = Formato dos dados para as instâncias Manufacturer Specific 8W (8 palavras):
10 = Formato dos dados para as instâncias Manufacturer Specific 9W (9 palavras):
11 = Formato dos dados para as instâncias Manufacturer Specific 10 (10 palavras):
12 = Formato dos dados para as instâncias Manufacturer Specific 11W (11 palavras):
13 = Formato dos dados para as instâncias Manufacturer Specific 12W (12 palavras):
14 = Formato dos dados para as instâncias Manufacturer Specific 13W (13 palavras):
15 = Formato dos dados para as instâncias Manufacturer Specific 14W (14 palavras):

Chamada de Manufacturer Specific, estas instâncias representam a mais simples interface de operação do equipamento segundo o perfil do servoconversor SCA06. O mapeamento dos dados é mostrado abaixo. Além das palavras de controle e estado, referência e valor atual da velocidade, é possível programar até 24 parâmetros do próprio equipamento para leitura e/ou escrita via rede, através dos parâmetros P0711 até P0734.

Monitoramento (Entrada)

Tabela 4.5: *Programação das palavras de I/O*

	Instância	Palavras de 16 bits (word)	Função
Fixo		#1	Palavra de estado (P0736)
		#2	Velocidade/corrente Atual(*)
Programável		#3	Leitura #3 DeviceNet
	150	#4	Leitura #4 DeviceNet
		#5	Leitura #5 DeviceNet
		⋮	⋮
		#14	Leitura #14 DeviceNet

Opção do P0710: 3, 4, 5, 6, 15

Controle (Saída)

Tabela 4.6: *Programação das palavras de I/O*

	Instância	Palavras de 16 bits (word)	Função
Fixo	100	#1	Palavra de controle (P0735)
		#2	Referência velocidade/corrente(*)
Programável		#3	Escrita #3 DeviceNet
		#4	Escrita #4 DeviceNet
		#5	Escrita #5 DeviceNet
		⋮	⋮
		#14	Escrita #14 DeviceNet

Opção do P0710: 3, 4, 5, 6, 15

* O conteúdo desta palavra é definido conforme o modo de operação programado nos bits 8 a 10 na palavra de controle (P0735):

- Modo velocidade: a referência de velocidade e a velocidade do motor é informada em RPM.
- Modo torque: a referência de corrente (proporcional ao torque) e a corrente do motor é informada conforme escala do P0003.

P0711 – LEITURA #3 DEVICENET

P0712 – LEITURA #4 DEVICENET

P0713 – LEITURA #5 DEVICENET

P0714 – LEITURA #6 DEVICENET

P0715 – LEITURA #7 DEVICENET

P0716 – LEITURA #8 DEVICENET

P0717 – LEITURA #9 DEVICENET

P0718 – LEITURA #10 DEVICENET

P0719 – LEITURA #11 DEVICENET

P0720 – LEITURA #12 DEVICENET

P0721 – LEITURA #13 DEVICENET

P0722 – LEITURA #14 DEVICENET

Faixa de Valores:	0 a 1249	**Padrão:** 0
Propriedades:	RW	

Descrição:

Estes parâmetros permitem programar o conteúdo das palavras 3 a 14 de entrada (input: escravo envia para o mestre). Utilizando estes parâmetros, é possível programar o número de outro parâmetro cujo conteúdo deve ser disponibilizado na área de entrada do mestre da rede.

Por exemplo, caso se deseje ler do drive a corrente do motor em amperes, deve-se programar em algum dos parâmetros o valor 3, pois o parâmetro P0003 é o parâmetro que contém esta informação. Vale lembrar que o valor lido de qualquer parâmetro é representado com uma palavra de 16 bits. Mesmo que o parâmetro possua resolução

decimal, o valor é transmitido sem a indicação das casas decimais. Por exemplo, se o parâmetro P0003 possuir o valor 4.7 A, o valor fornecido via rede será 47

Estes parâmetros são utilizados somente se o drive for programado no parâmetro P0710 para utilizar as opções 3 até 15. De acordo com a opção selecionada, são disponibilizadas até 14 palavras para leitura pelo mestre da rede.

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

P0723 – ESCRITA #3 DEVICENET

P0724 – ESCRITA #4 DEVICENET

P0725 – ESCRITA #5 DEVICENET

P0726 – ESCRITA #6 DEVICENET

P0727 – ESCRITA #7 DEVICENET

P0728 – ESCRITA #8 DEVICENET

P0729 – ESCRITA #9 DEVICENET

P0730 – ESCRITA #10 DEVICENET

P0731 – ESCRITA #11 DEVICENET

P0732 – ESCRITA #12 DEVICENET

P0733 – ESCRITA #13 DEVICENET

P0734 – ESCRITA #14 DEVICENET

Faixa de Valores:	0 a 1249	**Padrão:** 0
Propriedades:	RW	

Descrição:

Estes parâmetros permitem programar o conteúdo das palavras 3 a 14 de saída (output: mestre envia para o escravo). Utilizando estes parâmetros, é possível programar o número de outro parâmetro cujo conteúdo deve ser disponibilizado na área de saída do mestre da rede.

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

Estes parâmetros são utilizados somente se o drive for programado no parâmetro P0710 para utilizar as opções 3 até 15. De acordo com a opção selecionada, são disponibilizadas até 14 palavras para escrita pelo mestre da rede.

As duas primeiras palavras de saída são fixas.

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

P0735 – PALAVRA DE CONTROLE

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

Descrição:

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

Para que os comandos escritos neste parâmetro sejam executados, é necessário que o equipamento esteja programado para ser controlado via DeviceNet. Esta programação é feita através do parâmetro P0202.

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

Bits	15 a 11	10 a 8	7	6	5 a 3	2	1	0
Função	Reservado	Modo de operação	Reset Falha	Reset Drive	Reservado	Sentido de Giro	STOP	Habilita

Bit	Valor/Descrição
Bit 0 Habilita	**0**: Desabilita drive. **1**: Habilita drive.
Bit 1 STOP	**0**: Não executa função STOP **1**: Executa função STOP.
Bit 2 Sentido de Giro	**0**: Forward **1**: Resverse
Bits 3 a 5	Reservado.
Bit 6 Reset Drive	**0**: sem função **1**: Executa Reset do drive.
Bit 7 Reset de Falha	**0**: sem função **1**: Se em estado de falha, executa Reset da falha.
Bits 8 a 10 Modo de Opeação	**1**: Modo Torque **2**: Modo Velocidade
Bits 11 a 15	Reservado.

P0736 – ESTADO LÓGICO

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

Descrição:

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

Bits	15 a 11	10 a 8	7	6	5 a 3	2	1	0
Função	Reservado	Modo de operação	Falha	Atingiu Ref.	Reservado	Sentido de Giro	STOP	Habilitado

Bit	Valor/Descrição
Bit 0 Habilitado	**0**: Drive não está habilitado. **1**: Drive está habilitado.
Bit 1 STOP	**0**: Sem função **1**: Executando a função STOP.
Bit 2 Sentido de Giro	**0**: Horário **1**: Anti-horário
Bits 3 a 5	Reservado.
Bit 6 Atingiu Ref.	**0**: Valor da referência não atingido. **1**: Valor da referência atingido.
Bit 7 Falha	**0**: Drive sem falha **1**: Drive está em falha.
Bits 8 a 10 Modo de Opeação	**1**: Modo Torque **2**: Modo Velocidade
Bits 11 a 15	Reservado.

5 CLASSE DE OBJETOS SUPORTADAS

Todo dispositivo DeviceNet é modelado por um conjunto de objetos. São eles os responsáveis por definir que funções determinado equipamento terá. Ou seja, de acordo com os objetos implementados, este equipamento poderá ser um adaptador de comunicação, um drive AC/DC, um sensor fotoelétrico, etc.. Objetos obrigatórios e opcionais são definidos em cada um destes perfis de dispositivos (Device Profile). O servoconversor SCA06 suporta todas as classes obrigatórias do perfil AC/DC Drive Profile. Suporta também classes específicas do fabricante. Detalhes de cada uma delas são apresentados nas seções a seguir.

5.1 CLASSE IDENTITY (01H)

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

Tabela 5.1: *Atributos da instância da classe Identity*

Atributo	Método	Nome	Padrão	DescriçãoBit
1	GET	Vendor ID	355h	Identificador do Fabricante
2	GET	Product Type	2h	Tipo do produto
3	GET	Product Code		Código do produto
4	GET	Vendor Revision		Revisão do firmware
5	GET	Status		Estado atual do dispositivo
6	GET	Serial Number		Número serial
7	GET	Product Name	SCA-06	Nome do produto

5.2 CLASSE MESSAGE ROUTER (02H)

Fornece informações sobre o objeto roteador de mensagens do tipo explicit. No SCA06, esta classe não possui qualquer atributo implementado.

5.3 CLASSE DEVICENET (03H)

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

Tabela 5.2: *Atributos da classe DeviceNet*

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

Tabela 5.3: *Atributos da instância da classe DeviceNet*

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

5.4 CLASSE ASSEMBLY (04H)

Classe cuja função é agrupar diversos atributos numa única conexão. No SCA06 apenas o atributo Data (3) está implementado (5.4).

Tabela 5.4: *Atributos das instâncias da classe Assembly*

Atributo	Método	Nome	Descrição
3	GET/SET	Data	Dados da instância.

No SCA06, a classe Assembly contém as seguintes instâncias:

Tabela 5.5: *Atributos das instâncias da classe Assembly*

Instância	Tamanho	Descrição
20	2 palavras	ODVA Basic Speed Control Output.
70	2 palavras	ODVA Basic Speed Control Output.
21	2 palavras	ODVA Extended Speed Control Output.
71	2 palavras	ODVA Extended Speed Control Output.
23	3 palavras	ODVA Extended Speed/Torque Control Output.
73	3 palavras	ODVA Extended Speed/Torque Control Output.
100	2 - 14 palavras	Especifica do fabricante Saída
150	2 - 14 palavras	Especifica do fabricante Entrada

5.5 CLASSE CONNECTION (05H)

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

5.5.1 Instância 1: Explicit Message

Tabela 5.6: *Classe Connection – Instância 1: Explicit Message*

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

5.5.2 Instância 2: Polled

Tabela 5.7: Classe Connection – Instância 2: Polled

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

5.5.3 Instância 4: Change of State/Cyclic

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

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

5.6 CLASSE MOTOR DATA (28H)

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

Tabela 5.9: Atributos da classe Motor Data

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

Tabela 5.10: *Atributos da instância da classe Motor Data*

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

5.7 CLASSE MOTOR DATA (29H)

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

Tabela 5.11: *Atributos da classe Control Supervisor*

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

Tabela 5.12: *Atributos da instância da classe Control Supervisor*

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

5.8 CLASSE AC/DC DRIVE (2AH)

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

Tabela 5.13: Atributos da classe AC/DC Drive

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

Tabela 5.14: Atributos da instância da classe Control Supervisor

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

NOTA!
O SCA06 irá operar em modo de velocidade independente do conteúdo do atributo DriveMode.

5.9 CLASSE ACKNOWLEDGE HANDLER (2BH)

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

Tabela 5.15: Atributos da instância da classe Acknowledge Handler

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

5.10 CLASSE ESPECIFICAS DO FABRICANTE

As classes específicas do fabricante são utilizadas para mapear todos os parâmetros do SCA06. Elas permitem que o usuário leia e escreva em qualquer parâmetro através da rede. Para isto mensagens DeviceNet do tipo explicit são usadas. Existem faixas separadas para cada grupo de parâmetros, conforme mostrado na tabela 5.16:

Tabela 5.16: Classes específicas do fabricante

Classe	Nome	Faixa de valores
Classe 100 (64h)	VENDOR CLASS F1	Parametro 000 - 099
Classe 101 (65h)	VENDOR CLASS F2	Parametro 100 - 199
Classe 102 (66h)	VENDOR CLASS F3	Parametro 200 - 299
Classe 103 (67h)	VENDOR CLASS F4	Parametro 300 - 399
Classe 104 (68h)	VENDOR CLASS F5	Parametro 400 - 499
Classe 105 (69h)	VENDOR CLASS F6	Parametro 500 - 599
Classe 106 (6Ah)	VENDOR CLASS F7	Parametro 620 - 699
Classe 107 (6Bh)	VENDOR CLASS F8	Parametro 700 - 799
Classe 108 (6Ch)	VENDOR CLASS F9	Parametro 800 - 899
Classe 109 (6Dh)	VENDOR CLASS F10	Parametro 900 - 999
Classe 110 (6Eh)	VENDOR CLASS F11	Parametro 1000 - 1099
Classe 111 (6Fh)	VENDOR CLASS F12	Parametro 1100 - 1199

Tabela 5.17: *Parâmetros das classes específicas do fabricante*

Parâmetro	Classe	Instância	Atributo
P0000	Classe 100 (64h)	1	100
P0001	Classe 100 (64h)	1	101
P0002	Classe 100 (64h)	1	102
...	...	...	...
P0100	Classe 101 (65h)	1	100
P0101	Classe 101 (65h)	1	101
P0102	Classe 101 (65h)	1	102
...	...	...	...
P0200	Classe 102 (66h)	1	100
P0201	Classe 102 (66h)	1	101
P0202	Classe 102 (66h)	1	102
...	...	...	...
P0300	Classe 103 (67h)	1	100
P0301	Classe 103 (67h)	1	101
P0302	Classe 103 (67h)	1	102
...	...	...	...

NOTA!

- Para estas classes, o SCA06 utiliza apenas a instância 1.
- Para estas classes, os parâmetros são acessados adicionando o valor decimal 100 aos dígitos da dezena de qualquer parâmetro. Este novo número é chamado de atributo.

Por exemplo:

Parâmetro 23: classe 64h, instância 1, atributo 123. Este caminho dá acesso ao P0023.
Parâmetro 100: classe 65h, instância 1, atributo 100. Este caminho dá acesso ao P0100.
Parâmetro 202: classe 66h, instância 1, atributo 102. Este caminho dá acesso ao P0202.

6 FALHAS E ALARMES

A133/F233 - SEM ALIMENTAÇÃO NA INTERFACE CAN

Descrição:

Indica que a interface CAN não possui alimentação entre os pinos 6 e 10 do conector.

Atuação:

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

Se a interface CAN estiver alimentada e for detectada a falta de alimentação na interface CAN, será sinalizada através da HMI a mensagem de alarme A133 – ou falha F233, dependendo da programação feita no P313. 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 6 e 10 do conector da interface CAN.
- Verificar se os cabos de alimentação não estão trocados ou invertidos.
- Verificar problemas de contato no cabo ou no conector da interface CAN.

A134/F234 - BUS OFF

Descrição:

Detectado erro de bus off na interface CAN.

Atuação:

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

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

Possíveis Causas/Correção:

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

A136/F236 - MESTRE EM IDLE

Descrição:

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

Atuação:

Atua quando o SCA06 detectar que o mestre da rede foi para o modo Idle. Neste modo, apenas as variáveis lidas do escravo continuam sendo atualizadas na memória do mestre. Nenhum dos comandos enviados ao escravo é processado.

Neste caso será sinalizada através da HMI a mensagem de alarme A136 – ou falha F236 dependendo da programação feita no P0313.

Possíveis Causas/Correção:

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

A137/F237 - TIMEOUT NA CONEXÃO DEVICENET

Descrição:

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

Atuação:

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

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

Possíveis Causas/Correção:

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

WEG 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