

A photograph of a large industrial facility at night, with numerous illuminated structures and pipes against a dark sky.

Rede
CANopen

Rodrigo Borges

CANopen®

& CAN

Apresentação



CANopen®

- CANopen é um protocolo de comunicação baseado em CAN, do qual a organização CAN in Automation (CiA) cuida desde 1995 e é definido como um padrão pela norma europeia EN 50325-4.
- O protocolo foi desenvolvido como uma rede incorporada padronizada com opções de configuração altamente flexíveis.

CANopen®

Mas será que vale a pena estudar um protocolo serial ainda hoje?



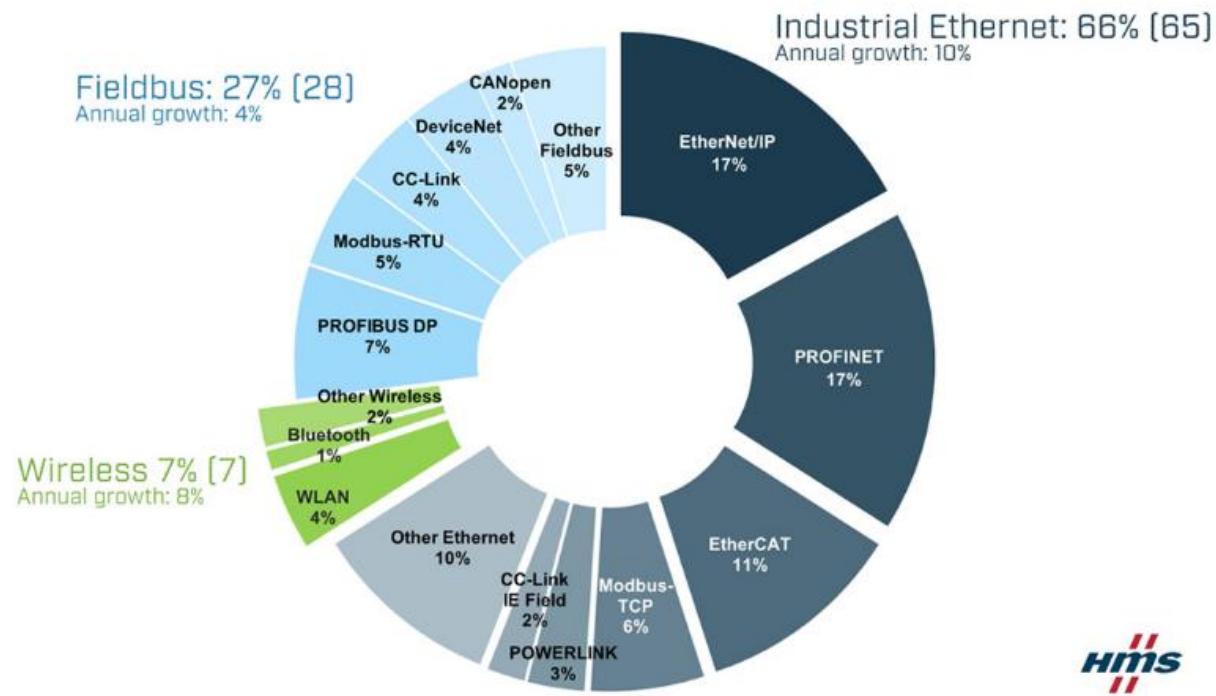
CANopen®

- Talvez você esteja se perguntando exatamente isso agora.
- Vamos analisar juntos a resposta!!!!

CANopen®

Mas será que vale a pena estudar um protocolo serial ainda hoje?

2022 Industrial Network Market Shares according to HMS Networks



- CANopen é um protocolo de comunicação industrial de relevância mundial!!!!



canopen®

Mas será que vale a pena estudar um protocolo serial ainda hoje?



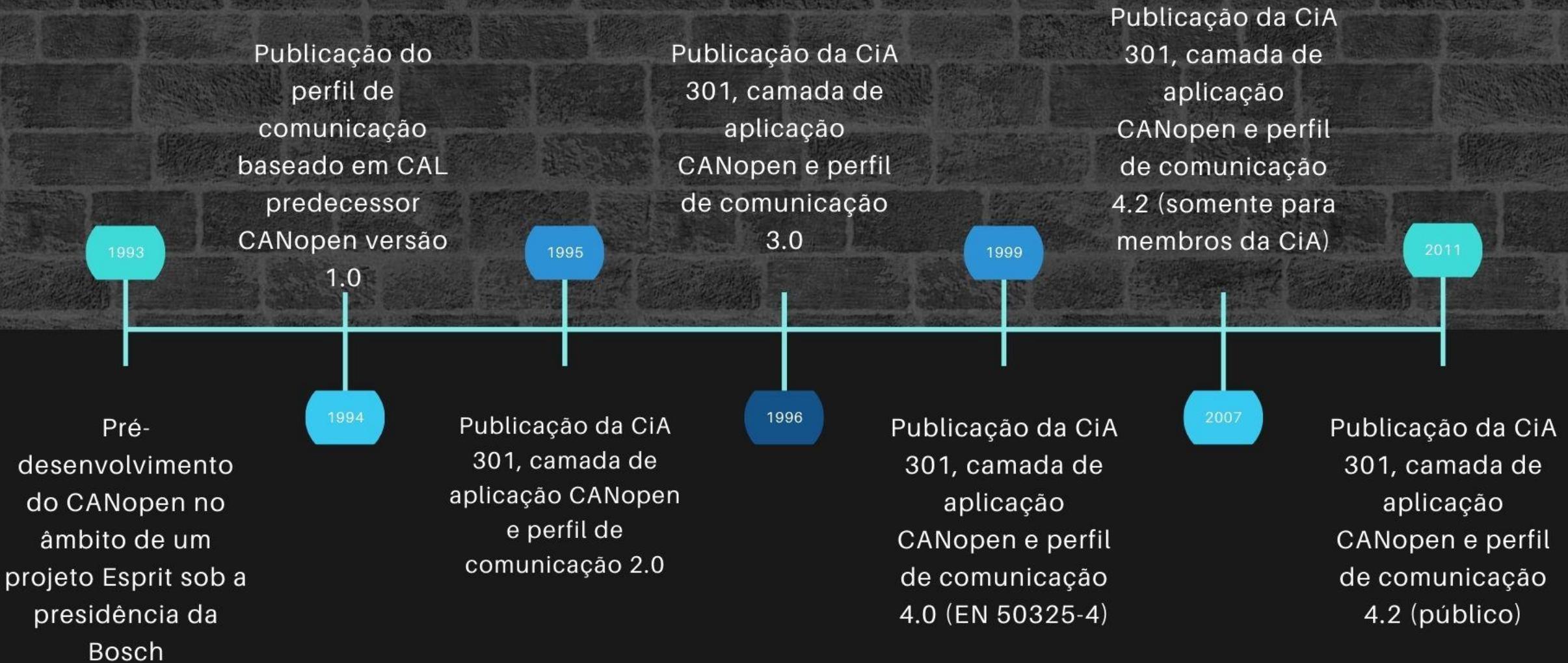
Redes de alto desempenho da nova geração trabalham baseadas no CANopen!!!



Resumindo: CANopen ainda é uma rede muito importante!



História do CANopen



CiA – CAN in AUTOMATION

- **CAN in Automation** (CiA)391 é a organização internacional de usuários e fabricantes que desenvolve e oferece suporte a protocolos de camada superior baseados em [CAN](#) .
- Cerca de 745 (2023) empresas de todo o mundo aderiram à organização sem fins lucrativos.
- As especificações CAN são desenvolvidas dentro dos grupos de interesse da CiA (IG) em cooperação com funcionários das empresas membros.
- Os seguintes IGs foram estabelecidos: IG [CANopen](#) , IG CANopen FD, IG profiles, IG layer 1/2, IG safety/security, IG [J1939](#) .
- Os grupos de interesse gerenciam seus grupos de interesse especiais (SIG) relacionados. Esses grupos de interesse especial (SIG) desenvolvem, por exemplo, especificações e recomendações CiA dedicadas (por exemplo, perfis de dispositivos e aplicativos para CANopen e CANopen FD).
- Os representantes da CiA apóiam muitas atividades internacionais de padronização ([ISO](#) , [IEC](#) , [CEN](#) , [Cenelec](#) e [SAE](#)), lidando com o CAN. Além disso, a CiA publica a revista técnica CAN Newsletter (arquivo PDF para download), bem como a CAN Newsletter Online.

CiA – CAN in AUTOMATION

Lenze



BOSCH



BALLUFF

BECKHOFF

**Bihl
+ Wiedemann**
...

Danfoss

FESTO

**Schneider
Electric**

**SEW
EURODRIVE**

ABB



O que vem a ser o CANopen afinal?

- CANopen é um protocolo de comunicação baseado em CAN.
- O padrão CANopen é útil, pois permite a interoperabilidade padrão entre dispositivos (nós) em, por exemplo, maquinário industrial. Além disso, fornece métodos padrão para configurar dispositivos, também após a instalação.
- O CANopen foi originalmente projetado para sistemas de controle de máquinas orientados a movimento.
- Hoje, o CANopen é amplamente utilizado no controle de motores (passo/servomotores) - mas também em uma ampla gama de outras aplicações:



Robótica automatizada, correias
transportadoras e outras máquinas
industriais

Geradores de raios X, injetores,
mesas de pacientes e
dispositivos de diálise





Agricultura, ferrovia, reboques, serviço
pesado , mineração , marinha e muito mais...



CANopen no contexto do modelo OSI

- O CANopen é um "protocolo de camada superior" baseado no barramento CAN.
- Isso significa que o barramento CAN (ISO 11898) serve como 'veículo de transporte' (como um caminhão) para mensagens CANopen (como contêineres).

CANopen - protocolo de camada superior



CiA 401, 402, ...

Device profiles (I/O, ...)

CiA 303-2 (CANopen)

Managers & program. devices

CiA 303-1 (CANopen)

Application & communication

canopen[®]

ISO 11898-1 (CAN)

ISO 11898-2 (CAN)

Por que escolher a rede CANopen?

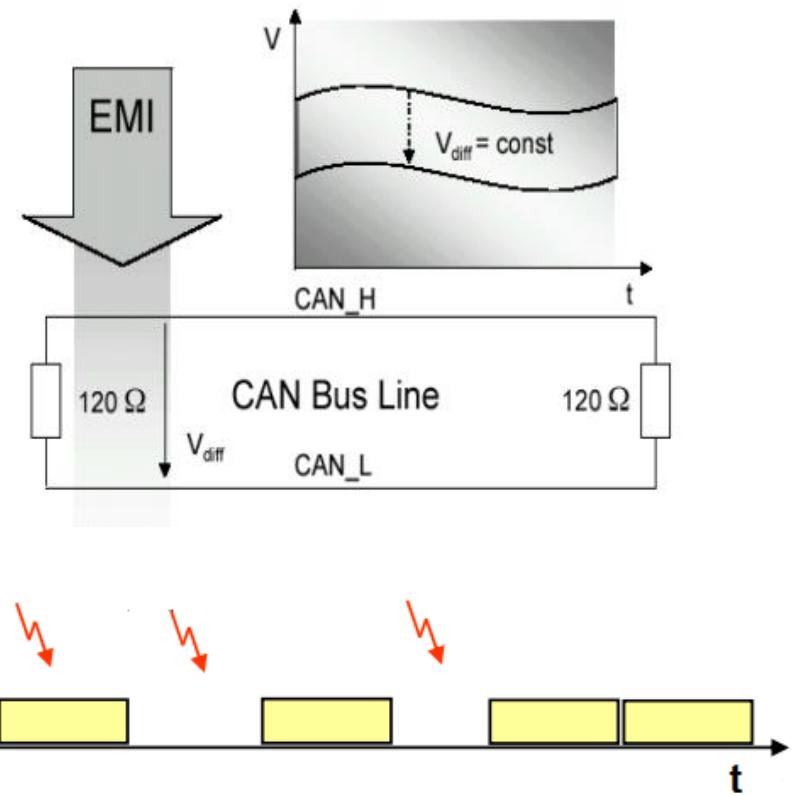
- Devido as Características Técnicas desta rede:
- Suporta ambientes com níveis de EMI elevados , característica intrínseca de imunidade
- Performance da Comunicação
- Acesso não destrutivo ao bus (Non-destructive bus access)
- Detecção de Erro e Localização

CANopen®



Ambientes com elevados níveis EMI

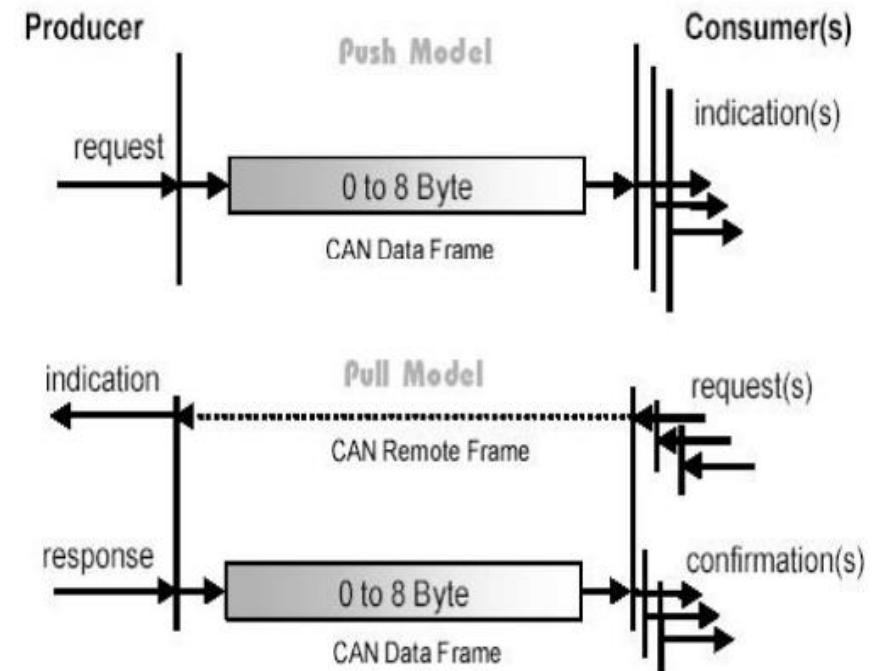
- Efeitos de扰动 EMI são minimizados.
- Linha de ground (GND) separada.
- Sinal com transmissão diferencial.
- Short data frames.
- Dados continuam sendo transmitidos em ambientes onde existem níveis de扰动 eletrônicos elevados.



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

Adaptação da Comunicação

- CANopen utiliza o modelo produtor/consumidor.
- Incrementa a eficiência.
- Comunicação direta peer-to-peer sem necessidade de sincronização pelo Mestre
- Transmissão de parâmetros e dados de configuração.



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

Performance de Comunicação

- Tempo de Resposta excelente
- <1 ms para 256 digital I/Os – 1Mbps
- 5 ms para 128 analog I/Os – 1Mbps
- Priorização de mensagens
- Processo de transmissão de dados Flexivel
- Change-of-state: transmissão somente quando requerido
- Cyclic: Base tempo pode ser definida para cada valor
- Broadcast: Transmissão simultanea a todos os dispositivos conectados a rede CANopen

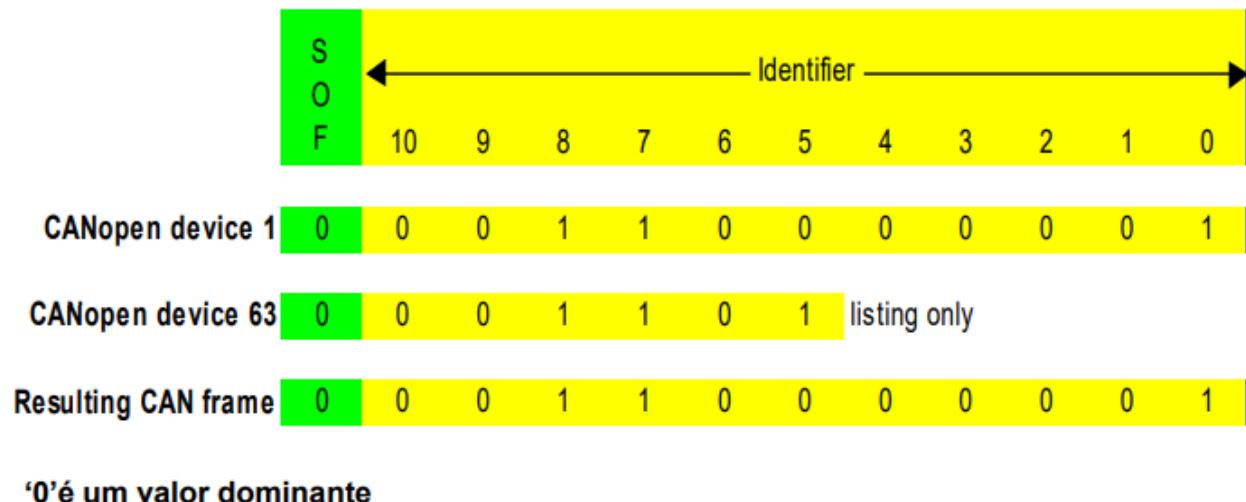


CANopen®

Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

Acesso não destrutivo ao bus

- Mensagens préias serão sempre transmitidas
- Não perde telegrama no caso de colisão
- Não é necessário retransmissão



Fonte: Redes & Arquiteturas CANopen Rede para
Máquinas e Instalações, Schneider electric

Erro detecção e Localização

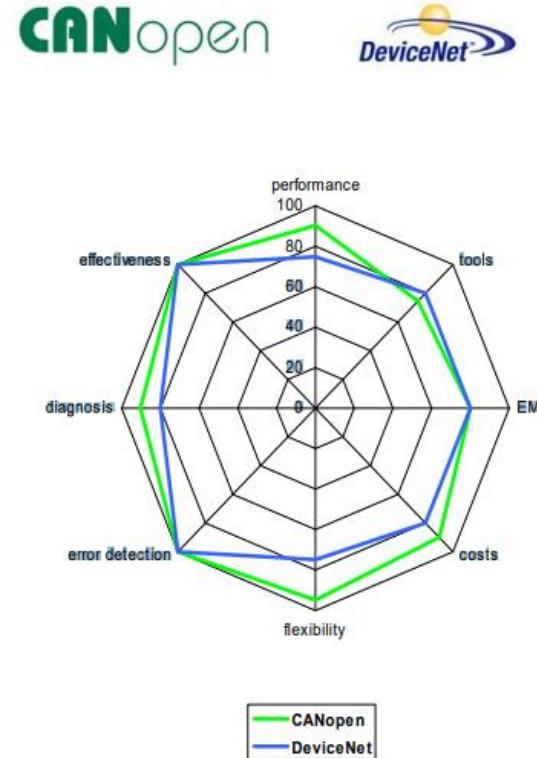
- Detecção Erro de Frame com alto grau de confiabilidade.
- 1 Erro não detectado a cada 1000 anos.
- Implementar Histórico de Erro
- Watchdog monitora a disponibilidade do dispositivo
- Informação detalhada de Diagnóstico



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

CANopen vs. DeviceNet

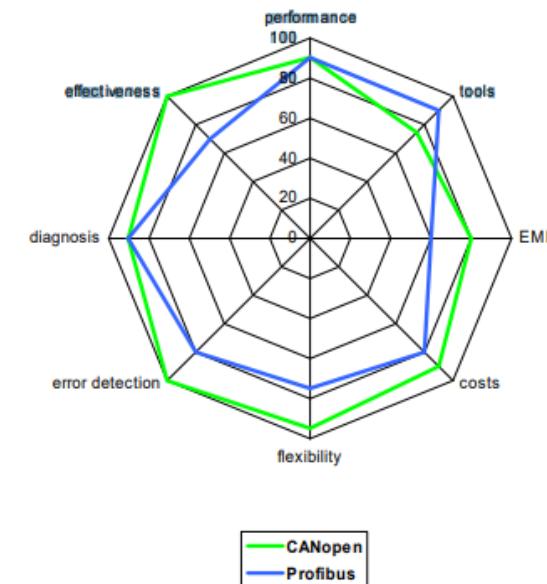
- OS benefícios da rede CANopen são:
- Performance
- Flexibilidade
- Diagnósticos
- Custo
- CANopen e DeviceNet são baseadas em CAN



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

CANopen vs. Profibus

- OS benefícios da rede CANopen são:
- Efetividade
- performance
- Imunidade EMI
- Flexibilidade
- Detecção de Erros
- Custo



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

CANopen: A escolha correta

- Devido aos benefícios ao usuário
- Armários menores podem ser utilizados
- Pode ser adaptada as suas necessidades
- Toda performance e flexibilidade
- Confiabilidade na transmissão de dados
- Melhora dos tempos de diagnóstico e recolocação em funcionamento



Fonte: Redes & Arquiteturas CANopen Rede para Máquinas e Instalações, Schneider electric

Conceitos básicos do CANopen

- Até aqui aprendemos que:
- O CANopen é um protocolo de comunicação baseado em redes CAN.
- Ele originalmente foi projetado para sistemas de controle de máquinas orientadas a movimento.
- É muito utilizado em controle de motores, motores de passo e servomotores.
- Habilita interoperabilidade transparente entre dispositivos.
- É um protocolo complexo porém poderoso.



CANopen®

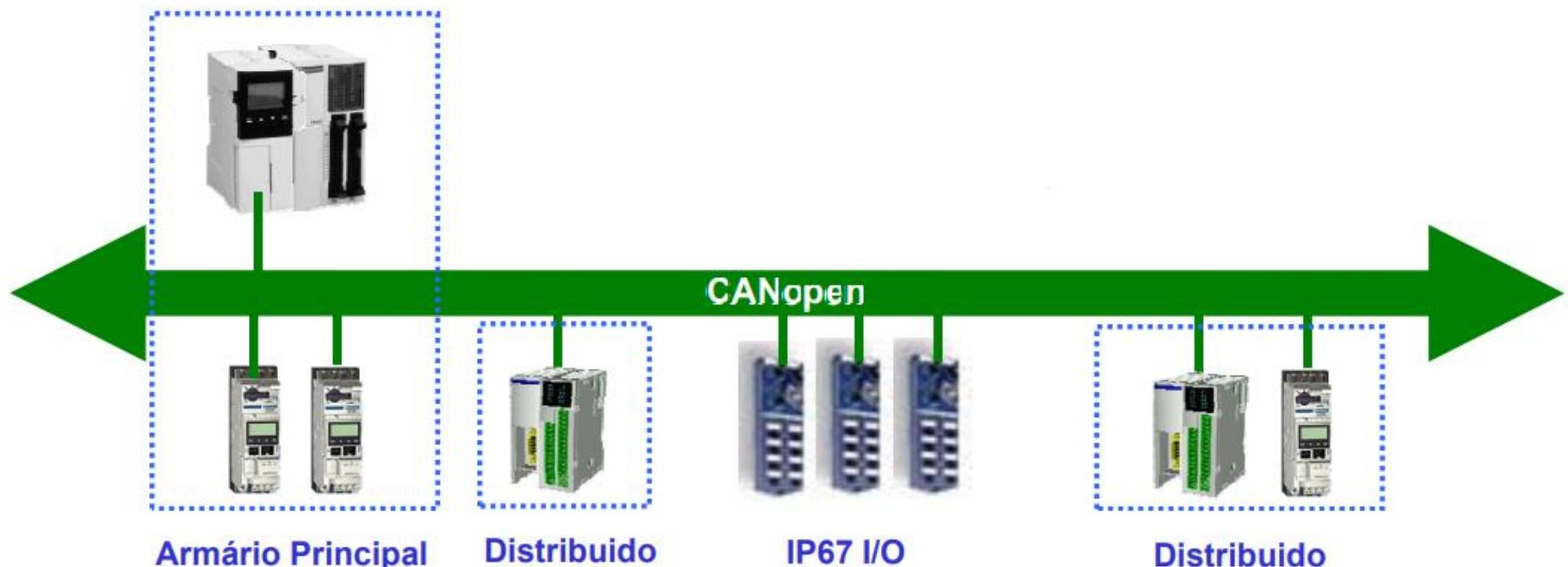


CANopen em Arquiteturas de Máquinas



Transportadores

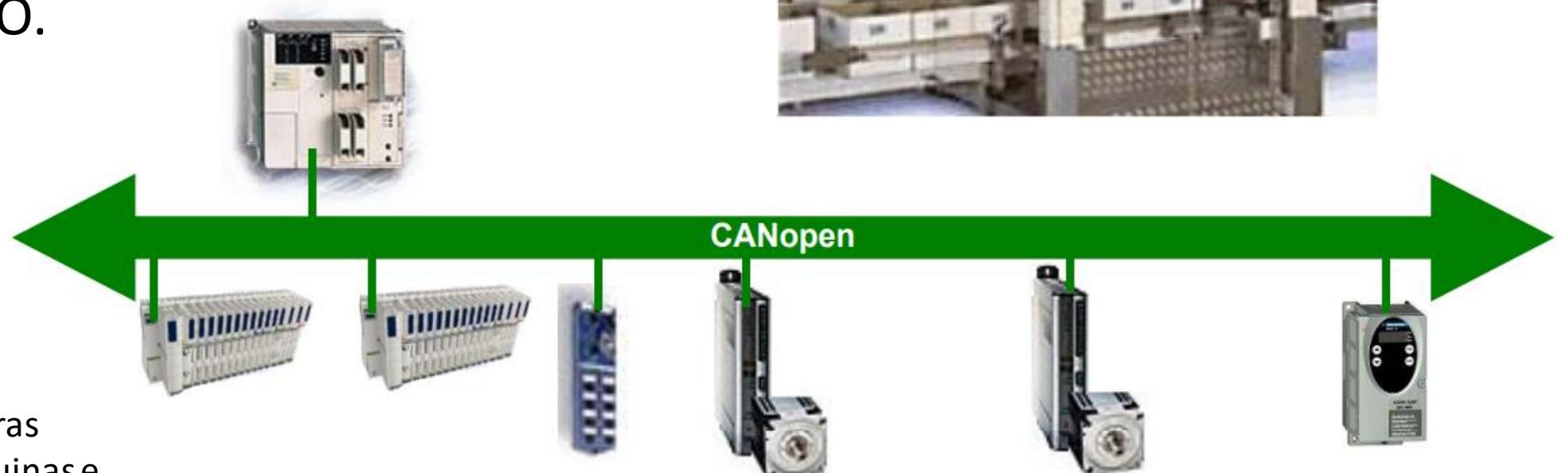
- Controle:
- CLP de pequeno porte.
- Tempo de Reação \approx 50 -100ms.
- Partidas de Motores CANopen.
- IP20 e IP67 I/O.



Fonte: Redes & Arquiteturas
CANopen Rede para Máquinas e
Instalações, Schneider electric

Máquina embalagem vertical

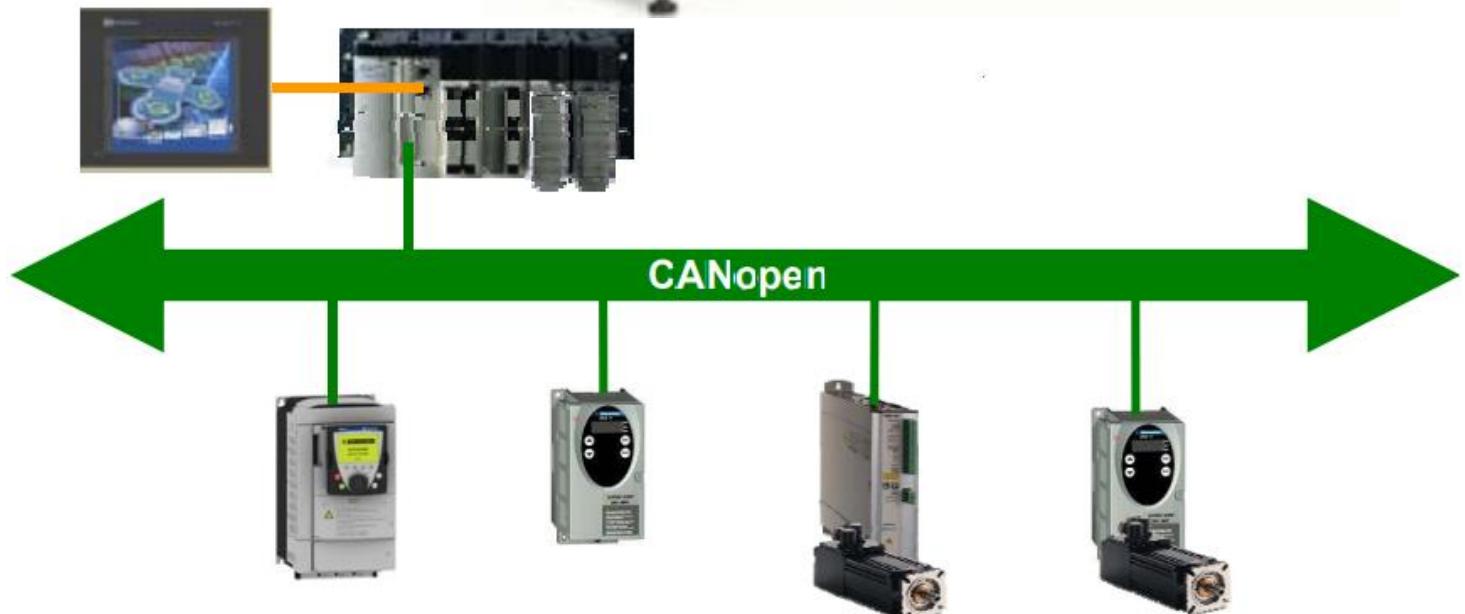
- Controle:
- CLP de tamanho médio.
- Tempo de Reação \approx 30ms.
- Drives & Motion.
- IP67 e IP20 I/O.



Fonte: Redes & Arquiteturas
CANopen Rede para Máquinas e
Instalações, Schneider electric

Máquina embalagem Horizontal

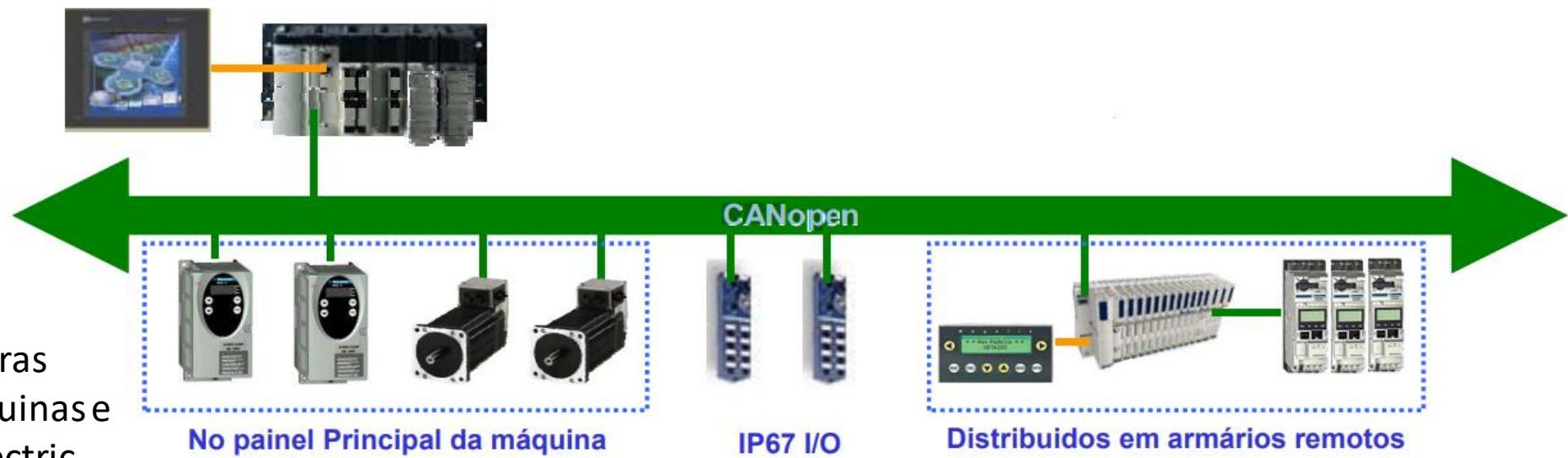
- Controle:
- CLP Potente.
- Tempo de Reação $\approx 10\text{ms}$.
- Drives & Motion.
- Fácil expandir e modificar , incluir novos dispositivos.



Fonte: Redes & Arquiteturas
CANopen Rede para Máquinas e
Instalações, Schneider electric

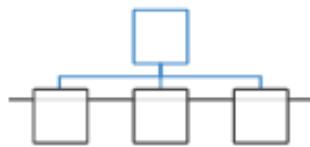
Máquina processamento Madeira

- Controle e Visualização:
- Tempo de reação $\approx 3\text{ms}$.
- Drives.
- IP67 I/O e Armários Distribuidos.



Seis conceitos básicos do CANopen

- Mesmo se você estiver familiarizado com o barramento CAN e, por exemplo, J1939, o CANopen adiciona uma série de novos conceitos importantes:



Modelos de Comunicação

Existem 3 modelos de comunicação dispositivo/nó: mestre/escravo, cliente/servidor e produtor/consumidor



Protocolos de Comunicação

Os protocolos são usados na comunicação, por exemplo, configurar nós (SDOs) ou transmitir dados em tempo real (PDOs)



Estados do dispositivo

Um dispositivo suporta diferentes estados. Um nó 'mestre' pode alterar o estado de um nó 'escravo' - por exemplo, redefinindo-o



dicionário de objetos

Cada dispositivo possui um OD com entradas que especificam, por exemplo, a configuração do dispositivo. Pode ser acessado via SDOs



Folha de Dados Eletrônica

O EDS é um formato de arquivo padrão para entradas OD - permitindo, por exemplo, que as ferramentas de serviço atualizem os dispositivos



Perfis de dispositivo

Os padrões descrevem, por exemplo, módulos de E/S (CiA 401) e controle de movimento (CiA 402) para independência do fornecedor

Modelos de comunicação CANopen

- Em uma rede CANopen, vários dispositivos precisam se comunicar.
- Por exemplo, em uma configuração de automação industrial, você pode ter um braço de robô com vários nós de servomotores e uma interface de controle/nó de PC.
- Para facilitar a comunicação, **existem três modelos** dentro do CANopen - cada um intimamente ligado aos protocolos CANopen.



Modelos de comunicação CANopen

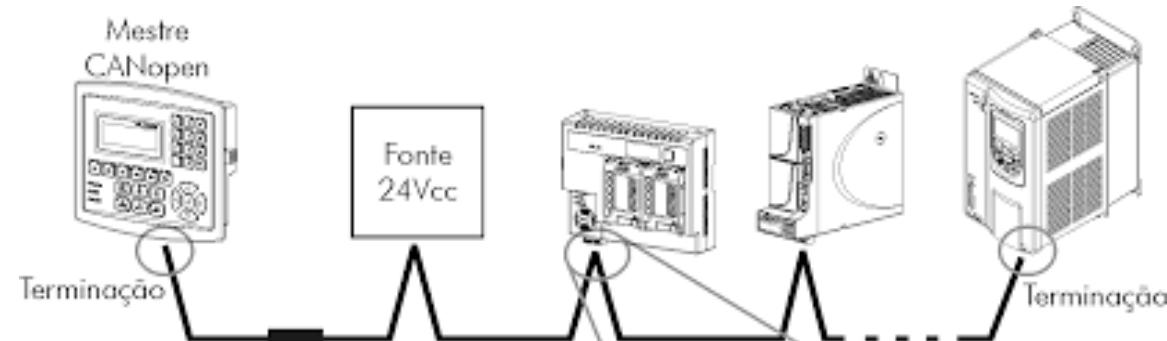
Uma rede CANopen pode conter dispositivos utilizando 3 modelos de comunicação ao mesmo tempo:

- Cliente/Servidor
- Mestre/escravo
- Produtor/Consumidor



Modelo de comunicação Mestre/Escravo

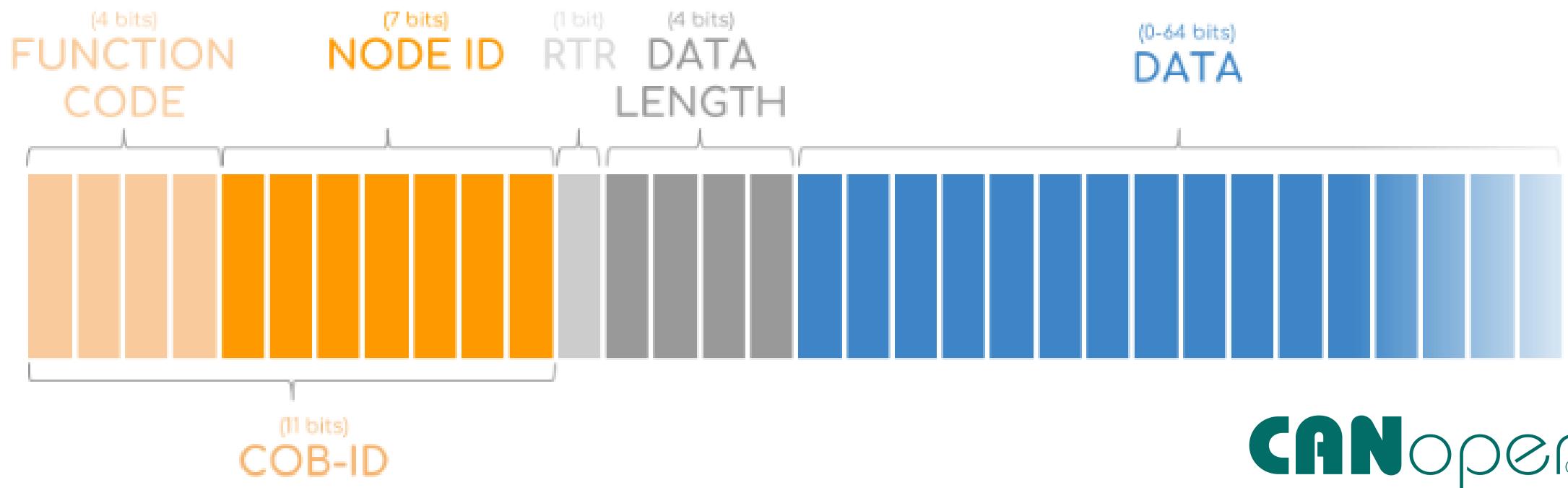
- É usado para diagnóstico e gerenciamento de estado dos dispositivos em uma rede CANopen.
- Cada dispositivo tem um identificador único, que vai de 0 a 127.
- Não podem haver identificadores duplicados.
- 127 é um limite lógico
- O limite pode ocorrer antes, dependendo do transceiver utilizado.



CANopen®

O quadro CANopen

- Para entender a comunicação CANopen, é necessário quebrar o **quadro CANopen CAN** :
- O CAN ID de 11 bits é chamado de identificador de objeto de comunicação (**COB-ID**) e é dividido em duas partes:
- Por padrão, os primeiros 4 bits equivalem a um **código de função** e os próximos 7 bits contêm o **ID do nó** .



Modelo de comunicação Cliente/Servidor

- Este é um protocolo de comunicação usado entre um único cliente e um único servidor. Um cliente emite um pedido (upload/download) fazendo com que o servidor execute uma determinada tarefa. Depois de terminar a tarefa o servidor responde ao pedido.

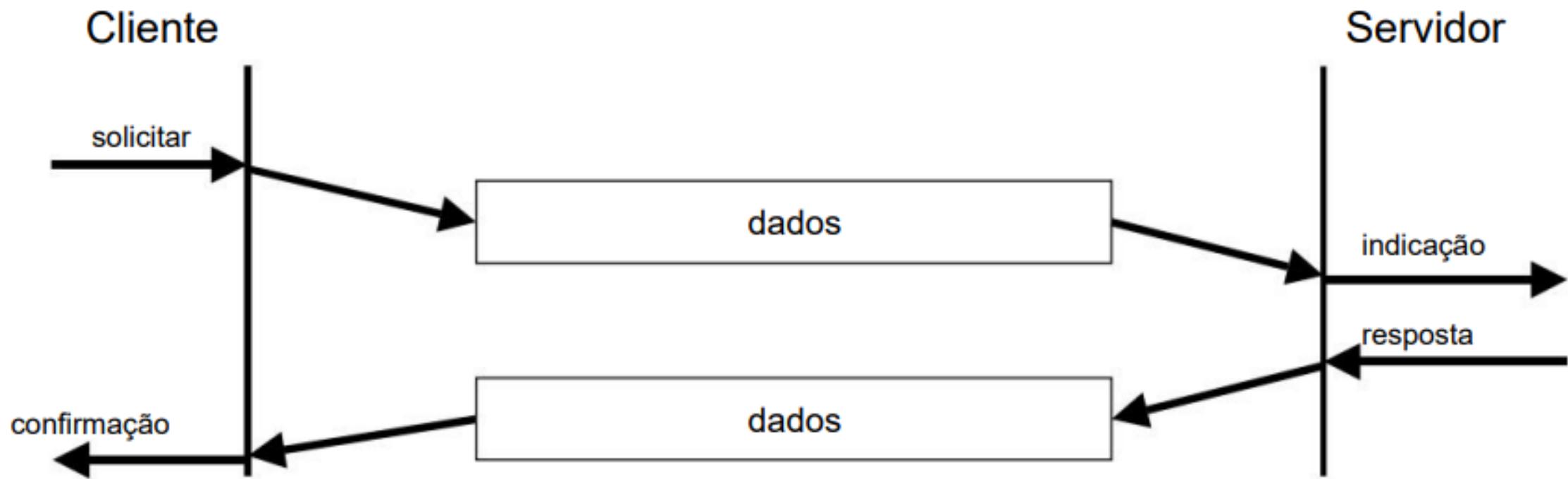
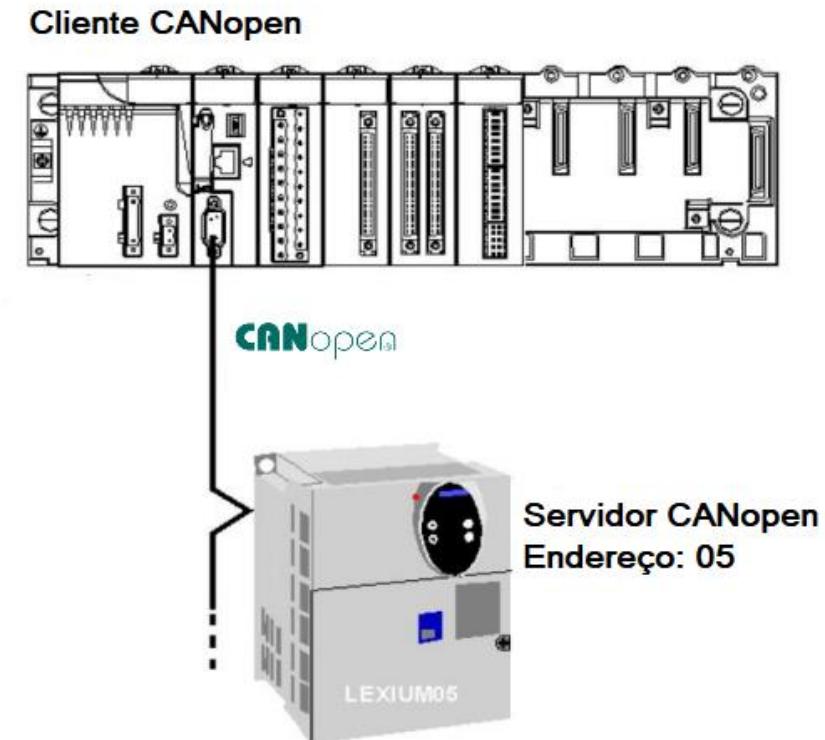


Figura 8: Protocolo de comunicação cliente/servidor

Modelo de comunicação Cliente/Servidor

- A diferença do mestre/escravo para o cliente/servidor no CANopen é o mesmo que para qualquer protocolo.
- O mestre é o gerenciador da rede, então um dispositivo mestre gerencia o estado dos escravos e pode solicitar que um escravo reinicie ou deixe de operar na rede.
- Já o cliente pode enviar ou ler dados do dicionário de objetos(modelo de dados do CANopen) de um servidor. Podem haver n últiplos clientes e servidores.



Modelo de comunicação produtor/consumidor

- O protocolo produtor/consumidor envolve um produtor e zero ou mais consumidores.
- O modelo push conforme definido na Figura 9 é caracterizado por um protocolo não confirmado solicitado pelo produtor.
- O modelo pull conforme definido na Figura 10 é caracterizado por um protocolo confirmado solicitado pelo consumidor.



Figura 9: Modelo push

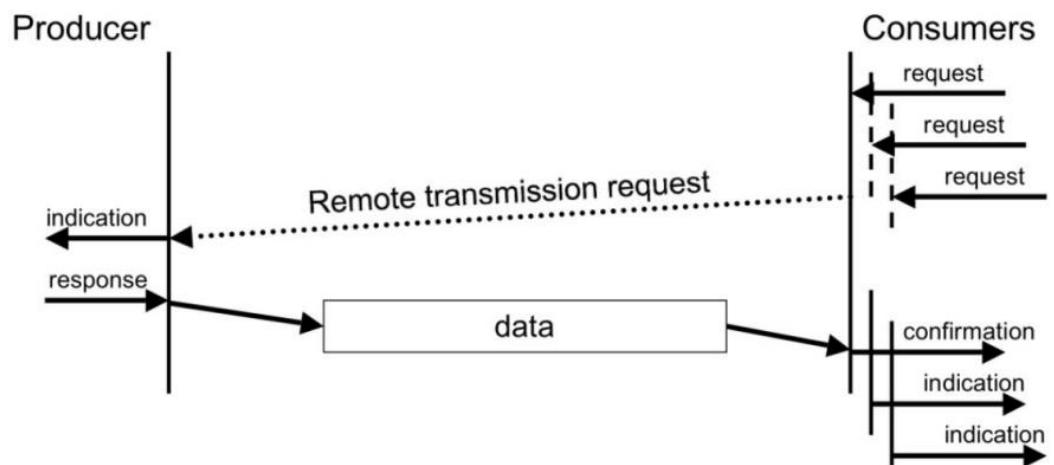
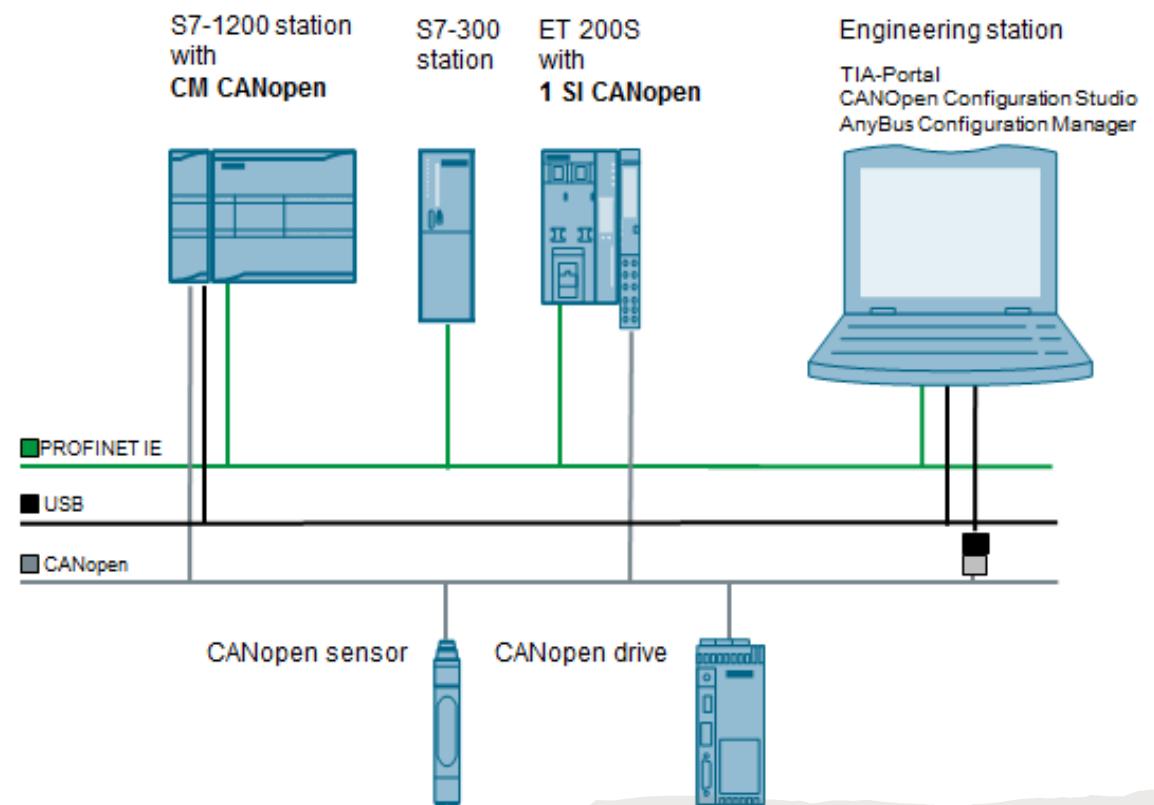


Figura 10: Modelo de tração

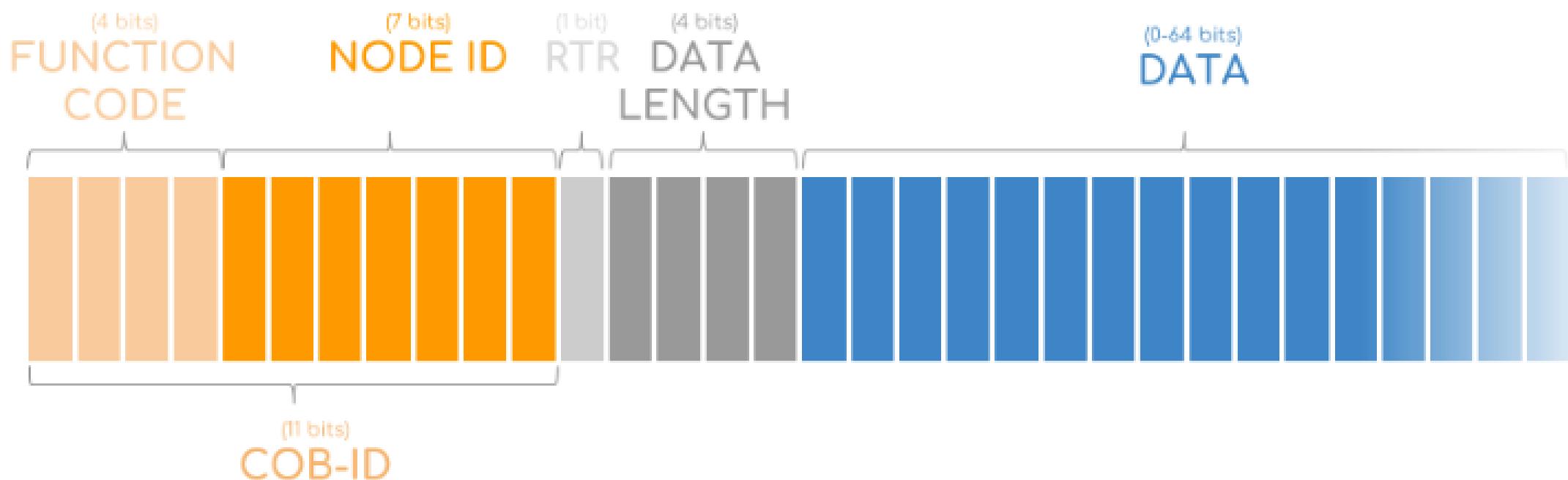
Modelo de comunicação produtor/consumidor

- Dispositivo produtor transmite os dados por broadcast
- Mas muitas vezes a comunicação é um multicast, pois somente alguns dispositivos tem interesse na informação transmitida.
- O produtor transmite dados sem uma requisição específica(modo push).
- Ou por uma requisição(modo pull).



FRAME CANopen

- COB-ID: CANopen Object Identification
- DLC: É usado dependendo do protocolo interno do CANopen de uma forma ou da forma tradicional do CAN.
- Data Field: são os dados transmitidos.



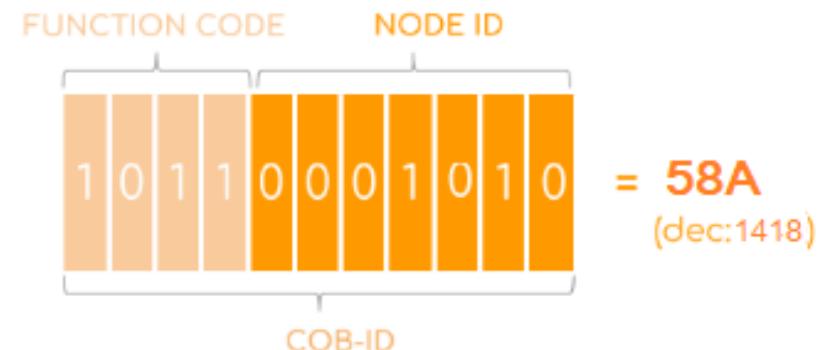
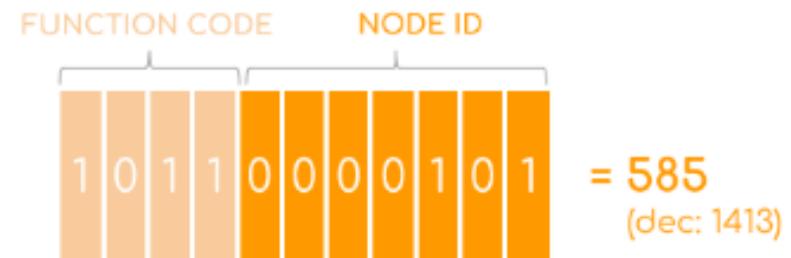
FRAME CANopen

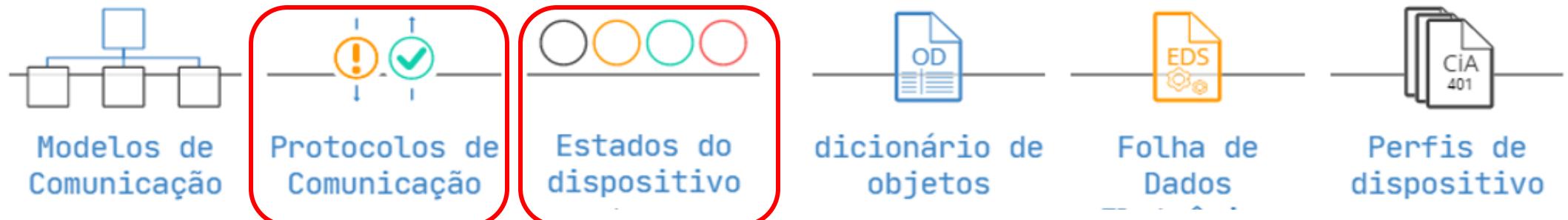
- Para entender como funciona o COB-ID, vamos começar na alocação pré-definida de identificadores usados em redes CANopen simples.
- Como é evidente, os COB-IDs (381, 581, ...) estão ligados aos serviços de comunicação (transmitir PDO 3, transmitir SDO, ...).
- Como tal, **o COB-ID detalha qual nó está enviando/recebendo dados - e qual serviço é usado .**

COMMUNICATION OBJECT	FUNCTION CODE (4 bit, bin)	NODE IDs (7 bit, bin)	COB-IDs (hex)	COB-IDs (dec)	#
1 NMT	0000	0000000	0	0	1
2 SYNC	0001	0000000	80	128	1
3 EMCY	0001	0000001-1111111	81 - FF	129 - 255	127
4 TIME	0010	0000000	100	256	1
5 Transmit PDO 1	0011	0000001-1111111	181 - 1FF	385 - 511	127
Receive PDO 1	0100	0000001-1111111	201 - 27F	513 - 639	127
Transmit PDO 2	0101	0000001-1111111	281 - 2FF	641 - 767	127
Receive PDO 2	0110	0000001-1111111	301 - 37F	769 - 895	127
Transmit PDO 3	0111	0000001-1111111	381 - 3FF	897 - 1023	127
Receive PDO 3	1000	0000001-1111111	401 - 47F	1025 - 1151	127
Transmit PDO 4	1001	0000001-1111111	481 - 4FF	1153 - 1279	127
Receive PDO 4	1010	0000001-1111111	501 - 57F	1281 - 1407	127
6 Transmit SDO	1011	0000001-1111111	581 - 5FF	1409 - 1535	127
Receive SDO	1100	0000001-1111111	601 - 67F	1537 - 1693	127
7 HEARTBEAT	1110	0000001-1111111	701 - 77F	1793 - 1919	127

FRAME CANopen

- Um dispositivo CANopen com ID de nó 5 transmitiria *um SDO via CAN ID 585* de 11 bits .
- Isso corresponde a um código de função binário de 1011 e um ID de nó de 5 (binário: 0000101).





canopen®

Protocolos de comunicação e Estados do dispositivo

Protocolos/serviços de comunicação CANopen

- Na rede CANopen nós temos 7 tipos de serviço de comunicação, que são:

#1 Network Management (NMT)

#2 Synchronization (SYNC)

#3 Emergency (EMCY)

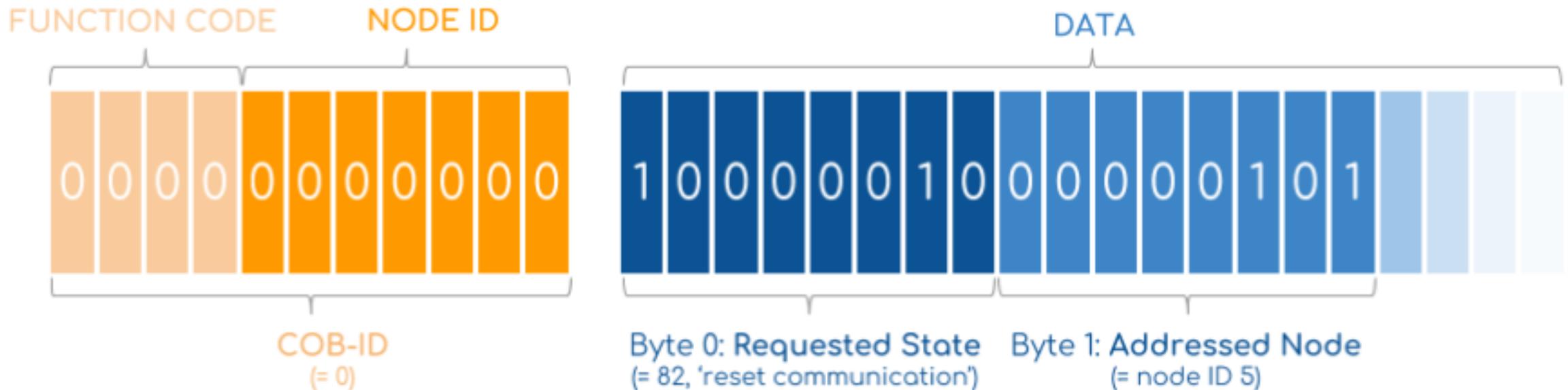
#4 Timestamp (TIME) [PDO]

#5 Process Data Object [PDO]

#6 Service Data Object [SDO]

#7 Node monitoring (Heartbeat) [SDO]





Protocolos de Comunicação Network Management (NMT)

- **O que vem a ser o protocolo NMT?**
- O serviço NMT é usado para controlar o estado dos dispositivos CANopen (por exemplo, pré-operacional, operacional, parado) por meio de comandos NMT (por exemplo, iniciar, parar, reiniciar).
- **Como funciona?**
- Para alterar o estado, o mestre NMT envia uma mensagem de 2 bytes com CAN ID 0 (ou seja, código de função 0 e ID do nó 0).
- Todos os nós escravos processam esta mensagem. O 1º byte de dados CAN contém o estado solicitado - enquanto o 2º byte de dados CAN contém o ID do nó de destino. O nó ID 0 indica um comando de broadcast.

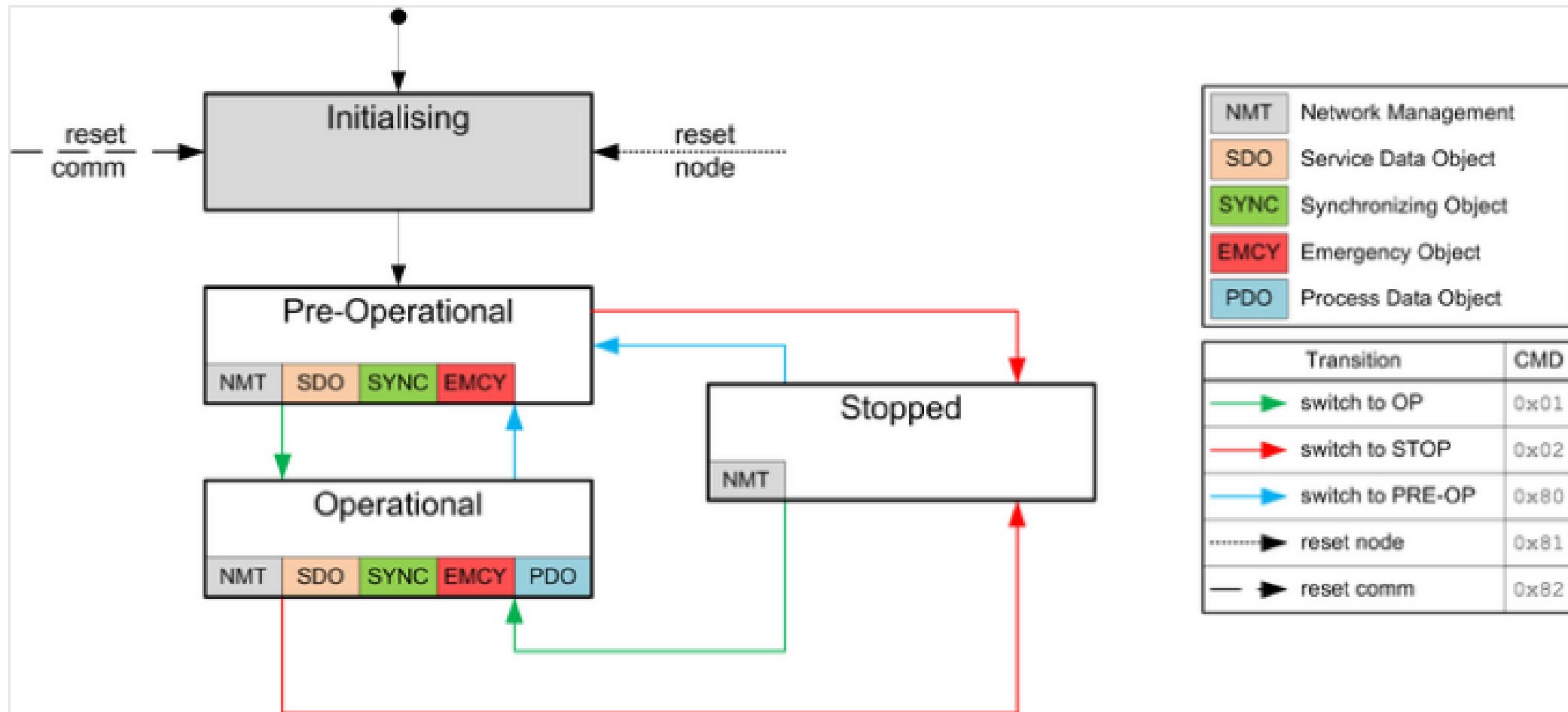
Protocolos de Comunicação Network Management (NMT)

- Os comandos possíveis incluem:

<CMD>	Significado
01h_	Alternar para o estado "Operacional"
02h_	Alternar para o estado "Parar"
80h_	Mude para o estado "Pré-Operacional"
81h_	Redefinir nó
82h_	Redefinir comunicação

- O valor para <Node-ID> pode ser 00_h , neste caso, o comando NMT se aplica a todos os dispositivos no barramento CAN (broadcast). Se for usado um número diferente de zero, apenas o dispositivo com o nó-ID correspondente é endereçado.

Estados dos dispositivos



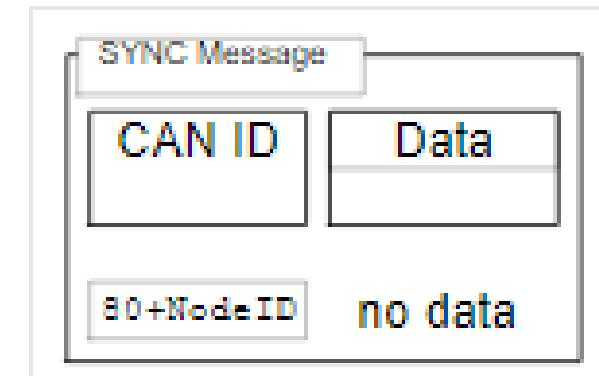
CANopen®

Estados dos dispositivos

Service	Initializing	Pre-Operational	Operational	Stopped
PDO			Active	
SDO		Active	Active	
SYNC		Active	Active	
EMCY		Active	Active	
BOOT-UP	Active			
NMT		Active	Active	Active

Protocolos de Comunicação Sincronização (SYNC)

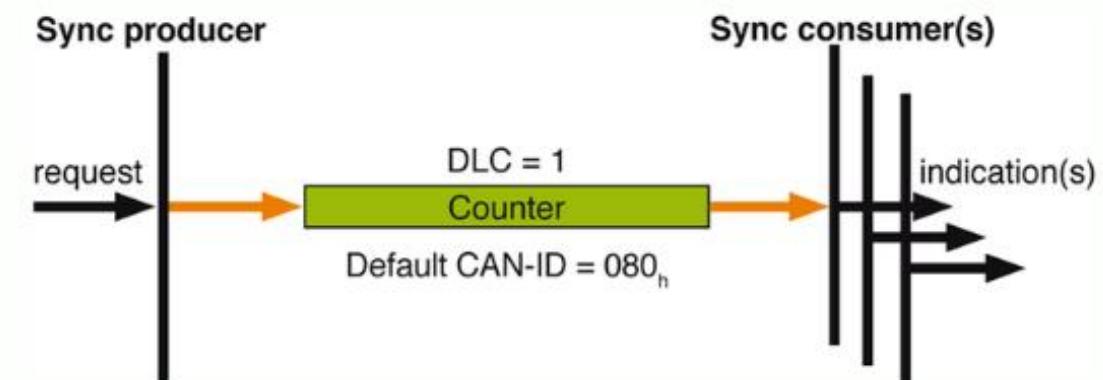
- **O que é o protocolo SYNC?**
- O protocolo SYNC fornece o mecanismo básico de sincronização de rede.
- O produtor SYNC dispara o objeto de sincronização (SYNC) periodicamente. O período de transmissão do objeto SYNC é configurável.
- Qualquer dispositivo CANopen pode ser configurado como consumidor SYNC.
- Em muitos casos, o protocolo SYNC é usado para fins de gerenciamento de carga de barramento. Em tais casos de uso, os PDOs nos consumidores SYNC são acionados pela recepção de um número pré-configurado de SYNCs.



CANopen®

Protocolos de Comunicação Sincronização (SYNC)

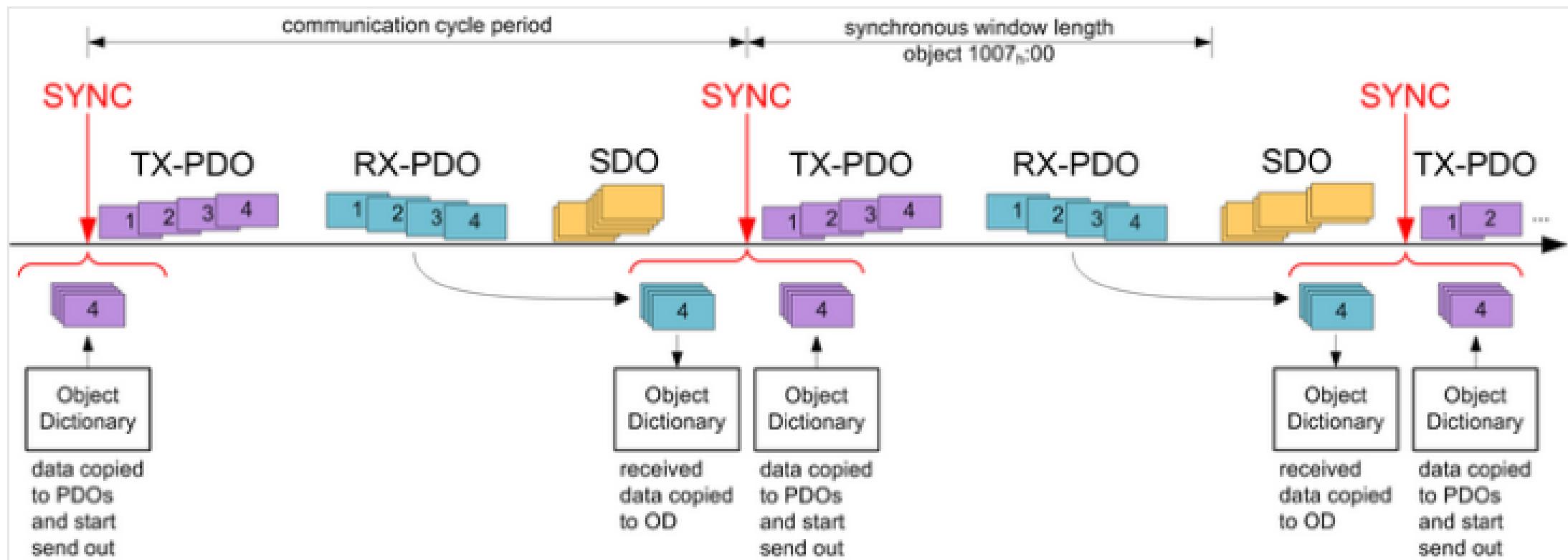
- A mensagem SYNC é mapeada por um único quadro CAN com identificador 80h. Por padrão, a mensagem SYNC não transporta nenhum dados (DLC=0). Dispositivos que suportam CiA 301 versão 4.1 ou superior podem opcionalmente oferecer uma mensagem SYNC, que fornece um valor de contador SYNC de 1 byte.
- Em geral é o mestre da rede CANopen que produz o objeto de sincronização (um escravo também pode produzir), mas o modelo de comunicação utilizado é o produtor/consumidor, não havendo confirmação de mensagem.



CANopen®

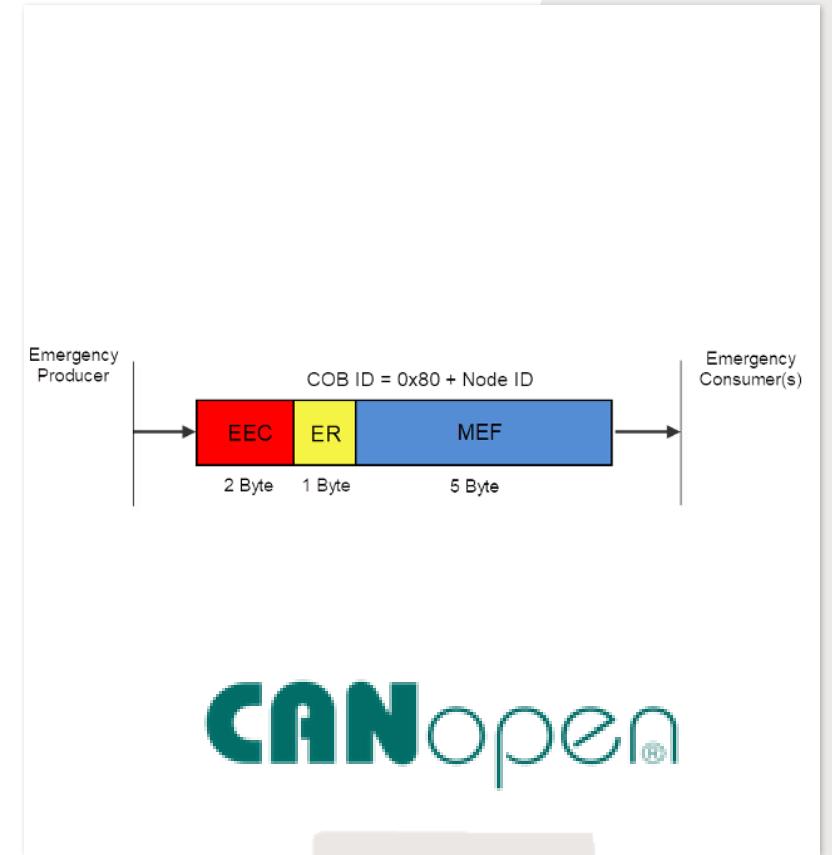
Protocolos de Comunicação Sincronização (SYNC)

- Um dos principais usos da mensagem de sincronização é sincronizar as mensagens do protocolo PDO.
 - O período do ciclo de sincronização(janela de sincronização) é configurável em microssegundos.



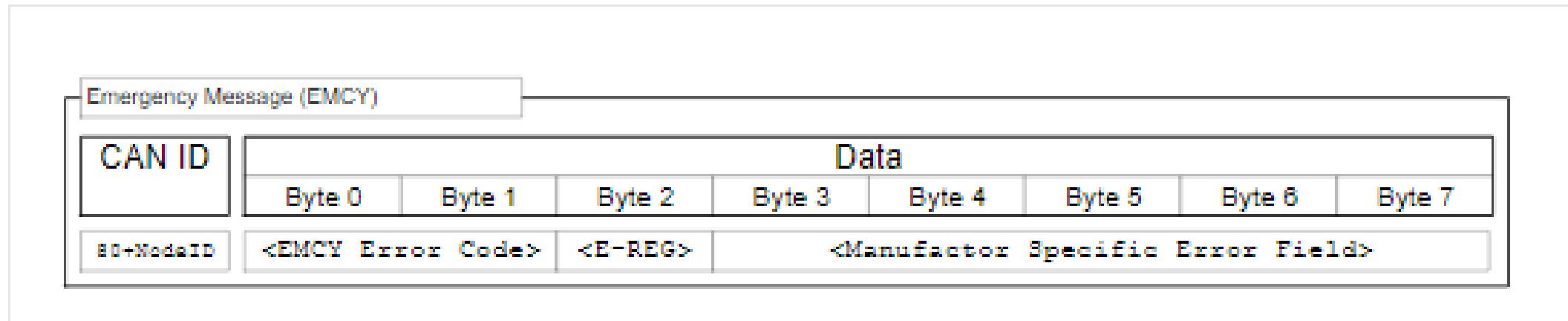
Protocolo de comunicação Emergency (EMCY)

- As mensagens de emergência são acionadas por um erro interno do dispositivo.
- A mensagem de emergência, transmitida pelo produtor de Emergência, é mapeada para um único quadro CAN, que contém até oito bytes de dados.
- Por padrão, um dispositivo que suporta a funcionalidade de produtor de Emergência atribui o CAN-identifier $80h + (\text{node-ID})$ à mensagem de Emergência.
- Uma mensagem de emergência é transmitida apenas uma vez por evento de erro.
- Desde que nenhum novo erro ocorra em um dispositivo, nenhuma outra mensagem de emergência será transmitida.
- Zero ou mais consumidores de emergência podem receber essas mensagens e podem iniciar contra-medidas adequadas e específicas da aplicação.



Protocolo de comunicação Emergency (EMCY)

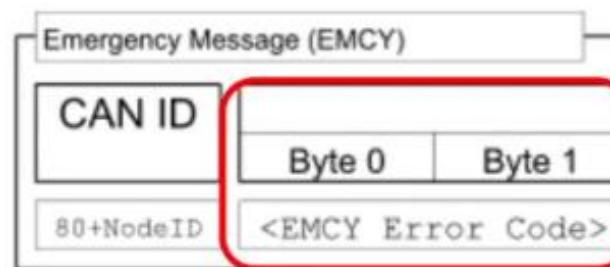
- A mensagem de emergência está estruturada da seguinte forma:



- Um total de três códigos de erro são transferidos aqui: o "Emergency Error Code" (<EMCY Error Code>), o conteúdo do objeto "Error Register" (1001 h, <E-REG>) e um código específico do fabricante (Erro específico do fabricante).

Emergency Error Code

O código de emergência é informado no objeto "preDefined Error Field" (1003h).

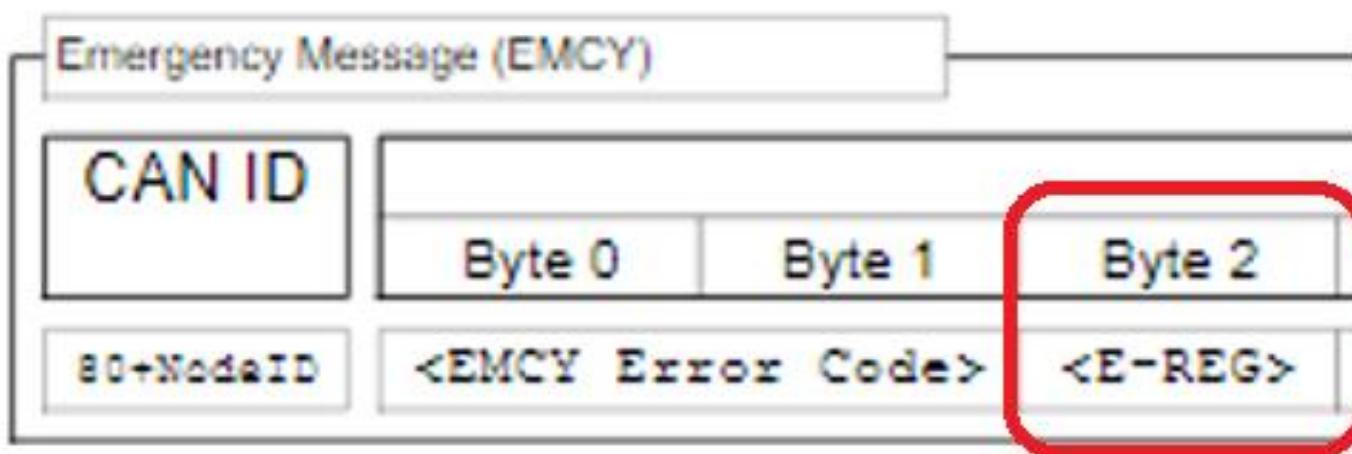


```
* Standard error codes according to CiA DS-301 and DS-401.  
*/  
#define CO EMC_NO_ERROR 0x0000U /*< 0x00xx, error Reset or No Error */  
#define CO EMC_GENERIC 0x1000U /*< 0x10xx, Generic Error */  
#define CO EMC_CURRENT 0x2000U /*< 0x20xx, Current */  
#define CO EMC_CURRENT_INPUT 0x2100U /*< 0x21xx, Current, device input side */  
#define CO EMC_CURRENT_INSIDE 0x2200U /*< 0x22xx, Current inside the device */  
#define CO EMC_CURRENT_OUTPUT 0x2300U /*< 0x23xx, Current, device output side */  
#define CO EMC_VOLTAGE 0x3000U /*< 0x30xx, Voltage */  
#define CO EMC_VOLTAGE_MAINS 0x3100U /*< 0x31xx, Mains Voltage */  
#define CO EMC_VOLTAGE_INSIDE 0x3200U /*< 0x32xx, Voltage inside the device */  
#define CO EMC_VOLTAGE_OUTPUT 0x3300U /*< 0x33xx, Output Voltage */  
#define CO EMC_TEMPERATURE 0x4000U /*< 0x40xx, Temperature */  
#define CO EMC_TEMP_AMBIENT 0x4100U /*< 0x41xx, Ambient Temperature */  
#define CO EMC_TEMP_DEVICE 0x4200U /*< 0x42xx, Device Temperature */  
#define CO EMC_HARDWARE 0x5000U /*< 0x50xx, Device Hardware */  
#define CO EMC_SOFTWARE_DEVICE 0x6000U /*< 0x60xx, Device Software */  
#define CO EMC_SOFTWARE_INTERNAL 0x6100U /*< 0x61xx, Internal Software */  
#define CO EMC_SOFTWARE_USER 0x6200U /*< 0x62xx, User Software */  
#define CO EMC_DATA_SET 0x6300U /*< 0x63xx, Data Set */  
#define CO EMC_ADDITIONAL_MODUL 0x7000U /*< 0x70xx, Additional Modules */  
#define CO EMC_MONITORING 0x8000U /*< 0x80xx, Monitoring */  
#define CO EMC_COMMUNICATION 0x8100U /*< 0x81xx, Communication */  
#define CO EMC_CAN_OVERRUN 0x8110U /*< 0x8110, CAN Overrun (Objects lost) */  
#define CO EMC_CAN_PASSIVE 0x8120U /*< 0x8120, CAN in Error Passive Mode */  
#define CO EMC_HEARTBEAT 0x8130U /*< 0x8130, Life Guard Error or Heartbeat Error */  
#define CO EMC_BUS_OFF_RECOVERED 0x8140U /*< 0x8140, recovered from bus off */  
#define CO EMC_PROTOCOL_ERROR 0x8150U /*< 0x8150, CAN-ID collision */  
#define CO EMC_PDO_LENGTH 0x8200U /*< 0x82xx, Protocol Error */  
#define CO EMC_PDO_LENGTH_EXC 0x8210U /*< 0x8210, PDO not processed due to length error */  
#define CO EMC_DAM_MPDO 0x8220U /*< 0x8220, PDO length exceeded */  
#define CO EMC_SYNC_DATA_LENGTH 0x8230U /*< 0x8230, DAM MPDO not processed, destination object not */  
#define CO EMC_RPDO_TIMEOUT 0x8240U /*< 0x8240, Unexpected SYNC data length */  
#define CO EMC_EXTERNAL_ERROR 0x8250U /*< 0x8250, RPDO timeout */  
#define CO EMC_ADDITIONAL_FUNC 0x9000U /*< 0x90xx, External Error */  
#define CO EMC_UNDEFINED 0xF000U /*< 0xF0xx, Additional Functions */
```

Error Register

O bit do respectivo erro é setado no objeto "Error Register"(1001h).

```
CO_ERR_REG_GENERIC_ERR = 0x01U, /*< bit 0, generic error */
CO_ERR_REG_CURRENT = 0x02U, /*< bit 1, current */
CO_ERR_REG_VOLTAGE = 0x04U, /*< bit 2, voltage */
CO_ERR_REG_TEMPERATUR = 0x08U, /*< bit 3, temperature */
CO_ERR_REG_COMM_ERR = 0x10U, /*< bit 4, communication error (overrun, error state) */
CO_ERR_REG_DEV_PROFILE = 0x20U, /*< bit 5, device profile specific */
CO_ERR_REG_RESERVED = 0x40U, /*< bit 6, reserved (always 0) */
CO_ERR_REG_MANUFACTURER = 0x80U /*< bit 7, manufacturer specific */
```





Manufacturer Specific Error Field

O número do erro é informado no objeto "manufacturer Status Register" (1002h).

```
#define CO_EM_10_unused          0x10U /*<< 0x10, (unused) */
#define CO_EM_11_unused          0x11U /*<< 0x11, (unused) */
#define CO_EM_CAN_TX_BUS_OFF     0x12U /*<< 0x12, communication, critical, CAN transmit bus is off */
#define CO_EM_CAN_RXB_OVERFLOW    0x13U /*<< 0x13, communication, critical, CAN module receive buffer overflow */
#define CO_EM_CAN_TX_OVERFLOW     0x14U /*<< 0x14, communication, critical, CAN transmit buffer has overflow */
#define CO_EM_TPDO_OUTSIDE_WINDOW 0x15U /*<< 0x15, communication, critical, TPDO is outside SYNC window */
#define CO_EM_16_unused          0x16U /*<< 0x16, (unused) */
#define CO_EM_17_unused          0x17U /*<< 0x17, (unused) */
#define CO_EM_SYNC_TIME_OUT      0x18U /*<< 0x18, communication, critical, SYNC message timeout */
#define CO_EM_SYNC_LENGTH         0x19U /*<< 0x19, communication, critical, Unexpected SYNC data length */
#define CO_EM_PDO_WRONG_MAPPING   0x1AU /*<< 0x1A, communication, critical, Error with PDO mapping */
#define CO_EM_HEARTBEAT_CONSUMER 0x1BU /*<< 0x1B, communication, critical, Heartbeat consumer timeout */
#define CO_EM_HB_CONSUMER_REMOTE_RESET 0x1CU /*<< 0x1C, communication, critical, Heartbeat consumer detection */
#define CO_EM_1D_unused          0x1DU /*<< 0x1D, (unused) */
#define CO_EM_1E_unused          0x1EU /*<< 0x1E, (unused) */
#define CO_EM_1F_unused          0x1FU /*<< 0x1F, (unused) */

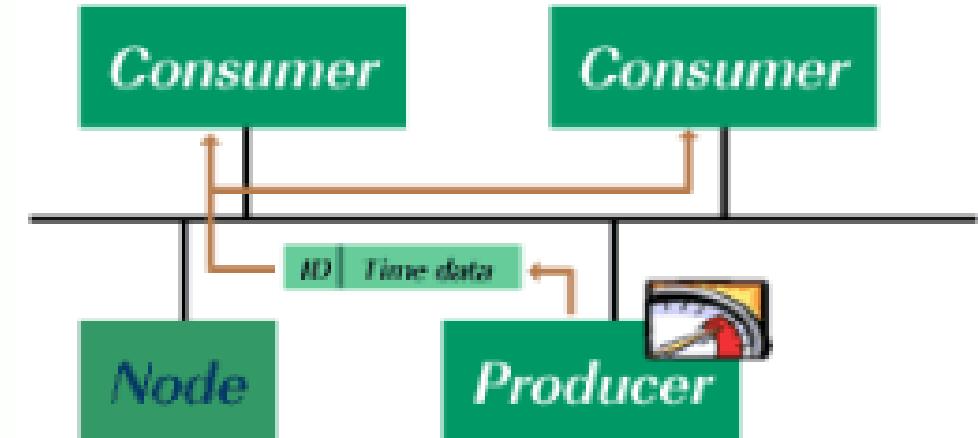
#define CO_EM_EMERGENCY_BUFFER_FULL 0x20U /*<< 0x20, generic, info, Emergency buffer is full, Emergency */
#define CO_EM_21_unused          0x21U /*<< 0x21, (unused) */
#define CO_EM_MICROCONTROLLER_RESET 0x22U /*<< 0x22, generic, info, Microcontroller has just started */
#define CO_EM_23_unused          0x23U /*<< 0x23, (unused) */
#define CO_EM_24_unused          0x24U /*<< 0x24, (unused) */
#define CO_EM_25_unused          0x25U /*<< 0x25, (unused) */
#define CO_EM_26_unused          0x26U /*<< 0x26, (unused) */
#define CO_EM_27_unused          0x27U /*<< 0x27, (unused) */

#define CO_EM_WRONG_ERROR_REPORT 0x28U /*<< 0x28, generic, critical, Wrong parameters to CO_errorReport */
#define CO_EM_ISR_TIMER_OVERFLOW 0x29U /*<< 0x29, generic, critical, Timer task has overflowed */
#define CO_EM_MEMORY_ALLOCATION_ERROR 0x2AU /*<< 0x2A, generic, critical, Unable to allocate memory for allocation */
#define CO_EM_GENERIC_ERROR        0x2BU /*<< 0x2B, generic, critical, Generic error, test usage */
#define CO_EM_GENERIC_SOFTWARE_ERROR 0x2CU /*<< 0x2C, generic, critical, Software error */
#define CO_EM_INCONSISTENT_OBJECT_DICT 0x2DU /*<< 0x2D, generic, critical, Object dictionary does not match */
#define CO_EM_CALCULATION_OF_PARAMETERS 0x2EU /*<< 0x2E, generic, critical, Error in calculation of device parameters */
#define CO_EM_NON_VOLATILE_MEMORY 0x2FU /*<< 0x2F, generic, critical, Error with access to non volatile memory */
```

Emergency Message (EMCY)							
CAN ID	Data						
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
80+NodeID	<EMCY Error Code>	<E-REG>	<Manufacturer Specific Error Field>				

Protocolo de comunicação Timestamp

- O protocolo Time-stamp permite que o usuário dos sistemas CANopen ajuste um horário de rede único.
- O carimbo de hora é mapeado para um único quadro CAN com um código de comprimento de dados de 6 bytes.
- O produtor da estampa de tempo(mestre ou escravo) fornece informações de tempo absoluto, que podem ser usadas como relógio de rede global.



CANopen®

Protocolo de comunicação Timestamp

- Esses seis bytes de dados fornecem a informação "Hora do dia".
- Essas informações são fornecidas em milissegundos após a meia-noite (tipo de dados: Unsigned28) e dias desde 1º de janeiro de 1984 (tipo de dados: Unsigned16).
- O quadro CAN associado possui por padrão o CAN-Identifier 100_h.

```
typedef union {
    unsigned long long ullValue;
    struct {
        unsigned long ms:28;
        unsigned reserved:4;
        unsigned days:16;
        unsigned reserved2:16;
    };
}timeOfDay_t;
```

CANopen®

Protocolo de comunicação Timestamp

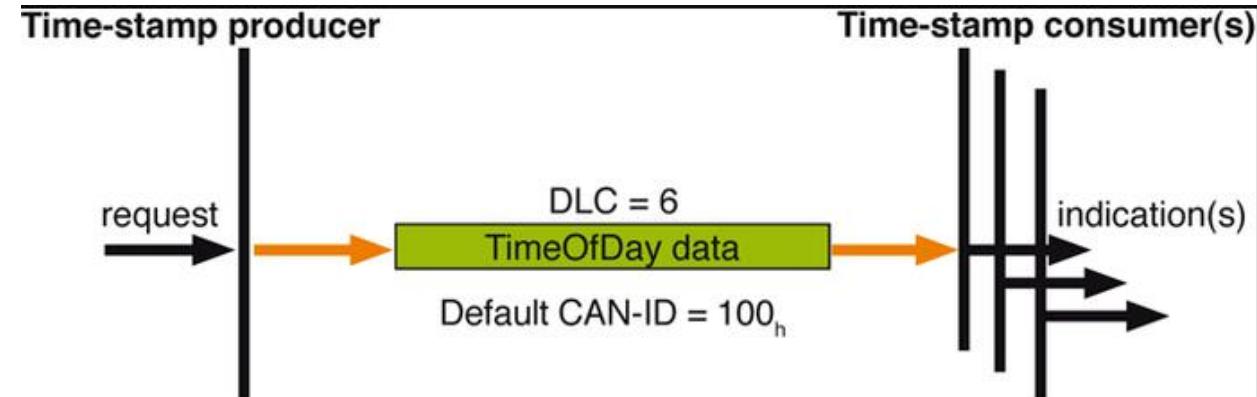
- Os jitters de tempo devem ser considerados, pois atrasos na transmissão podem ocorrer caso a rede não esteja ociosa ou a arbitragem seja perdida devido a COBs anteriores mais elevados.

```
typedef union {
    unsigned long long ullValue;
    struct {
        unsigned long ms:28;
        unsigned reserved:4;
        unsigned days:16;
        unsigned reserved2:16;
    };
}timeOfDay_t;
```

CANopen®

Protocolo de comunicação Timestamp

- O tempo entre as mensagens das estampas de tempo dependem da aplicação CANopen implementada, ou seja, é definido pelo produtor da estampa de tempo.



CANopen®

HEARTBEAT

- Os protocolos de controle de erros permitem o monitoramento de uma rede CANopen.
- Eles compreendem o Heartbeat, Node/Life-Guarding, bem como o protocolo de inicialização (Boot-up).
- O protocolo Heartbeat é usado para verificar se todos os participantes da rede ainda estão disponíveis em uma rede CANopen e se ainda estão em seu estado NMT FSA pretendido.
- Em sistemas CANopen antigos, o protocolo Node/Life-guarding baseado em quadros remotos CAN é usado para essa finalidade, em vez do protocolo Heartbeat.



HEARTBEAT

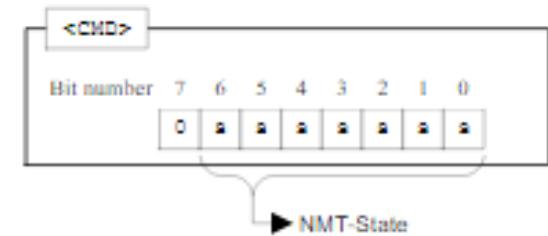
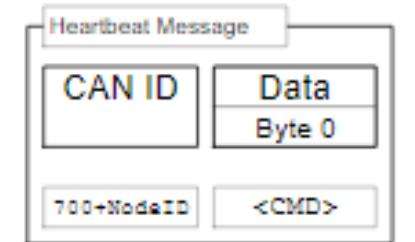
- O protocolo Heartbeat é uma mensagem transmitida ciclicamente que informa todos os consumidores de heartbeat sobre a disponibilidade do produtor de heartbeat.
- Ela tem dois objetivos bem definidos:
- 1º - Enviar uma mensagem de atividade (informar que está vivo).
- 2º - Responder comandos NMT.



CANopen®

HEARTBEAT

- Se o heartbeat estiver ativado, o escravo envia seu estado NMT para o barramento CAN sem solicitação e ciclicamente.
- Este serviço é ativado definindo o tempo Producer Heartbeat Time no objeto $1017_{\text{h}} : 00_{\text{h}}$ para um valor diferente de zero.
- A mensagem enviada pelo escravo tem a seguinte forma:



CANopen®

HEARTBEAT

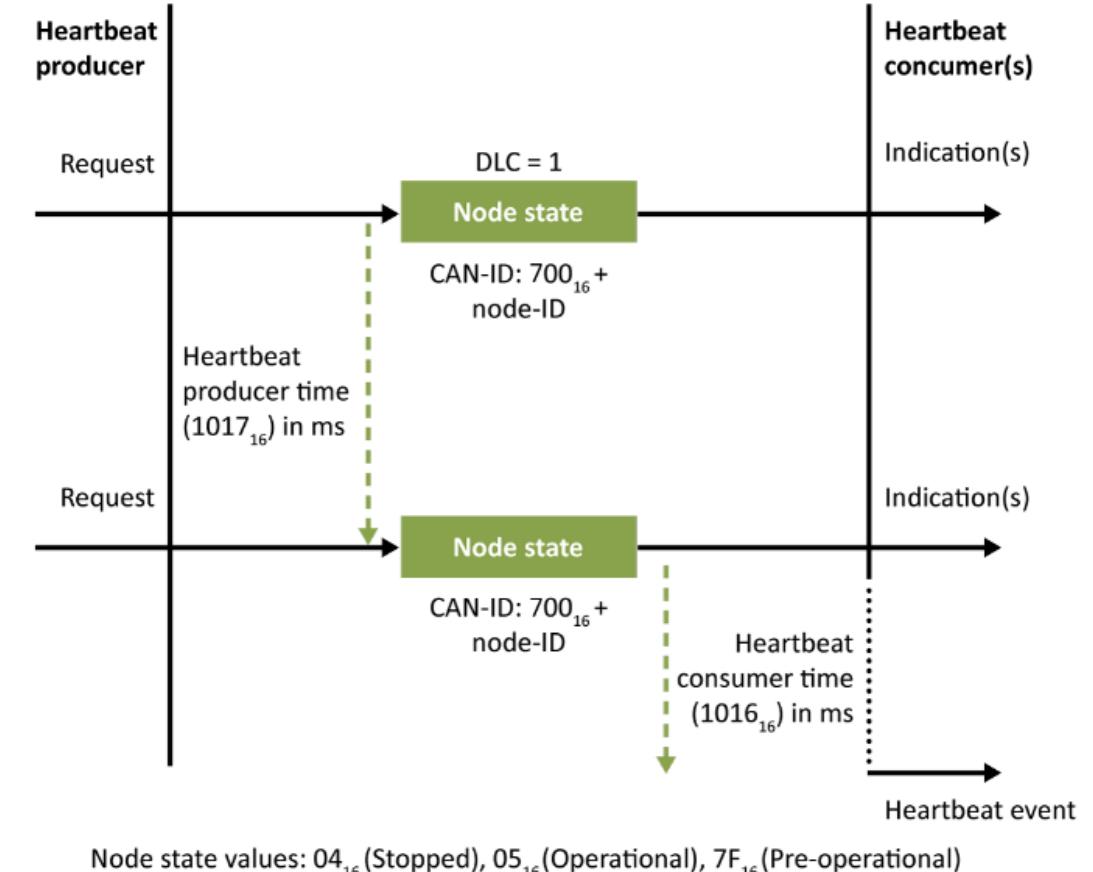
- O período do tempo de transmissão é definido por um parâmetro heartbeat-produce-time no dicionário de objetos.
- Como regra geral, o timeout do consumidor deve ser duas vezes o período do produtor.
- Os tempos do produtor e consumidor são altamente atrelados a uma aplicação específica.



CANopen®

HEARTBEAT

- A mensagem heartbeat de 1 byte usa um quadro CAN com identificador 700 somado ao ID do nó produtor.
- O conteúdo de um byte indica o estado NMT.
- Isso significa que o consumidor pode também detectar que PDOs não estão sendo gerados devido ao estado NMT do nó produtor.



HEARTBEAT

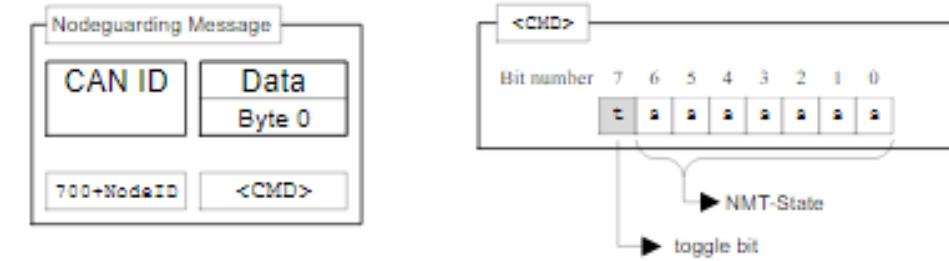
- Uma curiosidade sobre o Heartbeat é que geralmente sensores não consomem heartbeats, eles somente produzem seus dados e enviam mensagens heartbeat para indicar atividade e responder comandos NMT.



CANopen®

Node Guarding

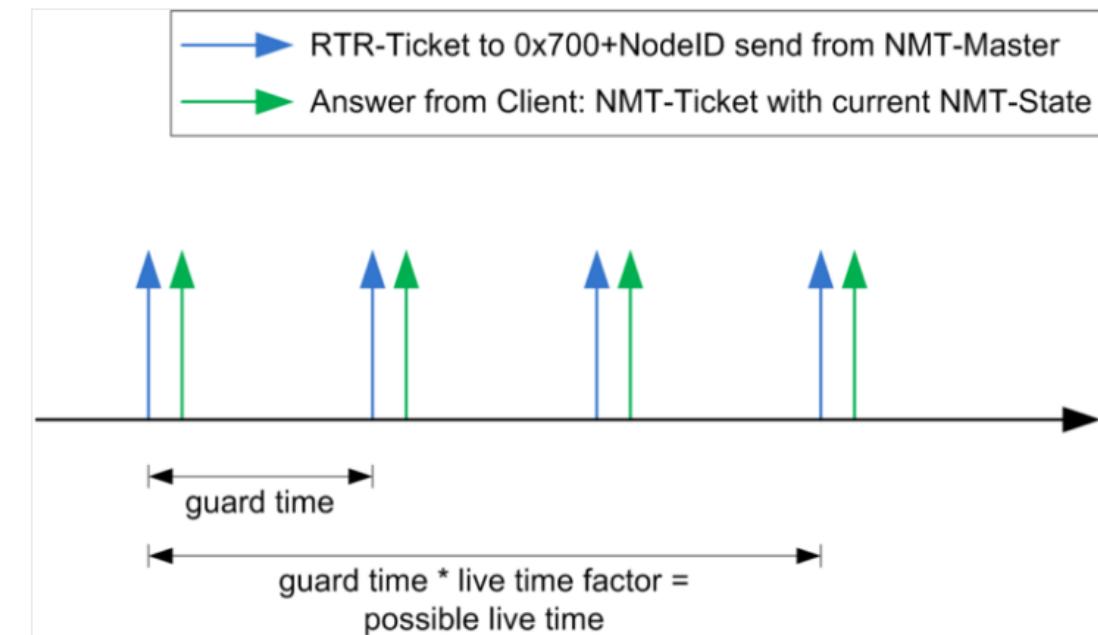
- A Node Guarding é um método desatualizado de verificar se um dispositivo CANopen ainda está funcionando no estado de rede correto.
- Este serviço baseia-se no facto de o mestre NMT enviar uma mensagem RTR com CAN-ID $700_h + \text{nodeID}$ ao respetivo escravo.
- O escravo deve então enviar uma mensagem como resposta; esta mensagem está estruturada da seguinte forma.
- O bit 7 alterna aqui em cada transferência, permitindo assim determinar se uma mensagem foi perdida.
- Inserido nos bits 6 a 0 é o status NMT atual do escravo.



CANopen®

Node Guarding

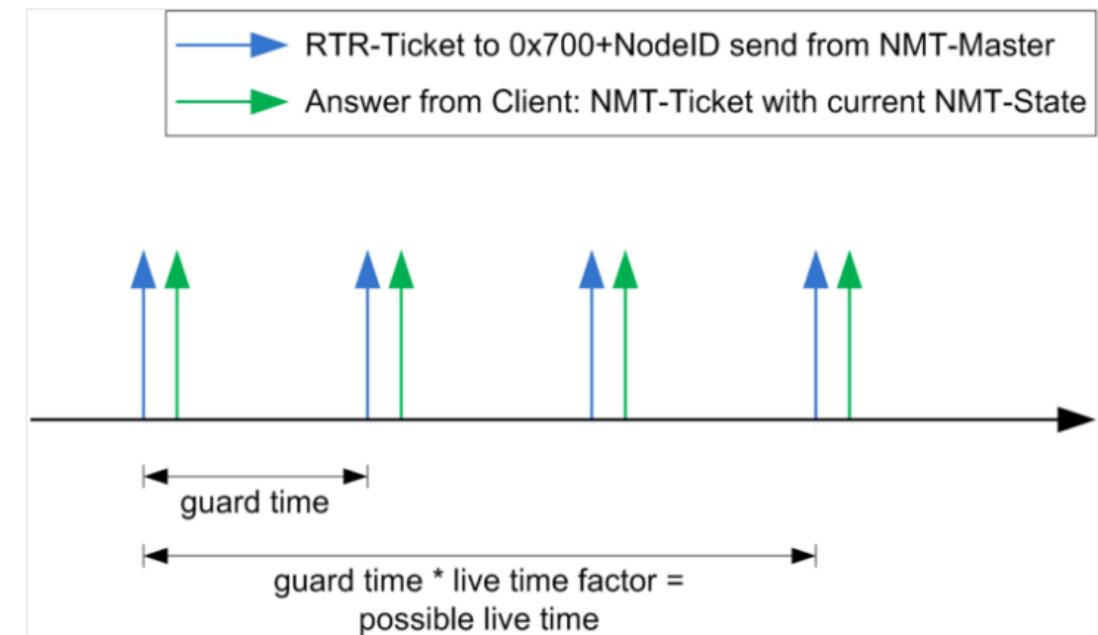
- Com nodeguarding, existem três intervalos de tempo:
- Guard time: O tempo entre duas mensagens RTR. Isso pode ser diferente para cada nodo CAN e é armazenado no escravo no objeto $100C_h :00$ (unidade: milissegundos)
- Live time factor: Um multiplicador para o tempo de guarda; Isso é armazenado no escravo CAN no objeto $1\ 00D_h :00$ e pode ser diferente para cada escravo no barramento CAN.
- Possible live time: O tempo produzido multiplicando o Guard time e Live time factor.



CANopen®

Node Guarding

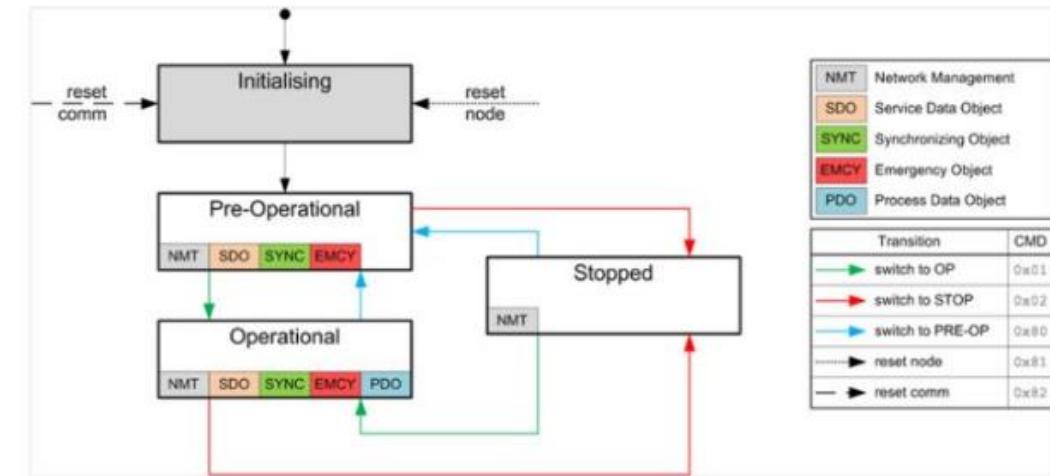
- As seguintes condições são verificadas durante o node guarding:
- O mestre NMT deve enviar a requisição RTR dentro do "possible live time".
- O escravo deve enviar a resposta ao pedido RTR dentro do "possible live time".
- O escravo deve responder com seu estado NMT. Além disso, o "bit de alternância" deve ser definido corretamente.



CANopen®

Protocolo de inicialização (Boot-up)

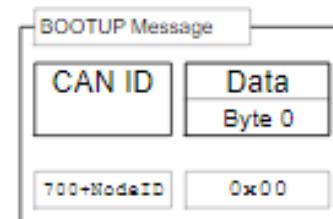
- O protocolo de inicialização (Boot-up) representa um tipo especial de protocolo de controle de erros.
- É transmitido como a ação final na inicialização do estado NMT FSA, antes de entrar no estado NMT FSA pré-operacional.
- A recepção desta mensagem indica que um novo equipamento foi cadastrado na rede CANopen.



CANopen®

Protocolo de inicialização (Boot-up)

- A recepção não intencional de tal protocolo durante o tempo de execução indica uma alteração na configuração da rede (por exemplo, devido à adição de um novo dispositivo CANopen).
- Ou é considerada um sinal de condição de erro (por exemplo, falha na alimentação do dispositivo CANopen relacionado).
- O protocolo usa o mesmo identificador de qualquer outro protocolo de controle de erros, como, por exemplo, o protocolo de Heartbeat.
- O campo de dados de 1 byte tem um valor fixo de zero.



CANopen®

Dicionário de Objetos



Os serviços PDO e SDO são particularmente importantes, pois formam a base para a maioria das comunicações CANopen.
Porém antes de estuda-los é importante entender o conceito de dicionários de objetos.

CANopen Object Dictionary

- Todos os nós CANopen devem ter um dicionário de objetos (OD).

Mas o que é um OD?

*O **dicionário de objetos** é uma estrutura padronizada contendo todos os parâmetros que descrevem o comportamento de um nó CANopen.*



CANopen Object Dictionary

- As entradas OD são pesquisadas por meio de um índice de 16 bits e um subíndice de 8 bits.
- Por exemplo, o índice 1008 (subíndice 0) de um nó OD compatível com CANopen contém o *nome do dispositivo* do nó .



CANopen Object Dictionary

- Especificamente, uma entrada no dicionário de objetos é definida por atributos:
- **Índice:** endereço base de 16 bits do objeto.
- **Nome do objeto:** nome do fabricante do dispositivo.
- **Código do objeto:** Array, variável ou registrador.
- **Tipo de dados:** VISIBLE_STRING ou UNSIGNED32 ou nome do registrador.
- **Acesso:** rw (leitura/gravação), ro (somente leitura), wo (somente gravação)
- **Categoría:** Indica se este parâmetro é obrigatório/opcional (M/O)



Seções padronizadas no OD

- O dicionário de objetos é dividido em **seções padronizadas** onde algumas entradas são obrigatórias e outras são totalmente personalizáveis.
- É importante ressaltar que as entradas OD de um dispositivo (por exemplo, um escravo) podem ser acessadas por outro dispositivo (por exemplo, um mestre) via CAN usando, por exemplo, SDOs.
- Por exemplo, isso pode permitir que um mestre da aplicação altere se um nó escravo registra dados por meio de um sensor de entrada específico - ou com que frequência o escravo envia uma heartbeat.

OD INDEX (16 bits, hex)	DESCRIPTION
0000	Reserved
0001 - 025F	Data types
0260 - 0FFF	Reserved
1000 - 1FFF	Communication object area
2000 - 5FFF	Manufacturer specific area
6000 - 9FFF	Device profile specific area
A000 - BFFF	Interface profile specific area
C000 - FFFF	Reserved

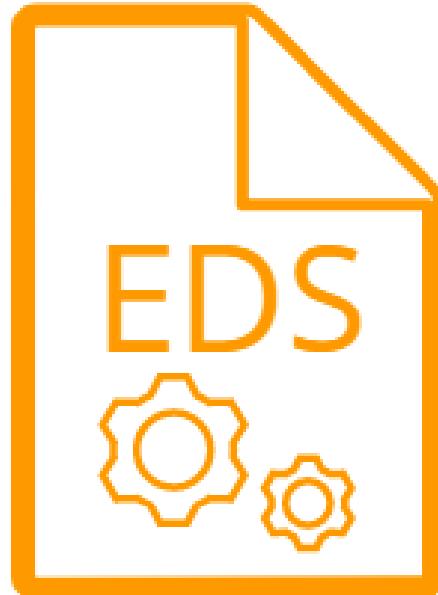


Electronic Data Sheet e Device Configuration File

- Para entender o OD, é útil consultar o 'formulário legível por humanos': O Electronic Data Sheet e o Device Configuration File.



Object Dictionary



Electronic Data Sheet



Device Configuration File



Electronic Data Sheet (EDS)

- Na prática, a configuração/gerenciamento de redes CANopen complexas será feita com ferramentas de software adequadas.
- Para simplificar isso, o padrão CiA 306 define um formato de arquivo INI legível por humanos (e amigável para máquinas), atuando como um "modelo" para o OD de um dispositivo - por exemplo, o "ServoMotor3000".
- Esse EDS normalmente é fornecido pelo fornecedor e contém *informações* sobre todos os objetos de dispositivo (mas não valores).





Device Configuration File

Device Configuration File

- Suponha que uma fábrica comprou um ServoMotor3000 para integrar em sua correia transportadora.
- Ao fazer isso, o operador edita o dispositivo EDS e adiciona *valores de parâmetros específicos* e/ou altera os nomes de cada objeto descrito no EDS.
- Ao fazer isso, o operador cria efetivamente o que é conhecido como Arquivo de configuração de dispositivo (DCF).
- Com isso instalado, o ServoMotor3000 está pronto para integração na rede CANopen específica no local.



Exemplo de EDC e DCF

EDS OBJECT 1006
; value for object 1006h
[1006]
SubNumber=0
ParameterName=ParaName
ObjectType=0x7
DataType=0x0007
LowLimit=1000
HighLimit=100000
DefaultValue=20000
AccessType=ro
PDOMapping=0

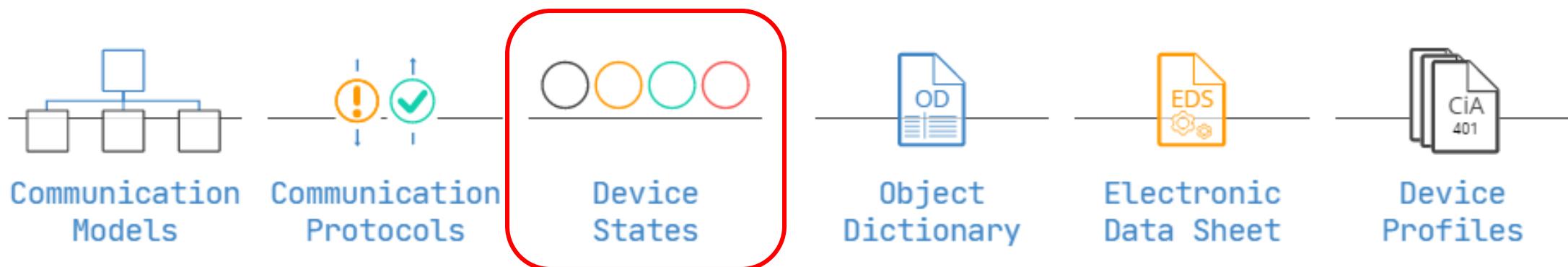
Comments
Comment field
16 bit object index
Number of sub indices
The object's name
Object type (7 = VAR)
OD index of data type
Minimum (if applicable)
Maximum (if applicable)
Used if no value given
Access: ro = read only
PDO mappable? (0 = no)
Specific parameter value

DCF OBJECT 1006
; value for object 1006h
[1006]
SubNumber=0
ParameterName=ParaName
ObjectType=0x7
DataType=0x0007
LowLimit=1000
HighLimit=100000
DefaultValue=20000
AccessType=ro
PDOMapping=0
ParameterValue=15000

- Revisar **exemplos reais de EDS/DCF** é uma das melhores maneiras de realmente entender o dicionário de objetos do CANopen - veja, por exemplo, a diferença entre uma entrada de objeto EDS e DCF. Recomendamos verificar o padrão CiA 306 para obter uma compreensão mais profunda do OD, EDS e DCF com exemplos práticos.
- Conforme mencionado, o DCF normalmente é criado na integração do dispositivo. No entanto, muitas vezes será necessário ler e/ou alterar os valores do objeto de um nó **após a configuração inicial** - é aqui que o serviço CANopen SDO entra em ação.

SDO - Configurando a rede CANopen

- Agora vamos dar dois passos atrás e voltar a falar dos serviços de comunicação SDO e PDO.



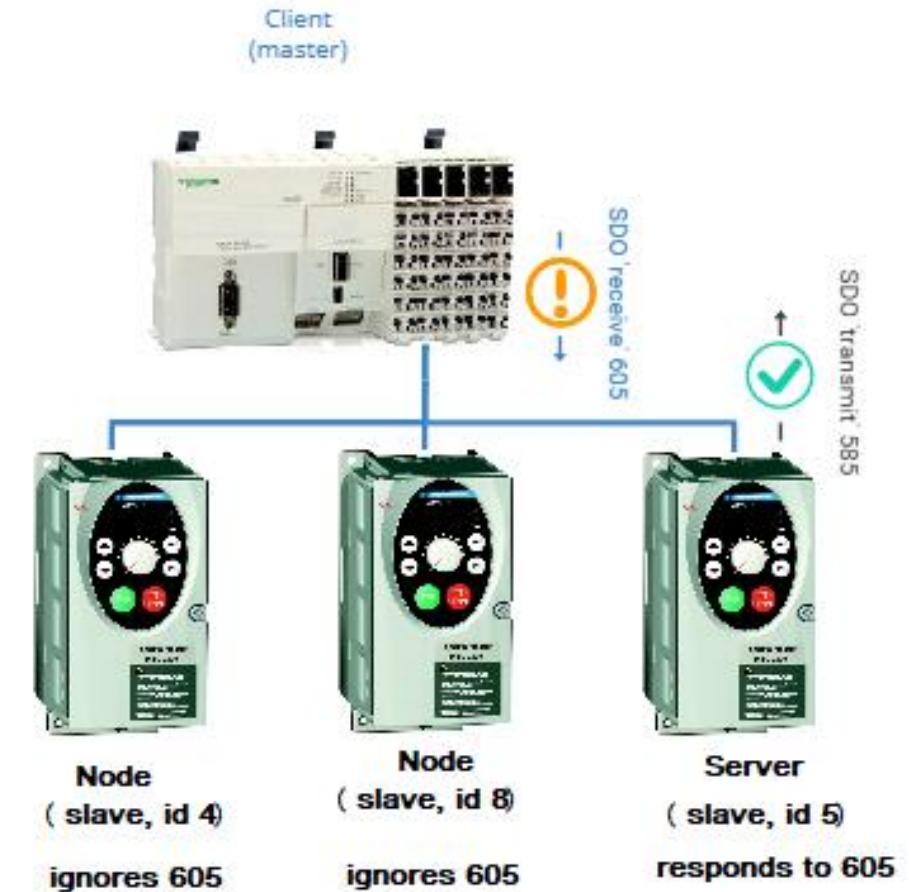
CANopen®

SDO - Configurando a rede CANopen

- O que é o serviço SDO?

O serviço SDO permite que um dispositivo CANopen leia/edita valores do dicionário de objetos de outro dispositivo pela rede CAN.

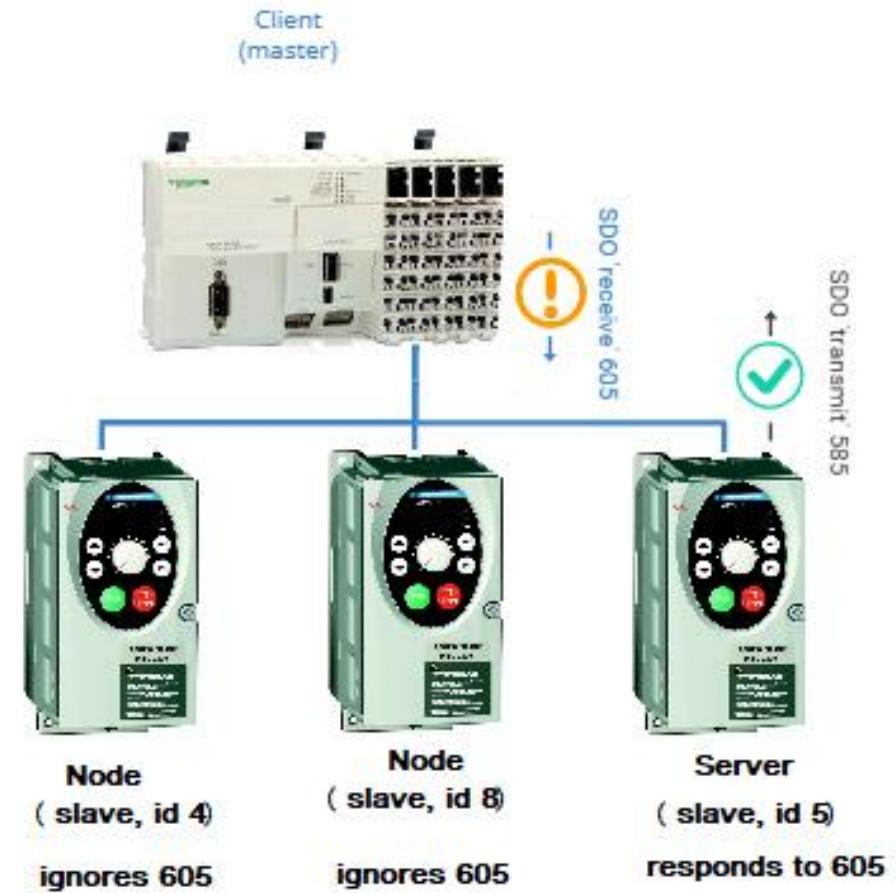
Conforme mencionado em 'modelos de comunicação', os serviços CANopen SDO utilizam um comportamento "cliente/servidor".



CANopen®

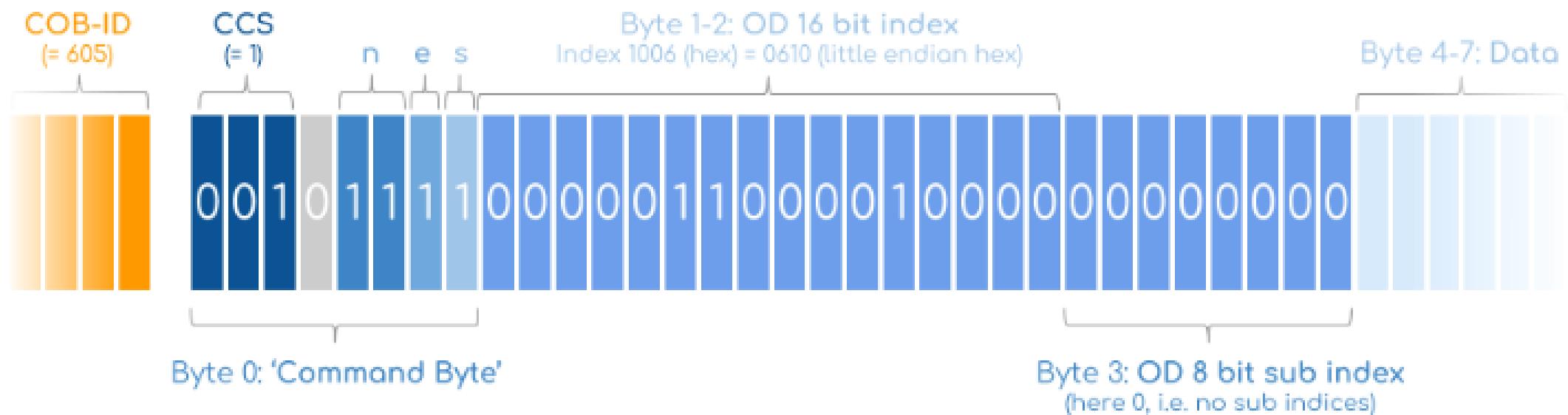
SDO Download e Upload

- Especificamente, um "cliente" SDO inicia a comunicação com um "servidor" SDO dedicado.
- O objetivo pode ser atualizar uma entrada OD (chamada de "download SDO") ou ler uma entrada ("upload SDO").
- Em redes mestre/escravo simples, o nó com funcionalidade de mestre NMT atua como o cliente para todos os nós escravos NMT lendo ou gravando em seus ODs.



Exemplo: download de SDO do nó cliente

- O nó cliente pode iniciar um download SDO para o nó 5 transmitindo abaixo do quadro CAN - o que acionará o nó 5 (e será ignorado por outros nós).
 - O quadro CAN SDO 'receive' (ou seja, request) tem a seguinte aparência:



Exemplo: download de SDO do nó cliente

- Primeiro, o **COB-ID** 605 reflete o uso de um 'SDO receive' (COB-ID 600 + ID do nó).
- O **CCS** (especificador de comando do cliente) é o tipo de transferência (por exemplo, **1**: Download, **2**: Upload).
- **n** é o #bytes nos bytes de dados 4-7 que *não* contêm dados (válido se **e** & **s** forem definidos).
 - ✓ n = 3 (11b) 1 Byte de dados sendo transmitido.
 - ✓ n = 2 (10b) 2 Bytes de dados sendo transmitidos.
 - ✓ n = 1 (01b) 3 Bytes de dados sendo transmitidos.
 - ✓ N = 0 (00b) 4 Bytes de dados sendo transmitidos.
- Se definido, **e** indica uma "transferência acelerada" (todos os dados estão em um único quadro CAN).
- Se definido, **s** indica que o tamanho dos dados é mostrado em **n**.
- **Índice** (16 bits) e **subíndice** (8 bits) refletem o endereço OD a ser acessado.
- Finalmente, os bytes 4-7 contêm os dados a serem baixados para o nó 5.

Comentários sobre a resposta SDO do servidor

- Depois que o quadro CAN é enviado pelo mestre (cliente), o nó escravo 5 (servidor) responde por meio de uma 'transmissão SDO' com COB-ID 585.
- A resposta contém o índice/subíndice e 4 bytes de dados vazios.
- Naturalmente, se o nó cliente solicitou um *upload* (ou seja, lendo dados do nó 5 OD), o nó 5 responderia com os dados relevantes contidos nos bytes 4-7.
- Alguns comentários:
 - Como fica evidente, cada SDO usa 2 identificadores, criando um "canal SDO"
 - O exemplo é simplificado, pois é "acelerado" (os dados estão contidos nos 4 bytes)
 - Para cenários de dados maiores, as transferências de bloco/segmentação SDO podem ser usadas
- Os SDOs são flexíveis, mas carregam muita sobrecarga - tornando-os menos ideais para dados operacionais em tempo real.

PDO - operando a rede CANopen

- Antes de mais nada: O que é o serviço CANopen PDO?

O serviço CANopen PDO é usado para compartilhar efetivamente dados operacionais em tempo real entre nós CANopen.

Por exemplo, o PDO transportaria dados de pressão de um transdutor de pressão - ou dados de temperatura de um sensor de temperatura.



Transmissor de pressão com interface CANopen® - D-20-9, D-21-9 - WIKA Brasil

Mas espere: o serviço SDO não pode fazer isso?



CANopen®

- Sim, a princípio o serviço SDO poderia ser usado para isso.
- No entanto, uma única resposta SDO pode transportar apenas 4 bytes de dados devido ao overhead (byte de comando e endereços OD).
- Além disso, digamos que um nó mestre precise de dois valores de parâmetro (por exemplo, "SensTemp2" e "Torque5") do nó 5.
- Para obtê-lo via SDO, seriam necessários 4 quadros CAN completos (2 solicitações, 2 respostas).

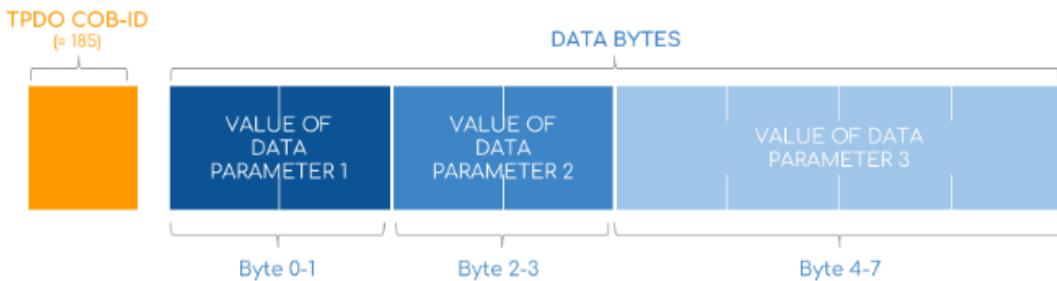


CANopen®

*Mas espere: o
serviço SDO não
pode fazer isso?*

- Em contraste, uma mensagem PDO pode conter 8 bytes completos de dados - e pode conter vários valores de parâmetro de objeto em um único quadro.
- Assim, o que exigiria pelo menos 4 quadros com SDO poderia potencialmente ser feito com 1 quadro no serviço PDO.
- O PDO é frequentemente visto como o protocolo CANopen mais importante, pois carrega a maior parte das informações.

Como funciona o serviço PDO no CANopen?

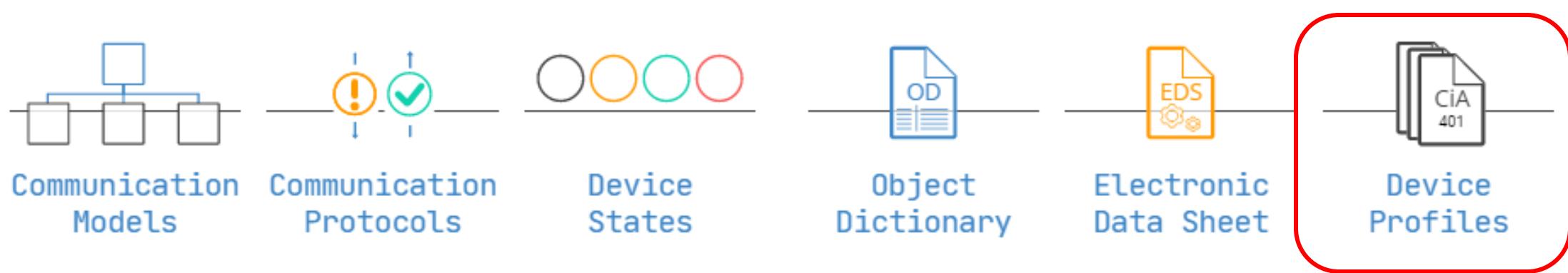


- Para PDOs, a terminologia consumidor/produtor é usada.
- Assim, um produtor 'produz dados', que transmite a um 'consumidor' (mestre) usando um PDO de transmissão (TPDO).
- Por outro lado, ele pode receber dados do consumidor por meio de um PDO de recebimento (RPDO).
- Os nós produtores podem, por exemplo, ser configurados para responder a um gatilho SYNC transmitido pelo consumidor a cada 100 ms.
- O nó 5 pode então, por exemplo, transmitir um PDO com COB-ID 185:
- Observe como os bytes de dados são compactados com 3 valores de parâmetro.
- Esses valores refletem dados em tempo real de entradas OD específicas do nó 5.
- Os nós que usam essas informações (os consumidores) obviamente precisam saber como interpretar os bytes de dados PDO.

Diferenças do PDO x SDO

- SDO
 - Acesso a todos os parâmetros no dicionário de objetos.
 - Opera por requisição e resposta.
 - Usado para configuração e diagnóstico.
 - Pode acessar dados de qualquer tamanho.
- PDO
 - Otimiza o acesso para parâmetros pré-selecionados do dicionário de objetos.
 - Self triggered Broadcast
 - Trigger por tempo, evento, mensagem de sincronismo ou outros.

Device Profiles e Application Profiles



canopen

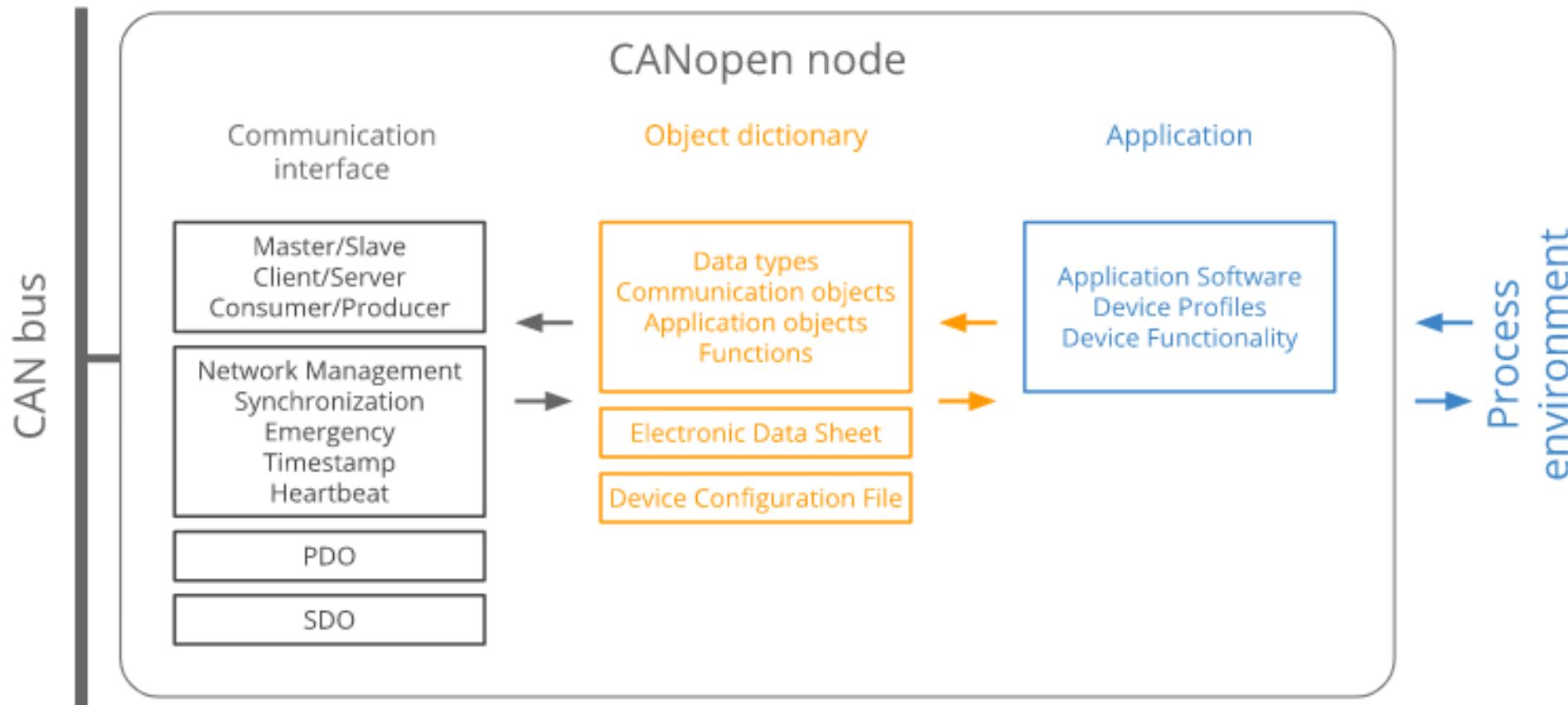


CANopen Node Profiles

- CiA Profiles:
- CiA401 Generic I/O
- CiA402 Drives and motion control
- CiA406 Encoders
- CiA410 inclinometer
- CiA418/419 Batteries/Charger
- CiA420 Extruder
- CiA450 Pumps
- CiA Application Profiles
 - CiA417 Lift control systems
 - CiA420/423/424/426/430/433 Train/rail vehicle
 - CiA422 Municipal vehicles (CleANopen)
 - CiA437 Photovoltaic systems
 - CiA447 Special purpose vehicles (Emergency response)
 - CiA454 Energy Management (EnergyBus – e.g. E-Bikes).

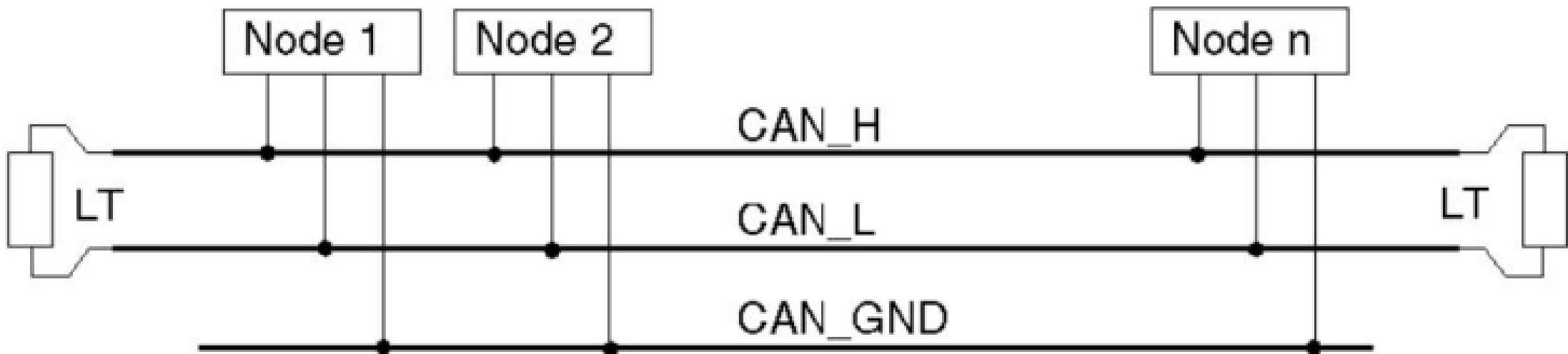
CANopen®

Como os conceitos do CANopen se conectam



Arquitetura geral de uma rede CANopen

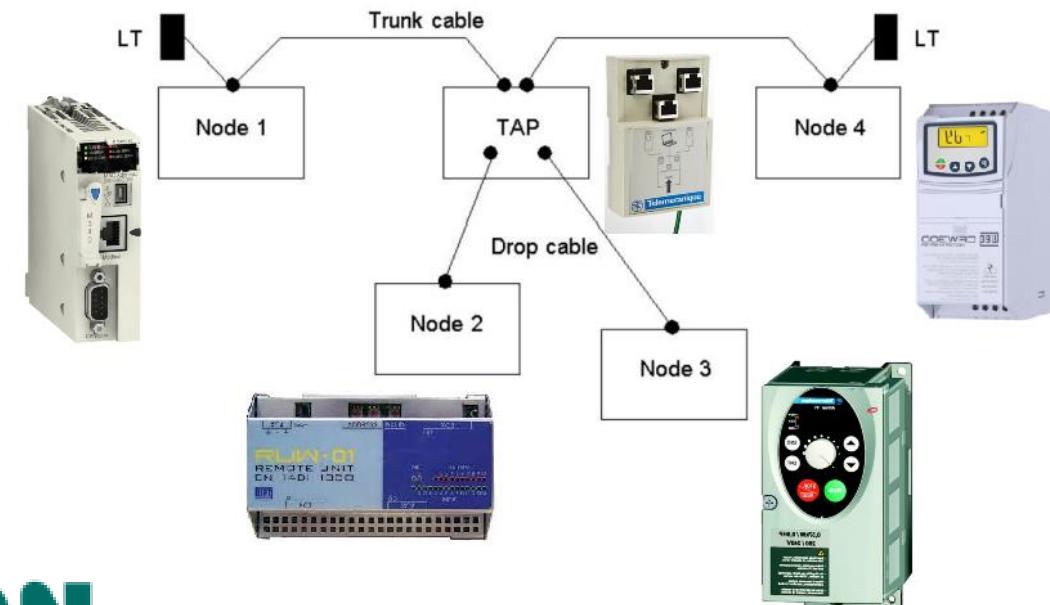
- A rede CANopen utiliza um cabo de par trançado para transmitir os sinais diferenciais, terminados em ambas as extremidades físicas com resistores de 120 ohms.
- Um sinal de terra separado é usado como referência comum para os nós CANopen.
- Além dos 3 fios mencionados acima, alguns cabos fornecem este recurso com um quarto fio para alimentação remota dos dispositivos.



CANopen®

Topologia Básica

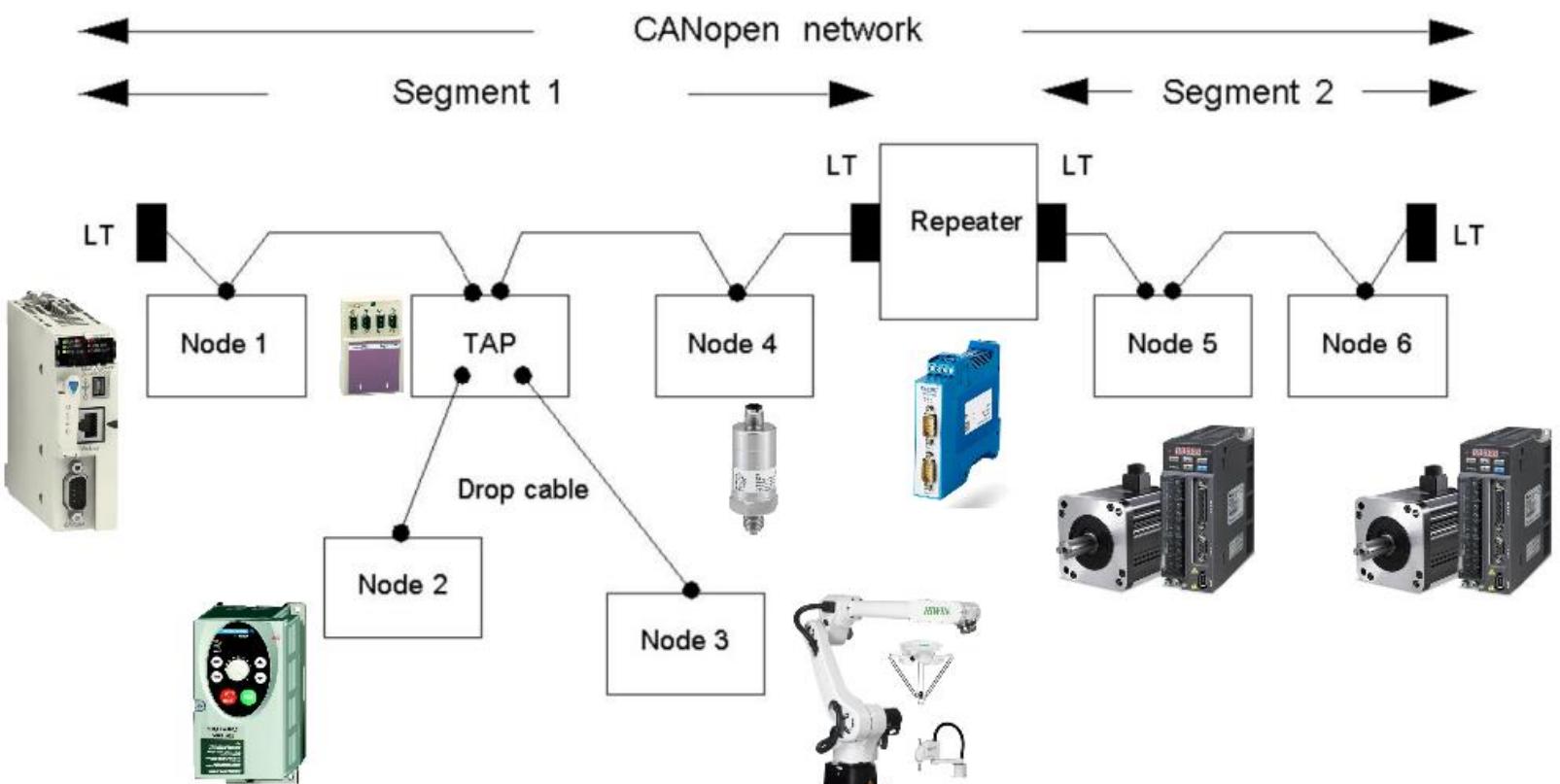
- A rede CANopen consiste em uma linha de transmissão que deve ser terminada em ambas as extremidades físicas com resistores de terminação.
- Um TAP em combinação com cabos drop formam uma topologia em estrela parcial.
- Para minimizar as reflexões, mantenha os cabos drop o mais curtos possível.
- O comprimento máximo dos cabos drop depende da velocidade de transmissão.



CANopen®

Topologia com repetidor

- A rede CANopen pode ser composta por 1 ou vários segmentos, interligados fisicamente através de um repetidor CAN.



CANopen®

Funções do repetidor

Um repetidor:

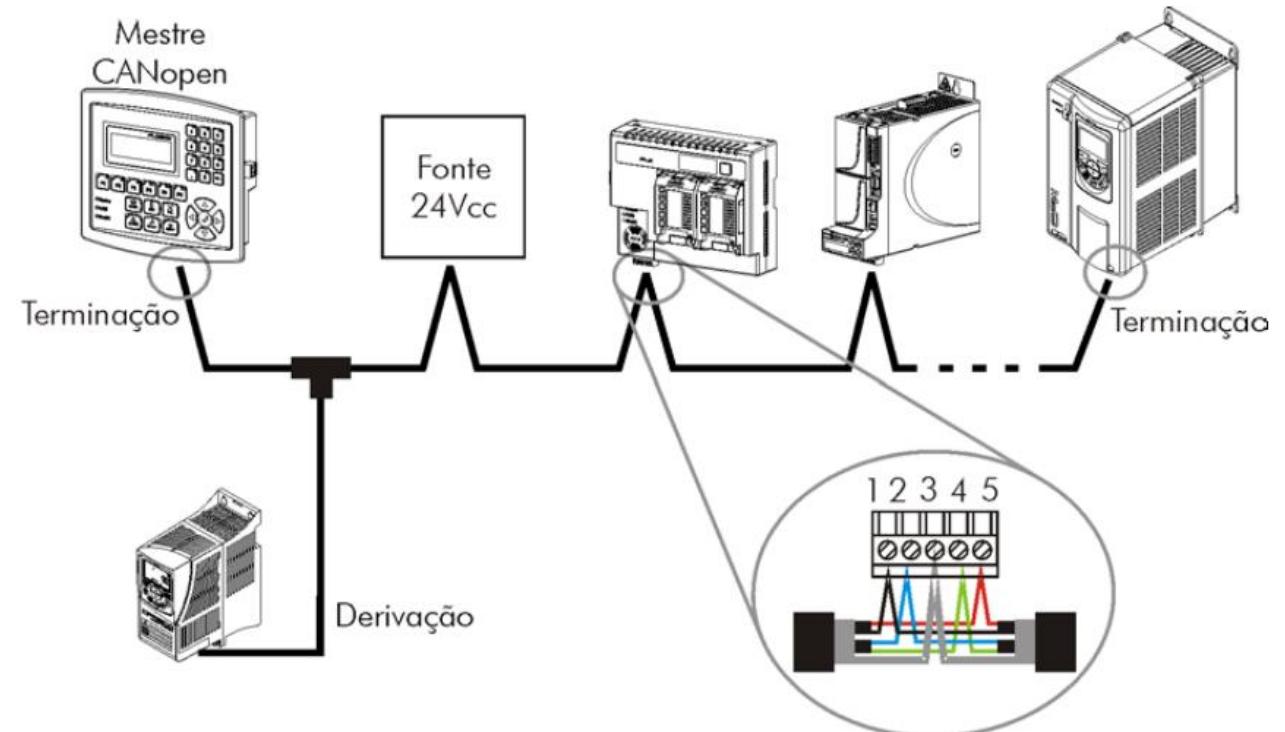
- Fornece uma regeneração dos sinais CAN, permitindo assim mais de 64 nós.
- Pode fornecer isolamento entre os segmentos, sendo que cada um desses segmentos deve ser terminado.
- É transparente, do ponto de vista da rede, pois simplesmente encaminha os sinais CAN. Isso significa que os dispositivos conectados ao barramento participam da mesma arbitragem.
- Não permite aumentar o comprimento total do cabo.



CANopen®

Conexão por Encadeamento de cabo

- O encadeamento do cabo de 1 nó para o próximo é realizado através dos conectores de 2 maneiras diferentes:
- Conectando 2 cabos ao mesmo conector de cabo.
- Esta técnica de encadeamento amplamente utilizada permite desconectar o conector do cabo do dispositivo (ou seja, substituição do dispositivo) sem interrupção da rede.

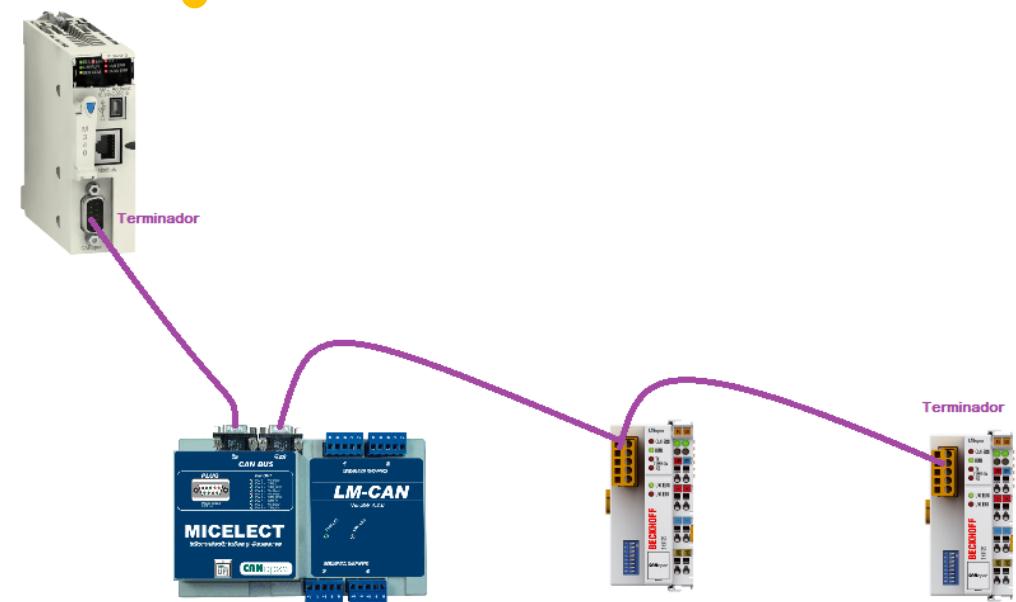


Exemplo de instalação em rede CANopen

CANopen®

Conexão por Encadeamento de cabo

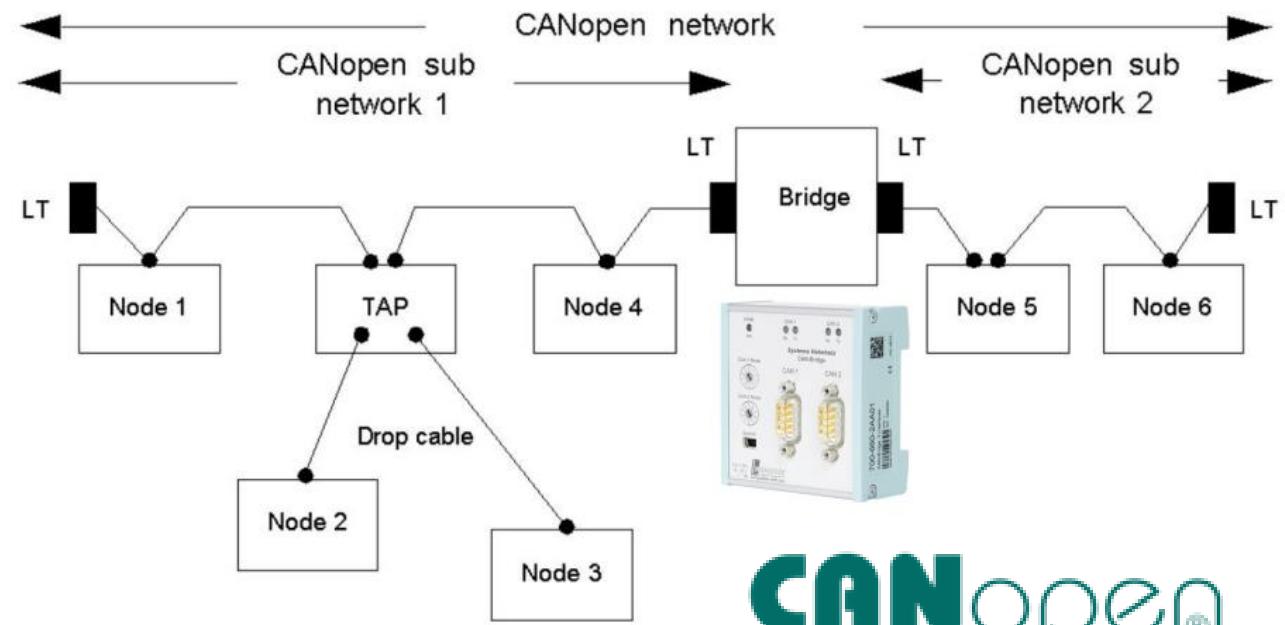
- Conectando os 2 cabos aos conectores de cabos individuais em dispositivos que fornecem 2 conectores.
- Esta técnica de encadeamento é usada especialmente em dispositivos de alta proteção (ou seja, dispositivos IP67) ou para sistemas de cabeamento otimizados no gabinete.



CANopen®

Topologia com uma Bridge (ponte)

- Uma rede geral CANopen pode ser separada em sub-redes mais ou menos independentes através de uma ponte CAN.



Topologia com uma Bridge (ponte)

Funções de uma bridge:

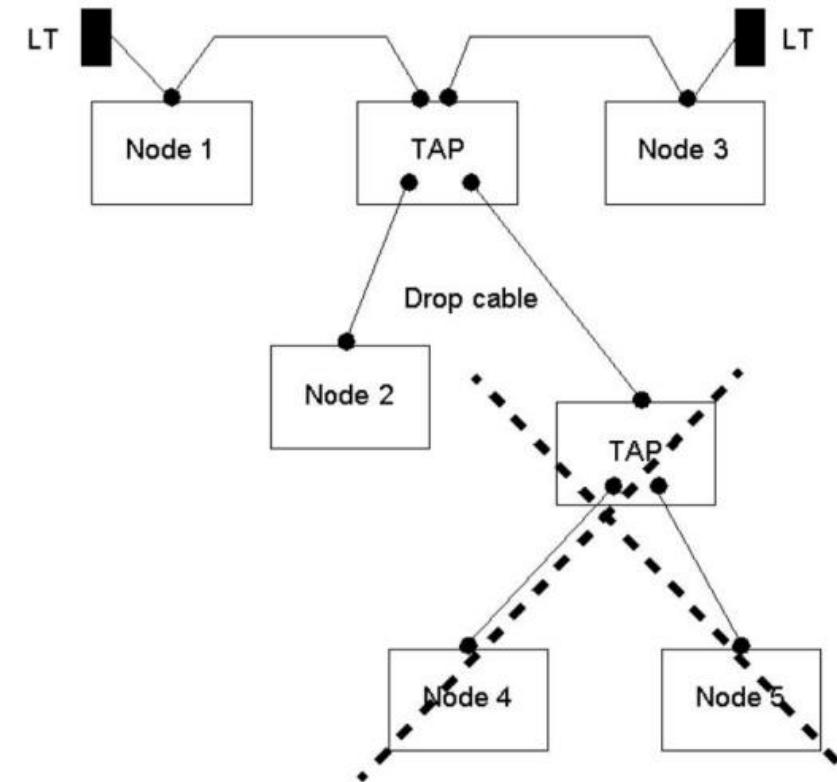
- Separa a rede CAN geral em sub-redes mais ou menos independentes.
- Fornece uma arbitragem individual para cada sub-rede.
- Oferece a possibilidade de cada sub-rede ter sua própria velocidade de transmissão.
- É baseado no princípio de armazenamento e encaminhamento, ou seja, as mensagens CAN são recebidas de uma sub-rede e são então encaminhados para outra sub-rede.
- Permite o uso de regras de tradução e filtro.
- Permite realizar uma adaptação de protocolo entre as sub-redes.
- Ao contrário do repetidor CAN, a ponte CAN permite ampliar o tamanho máximo da rede.



CANopen®

Cascata de TAPs

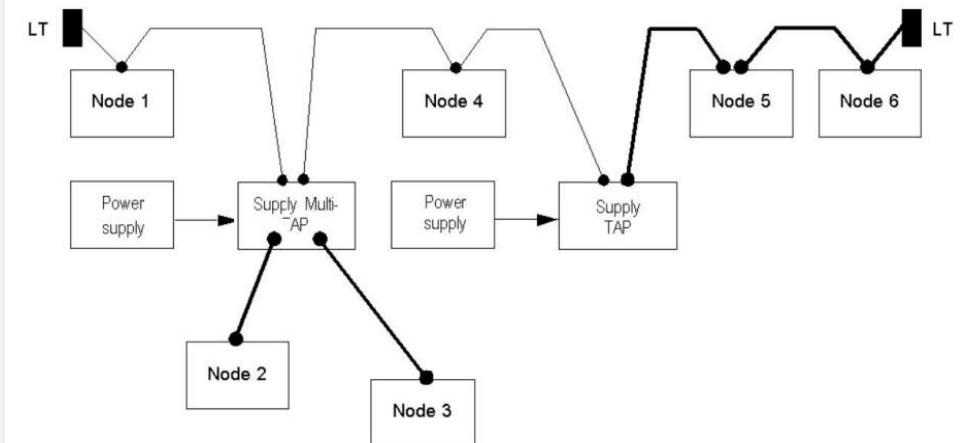
- Em redes CANopen não é permitido o cascateamento de TAPs, pois isso prejudicaria a característica da linha de transmissão.



CANopen®

Topologia com fonte de alimentação externa

- Para fornecer energia aos nós da rede CANopen, uma fonte de alimentação externa pode ser conectada a um TAP.
- A energia é transportada pelos sinais CAN_V+ e CAN_GND. Como esses sinais são fornecidos em cabos CAN padrão, não são necessários cabos especiais para alimentação.
- Para encaminhar a energia através do cabo, é necessário que o sinal CAN_V+ esteja conectado no conector do cabo de cada nó, mesmo que o respectivo nó não utilize a energia em si, mas a encaminhe para um nó seguinte.
- NOTA: Repetidores, pontes e cabos RJ45 não encaminham de forma alguma o sinal CAN_V+



CANopen®

Sistema de cabeamento CANopen

- Termos e definições da CiA 303-1:
- Bus cable: Cabo terminado em ambas as extremidades por resistores de terminação.
- Stub cable: Ramificação curta de um cabo de barramento não terminado por um resistor e conectado ao dispositivo CANopen.
- T-connector: Conector elétrico em forma de T com três pontos de conexão para um cabo de barramento.
- Trunk cable: O backbone do cabo do barramento sem os cabos stub.



CANopen®

Exemplo de
conector em T
para sistema de
cabeamento
CAN

CANopen®



Recomendações da CiA 303-1

- Os cabos de barramento, conectores e resistores de terminação devem corresponder em relação à impedância.
- Incompatibilidades de impedância causam reflexões, que podem levar a erros de comunicação.
- O cabo stub deve ser o mais curto possível.
- Os dispositivos CANopen podem ser conectados à rede diretamente ao T-conector ou com um cabo stub.



CANopen®

- Tabela 1 – Valores recomendados para redes CANopen

- Tabela 1 mostra alguns valores recomendados para redes CANopen com menos de 64 nós.
 - (fonte: Tabela 4 e Tabela 5 na AN 96116:1996). mm²

Bus length [m]	Bus cable		Termination resistance [Ω]	Bit rate [kbit/s]
	Length-related resistance [m Ω /m]	Cross-section [mm ²]		
0 to 40	70	0,25 to 0,34	124	1000 at 40 m
40 to 300	<60	0,34 to 0,6	150 to 300	\leq 500 at 100 m
300 to 600	<40	0,5 to 0,6	150 to 300	<100 at 500 m
600 to 1000	<26	0,75 to 0,8	150 to 300	<50 at 1 km



Tabela 2 - Comprimento máximo da fiação

- Tabela 2 recomenda o comprimento máximo da fiação fornecido para diferentes cabos de barramento e diferentes números (n) de nós de barramento conectados (fonte: Tabela 6 em AN 96116:1996).

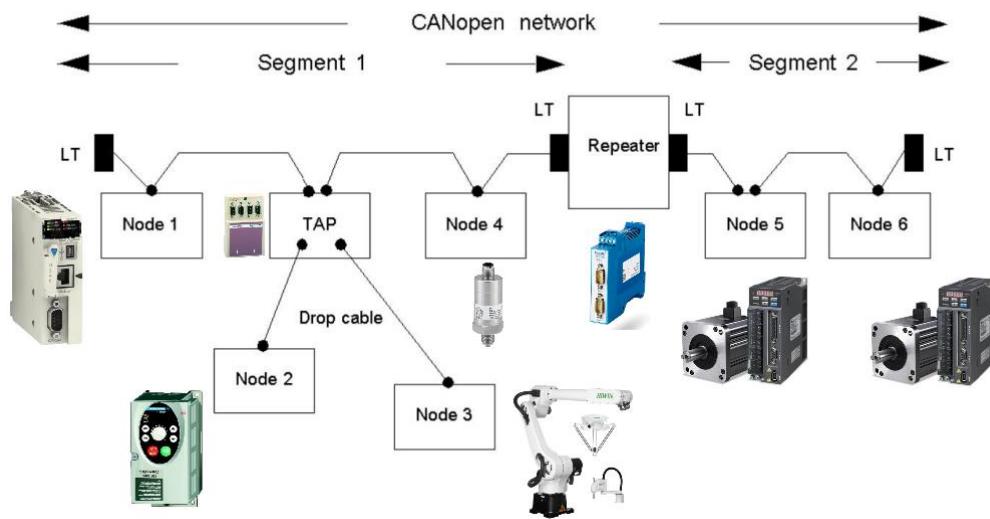
Wire cross-section [mm ²]	Maximum length [m] ^a			Maximum length [m] ^b		
	n = 32	n = 64	n = 100	n = 32	n = 64	n = 100
0,25	200	170	150	230	200	170
0,5	360	310	270	420	360	320
0,75	550	470	410	640	550	480

^a safety margin of 0,2 m

^b safety margin of 0,1 m

Design de rede

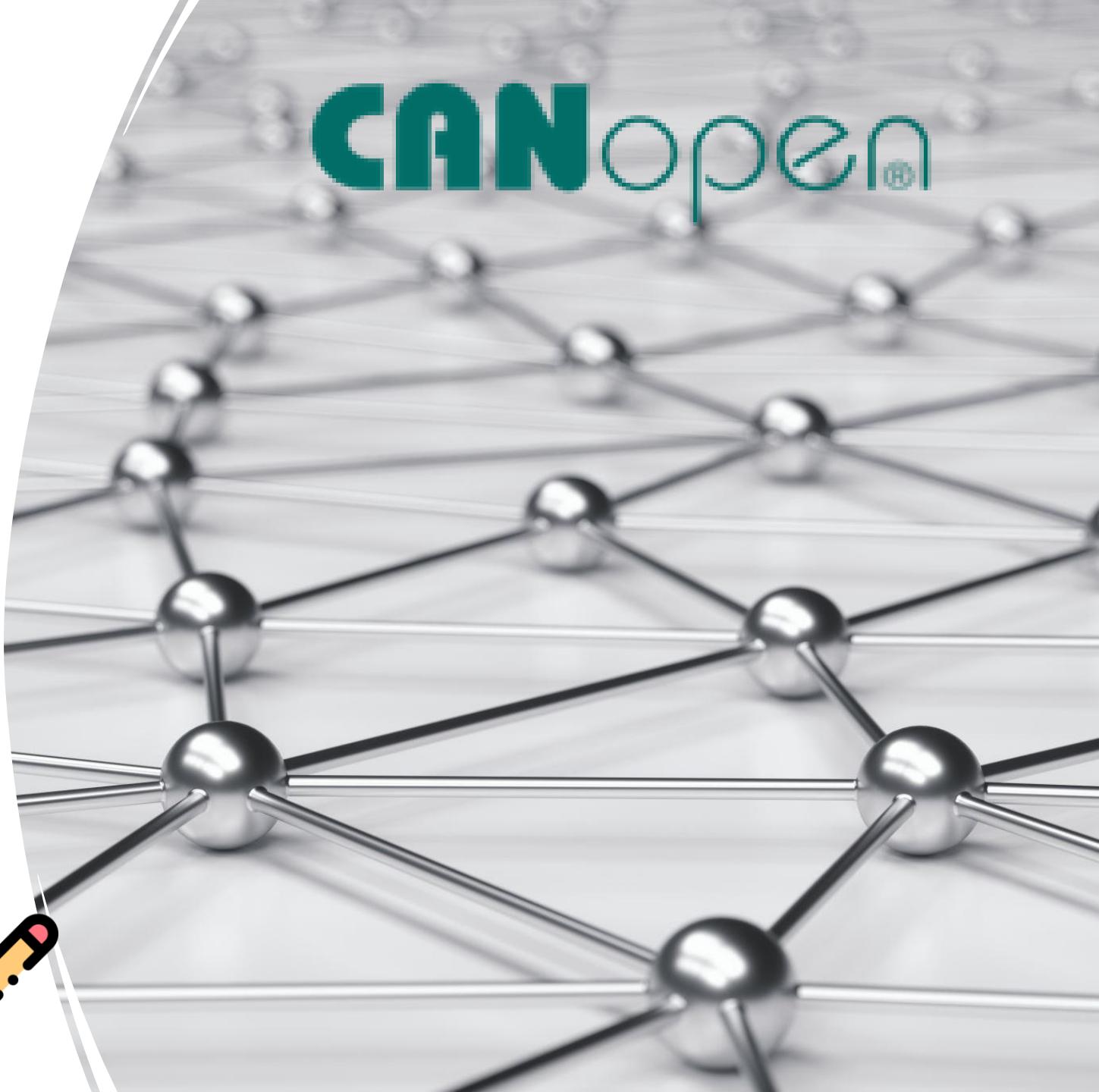
- Nessa aula vamos tratar das regras de projeto de uma rede CANopen, buscando descrever a relação entre o comprimento do cabo e a velocidade de transmissão, as limitações dos cabos drop, bem como as especificações aplicáveis às redes com fontes de alimentação externas.



CANopen®

Princípios de Design de Rede

- Documente e registre cuidadosamente o projeto de rede com cálculos associados, pois essa documentação será muito útil ao planejar mudanças futuras.
- Também ajudará a manter o desempenho do barramento.



Regras ao se projetar o barramento CANopen

- 1º - Atribuir a cada nó um endereço de nó CANopen exclusivo.
- 2º - Certifique-se de que os nós em uma rede tenham a mesma velocidade de transmissão.
- 3º - Verificar o comprimento dos cabos drops e a densidade desses drops.
- 4º - Verifique se os segmentos têm um resistor de terminação conectado em cada extremidade.



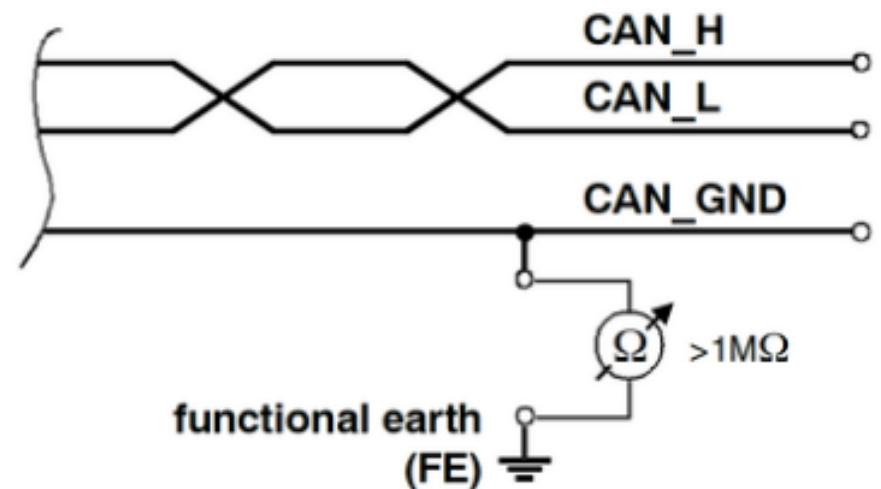
Instalação de cabos



- O barramento CANopen foi projetado para ser utilizado dentro de edifícios, em máquinas móveis, em um ambiente de oficina ou até mesmo em uma fábrica.
- Como para qualquer outro barramento industrial, você deve, no entanto, seguir regras rígidas de instalação para obter o desempenho total do barramento.

Blindagem e Aterramento

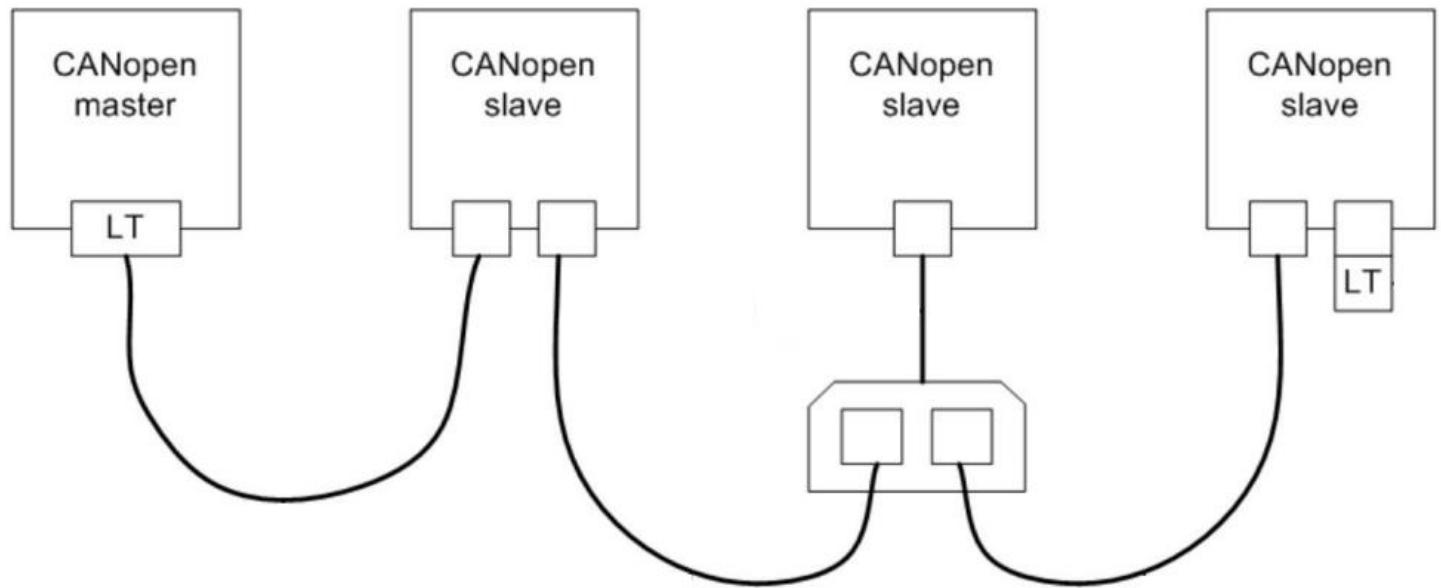
- Para limitar os distúrbios de modo comum e obter um alto nível de robustez contra EMC, tome as seguintes precauções:
- Em redes CANopen completamente isoladas galvanicamente, o sinal de terra CAN deve ser transportado na linha de cabo.
- Deve ser conectado em apenas um ponto ao potencial de terra do CAN.
- Se um dispositivo CANopen com uma interface não isolada galvanicamente for conectado à rede, a conexão com o potencial de terra CAN é fornecida.
- Portanto, apenas um dispositivo com uma interface não isolada galvanicamente deve ser conectado à rede.
- O projetista de rede é responsável por garantir que a rejeição de modo comum dos transceptores ainda atinja o limite superior.
- O CANopen usa cabos de par trançado blindados. Em cada dispositivo, a blindagem é conectada ao terra funcional. Isto é obtido, por exemplo, automaticamente através da caixa metálica do conector do cabo SUB-D 9.



CANopen®

Cabeamento em Gabinete CANopen

- Normalmente o cabeamento de dispositivos em gabinete é baseado em conectores RJ45.
- Se cada dispositivo CANopen usado no gabinete estiver equipado com um conector RJ45, a topologia da rede é uma simples cadeia sem TAPs.



CANopen®

Limitações desse tipo de ligação:

- Se você esses elementos de infraestrutura na linha tronco, as seguintes restrições se aplicam:
- O comprimento máximo do cabo é reduzido em 50% em comparação com um cabo CANopen padrão.
- Esses elementos de infraestrutura de cabeamento em gabinete devem ser usados somente dentro de 1 único gabinete.
- Para distribuir a rede CANopen em vários gabinetes diferentes, use o cabo CANopen padrão para conectar os gabinetes.
- Os conectores RJ45 não fornecem CAN_V+ e, portanto, não possibilitam a distribuição de energia.





CANopen®

Limitações da camada física

- Agora vamos listar as restrições que você deve conhecer ao configurar uma rede CANopen e também veremos um pouco de solução de problemas para ajudá-lo a resolver algo que possam surgir durante a instalação.

CANopen®

Velocidade de transmissão e comprimento do cabo

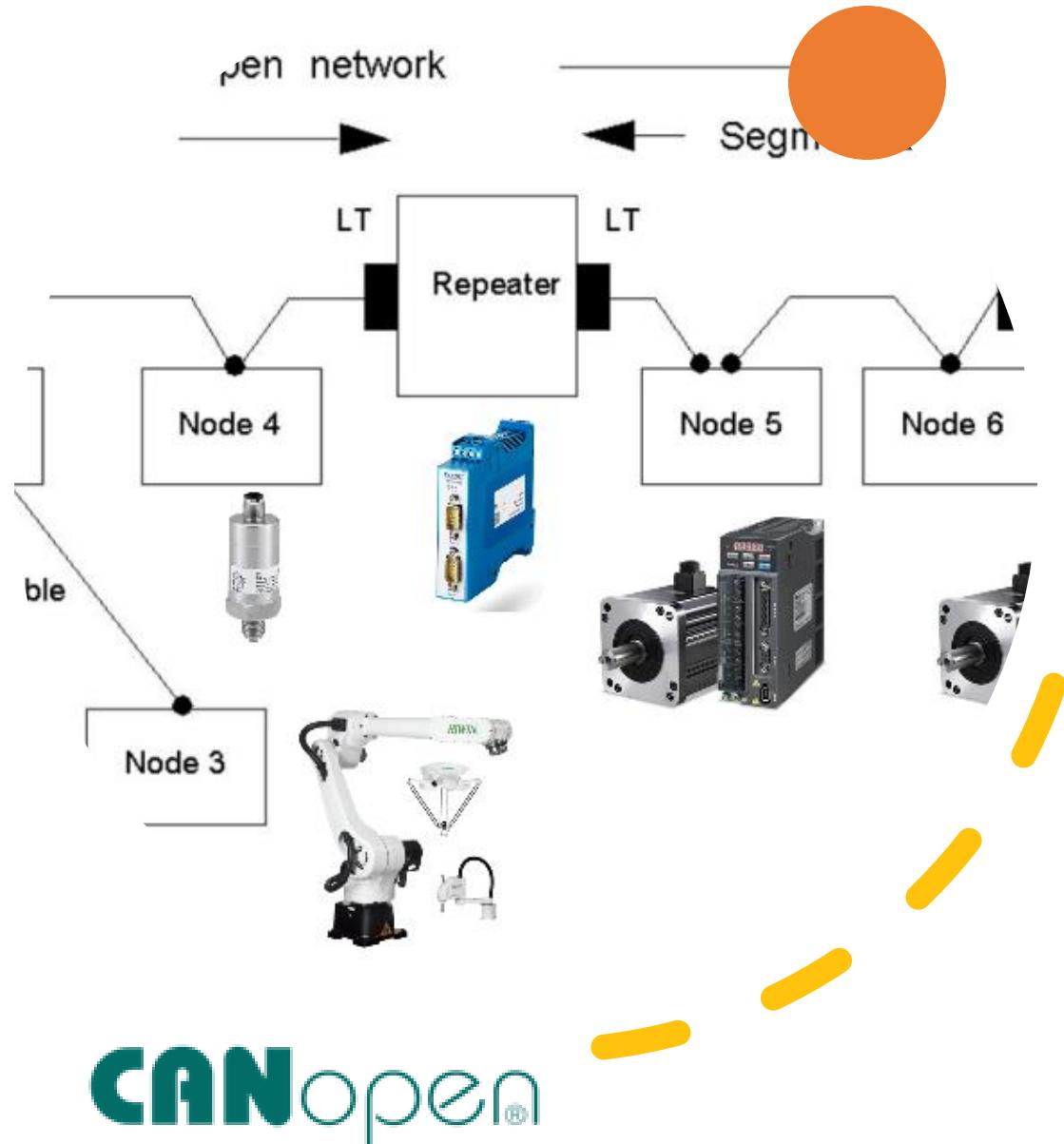
- Nós aprendemos que a rede CANopen permite 127 dispositivos (o mestre do barramento e 126 escravos remotos).
- A velocidade de transmissão depende estritamente do tipo de cabo utilizado e do comprimento da rede.
- A estimativa do comprimento do barramento é baseada em um atraso de propagação de 5 ns/m.
- Os tempos de atraso de controladores CAN usados, transceptores CAN e optoacopladores precisam ser considerados adicionalmente.

Bit rate	Bus length ⁽¹⁾
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
50 kbit/s	1.000 m
20 kbit/s	2.500 m
10 kbit/s	5.000 m

CANopen®

Repetidores reduzem o comprimento do cabo

- Os valores apresentados especificam o comprimento máximo do cabo sem qualquer repetidor.
- Como os repetidores adicionam um atraso de propagação no barramento, esse atraso reduz o comprimento máximo do barramento.
- Um atraso de propagação de 5 ns leva a uma redução de comprimento de 1 mt.
- Exemplo. Um repetidor com um atraso de propagação de 150 ns reduz o comprimento máximo do cabo em 30 mt.



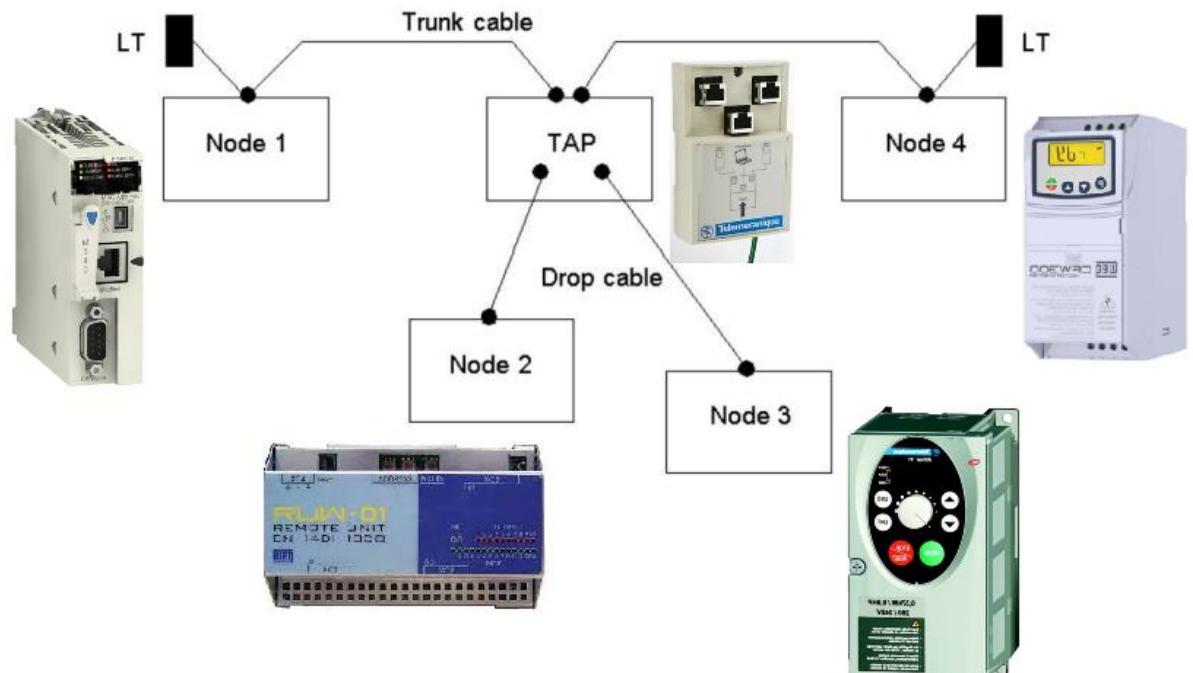
Isolamento Elétrico de Dispositivos CANopen

- Em documentos sobre CANopen você encontrará frequentemente o valor máximo de 40 mt a uma velocidade de transmissão de 1 Mbit/s.
- Este comprimento é calculado sem isolamento elétrico como usado nos dispositivos CANopen.
- Em qualquer caso, o número máximo de nós que podem ser conectados no mesmo segmento é restrito a 64.
- Para conectar mais nós a 1 segmento, use um repetidor.
- Exemplo: Um repetidor com um atraso de propagação de 150 ns reduz o comprimento máximo do cabo em 30 mt.
- Com esse isolamento elétrico, o comprimento mínimo da rede calculado é de 4 mt a uma velocidade de transmissão de 1 Mbit/s.
- No entanto, a experiência mostra que 20 mt são o comprimento prático que pode ser encurtado por quedas ou outras influências.



Limitações do cabo drop

- Um cabo drop cria uma reflexão de sinal na linha de transmissão característica do cabo tronco.
- Para limitar as reflexões, os cabos drop devem ser o mais curtos possível.



CANopen®

Comprimento máximo do cabo drop



Os seguintes valores devem ser respeitados:

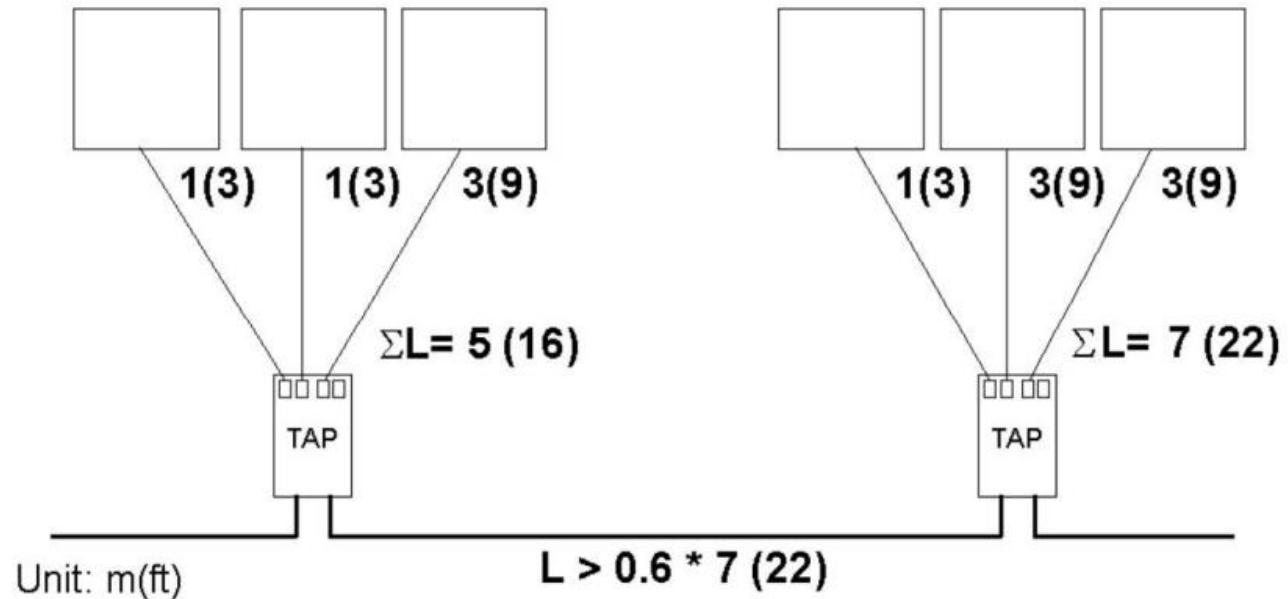
Transmission Rate	Lmax	ΣL_{max}	TAP Distance	ΣLG_{max}
1 Mbit/s	0.3 m (0.98 ft)	0.6 m (0.98 ft)		1.5 m (4.92 ft)
800 kbit/s	3 m (9.84 ft)	6 m (19.68 ft)	3.6 m (11.81 ft)	15 m (49.21 ft)
500 kbit/s	5 m (16.4 ft)	10 m (32.8 ft)	6 m (19.68 ft)	30 m (98.42 ft)
250 kbit/s	5 m (16.4 ft)	10 m (32.8 ft) 	6 m (19.68 ft)	60 m (196.84 ft)
125 kbit/s	5 m (16.4 ft)	10 m (32.8 ft)	6 m (19.68 ft)	120 m (393.69 ft)
50 kbit/s	60 m (196.84 ft)	120 m (393.69 ft)	72 m (236.21 ft)	300 m (984.24 ft)
20 kbit/s	150 m (492.12 ft)	300 m (984.24 ft)	180 m (590.54 ft)	750 m (2460.62 ft)
10 kbit/s	300 m (984.24 ft)	600 m (1968.49 ft)	360 m (1181.09 ft)	1500 m (4921.24 ft)

- **Lmax** é o comprimento máximo de 1 cabo drop.
- **ΣL_{max}** é o valor máximo da soma dos cabos drop no mesmo TAP.
- **TAP distance** é a distância mínima necessária entre 2 TAPs, podendo ser calculada para cada TAP (deve ser maior que 60% do maior dos 2 ΣL_{max}).
- **ΣLG_{max}** é o valor máximo da soma dos cabos drop na rede.

Exemplo de cálculo

CANopen®

- Exemplo de cálculo de distância TAP com 2 caixas de junção e 6 dispositivos:
- A distância TAP no exemplo é calculada da seguinte forma:
- 1º Calculando a soma dos comprimentos dos cabos drop para cada junção de derivação. Que dá 5 mt e 7 mt.
- 2º Mantendo o maior comprimento. Que é 7 mt.
- 3º Calculando o comprimento mínimo do cabo entre os 2 TAPs. Que dá 60% de 7 mt.



Respeite a distância TAP mesmo se um dispositivo estiver no meio.

Rede com fonte de alimentação externa

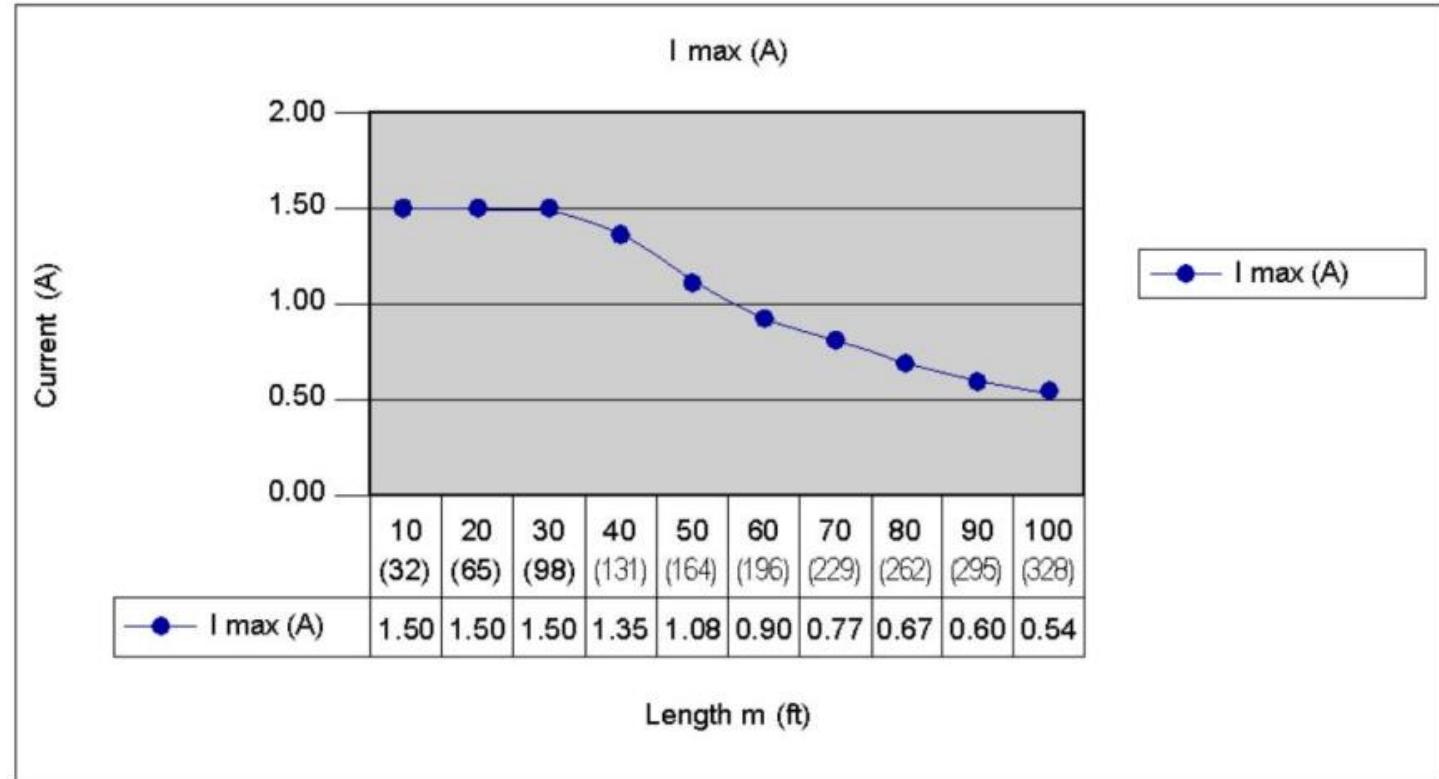
- Características básicas:
- A tensão nominal da fonte de alimentação deve ser de 24 V.
- A tensão de saída na fonte de alimentação opcional deve ser +18 VCC a +30VCC.
- A soma da corrente consumida pelos dispositivos alimentados de 1 fonte de alimentação não deve exceder 1500 mA.
- Isso também se aplica a um único dispositivo.

Standard	IEC61131-2:2003, PELV or SELV
Initial tolerance	24 V +/- 3% or better (no load voltage)
Line regulation	+/- 3% max
Load regulation	+/- 3% max
Output ripple	200 mV p-p max
Load capacitance capability	7000 µF max
Isolation	output isolated from AC and chassis ground
Minimum output voltage	19.2 V at full load
Current limit	2 A



Limitação do comprimento do cabo

- Com base na quantidade de corrente, uma certa queda de tensão no cabo aparecerá.
- Essa queda de tensão e, portanto, o comprimento do cabo, precisa ser limitada.



Verificações e solução de problemas

- Para comunicações de rede CANopen confiáveis, execute as verificações descritas a seguir.

Verificando a configuração do dispositivo:

- ✓ Verifique se os dispositivos conectados estão configurados para a mesma velocidade de transmissão.
- ✓ Verifique se cada dispositivo possui um endereço de nó exclusivo.



CANopen®

Verificações e solução de problemas

- Verificando a topologia :
 - ✓ Verifique o comprimento máximo do cabo versus a velocidade de transmissão.
 - ✓ Verifique o comprimento do segmento e o número de nós no segmento.
 - ✓ Verifique o comprimento dos cabos drop e a distância TAP versus velocidade de transmissão.

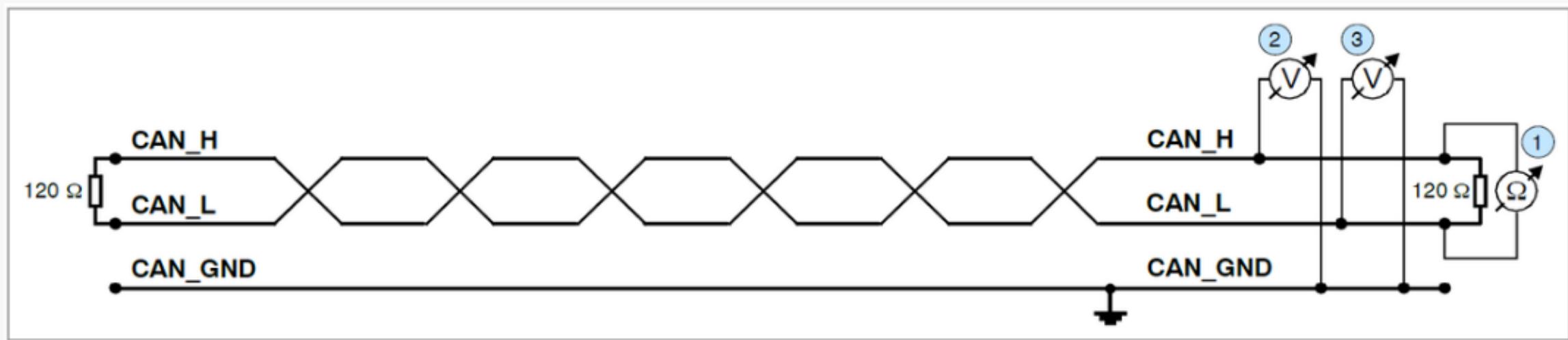


CANopen®

A terminação correta do barramento CAN

- A terminação do barramento é usada para casar a resistência de um nó com a resistência da linha do barramento usada.
- Se a impedância for incompatível, o sinal transmitido não será totalmente absorvido pela carga e será parcialmente refletido de volta na linha de transmissão.
- Se as impedâncias das fontes, linhas de transmissão e carga forem iguais, as reflexões são evitadas.
- Este teste mede a resistência total das duas linhas de dados CAN e os resistores de terminação conectados.

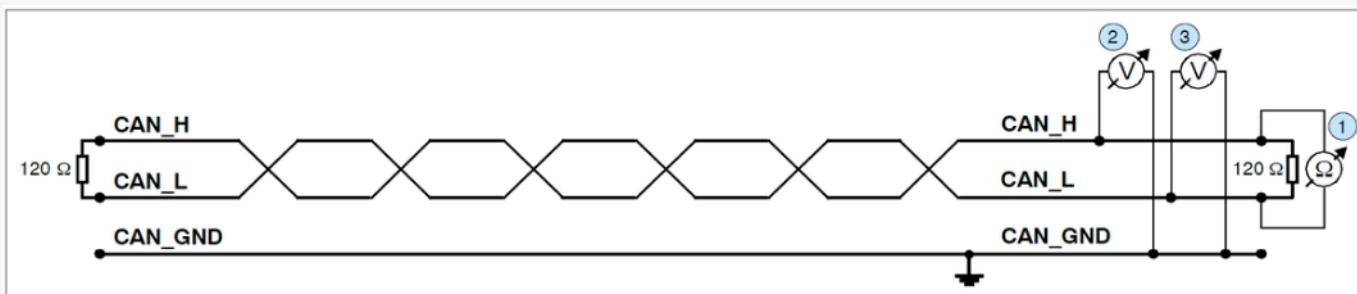
Diagrama de circuito simplificado de uma rede CAN



Para testar, proceda da seguinte forma:

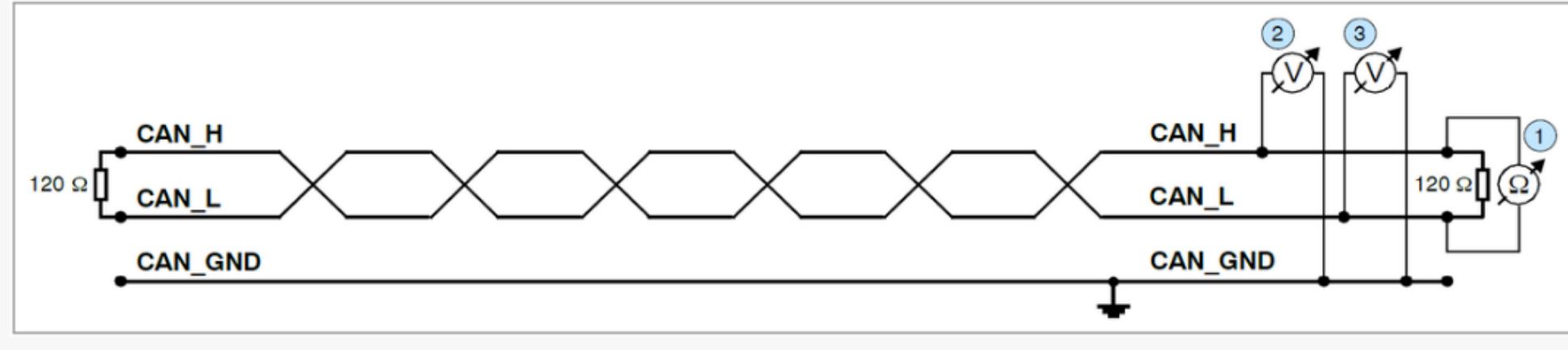
- Desligue as tensões de alimentação de todos os nós CAN conectados.
- Meça a resistência CC entre CAN_H e CAN_L em uma extremidade da rede (ponto de medição (1), consulte o diagrama de circuito de uma rede CAN acima).
 - Resultado esperado:
O valor medido deve estar entre 50 Ω e 70 Ω.

Diagrama de circuito simplificado de uma rede CAN



CANopen®

Diagrama de circuito simplificado de uma rede CAN



CANopen®

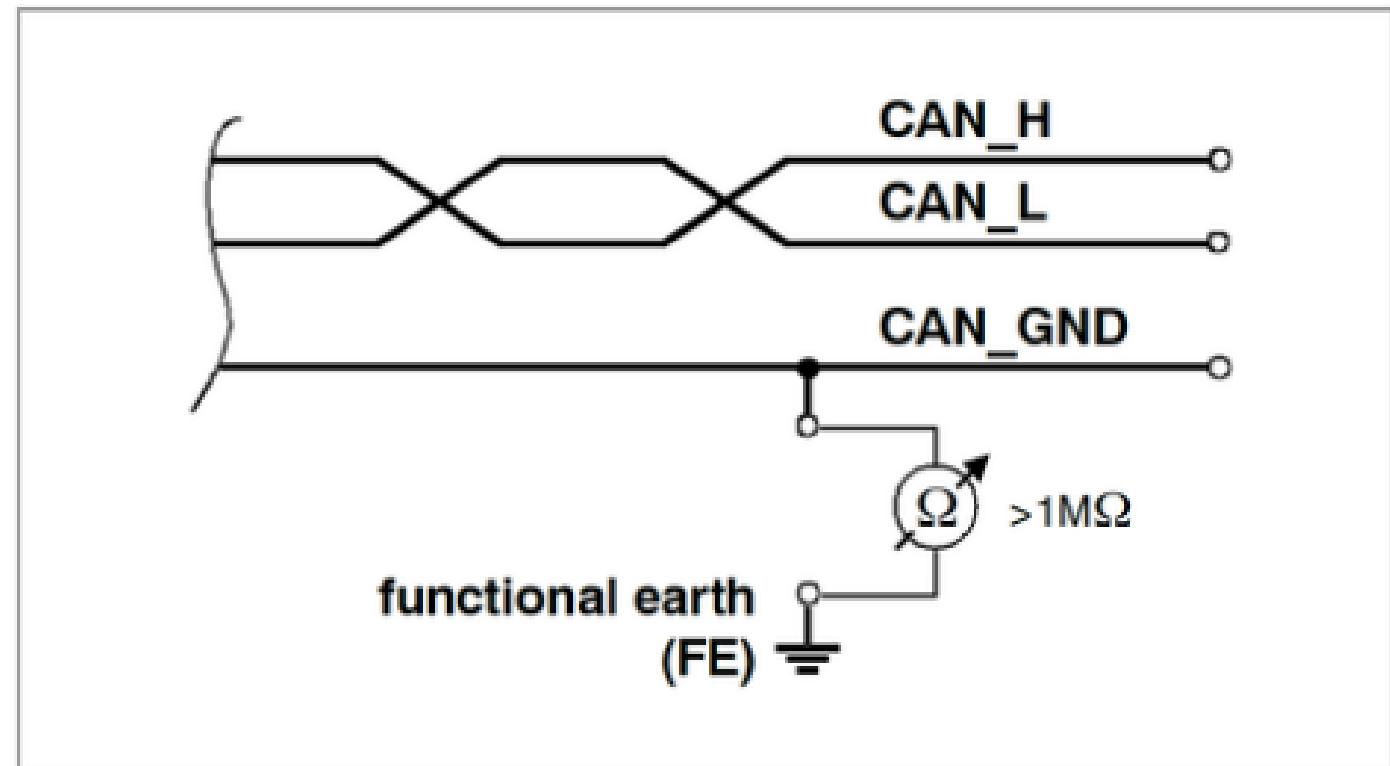
Possíveis causas de erro:

- **Se o valor determinado for inferior a 50 Ω, certifique-se de que:**
 - Não há curto-circuito entre a fiação CAN_H e CAN_L
 - Não mais do que dois resistores de terminação estão conectados
 - Os **Transceiver** dos nós individuais não estão com defeito.
- **Se o valor determinado for superior a 70 Ω, certifique-se de que:**
 - Todas as linhas CAN_H e CAN_L estão conectadas corretamente
 - Dois resistores de terminação de 120 Ω cada estão conectados à sua rede CAN (um em cada extremidade).

O aterramento correto do barramento CAN

- O CAN_GND da rede CAN deve ser conectado ao potencial de terra funcional (FE) em apenas um ponto.
Este teste indica se o CAN_GND está aterrado em um ou mais pontos.
- *Nota: Este teste só pode ser realizado com nodos CAN eletricamente isolados.*

Diagrama esquemático simplificado de medição de teste de solo



Para testar, proceda da seguinte forma:

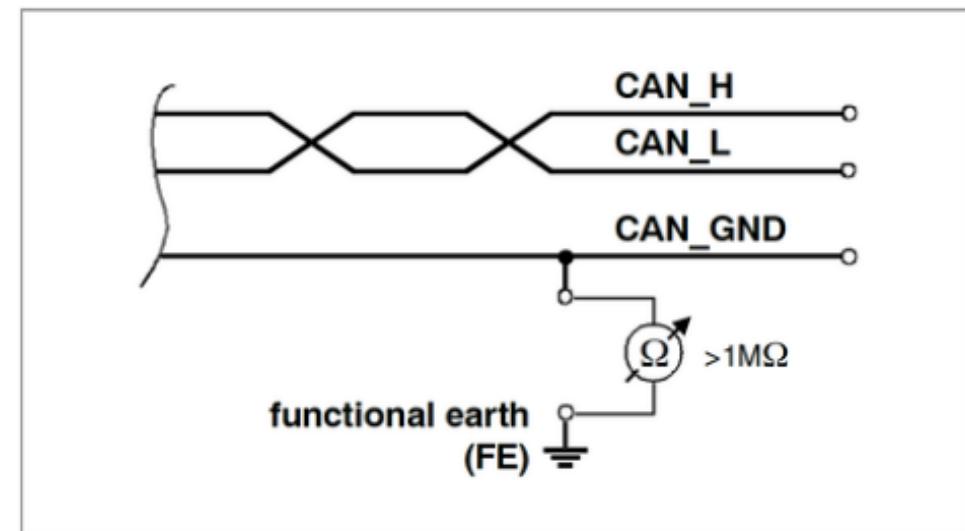
- Desconecte o CAN_GND do potencial de terra (FE).
- Meça a resistência CC entre CAN_GND e o potencial de aterramento (consulte a medição de aterramento do diagrama de circuito).

Resultado esperado:

A resistência medida deve ser maior que $1\text{ M}\Omega$. Se for menor, procure aterramento adicional dos fios CAN_GND.

Após o teste, não esqueça de reconectar CAN_GND ao potencial de terra.

Diagrama esquemático simplificado de medição de teste de solo



Curto-circuito na fiação CAN:

- Um barramento CAN ainda pode transmitir dados mesmo se CAN_GND e CAN_L estiverem em curto-circuito.
- No entanto, isso geralmente fará com que a taxa de erro aumente acentuadamente.
- Certifique-se de que não haja curto-circuito entre CAN_GND e CAN_L!

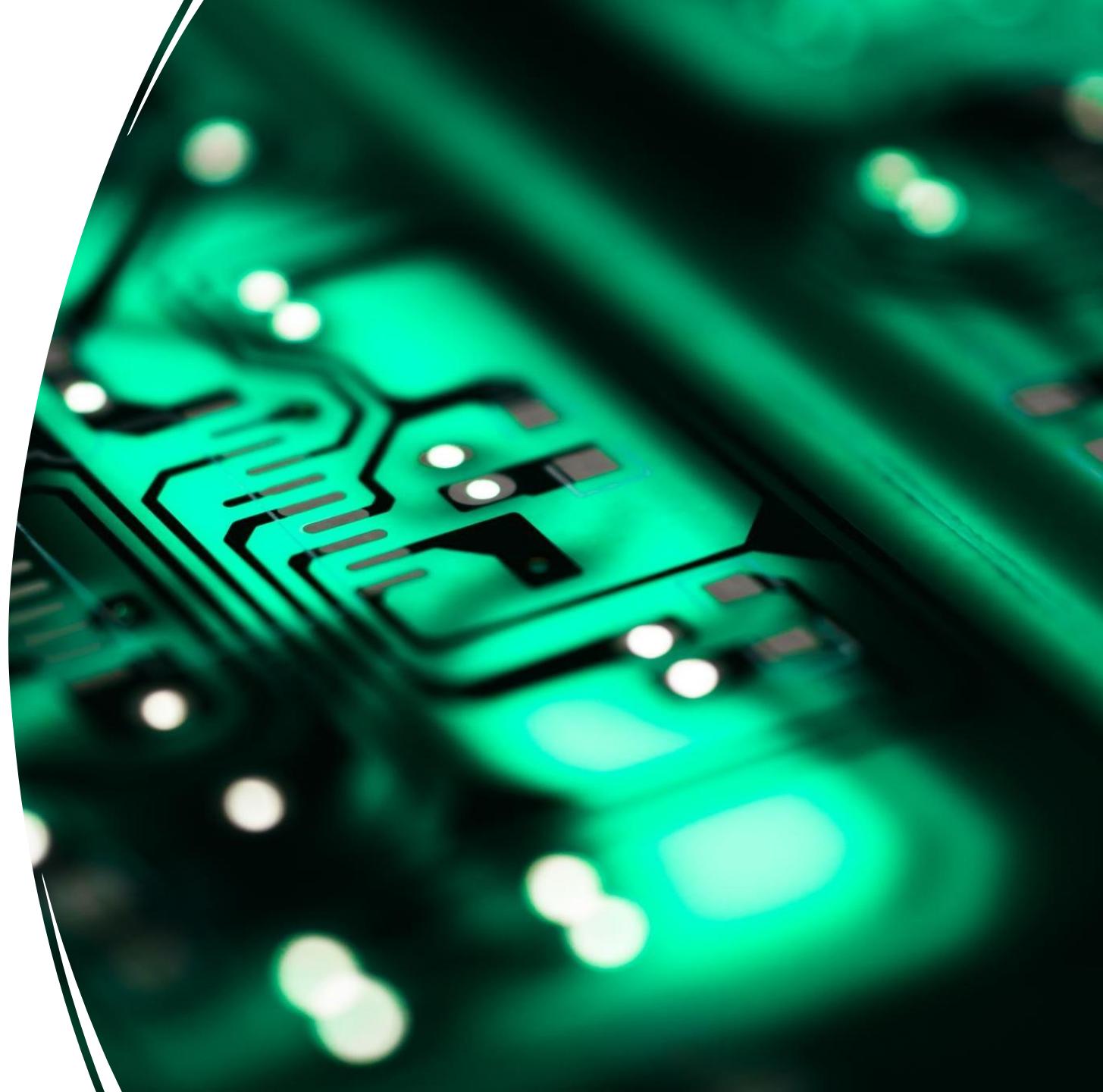
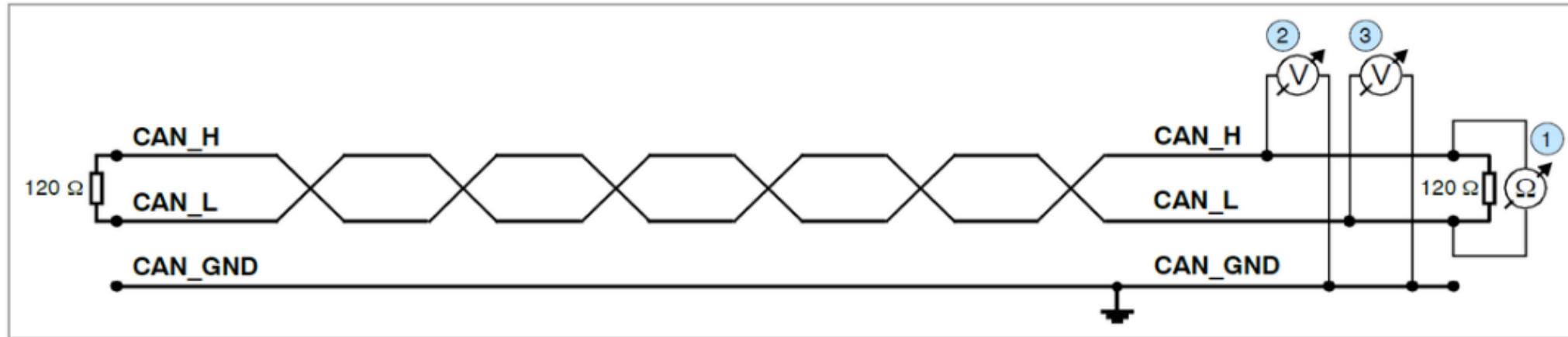


Diagrama de circuito simplificado de uma rede CAN



CANopen®

Os níveis de tensão corretos em CAN_H e CAN_L

- Cada nó possui um **Transceiver** CAN que gera sinais diferenciais nas linhas de dados.
- Quando a comunicação de rede está ociosa, as tensões CAN_H e CAN_L são cerca de 2,5 V para CAN_GND.
- **Transceiver** defeituosos podem alterar essas tensões e interromper a comunicação da rede.

Para testar Transceiver defeituosos, proceda da seguinte forma:

- Ligue todas as tensões de alimentação.
- Encerre todas as comunicações de rede.
- Meça a tensão CC entre CAN_H e GND (ponto de medição (2), consulte o diagrama de circuito de uma rede CAN).
- Meça a tensão CC entre CAN_L e GND (ponto de medição (3), consulte o diagrama de circuito de uma rede CAN).

Resultado esperado:

A tensão medida deve estar entre 2,0 V e 3,0 V.

Diagrama de circuito simplificado de uma rede CAN

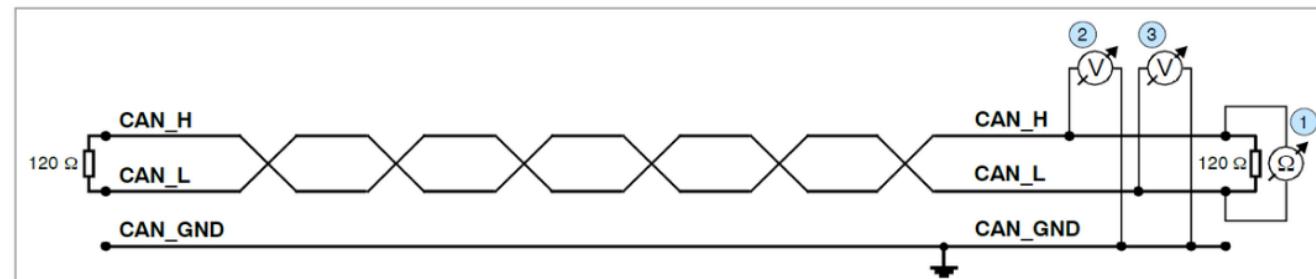
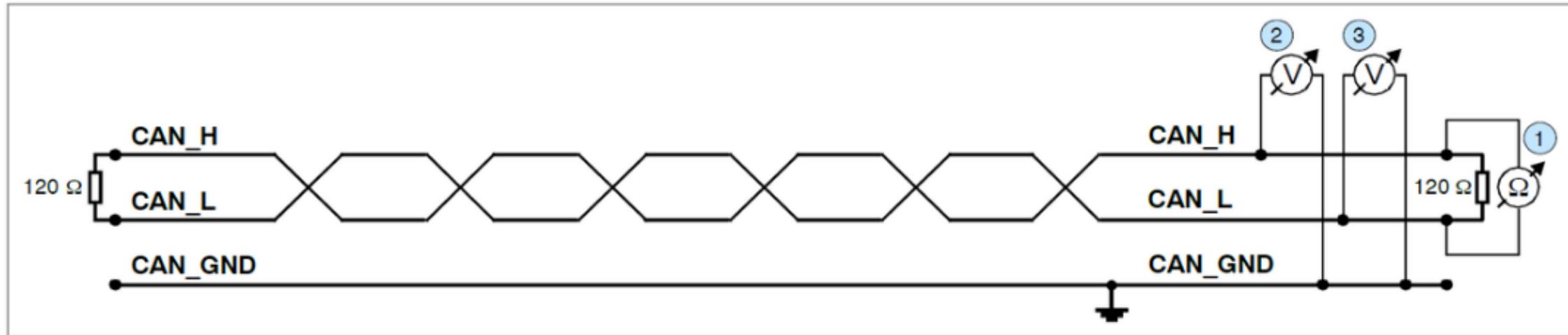


Diagrama de circuito simplificado de uma rede CAN

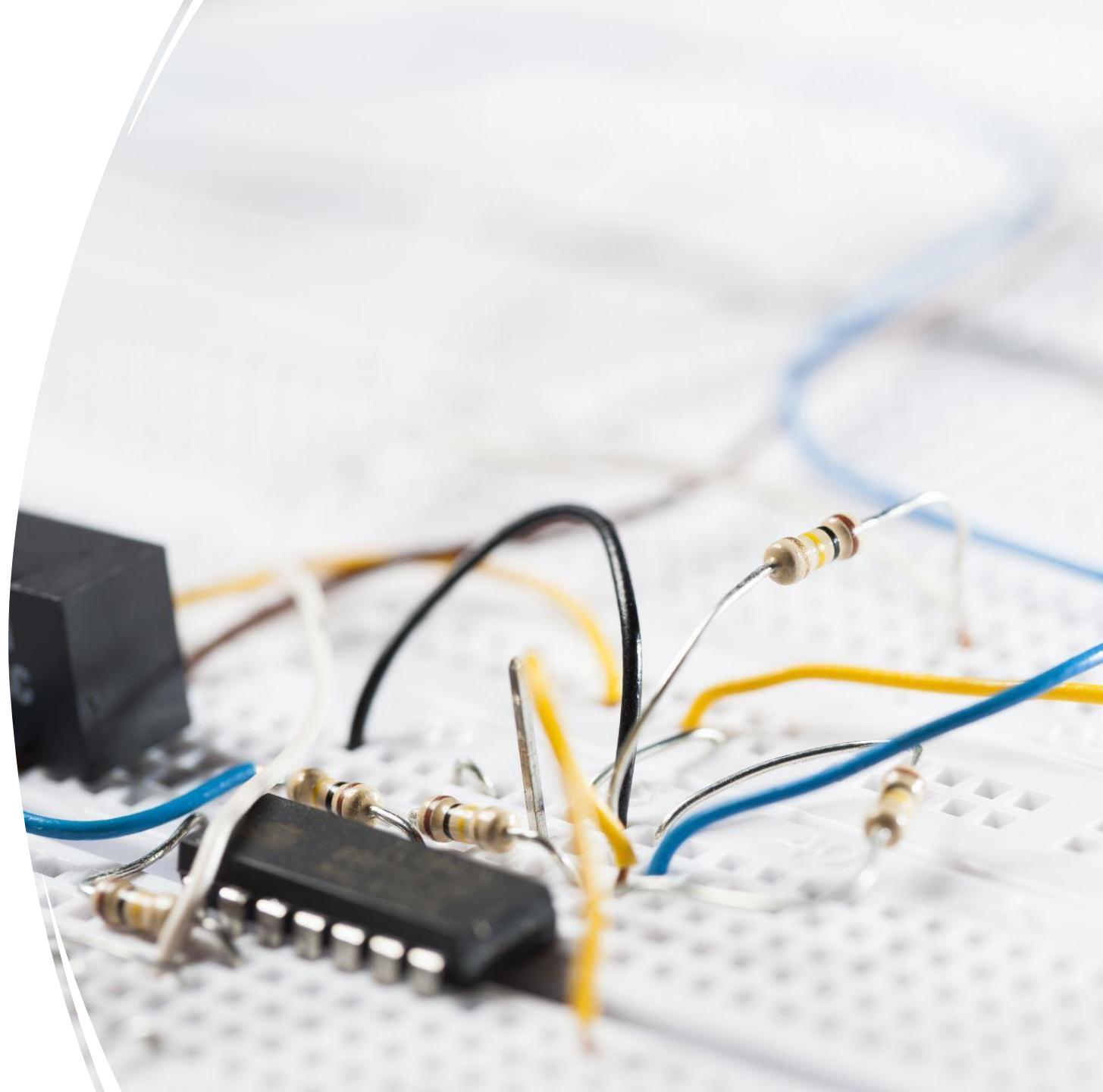


Possíveis causas de falhas:

- Se a tensão for menor que 2,0 V ou maior que 3,0 V, é possível que um ou mais nós tenham **Transceivers** defeituosos.
- Se a tensão for inferior a 2,0 V, verifique a conexão das linhas CAN_H e CAN_L.
- Para encontrar um nó com um **Transceiver** defeituoso em uma rede, verifique individualmente as resistências dos **Transceivers** CAN dos nós.

A resistência correta dos Transceivers CAN

- Transceivers CAN possuem circuitos que controlam CAN_H e CAN_L.
- A experiência mostra que danos elétricos podem aumentar a corrente de fuga nesses circuitos.



Para medir a corrente de fuga através dos circuitos CAN, use um ohmímetro e proceda da seguinte forma:

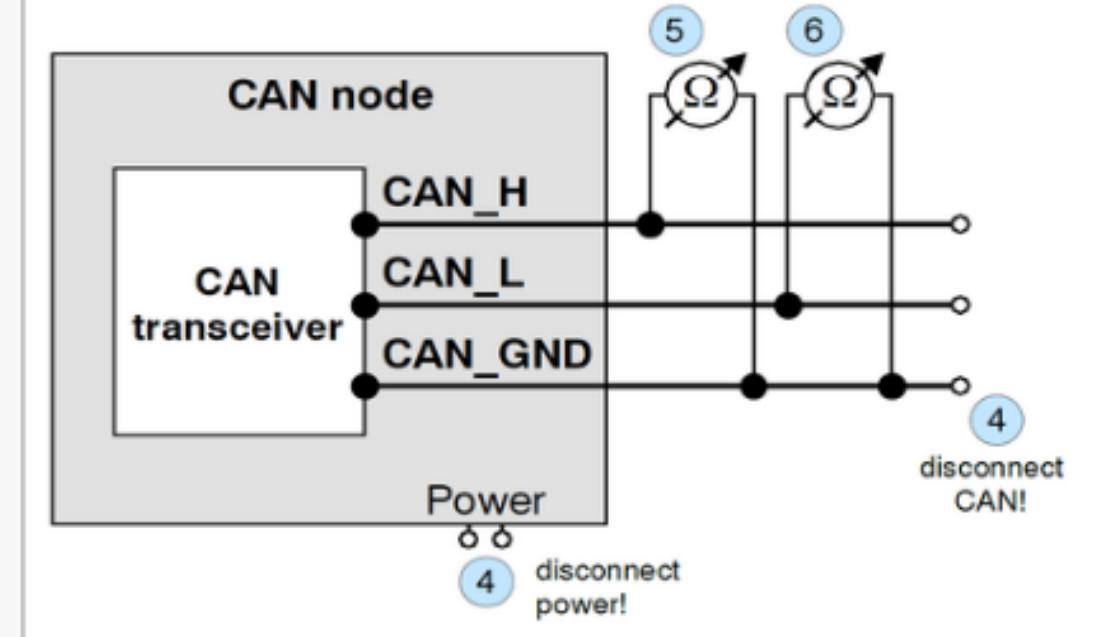
- Desligue o nó (4) e desconecte-o da rede CAN.
- Meça a resistência CC entre CAN_H e CAN_GND, ponto de medição (5).
- Meça a resistência CC entre CAN_L e CAN_GND, ponto de medição (6).

Para a posição do nó (4) e os pontos de medição (5) e (6), consulte o diagrama "Medição da resistência de entrada".

Resultado esperado:

A resistência medida deve ser superior a 10 k Ω para cada medição.

Medição da resistência de entrada do transceptor CAN



Possíveis causas de falhas:

- Se a resistência for significativamente menor, o **Transceiver** CAN pode estar com defeito.
- Outra indicação de um **Transceiver** CAN com defeito é um desvio muito alto das duas resistências de entrada medidas ($>> 200\%$).



Led indicativo de erros



- O LED **CAN ERROR** indica erros da camada física do barramento CAN, além de erros da comunicação CANopen.
 - Apagado: Sem erro.
 - Uma piscada: Atingido o estado de warning.
 - Duas piscadas: Falha no serviço de controle de erros (Node Guarding ou Heartbeat).
 - Aceso: Bus OFF.

Conectores de Cabo CANopen:

Nessa aula vamos ter uma visão geral dos diferentes conectores de cabo CANopen.

Para alcançar a intercambiabilidade dos dispositivos em relação à interface CAN, é necessário o uso de conexões físicas compatíveis. No caso de conectores, o plugue e o soquete precisam corresponder, incluindo a atribuição de pinos. Uma das primeiras recomendações lançadas pela CiA foi a atribuição de pinos para conectores DSUB de 9 pinos.

Ao longo dos anos, a CiA recomendou a atribuição de pinos para muitos tipos de conectores.

Essas recomendações foram publicadas no documento CiA 303-1, que faz parte da série clássica do CANopen.

A fim de generalizar essas recomendações, a CiA transferiu as recomendações de atribuição de pinos para o documento CiA106, que não está mais relacionado apenas ao CANopen clássico.

Agora, as recomendações de atribuição de pinos do conector fornecidas neste documento são válidas para as interfaces Classical CAN, CAN FD e CAN XL.

Eles também são referenciados por especificação de perfil para CANopen clássico e CANopen FD. Eles também são adequados para outras abordagens de camada superior baseadas em CAN, que não recomendam ou especificam atribuições de pinos contraditórias.



CANopen®

Conector SUB-D de 9 pinos:

- Conector DB9 segundo a **CiA106** e o padrão **DIN 41652**:

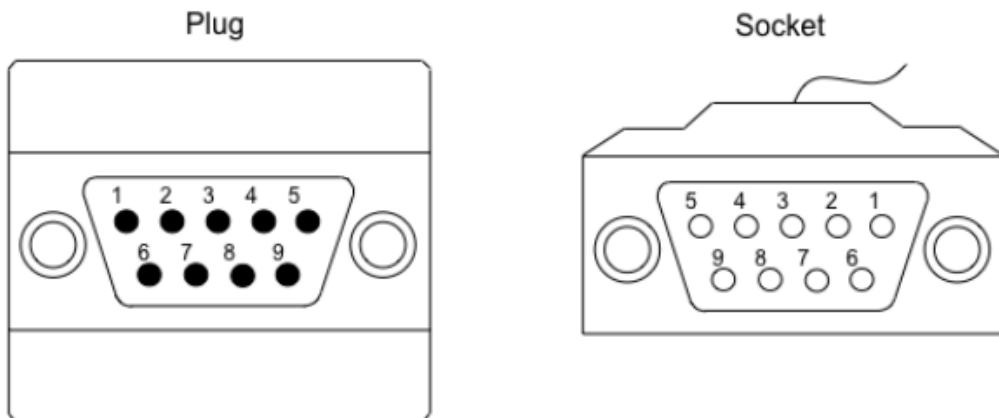


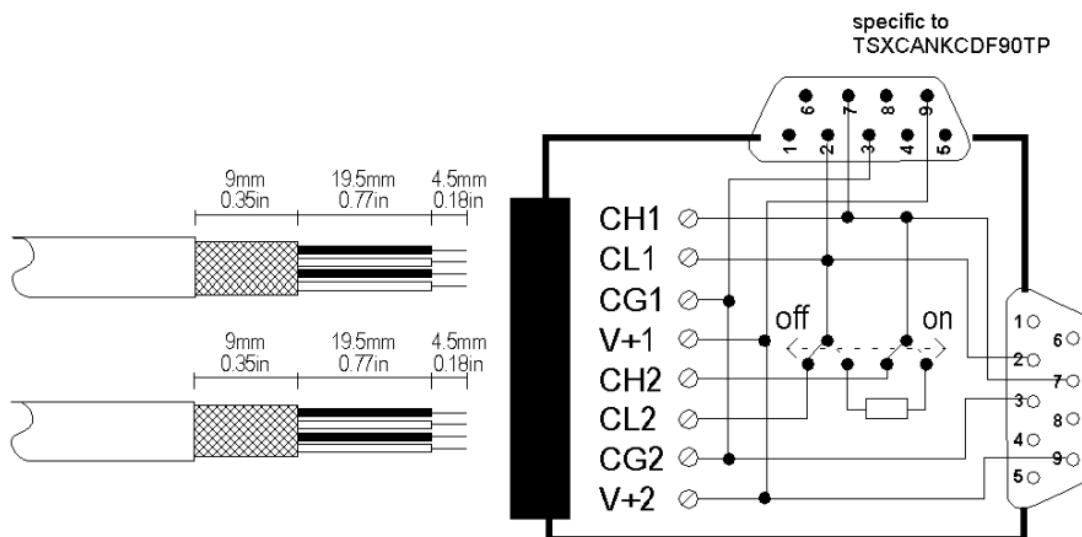
Figure 1 – D-SUB 9-pin connector pin-numbering

Table 2 – D-SUB 9-pin connector pin-assignment

Pin	Notation	Description
1	-	Reserved
2	CAN_L	CAN_L network line (dominant low)
3	CAN_GND	CAN ground
4	-	Reserved
5	(CAN_SHLD)	Optional CAN shield
6	(GND)	Optional ground
7	CAN_H	CAN_H network line (dominant high)
8	-	Reserved
9	(CAN_V+)	Optional CAN external positive supply

Conector SUB-D de 9 pinos:

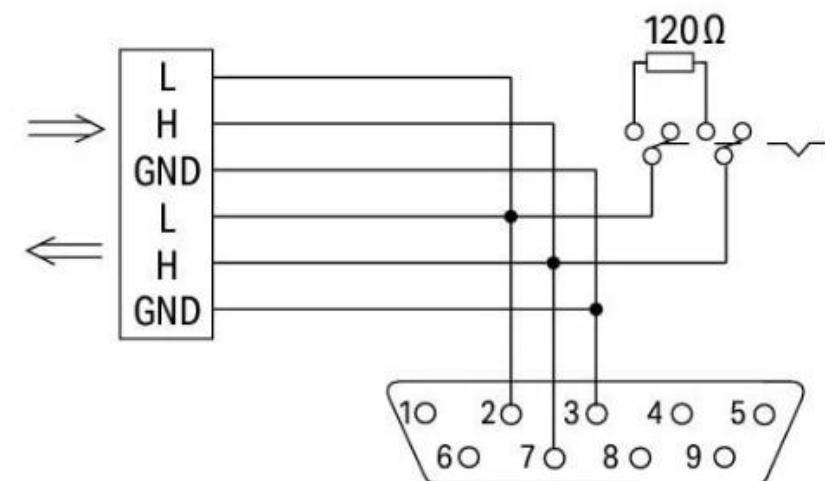
- Conector DB9 segundo o padrão **ISO 11898-2**:

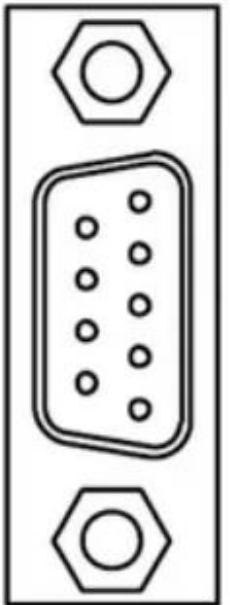


Signal	Terminal Block 1, Incoming Cable	Terminal Block 2, Outgoing Cable	Wire Color
CAN_H	CH1	CH2	white
CAN_L	CL1	CL2	blue
CAN_GND	CG1	CG2	black
CAN_V+	V+1	V+2	red

Conecotor SUB-D de 9 pinos:

CANopen®



DB9P plug	Pin	signal	description
 6 7 8 9	1	N.C.	Unused
	2	CAN_L	CAN_L Signal line
	3	CAN_GND	Reference ground
	4	N.C.	Unused
	5	CAN_SHIELD	Shielded line
	6	CAN_GND	Reference ground
	7	CAN_H	CAN_H Signal line
	8	N.C.	Unused
	9	N.C.	Unused



Conecotor SUB-D de 9 pinos:

 canopen®

Conector RJ45:

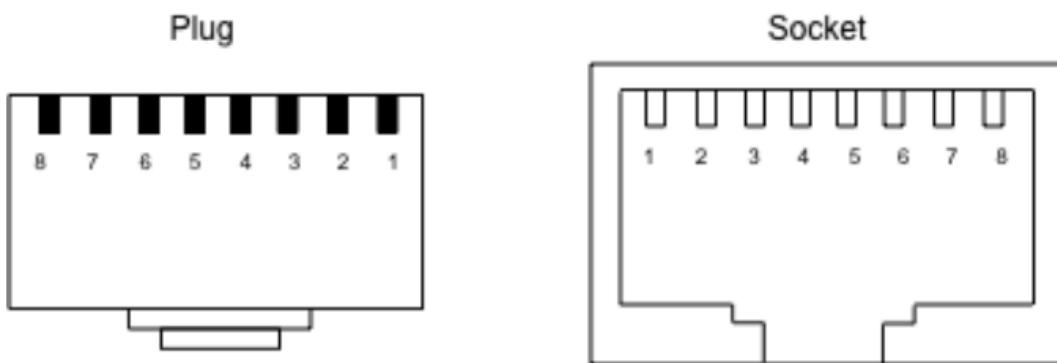


Figure 3 – RJ45 connector pin-numbering

Pin	Notation	Description
1	CAN_H	CAN_H network line (dominant high)
2	CAN_L	CAN_L network line (dominant low)
3	CAN_GND	Ground or 0 V or V- (optional CAN external negative supply)
4	-	Reserved
5	-	Reserved
6	(CAN_SHLD)	Optional CAN shield
7	(GND)	Optional ground
8	(CAN_V+)	Optional CAN external positive supply

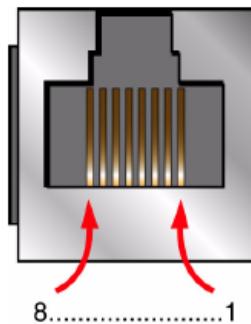
Conector RJ45:

- Exemplo de conexão com inversor de frequencia Altivar 312:



CANopen®

Pino	CANopen®
1	CAN_H
2	CAN_L
3	CAN_GND
4	NÃO UTILIZADO
5	NÃO UTILIZADO
6	NÃO UTILIZADO
7	NÃO UTILIZADO
8	NÃO UTILIZADO



Conector de estilo aberto

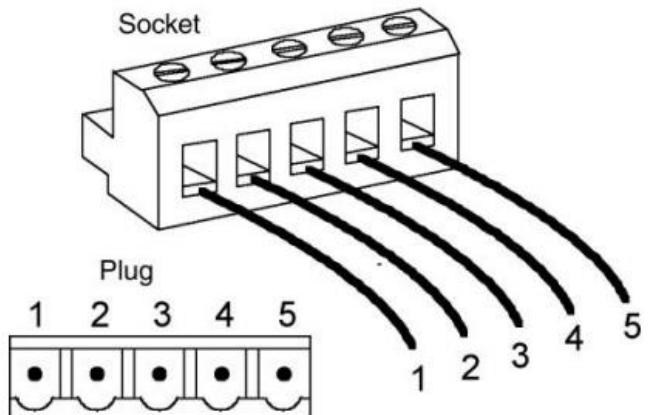
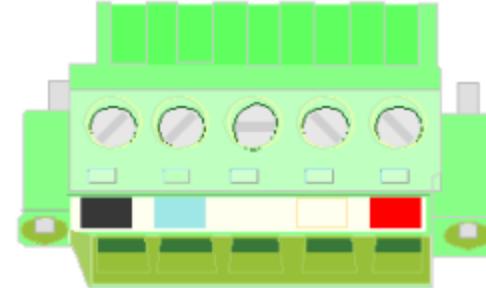


Figure 5 – Open-style connector pin-numbering

Pin	Signal	Connector Color Marking	Wire Color	Termination Resistor
1	CAN_GND	black	black	place a 120 Ω 0.25 W 5%
2	CAN_L	blue	blue	resistor between CAN_H
3	CAN_Shield	bare	tinned copper drain wire	and CAN_L if this is the
4	CAN_H	white	white	physical end of the trunk
5	CAN_V+	red	blue	cable

CANopen®

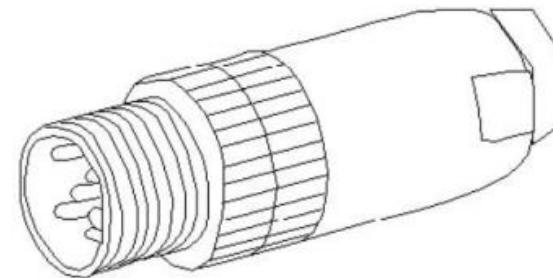
Conector de Cabo IP67 M12

- Como esses conectores permitem conectar apenas 1 cabo, o encadeamento do cabo é realizado pelo dispositivo.
- Ele fornece portas específicas para o cabo de entrada e saída.

CANopen®

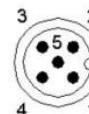


A figura abaixo mostra um conector de cabo IP 67 M12:



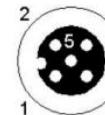
Conector BUS IN

A figura abaixo mostra o conector BUS IN macho M12 de 5 pinos:



Conector BUS OUT

A figura abaixo mostra o conector BUS OUT fêmea M12 de 5 pinos:



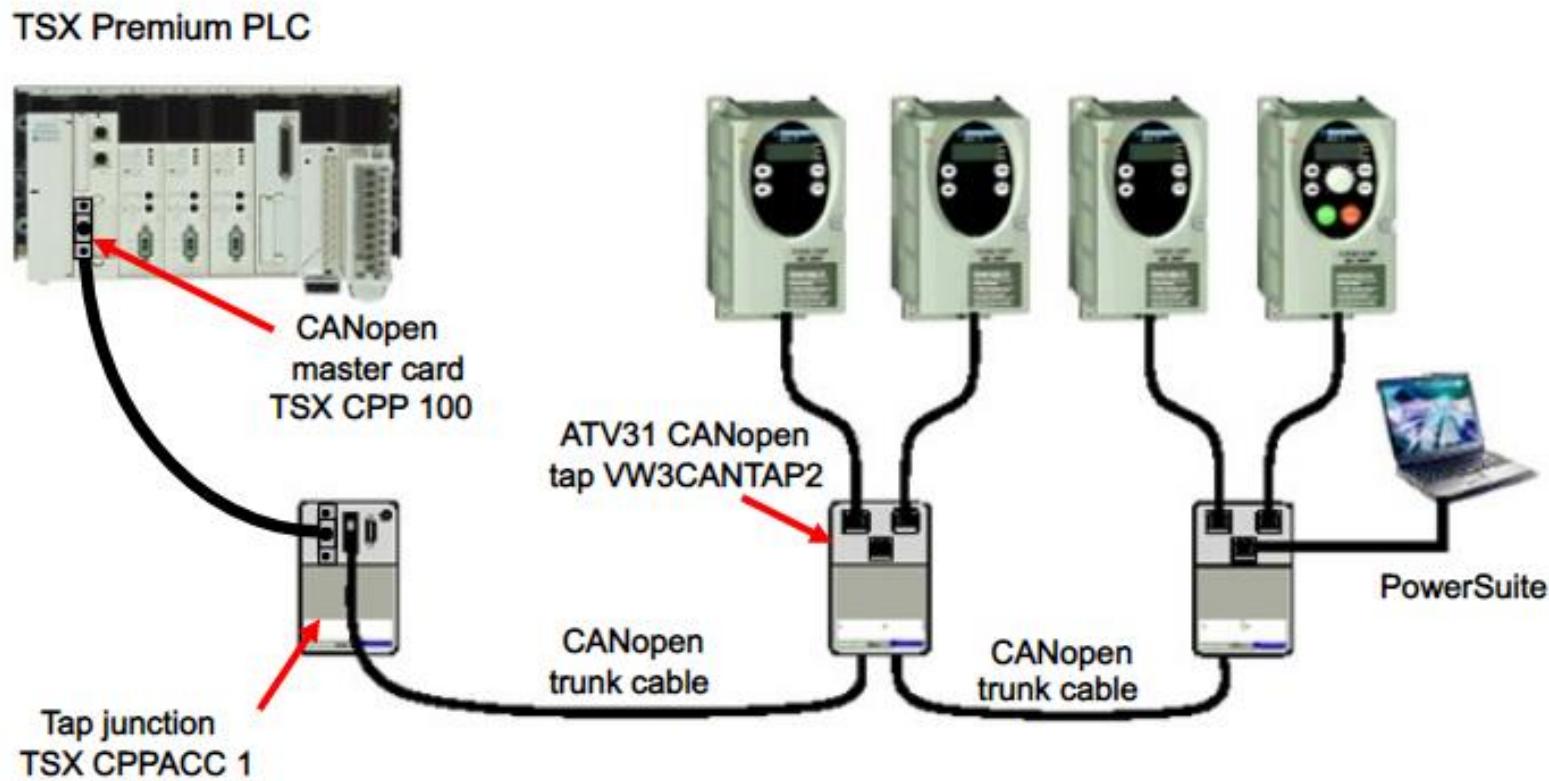
CANopen TAPs

Nessa aula vamos ter uma visão geral dos diferentes TAPs CANopen.

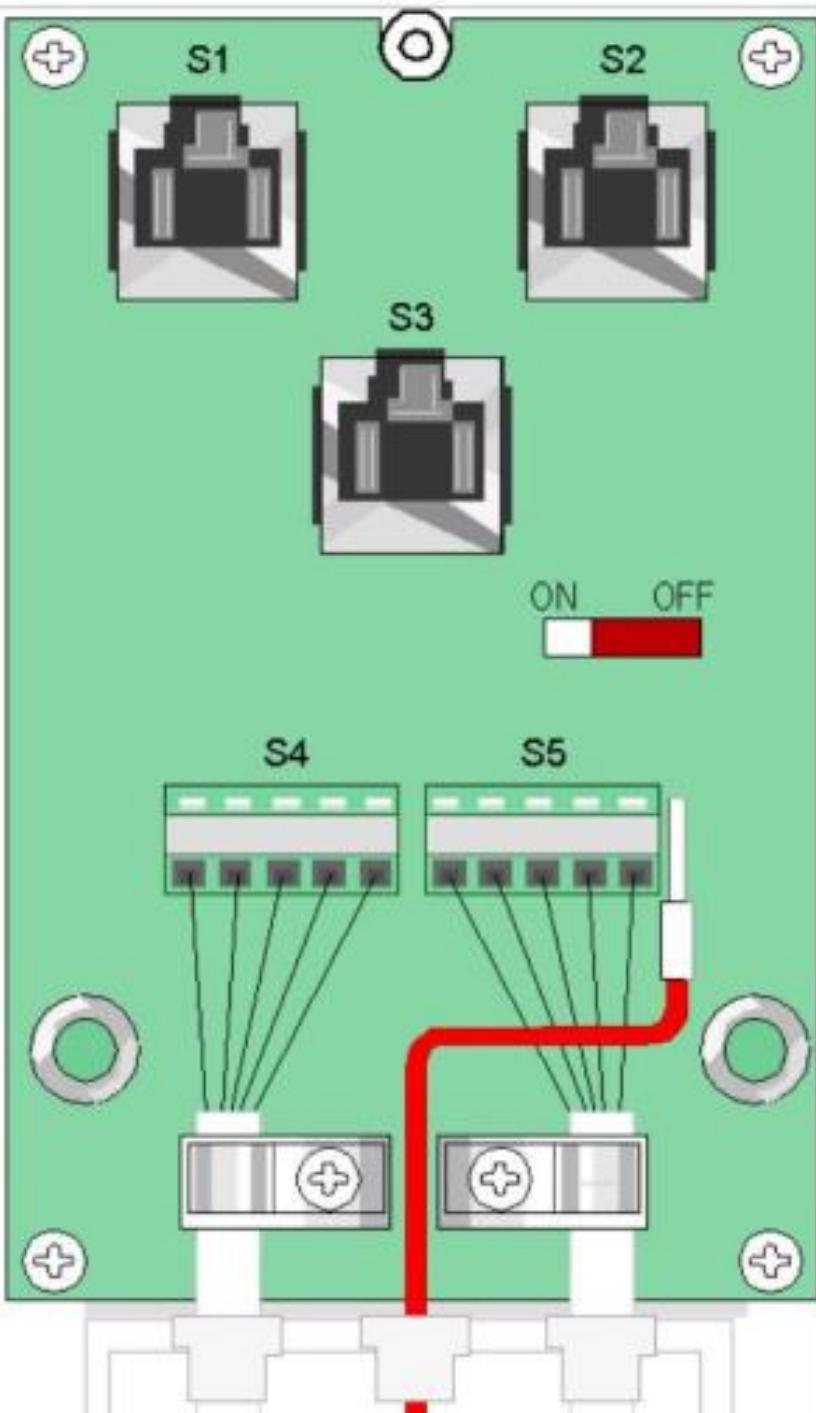


CANopen®

Exemplo de uma arquitetura CANopen com TAPs



CANopen®



Exemplo do TAP VW3CANTAP2

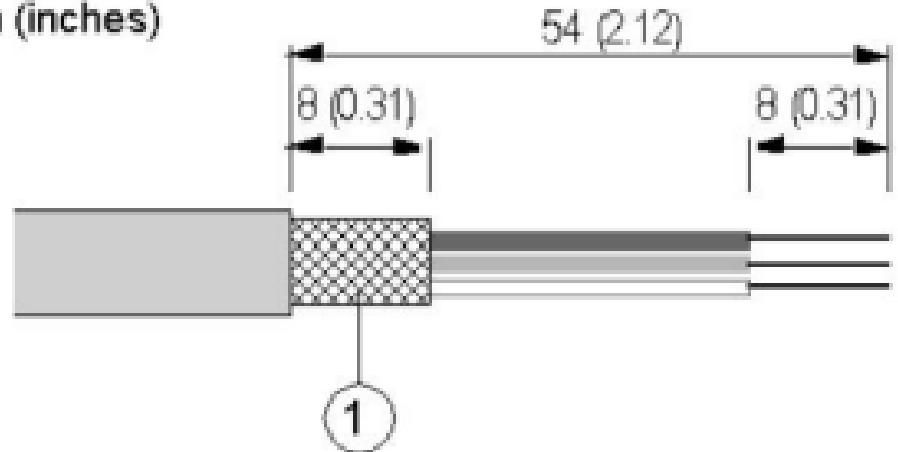
- O VW3CANTAP2 permite a conexão de 2 dispositivos como ATV31, ATV71, Lexium05 ramificando o cabo drop para os 2 conectores S1 e S2.
- Além disso, permite a conexão de uma ferramenta baseada em Modbus no conector S3.
- Os sinais CAN (CAN_H, CAN_L e CAN_GND) dos cabos de entrada e saída e os 2 conectores RJ45 (S1, S2) são interligados dentro da caixa.
- Da mesma forma, a blindagem do conector é conectada à blindagem do cabo.
- A conexão ao terminal PE (terra) deve utilizar o cabo verde-amarelo.
- Um interruptor de terminação de linha é fornecido para comutar um resistor de terminação embutido.

CANopen®

Exemplo do TAP VW3CANTAP2

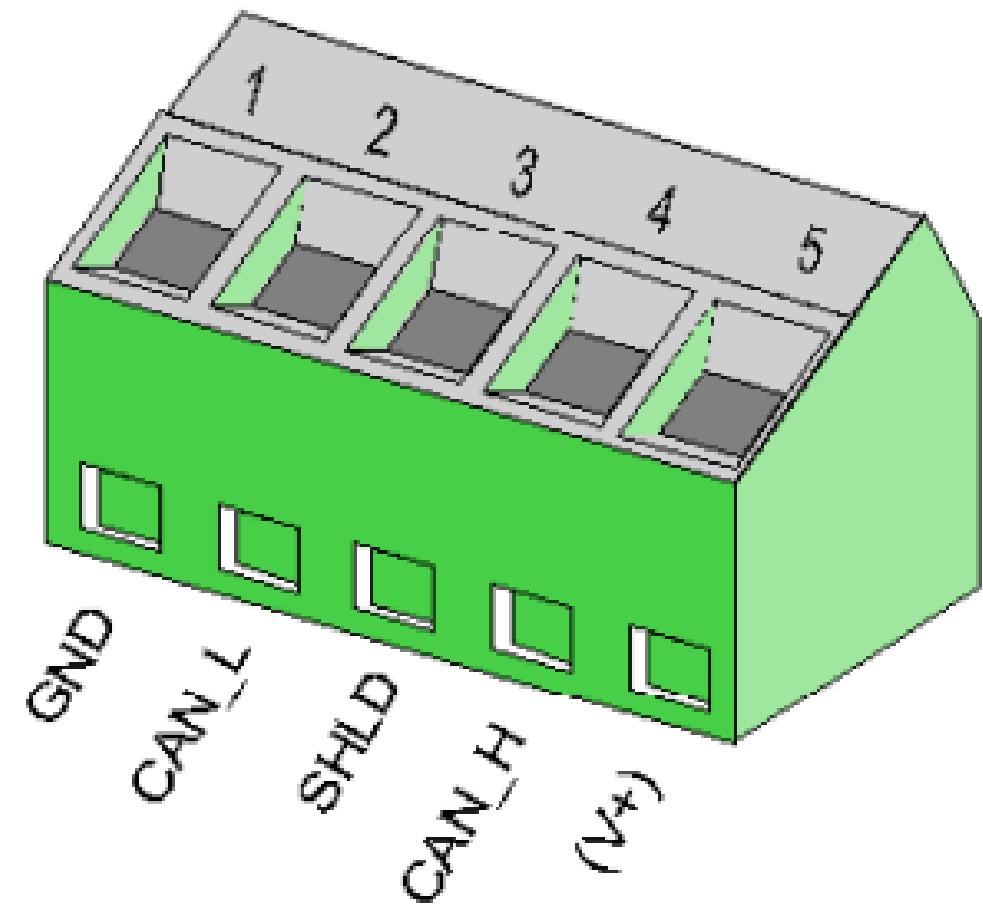
Modelo de preparação de cabos

mm (inches)



1 blindagem

Pin	Signal	Wire Color	Description
1	GND	black	ground
2	CAN_L	blue	CAN_L bus line
3	SHLD	(bare cableshield)	optional shield
4	CAN_H	white	CAN_H bus line
5	(V+)	red	optional positive power supply

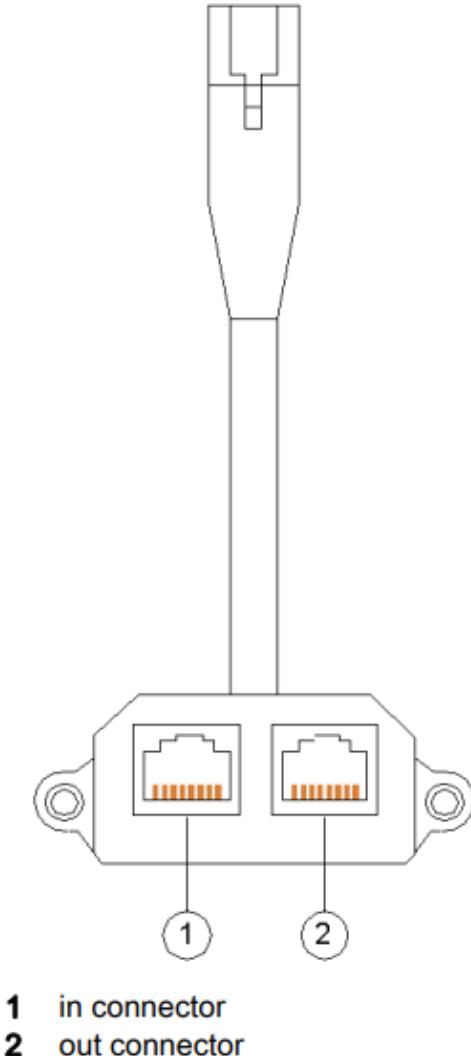


CANopen®

CANopen - Conector Daisy Chain

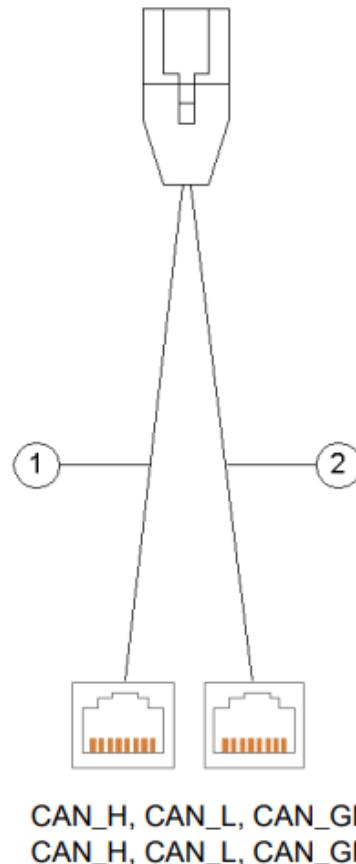
- Além dos Taps, nós também podemos utilizar o que chamamos de conectores Daisy Chain, que nos fornecem uma junção em Y para conectores RJ45, permitindo assim o encadeamento do cabo CAN.

CANopen®



CANopen - Conector Daisy Chain

- Mesmo que o comprimento visível do cabo deste conector seja de 0,30 m (0,98 pés), você deve considerar um comprimento absoluto de 0,60 m (1,97 pés) ao calcular o comprimento máximo do cabo devido à sua fiação elétrica.

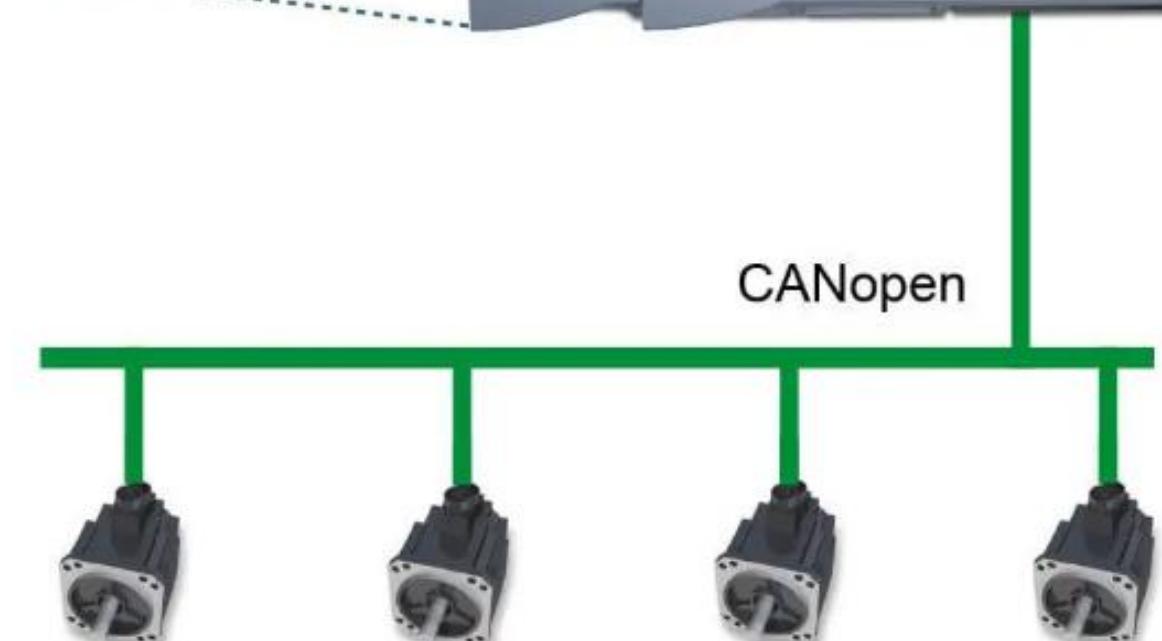
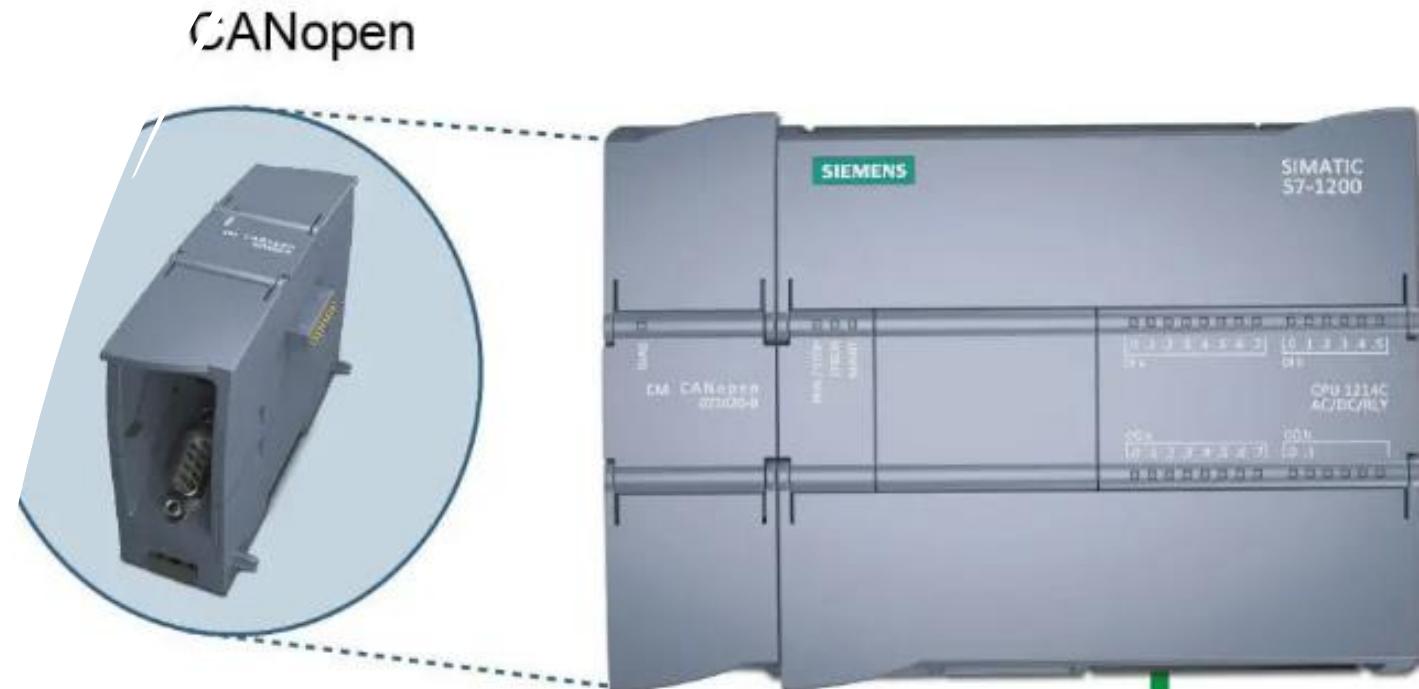


CANopen®

Dispositivos CANopen

- Os dispositivos que normalmente podem ser conectados a um barramento CANopen são divididos em grupos de acordo com suas funções:
 - ✓ Dispositivos de comando de movimento.
 - ✓ Dispositivos de entrada/saída.
 - ✓ Outros dispositivos.

CANopen®



Slave devices on the CANopen network

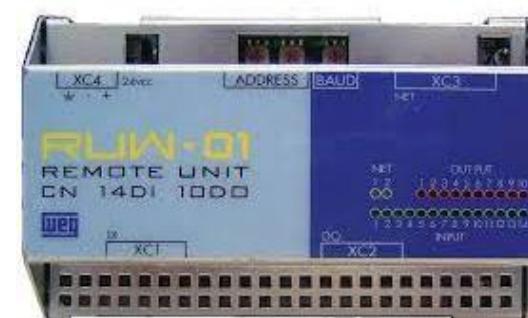
Dispositivos de comando de movimento

- Dispositivos de comando de movimento permitem que você controle motores.
- ✓ Inversor de frequência Altivar, Schneider.
- ✓ Servo drive Lexium, Schneider.
- ✓ SERVO DRIVE ASD-A2-1521-M, DELTA.
- ✓ CFW300 + CCAN, WEG.
- ✓ *MDXK61GN3 - Servo Motor integrado, MDX.*



Dispositivos de Input/Output

- Os módulos de entrada/saída funcionam como módulos remotos. Exemplos desses dispositivos:
 - ✓ Unidade Remota RUW-01, WEG.
 - ✓ MÓDULO OTB MODICON, Schneider.
 - ✓ SIMATIC ET200S + 1SI CANopen, Siemens.
 - ✓ 750-337 - Acoplador de Rede CANopen, WAGO



canopen®

Outros dispositivos

- Além dos dispositivos de comando de movimento e IOs podemos ter dispositivos diversos, como por exemplo:
 - ✓ Transmissor de pressão com interface CANopen, WIKA.
 - ✓ Sensor de Ângulo Indutivo com Interface CANopen, TURK.
 - ✓ CANopen sensor de vibração, whitebeam.
 - ✓ Encoders com interface CANopen, Posital



CANopen



CANopen®

Controladores

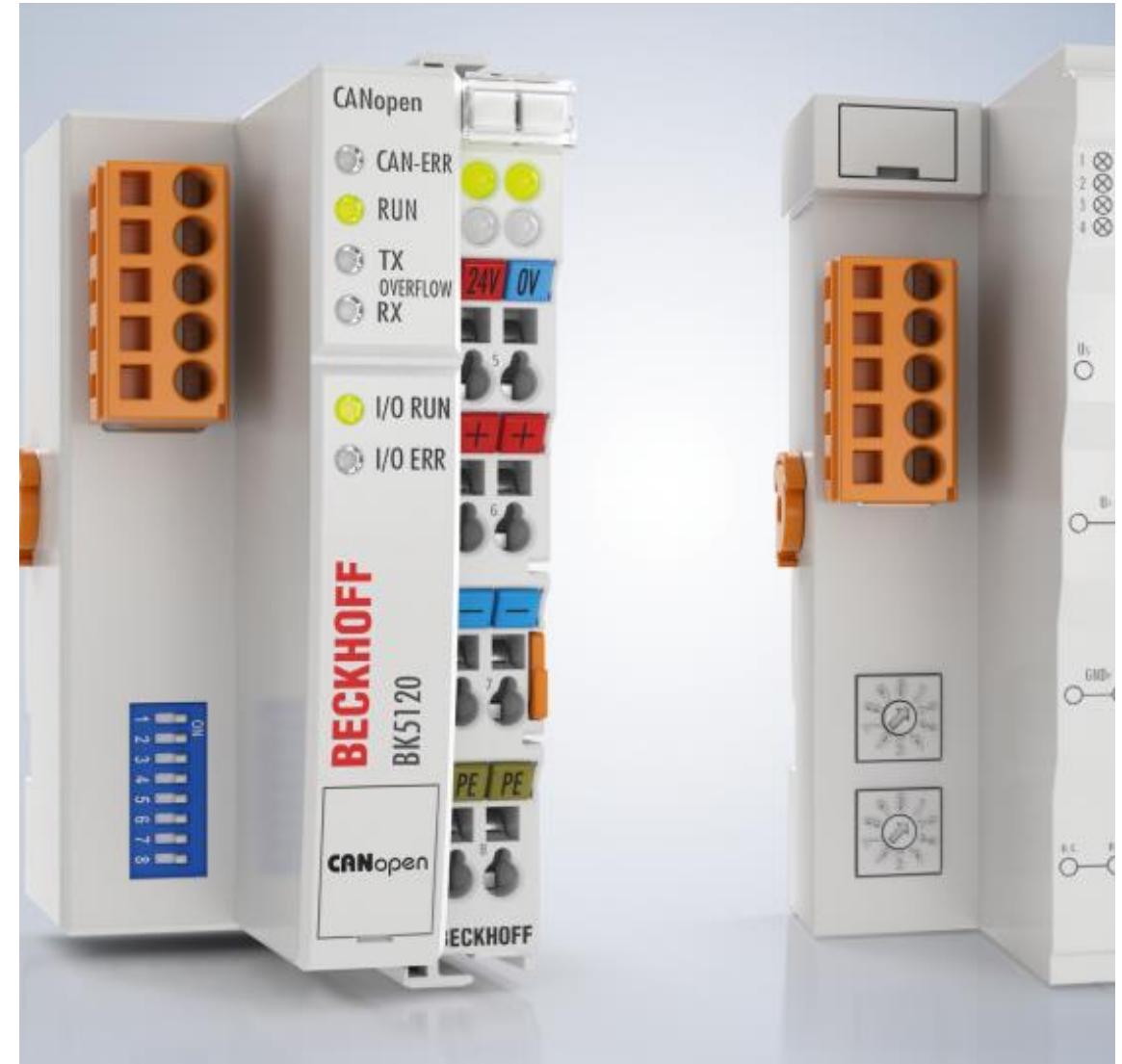
- Afim de gerenciar a rede temos os controladores, que podem ser por exemplo:
 - ✓ PAC MODICON M340, Schneider.
 - ✓ CLP MODICON M258, Schneider.
- DVPCOPM-SL Módulo de comunicação CANopen mestre para CLP DVP-SV, Delta.
- Módulo mestre/escravo CM CANopen CANopen para Simatic® S7-1200 CLP, Siemens.



canopen®

Endereçamento dos dispositivos CANopen

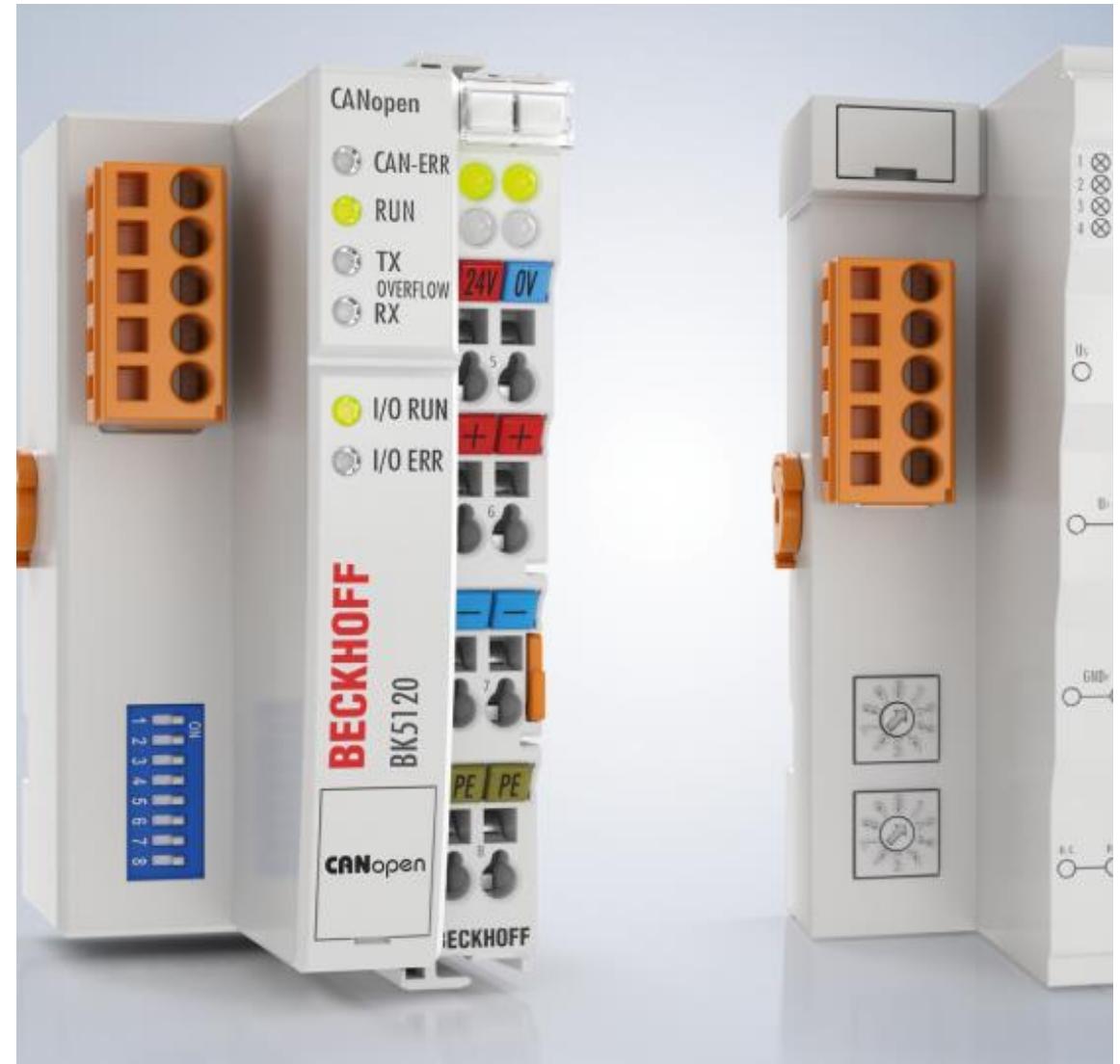
- Até aqui aprendemos sobre endereçamento de dispositivos na rede CANopen:
 - ✓ Cada dispositivo deve ter um endereço único.
 - ✓ A faixa de endereçamento vai de 1 a 127.
 - ✓ A quantidade de dispositivos vai depender do transceiver utilizado no controlador da rede.
 - ✓ Normalmente temos redes com até 63 devices podendo estender a capacidade utilizando um repetidor.



CANopen®

Tipos de endereçamento

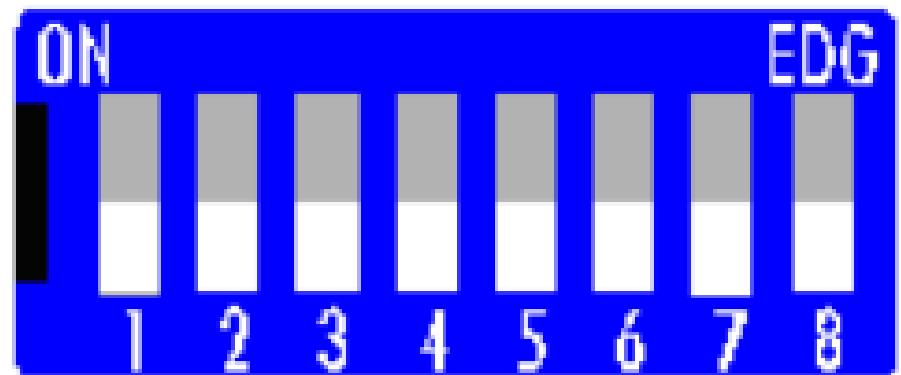
- Normalmente temos três maneira de atribuir um endereço a um dispositivo na rede CANopen:
 - ✓ Utilizando chaves dipSwitch;
 - ✓ Utilizando chaves rotativas;
 - ✓ Ou via software de configuração.



CANopen®

Endereçamento via DipSwitch

- Para exemplificar o funcionamento do endereçamento de dispositivos CANopen via DipSwitch vou utilizar o CANopen Bus Coupler do fabricante Beckhoff Automation.
- O ID do nó é definido com as chaves DIP 1 - 6.
- A chave 1 é o bit 2^0 de valor mais baixo e a chave 6 o bit 2^6 de valor mais alto.
- O bit é definido quando a chave está LIGADA.
- O ID do nó pode ser definido na faixa de 0 a 63 (por exemplo, ID do nó = 14 -> switch 2, 3, 4 para ON), mas o 0 não é permitido.
- Cada endereço só pode ocorrer uma vez na rede. Para alterar um endereço, o acoplador de barramento deve ser desligado.
- A mudança de endereço está ativa assim que o dispositivo é ligado.

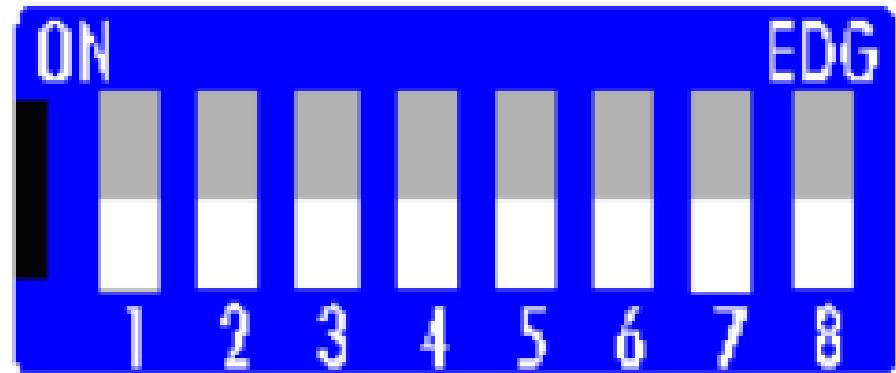


CANopen®

Endereçamento via DipSwitch

- As chaves 7 e 8 são responsáveis por definir a velocidade do dispositivo.

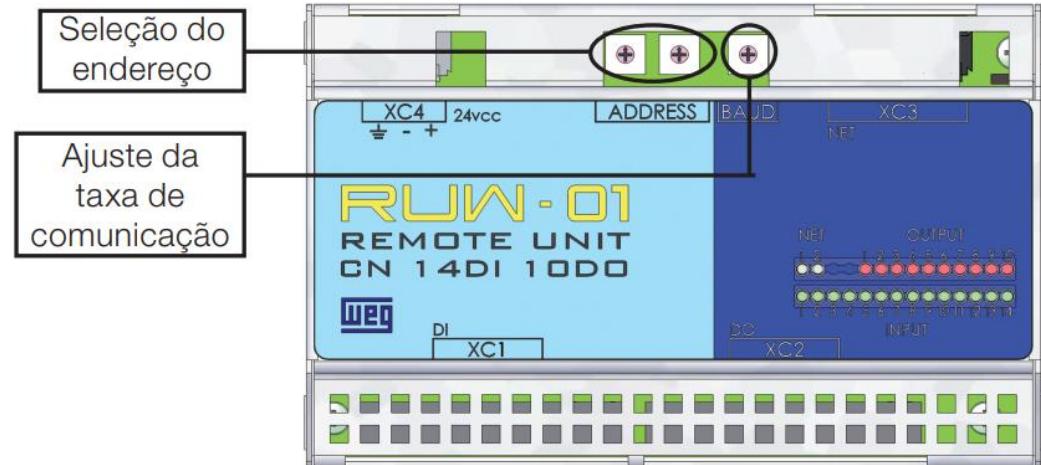
Baud rate	DIP7	DIP8
Auto	0	0
500 kbit/s	1	0
250 kbit/s	0	1
125 kbit/s	1	1



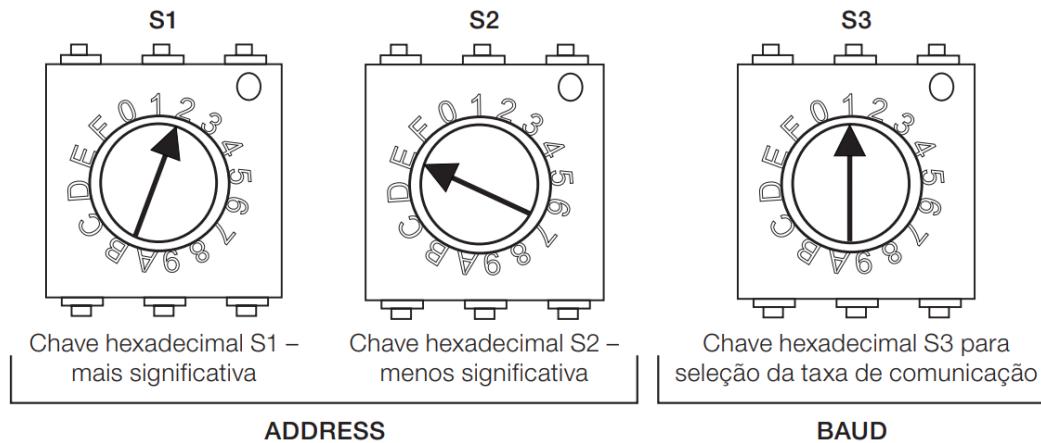
CANopen®

Endereçamento via chave rotativa

- Para exemplificar o endereçamento de dispositivos CANopen com chaves rotativas vou utilizar a remota RUW-01 do fabricante WEG.
- O endereço da unidade remota CANopen é configurado através de duas chaves hexadecimais S1 e S2.
- Endereços válidos: 1 a 127 (01h a 7Fh).



Chaves hexadecimal para configuração da RUW-01



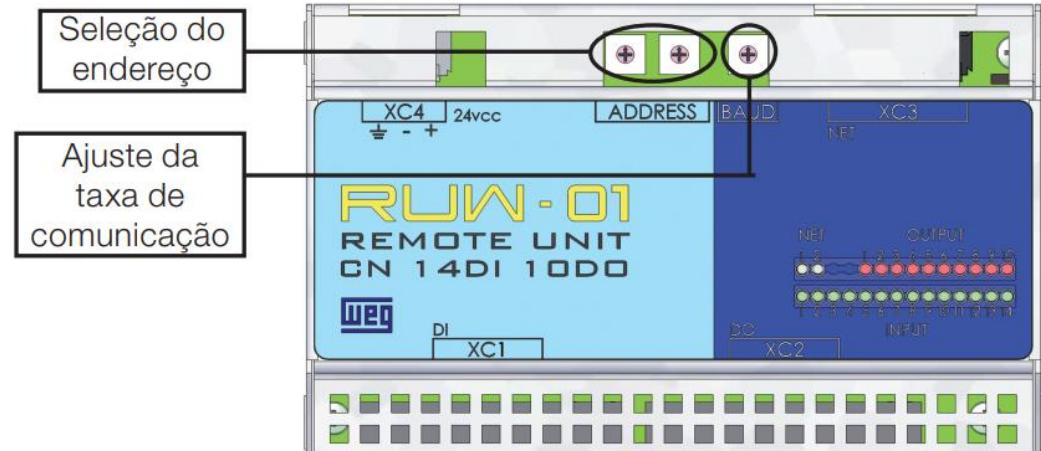
Chaves hexadecimal para configuração da RUW-01

CANopen®

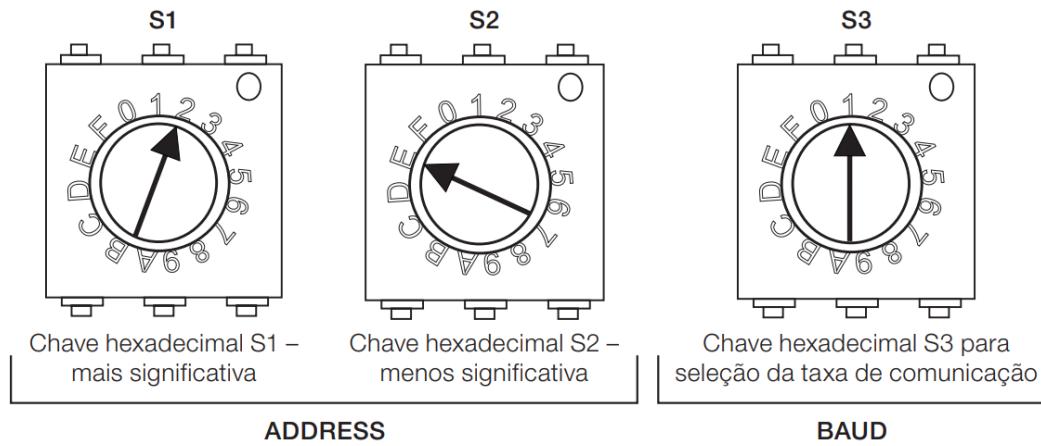
Endereçamento via chave rotativa

- A RUW-01 apresenta a chave hexadecimal S3 para a seleção da taxa de comunicação.

- ✓ 0 = 1 Mbit/s.
- ✓ 1 = 800 Kbit/s.
- ✓ 2 = 500 Kbit/s.
- ✓ 3 = 250 Kbit/s.
- ✓ 4 = 125 Kbit/s.
- ✓ 5 = 100 Kbit/s.
- ✓ 6 = 50 Kbit/s.
- ✓ 7 = 20 Kbit/s.
- ✓ 8 = 10 Kbit/s.



Chaves hexadecimais para configuração da RUW-01

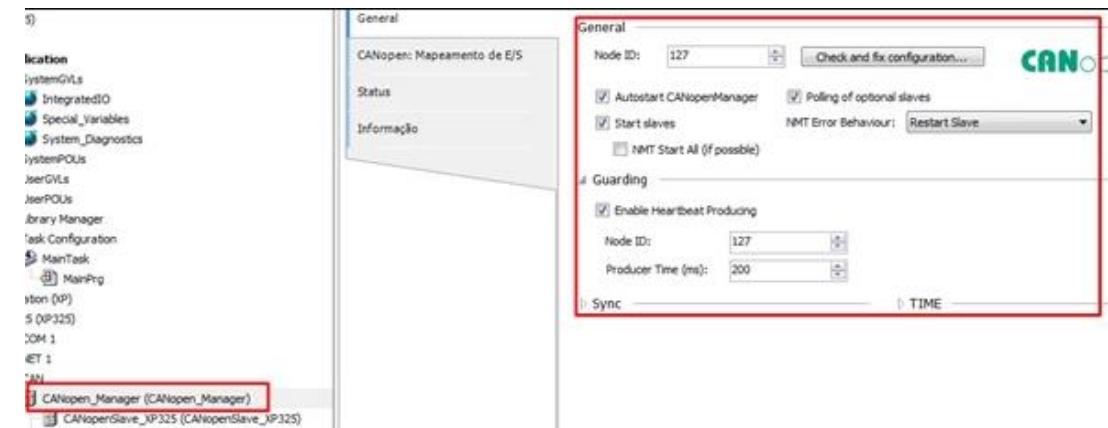
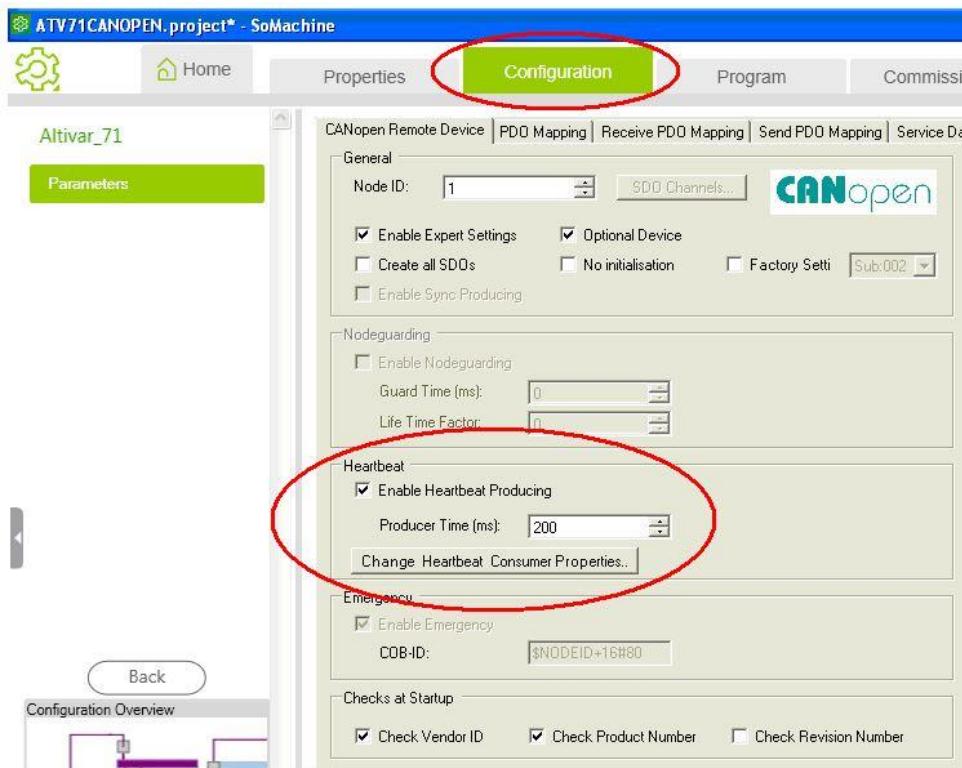


Chaves hexadecimais para configuração da RUW-01

CANopen®

Endereçamento via Software

- Quando o dispositivo não possui métodos físicos para endereçamento o mesmo deve ser configurado via software de programação, que normalmente é a plataforma de engenharia que se programa o controlador da rede.



CANopen®

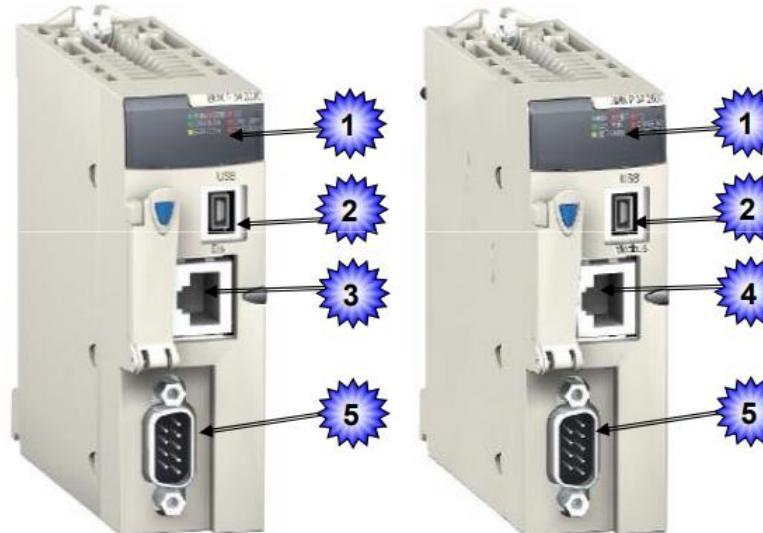
Aula prática

Conhecendo o hardware

Inicialmente vamos utilizar um PAC da Schneider da linha Modicon M340 com uma CPU BMX P34 2010 como mestre da nossa rede CANopen.

Processors with an integrated CANopen

■ Modules BMX P34 2010 / 2030



Value	Designation
1	Display
2	USB port
3	Eth port
4	Serial port
5	CANopen port

CANopen®

BMX P34 2010

A conexão é feita em um conector padrão DB9 macho que já é nativo do processador.



PIN	SIGNAL
1	
2	CAN-L
3	CAN-GND
4	
5	
6	GND
7	CAN-H
8	
9	

canopen®

Características do canal integrado

Character mode													
Baud rate	20, 50, 125, 250, 500, 1000 kbds (according with the lenght)												
Equipments	63 Equipments max												
Bus length	0 to 2500 meters												
Services	NMT, PDO, SDO, SYNC, EMCY												
Speed / Length <i>(the speed is directly linked with the bus's length)</i>	<table border="1"><tbody><tr><td>1 Mbits</td><td>20 m</td></tr><tr><td>500 Kbits</td><td>100 m</td></tr><tr><td>250 Kbits</td><td>250 m</td></tr><tr><td>125 Kbits</td><td>500 m</td></tr><tr><td>50 Kbits</td><td>1000 m</td></tr><tr><td>20 Kbits</td><td>2500 m</td></tr></tbody></table>	1 Mbits	20 m	500 Kbits	100 m	250 Kbits	250 m	125 Kbits	500 m	50 Kbits	1000 m	20 Kbits	2500 m
1 Mbits	20 m												
500 Kbits	100 m												
250 Kbits	250 m												
125 Kbits	500 m												
50 Kbits	1000 m												
20 Kbits	2500 m												

CANopen®

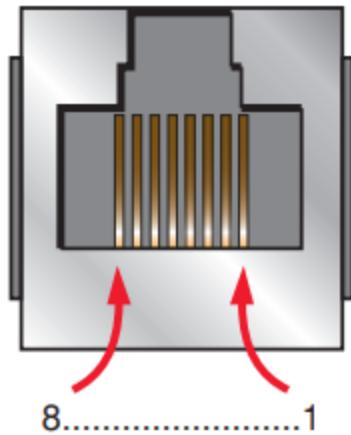


Primeiro escravo da rede inversor de frequência Altivar 31

- Os seguintes parâmetros foram modificados
- **Menu CtL**
 - LAC (nível de acesso) = **L3**
 - Fr1 (configuração de referência) = **CAn**
 - CHCF (modo misto) = **SEP**
 - Cd1 (configuração do canal de comando) = **CAn**
- **Menu Con**
 - AdCO (endereço do inversor) = **1**
 - bdCO (velocidade de transmissão) = **500kbps** (velocidade conforme configurada no “bus parameters” no M340)

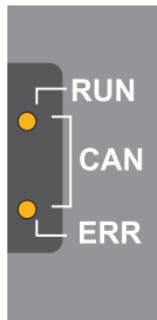
Pinagem e Leds

Pin out of the RJ45 connectors



(2)Modbus signals
(3)Supply for RS232 / RS485 converter or a remote terminal

Pin	Signal
1	CAN_H
2	CAN_L
3	CAN_GND
4	D1 (2)
5	D0 (2)
6	Not connected
7	VP (3)
8	Common (2)

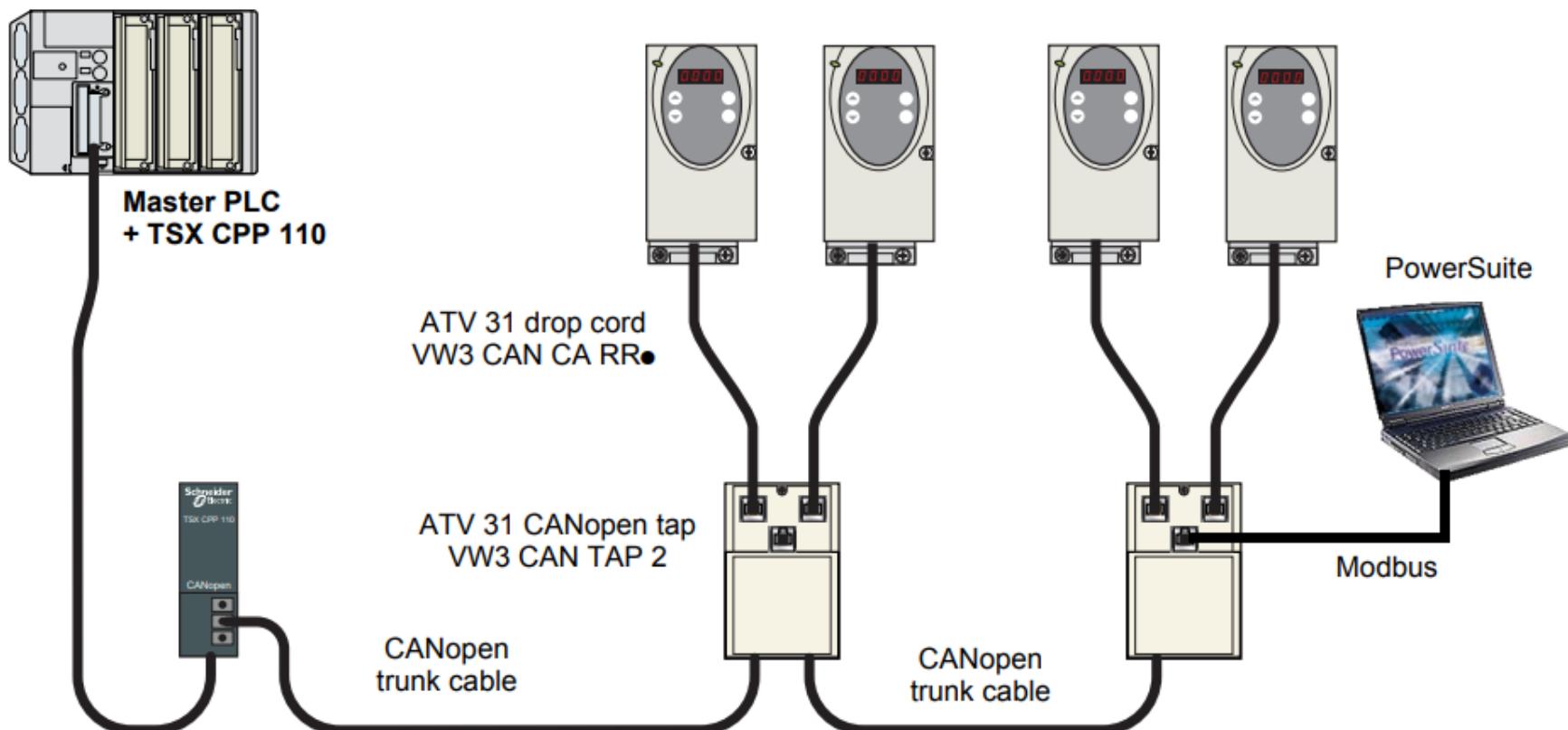


LED state	Altivar 31 / CANopen state
⊗	The CANopen controller is in "OFF" state
✗	The Altivar 31 is in "STOPPED" state
★	The Altivar 31 is in "PRE-OPERATIONAL" state
●	The Altivar 31 is in "OPERATIONAL" state
⊗	No error reported
✗	Warning reported by the CANopen controller of the Altivar 31 (e.g. too many error frames)
✗ ✗	Error due to the occurrence of a node-guarding event or a heartbeat event
●	The CANopen controller is in "bus-off" state

Description of the various LED states:

LED state	Visual description of the LED state
⊗	The LED is OFF
✗	The LED is SINGLE FLASHING (200 ms ON and 1 second OFF)
✗ ✗	The LED is DOUBLE FLASHING (200 ms ON, 200 ms OFF, 200 ms ON, and 1 second OFF)
★	The LED is BLINKING at 2.5 Hz (200 ms ON and 200 ms OFF)
●	The LED is ON

Conexão



CANopen®



Project Browser

Structural view

- Project
 - Configuration
 - 0 : PLC bus
 - 2 : EIO Bus**
 - Derived Data Types
 - Derived FB Types
 - Variables & FB instances
 - Elementary Variables
 - Derived Variables
 - Device DDT Variables
 - IO Derived Variables
 - Elementary FB Instances
 - Derived FB Instances
 - Motion
 - Communication
 - Programs
 - Tasks
 - MAST
 - Events
 - Timer Events
 - IO Events
 - Animation Tables
 - Operator Screens
 - Documentation

Software de
programação

EcoStruxure Control E

- Nesse primeiro momento estaremos trabalhando com o software EcoStruxure Control Expert V14.1 da Schneider.



Principais passos a serem respeitados para configurar a rede CANopen.

Configuration du Bus CANopen :

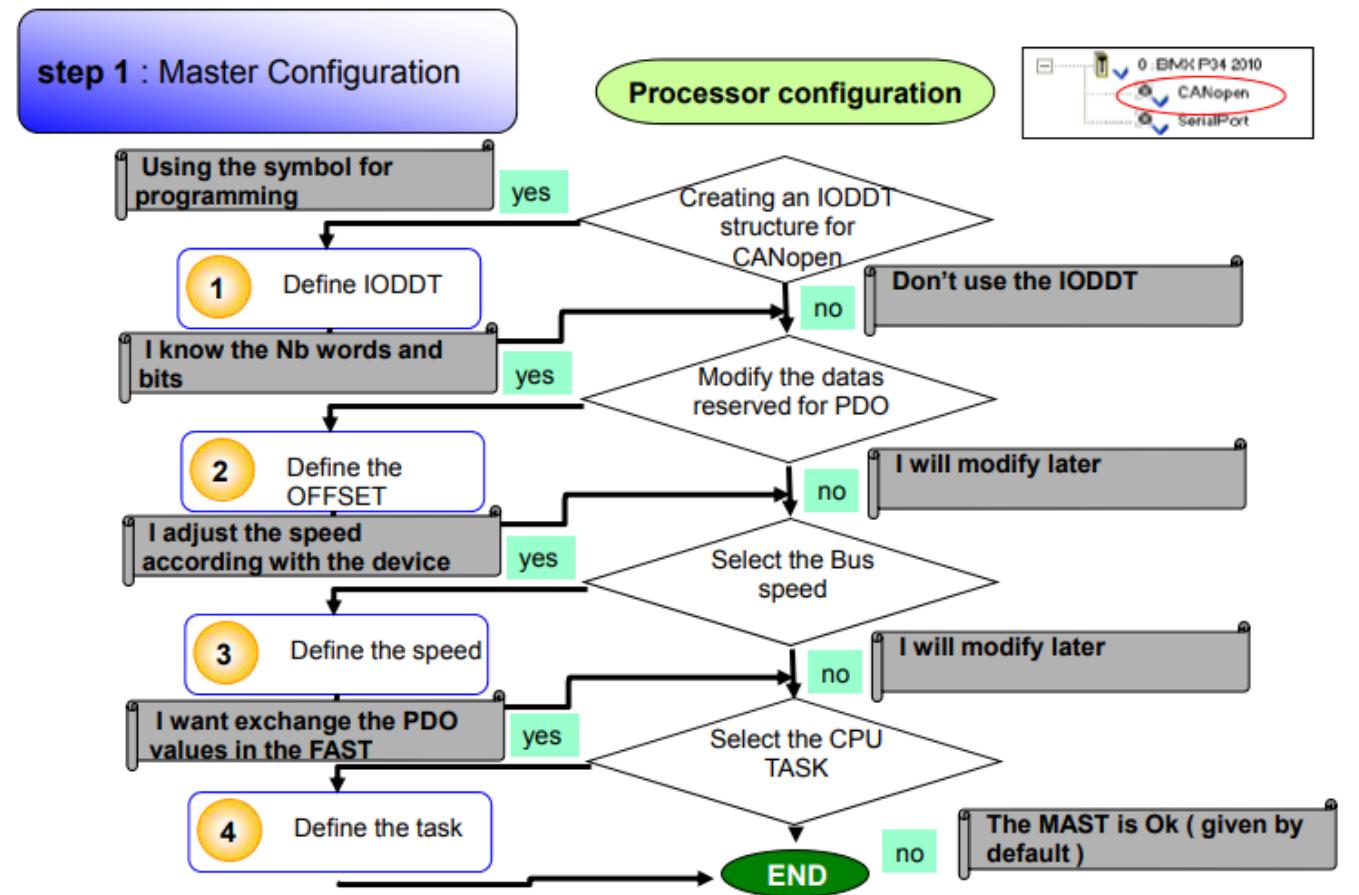
step 1 : Master Configuration (Speed, @, PDO Offset)

step 2 : Slave Configuration (PDO and exchanges)

- Standard
- Expert

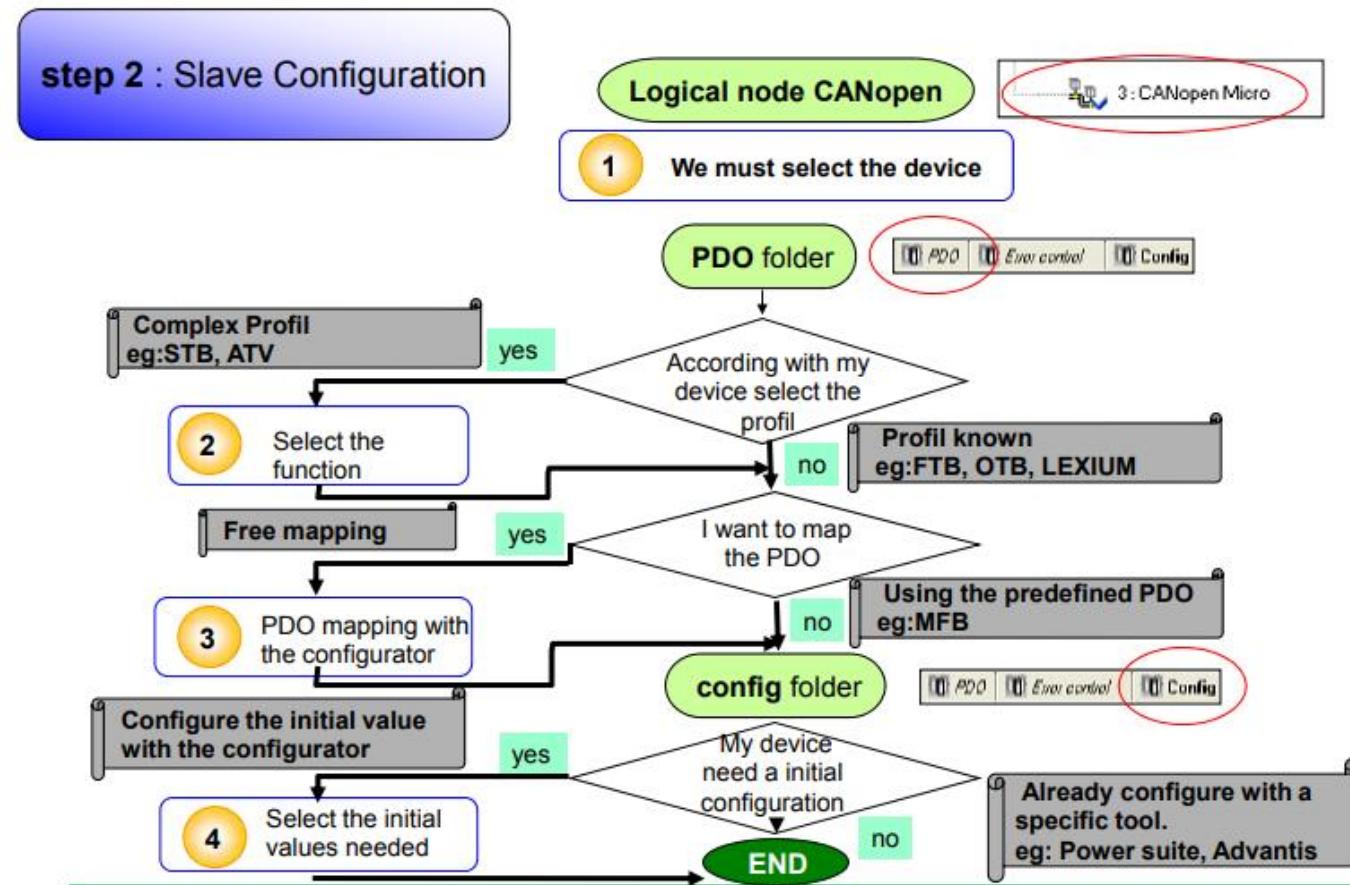


Configuração no mestre



CANopen®

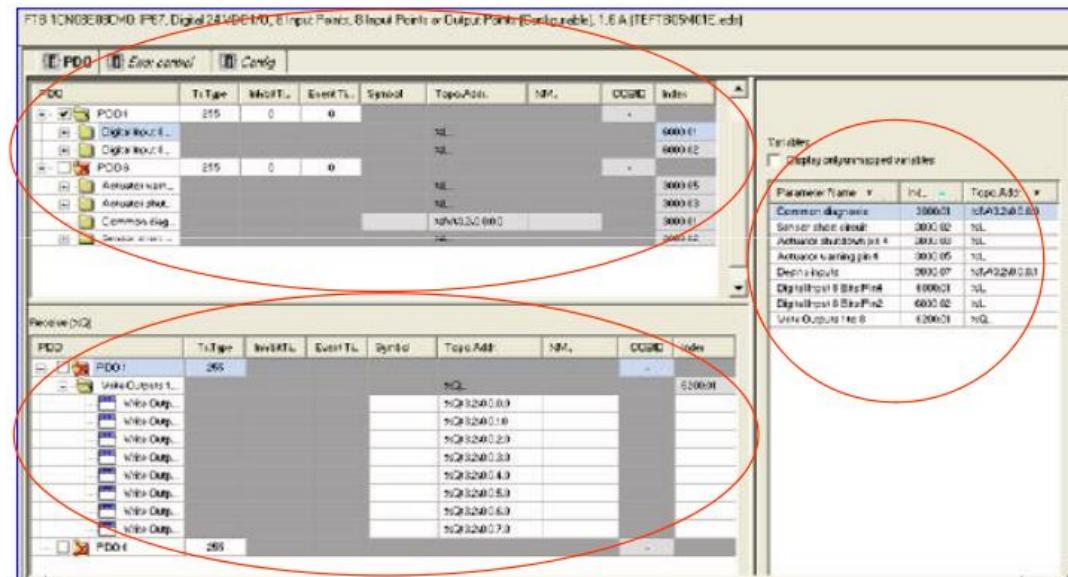
Configuração no escravo



CANopen®

Configuração no escravo

- PDO : In this screen it is possible to map the PDO
- communication.
- This screen is divided in 3 parts :
- ✓ Transmits PDOs (escravo transmite uma informação ao mestre).
- ✓ Receives PDOs (escravo recebe uma informação do Mestre).
- ✓ Uma lista das variáveis que podem ser mapeadas nos PDOs



CANopen®

Configuração no escravo

- PDO: Nesta tela é possível mapear a comunicação PDO.
- Esta tela é dividida em 3 partes:



FIB 1CN03E000MD IP67, Digital/24VDC/0-10V, 6 Input Points, 8 Input Points or Output Points (Selectable), 1.6 A (TEFTB05M01E.eds)

PDO

PDO	Ti Type	MinTtu	EventTtu	Symbol	TopoAdd.	NM	ODGID	Index
PDO1	255	0	0		10		4000.01	
							4000.02	
PDO8	255	0	0		10		3000.05	
							3000.03	
							3000.01	
							3000.02	

Variables

Parameter Name	HL	Type Add.
Common diagnosis	3000.01	%M432@0.000
Sensor short circuit	3000.02	%L
Actuator short circuit 1	3000.03	%L
Actuator warning point 1	3000.05	%L
Depot level	3000.07	%M432@0.001
Digital Out 8 Bits Pin1	4000.01	%L
Digital Out 8 Bits Pin2	4000.02	%L
Vario Output reg 0	4200.01	%Q

Receive (DIO)

PDO

PDO	Ti Type	MinTtu	EventTtu	Symbol	TopoAdd.	NM	ODGID	Index
PDO1	255				10L		5000.01	
					%Q@2@0.003			
					%Q@2@0.010			
					%Q@2@0.023			
					%Q@2@0.030			

Com o configurador CANopen se eu adicionar um PDO ou se eu modificar um PDO, o mapeamento para %Q/I e %MW nunca vai mudar, então não é necessário ajustar sua aplicação.

■ De acordo com o arquivo EDS, alguns PDOs já estão mapeados.

■ Os PDOs usam endereços topológicos %Ix, %Qx, %IW ou %QW.

E

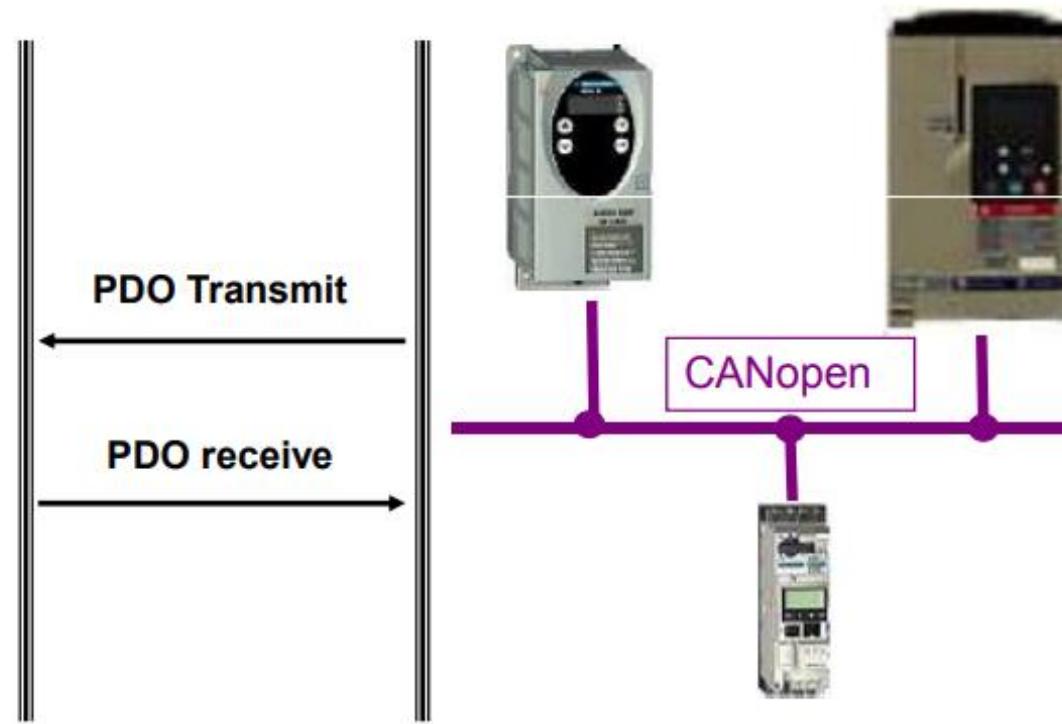
Variáveis internas %MW (estas variáveis estão sempre presentes no produto para a comunicação MODBUS)

Gerenciamento de trocas implícitas de PDO

- Os PDOs permitem o gerenciamento de dados implícitos entre o Mestre, no nosso caso a CPU M340 e o escravo por palavras de leitura/escrita.

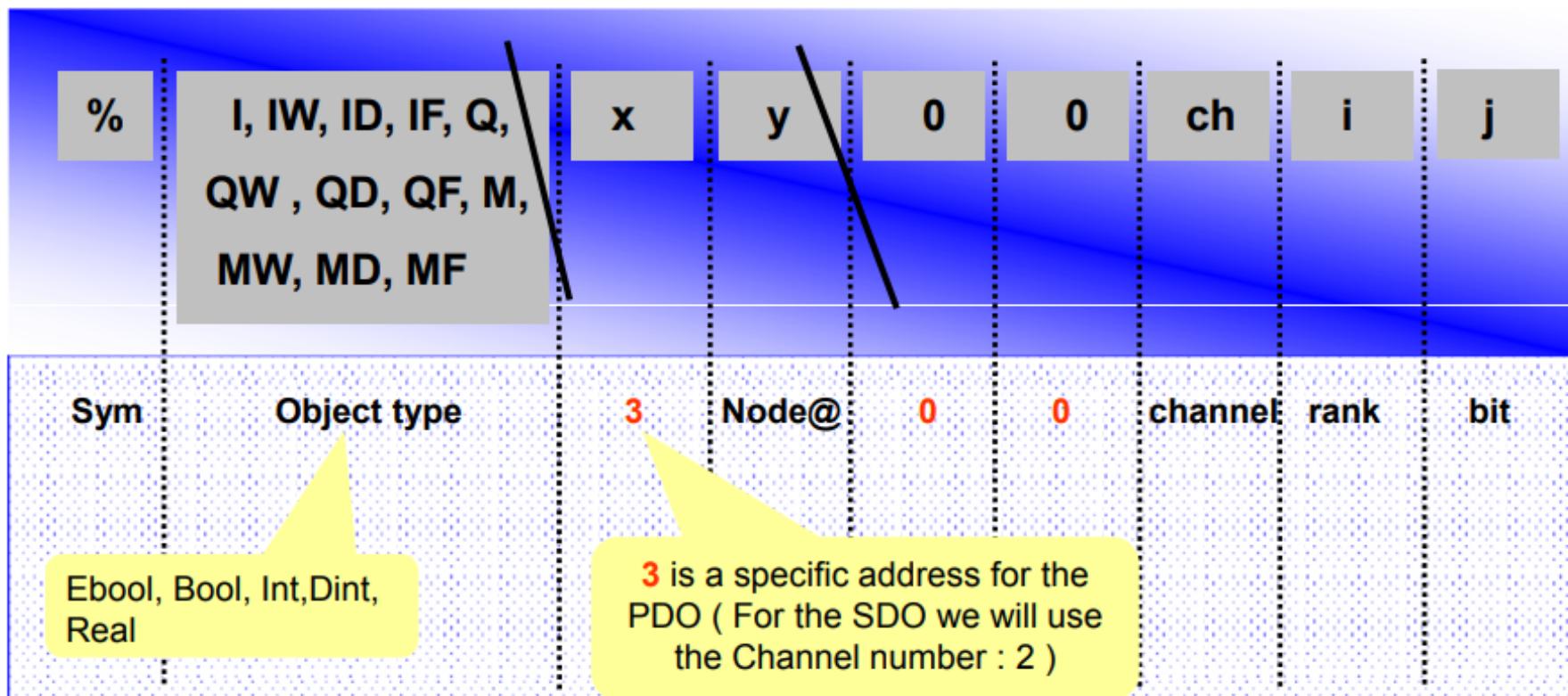


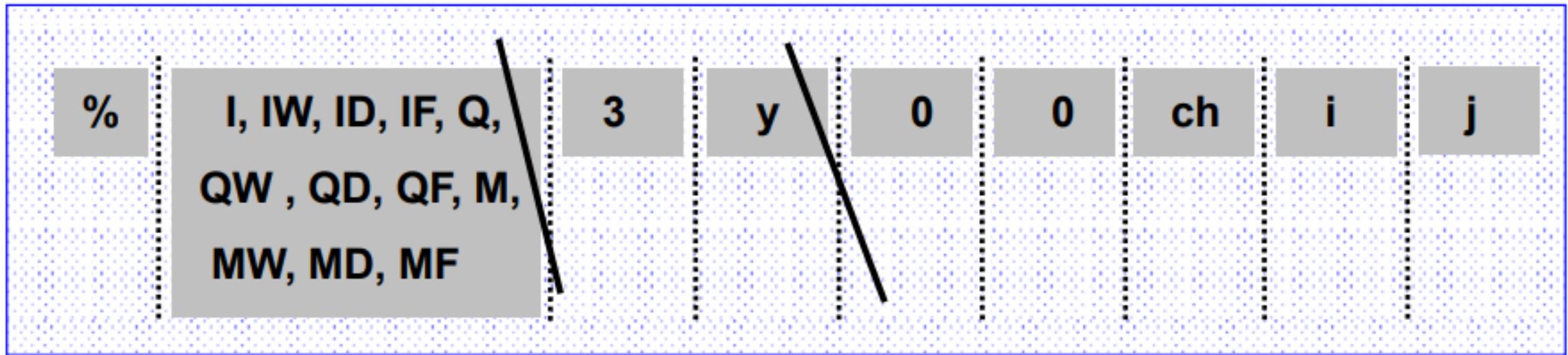
CANopen®



Endereçamento

- No Modicon M340, os endereços de variáveis PDO podem ser definidos como tipos de palavra (Word), palavra dupla (Double Word) e ponto flutuante(Real/Float).





Exemplo

➤ Leia uma entrada em um ADVANTYS @ 4, canal 2, Word 3, bit2

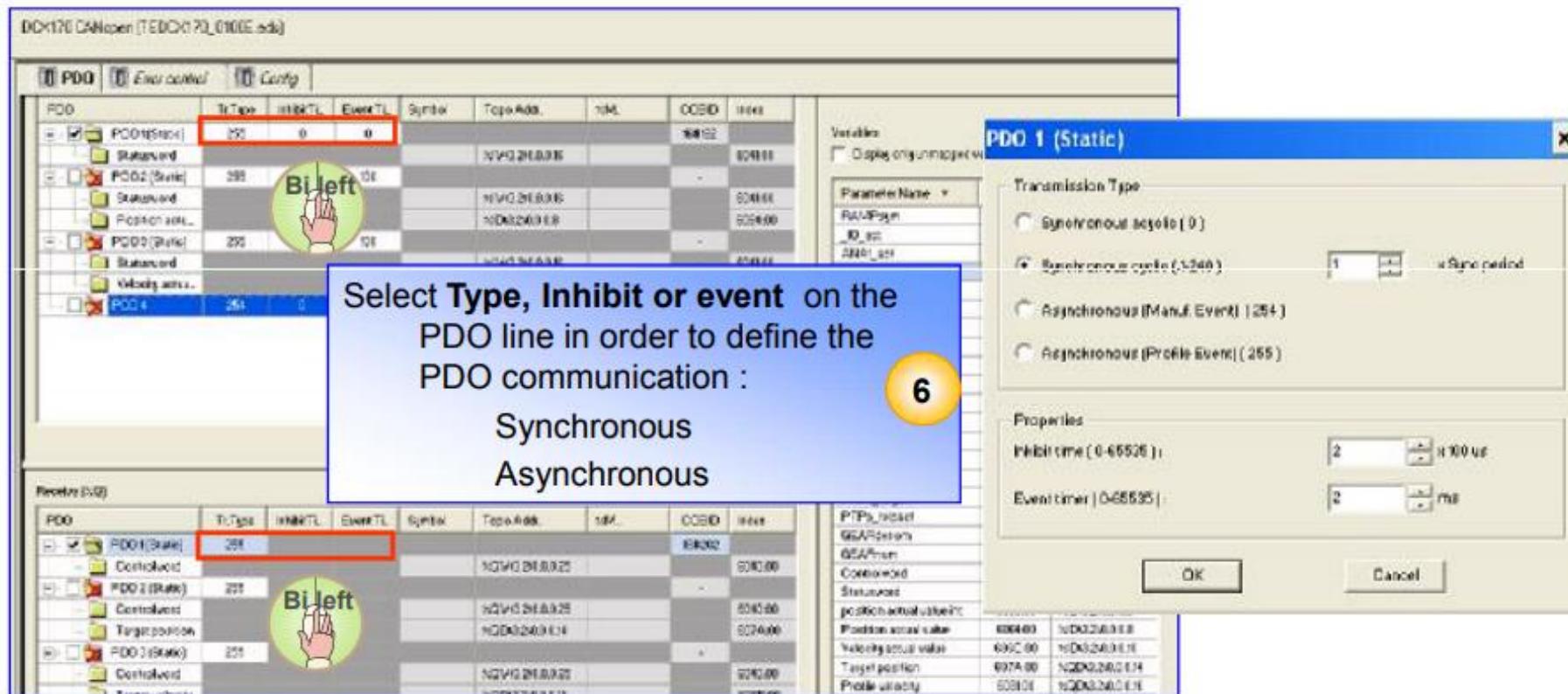
%IW \ 3.4 \ 0 . 0 . 2 . 3 . 2

➤ Escreva a palavra de controle em um LEXIUM @ 9, canal 4, Word 3

%QW \ 3.9 \ 0 . 0 . 4 . 3

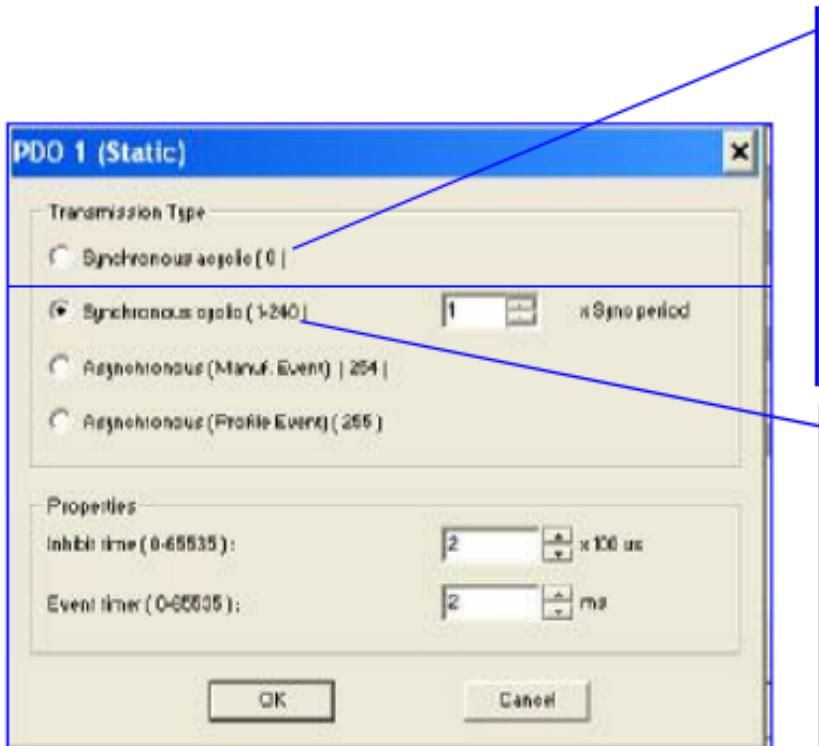
Configuração do escravo ‘EXPERT’

- Nesta tela é possível mapear a comunicação PDO.
 - Esta tela é dividida em 3 partes:



Configuração do escravo ‘EXPERT’

- Os PDO são trocados de acordo com a configuração feita. (exemplo tipo de transmissão).



Synchronous acyclic PDO :

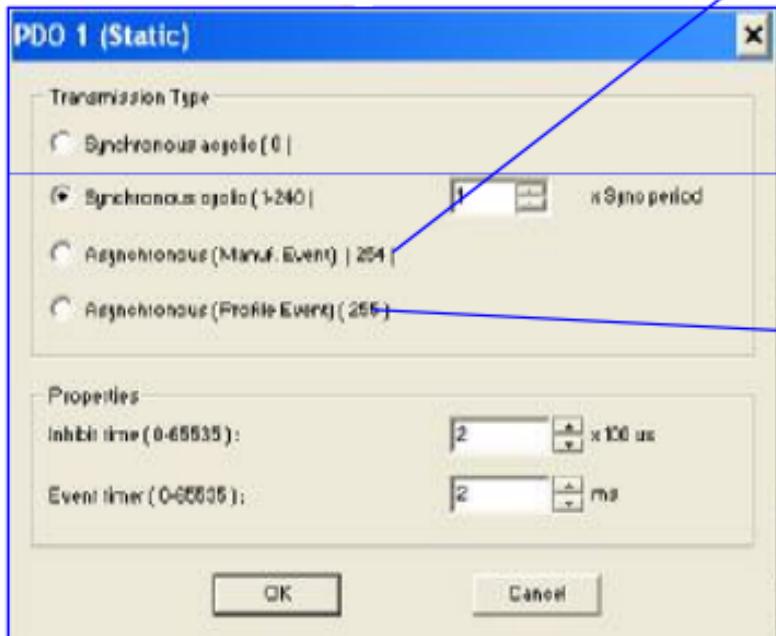
Um tipo de transmissão de 0 significa que a mensagem deve ser transmitida de forma síncrona com a mensagem SYNC, mas não periodicamente de acordo com o valor. Usado principalmente para drives.

Synchronous cyclic PDO :

Um valor entre 1 e 240 significa que o PDO é transmitido de forma síncrona e cíclica, o valor do tipo de transmissão indicando o número de mensagens SYNC entre duas transmissões PDO.

Configuração do escravo ‘EXPERT’

- Os PDO são trocados de acordo com a configuração feita. (exemplo tipo de transmissão).



Asynchronous PDO:

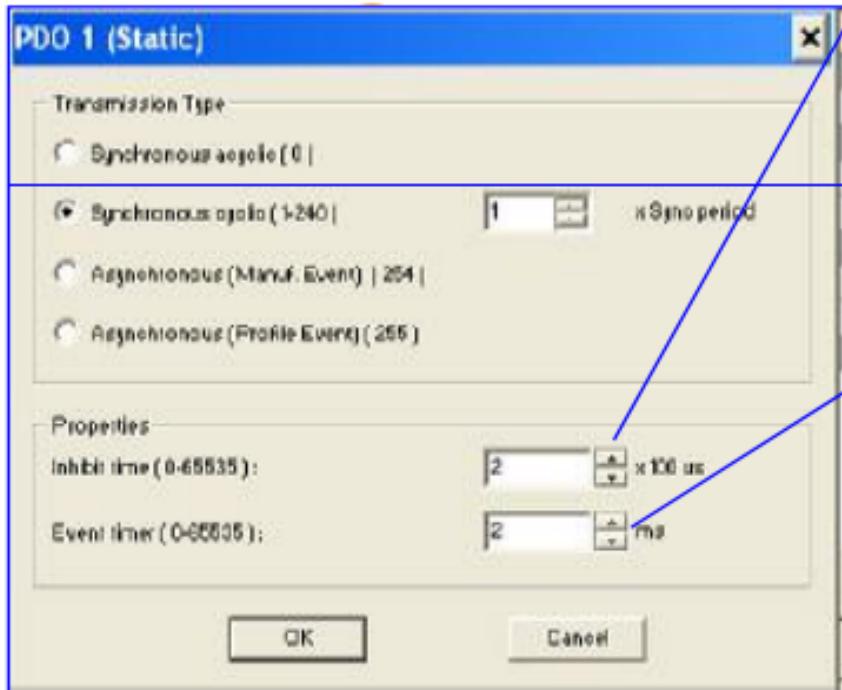
O tipo de transmissão 254 significa que o PDO é transmitido de forma assíncrona. Depende totalmente da implementação no dispositivo. Usado principalmente para I/O digital.

Asynchronous PDO:

O tipo de transmissão 255 significa que o PDO é transmitido assincronamente quando o valor muda.

Configuração do escravo ‘EXPERT’

- Os PDO são trocados de acordo com a configuração feita. (exemplo tipo de transmissão).



Inhibit time :

Mascara a comunicação durante este tempo.

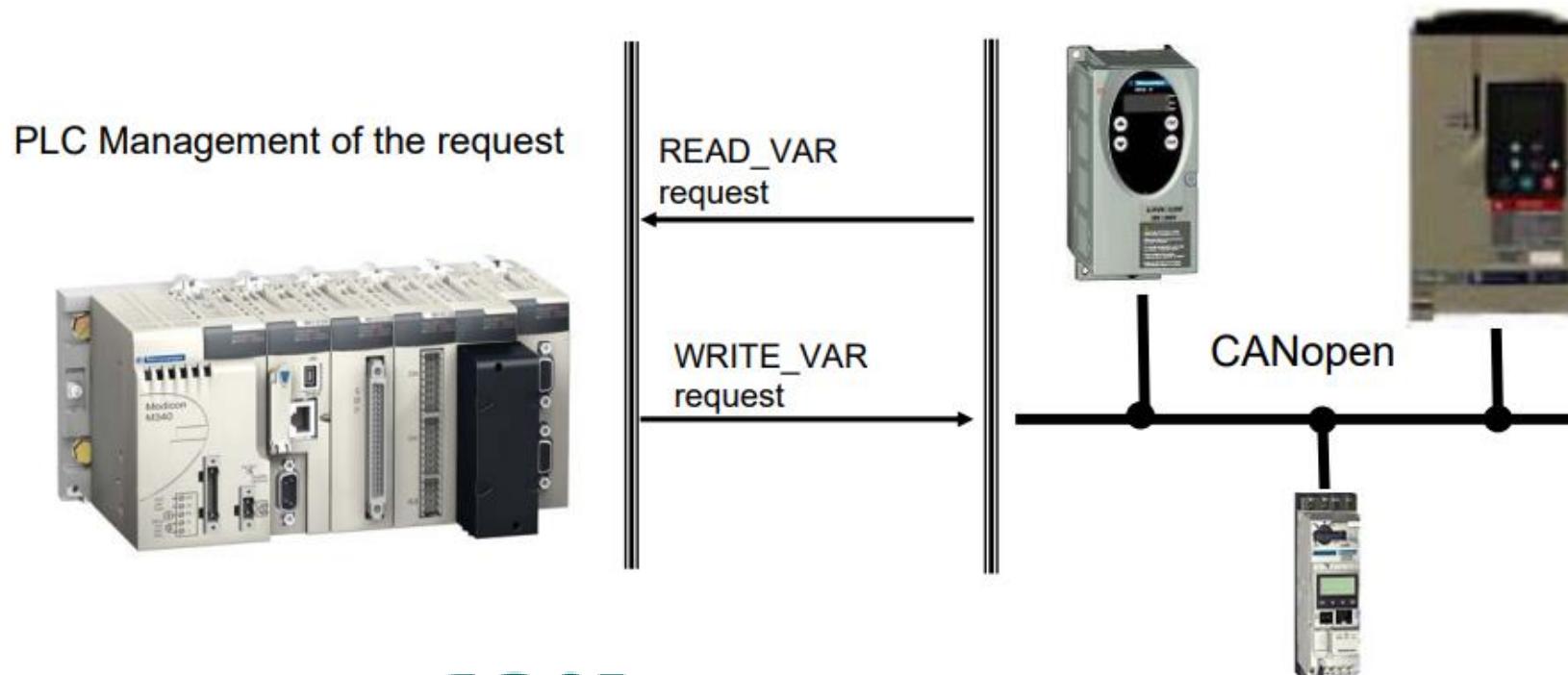
Nota: Com o MFB só podemos ajustar este valor de acordo com o tráfego e o comprimento do bus.

Event timer :

Tempo para gerenciar um evento para iniciar um PDO.

Gerenciamento de trocas de dados explícitas com SDO

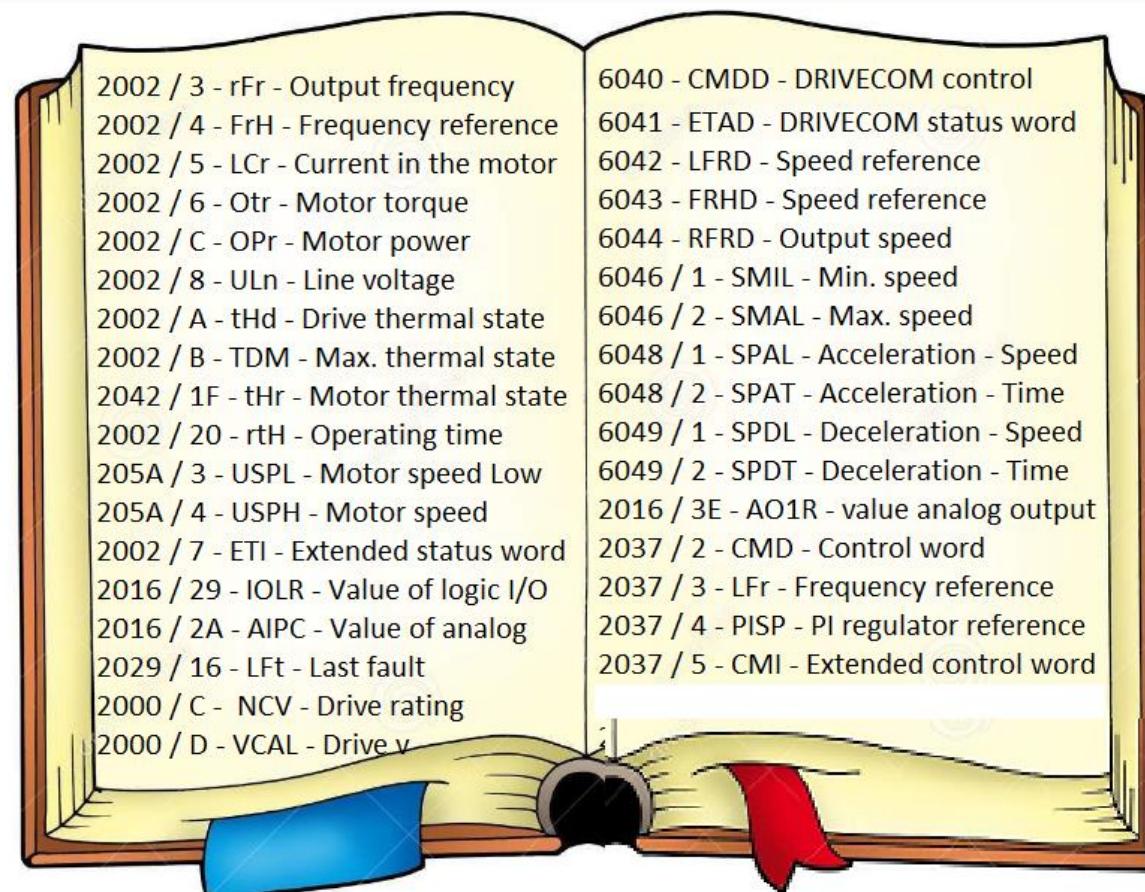
- Os SDOs permitem a gestão dos dados explícitos entre o Master, no nosso caso o M340 e o slave utilizando o request READ_var / WRITE_var.



CANopen®

Conceito de INDEX / SubINDEX

- Todas as requisições no CANopen são definidas com um INDEX/SubINDEX.
- Aqui está um exemplo em um dispositivo CANopen Altivar 31



2002 / 3 - rFr - Output frequency
2002 / 4 - FrH - Frequency reference
2002 / 5 - LCr - Current in the motor
2002 / 6 - Otr - Motor torque
2002 / C - OPr - Motor power
2002 / 8 - ULn - Line voltage
2002 / A - tHd - Drive thermal state
2002 / B - TDM - Max. thermal state
2042 / 1F - tHr - Motor thermal state
2002 / 20 - rtH - Operating time
205A / 3 - USPL - Motor speed Low
205A / 4 - USPH - Motor speed
2002 / 7 - ETI - Extended status word
2016 / 29 - IOLR - Value of logic I/O
2016 / 2A - AIPC - Value of analog
2029 / 16 - LfT - Last fault
2000 / C - NCV - Drive rating
2000 / D - VCAL - Drive v

6040 - CMDD - DRIVECOM control
6041 - ETAD - DRIVECOM status word
6042 - LFRD - Speed reference
6043 - FRHD - Speed reference
6044 - RFRD - Output speed
6046 / 1 - SMIL - Min. speed
6046 / 2 - SMAL - Max. speed
6048 / 1 - SPAL - Acceleration - Speed
6048 / 2 - SPAT - Acceleration - Time
6049 / 1 - SPDL - Deceleration - Speed
6049 / 2 - SPDT - Deceleration - Time
2016 / 3E - AO1R - value analog output
2037 / 2 - CMD - Control word
2037 / 3 - LFr - Frequency reference
2037 / 4 - PISP - PI regulator reference
2037 / 5 - CMI - Extended control word

Syntax para SDO (Write_Var)

- WRITE_VAR

(ADR := ADDM('0.0.2.5'), (*Endereço do dispositivo no barramento CANopen*))

OBJ := 'SDO', (*Objeto do tipo SDO*)

NUM := Subindex_Index, (*Subindex e Index do Objeto a ser escrito*)

NB := 5, (*NodeID do dispositivo na rede*)

EMIS := Sent1, (*Dados a serem enviados/escritos*)

GEST := GUEST1); (*Dados de controle do bloco*)

Syntax para SDO (Read_Var)

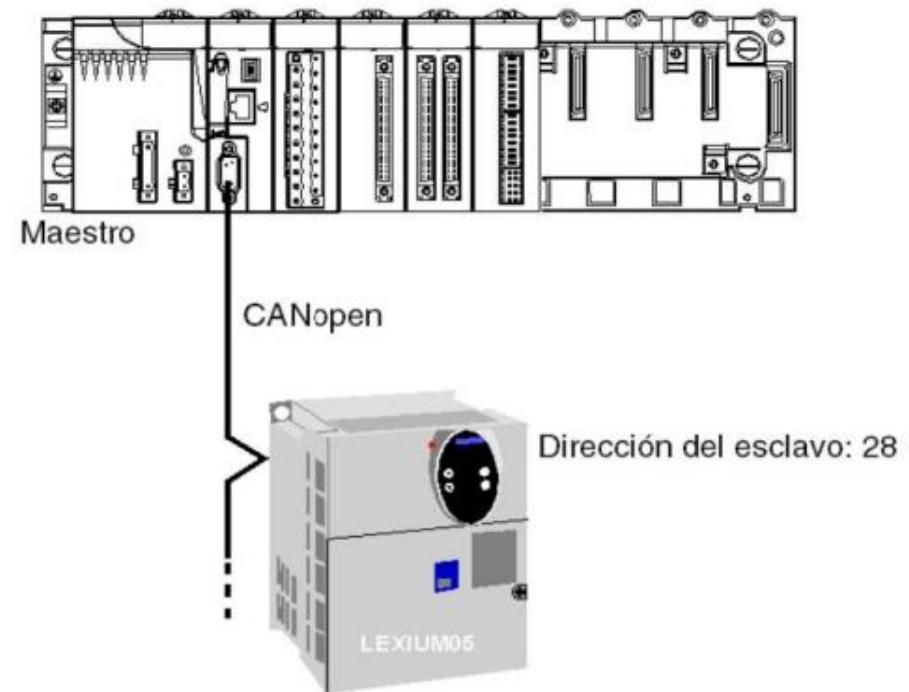
- READ_VAR
- (ADR := ADDM('0.0.2.5'),
(*Endereço do dispositivo no barramento CANopen*)
- OBJ := 'SDO', (*Objeto do tipo SDO*)
- NUM := Subindex_Index, (*Subindex e Index do Objeto a ser lido*)
- NB := 5, (*NodeID do dispositivo na rede*)
- GEST := GUEST1, (*Dados de controle do bloco*)
- RECP => IOLR_Altivar31); (*Dados Lidos*)

Endereçamento do ADDM

Endereçando um dispositivo em um barramento CANopen

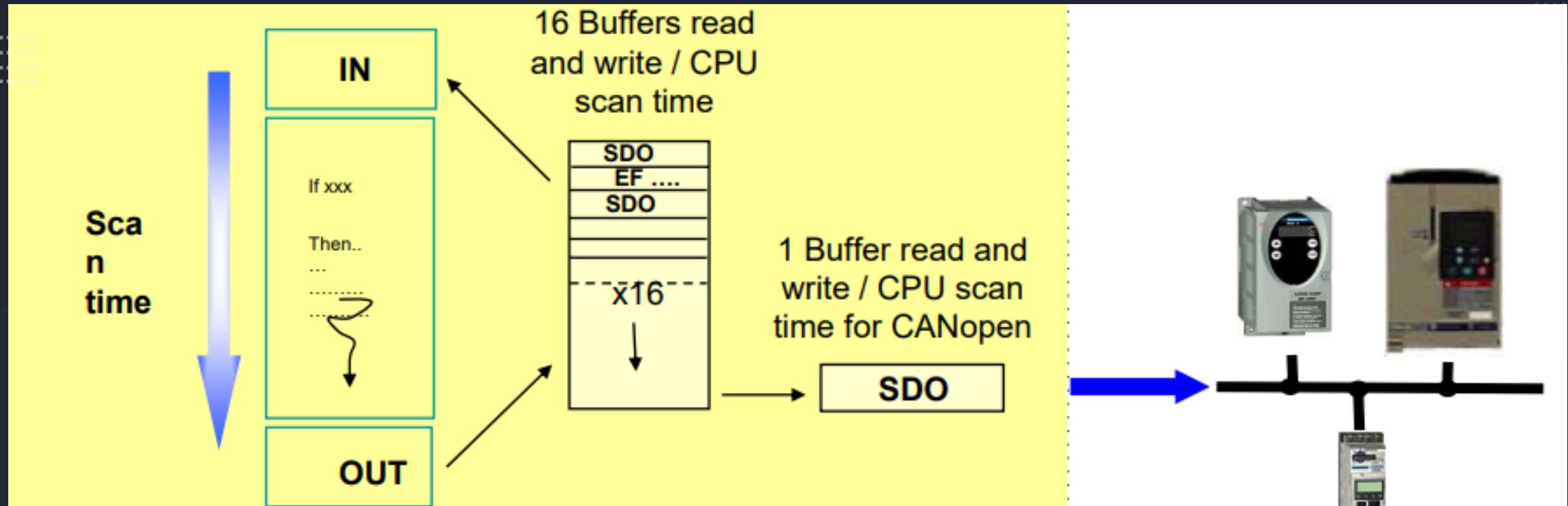
O endereço de um dispositivo em um barramento CANopen assume a forma ADDM('rmce'), onde:

r: número do quadro (rack). m:
posição do módulo rack. c: número do canal
(canal) da porta CANopen (2). e: nó escravo CANopen
(equipamento) (faixa 1 a 127).



Configurações de endereço do escravo 28: ADDM('0.0.2.28').

CANopen®



Performance

CANopen®

- Para todos os processadores podemos gerenciar:
- 16 mensagens em Entrada/Saída por tempo de varredura da tarefa MAST.
- Mas o ideal é que você gerencie seu projeto para enviar um SDO por vez.

Unidade Remota CANopen RUW-01

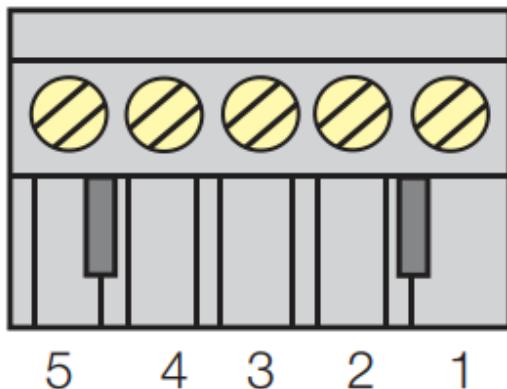


CANopen®

CONEXÃO DA REDE CANOPEN

- A RUW-01 possui um LED na cor verde para indicar que a interface está alimentada. O módulo para comunicação CAN possui um conector plug-in de 5 vias (XC3) com a seguinte pinagem:

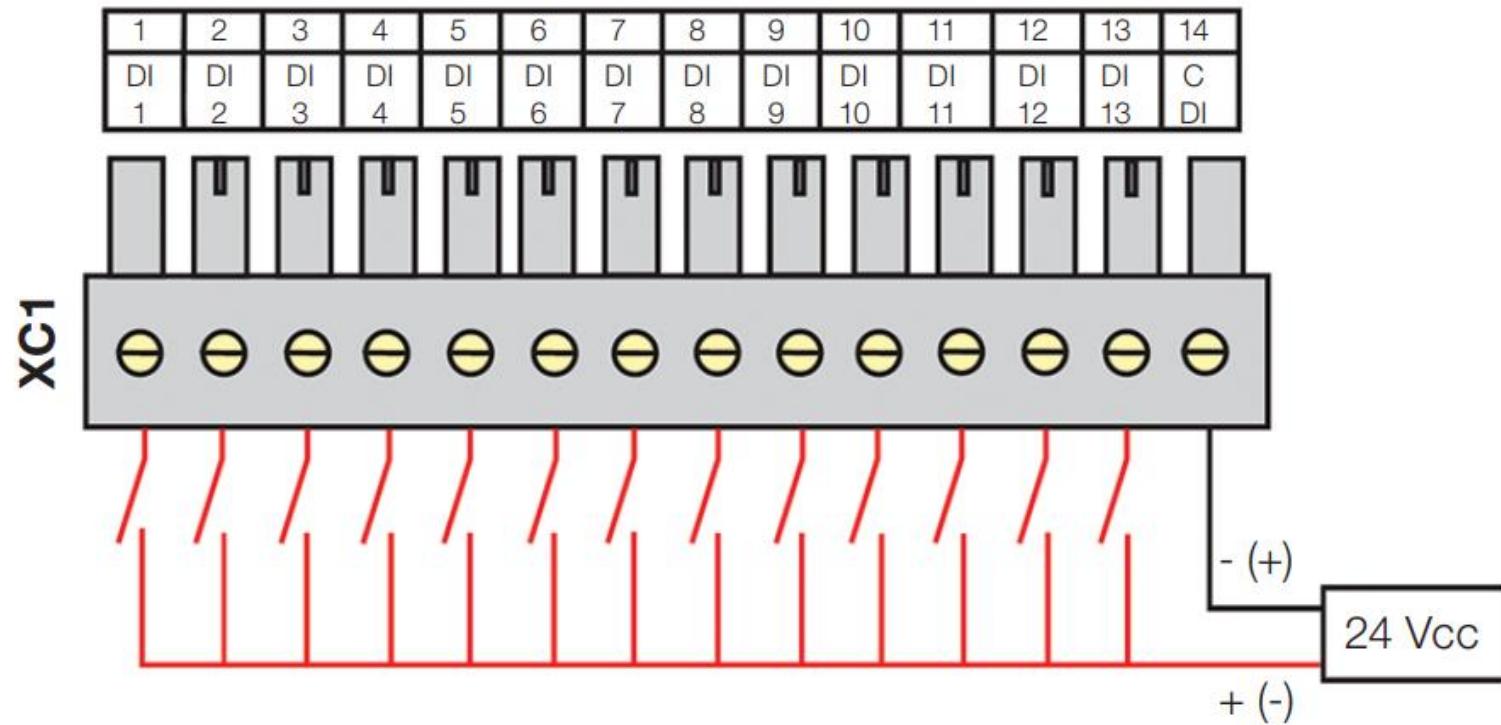
Tabela 1: Pinagem do conector XC3 para interface CAN



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

CONEXÃO DAS ENTRADAS DIGITAIS

- A RUW-01.01 apresenta 13 entradas digitais bidirecionais que podem ser acionadas através de uma fonte de tensão externa de 24 Vcc.



(a) XC1: Acionamento das entradas digitais da RUW-01.01

CONFIGURAÇÃO

- A configuração da RUW-01 é feita através de chaves hexadecimal com as seguintes funções:
- S1 e S2: Configuração do endereço na rede CANopen.
- S3: Seleção da taxa de comunicação.

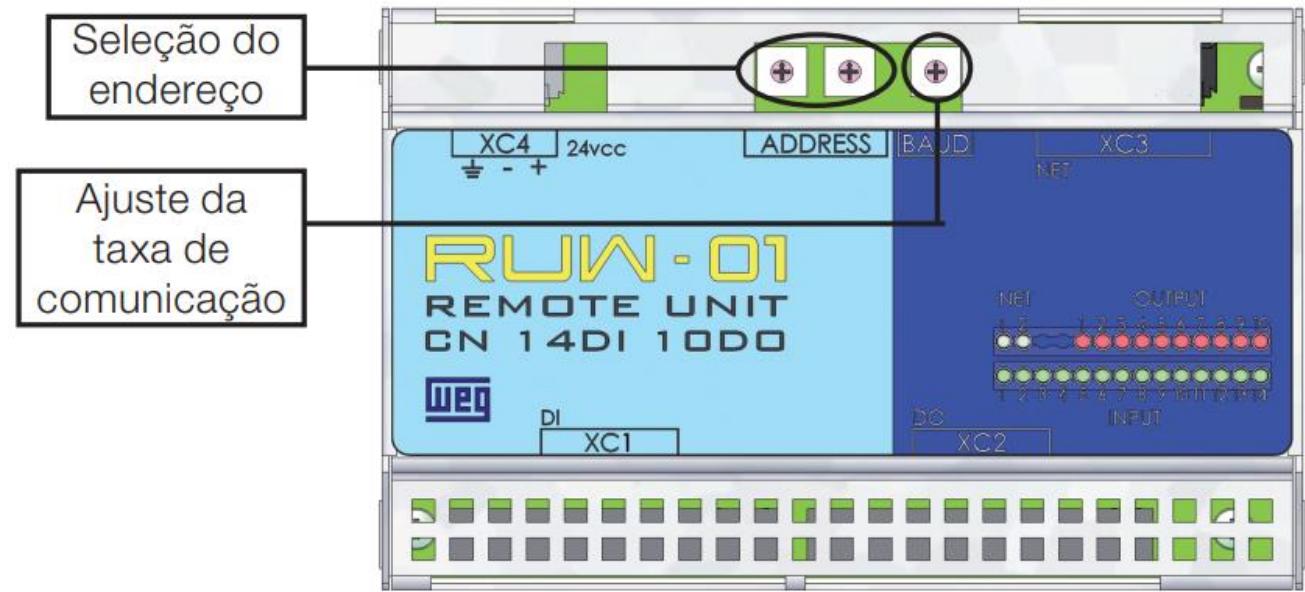
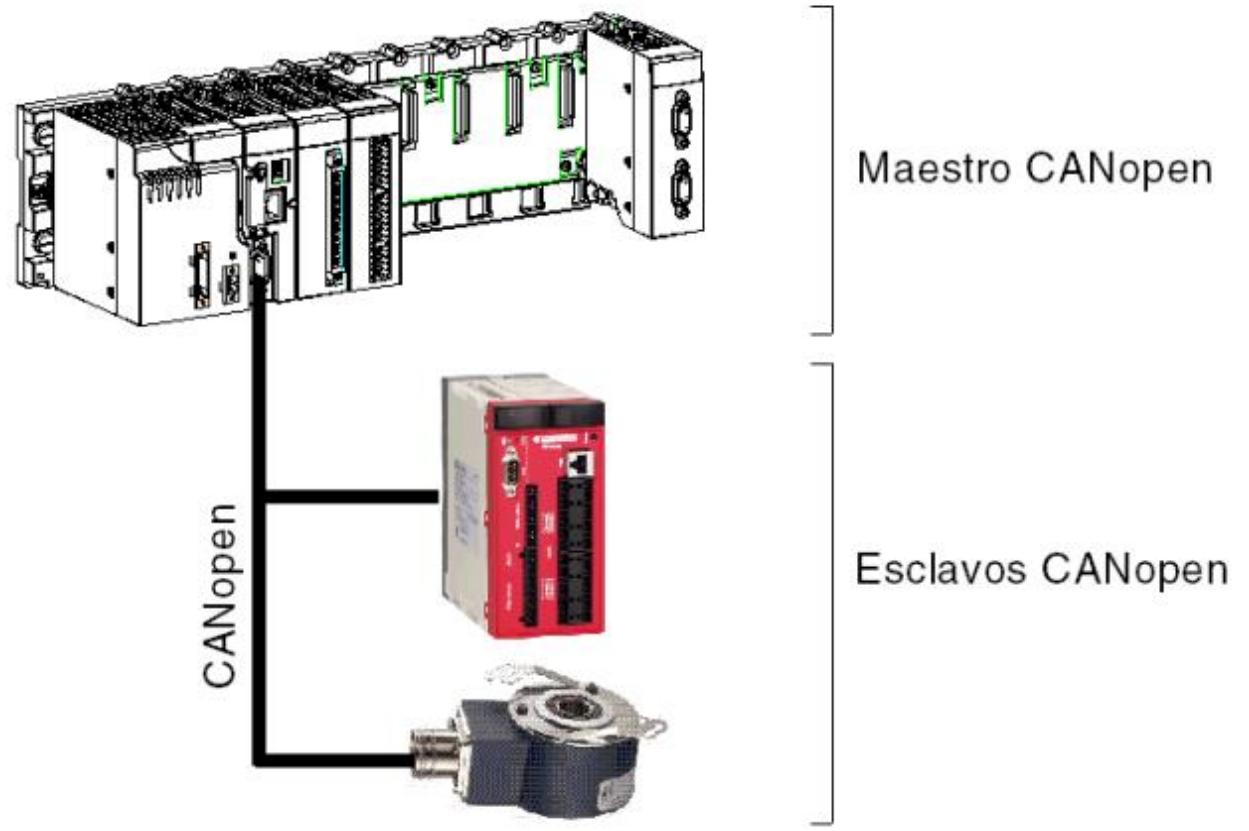


Figura 5: Chaves hexadecimais para configuração da RUW-01

CANopen®

Objetos de linguagem IODDT específicos do CANopen

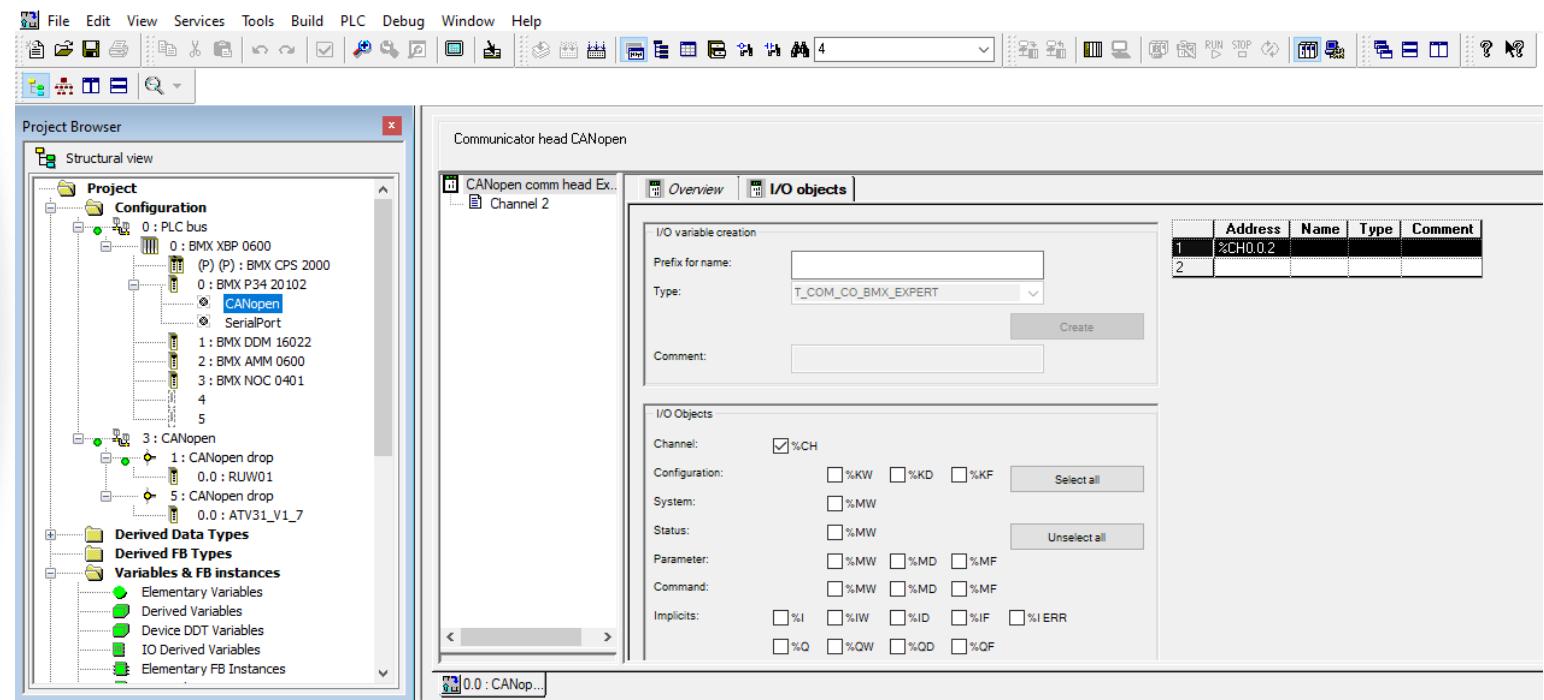
- Objetos de intercambio implícito IODDTT_COM_CO_BMX_EXPERT
- A ilustração a seguir mostra um exemplo de um indicador de status para o Mestre da rede.
- Neste exemplo, a palavra %IW0.0.2.1 fornece o status do mestre CANopen.
- Os parâmetros são os seguintes:
- r: '0'
- m: '0'
- c: '2' (canal CANopen).
- O último parâmetro ('1') indica a palavra utilizada (CAN_STS).



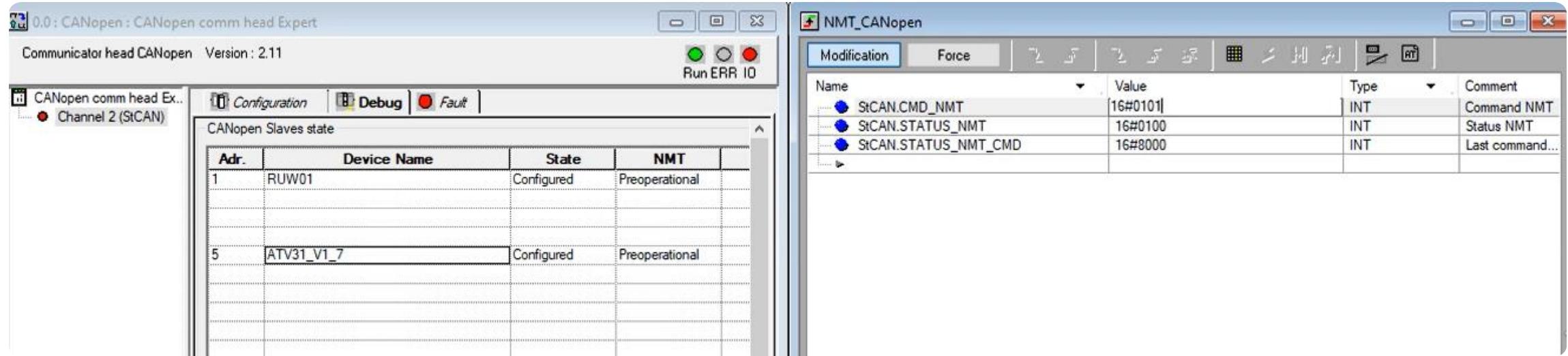
CANopen®

T_COM_CO_BMX_EXPERT

- Com esse IODDT é possível trabalhar com diversos status da nossa rede CANopen, como por exemplo o estado dos dispositivos da rede.
- Para habilitá-lo basta configurar o I/O objects da interface de comunicação CANopen da CPU criando uma variável como %CH.



CANopen®



Comandos NMT no Control Expert

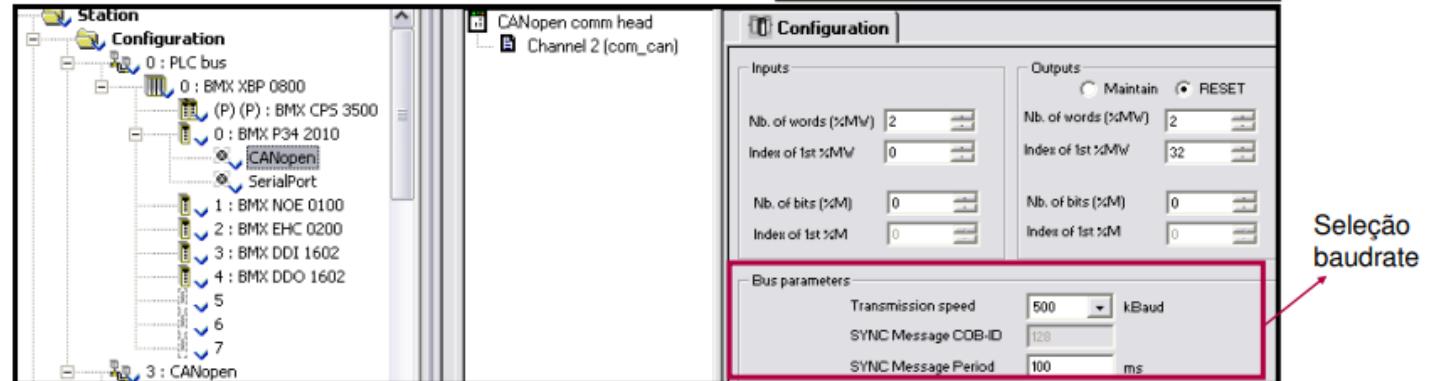
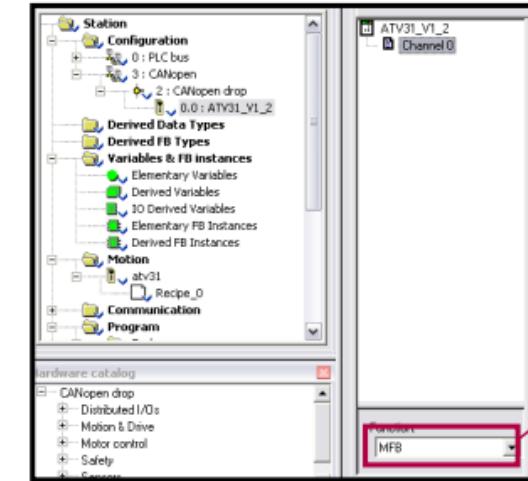
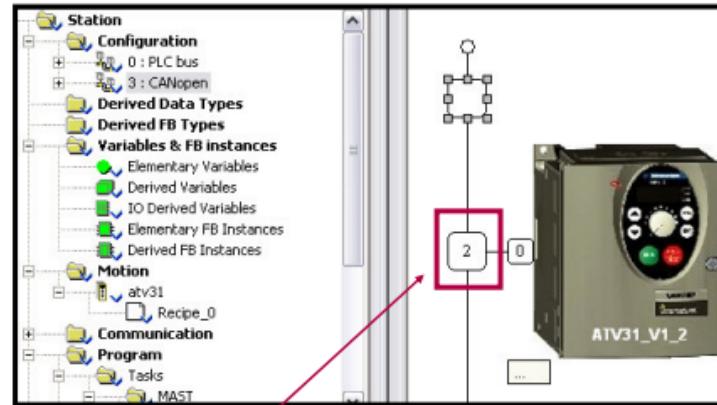
- Ao habilitar o IODDT T_COM_CO_BMX_EXPERT
- Podemos utilizar os comandos NMT:
- 01h – Start,
- 02h – Stop,
- 80h – Pre-Operational,
- 81h – Reset Node,
- 82 – Reset Com.

CANopen®

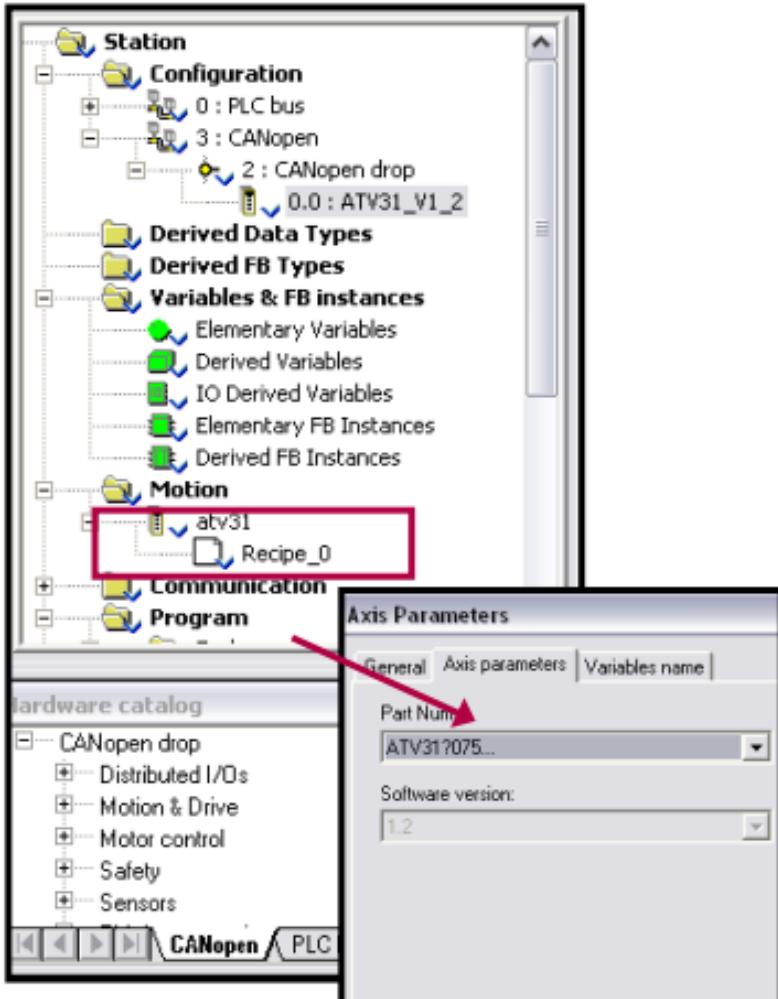
Comunicação CanOpen utilizando a biblioteca PLC Open - MFB (Motion Function Block) com ATV31.

CANopen®

• Configuração no PLC



• Configuração no PLC



• Configuração do “Axis”

Part Num:

Atentar para a referência do inversor
Ex.: ATV31075M2A

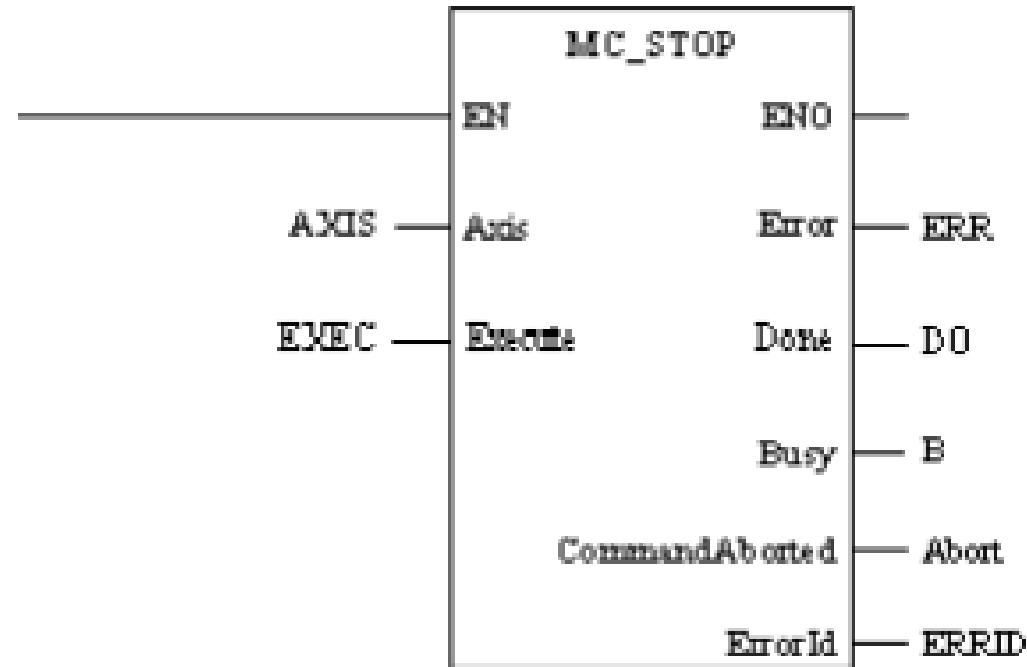
Software version:

1.2

A função MFB está disponível para ATV31 V1.2 . É possível trabalhar com inversores cuja versão de firmware é V1.7

Função dos blocos

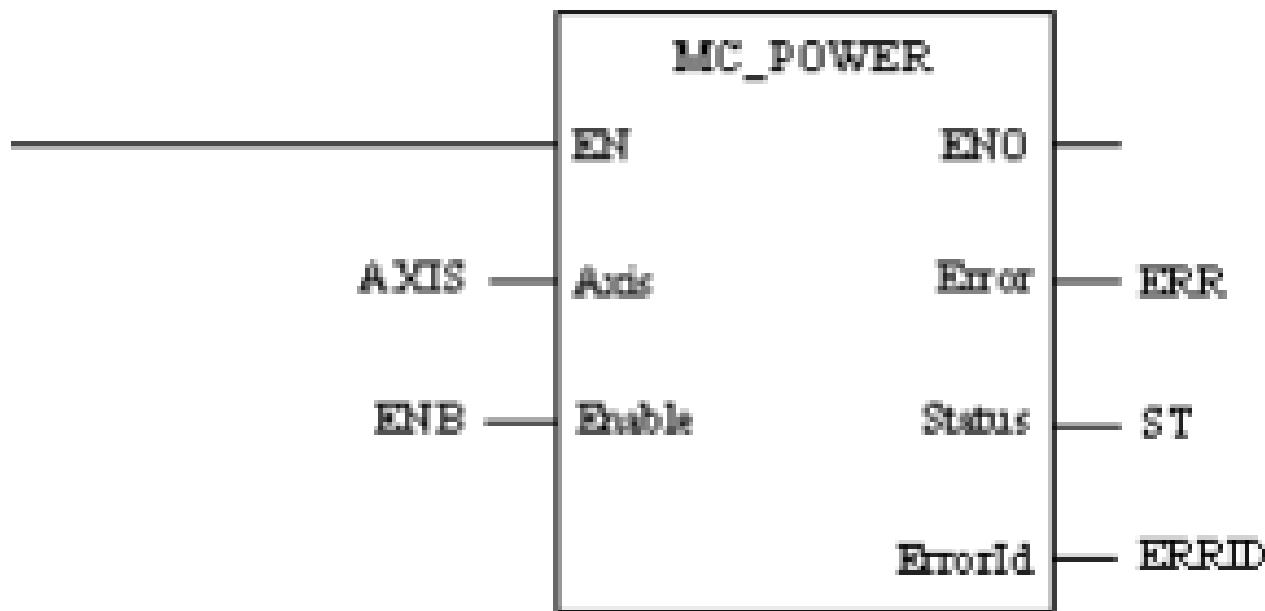
- MC_POWER
- A função do MC_POWER é habilitar o inversor no barramento CANopen, baseado no eixo configurado.



CANopen®

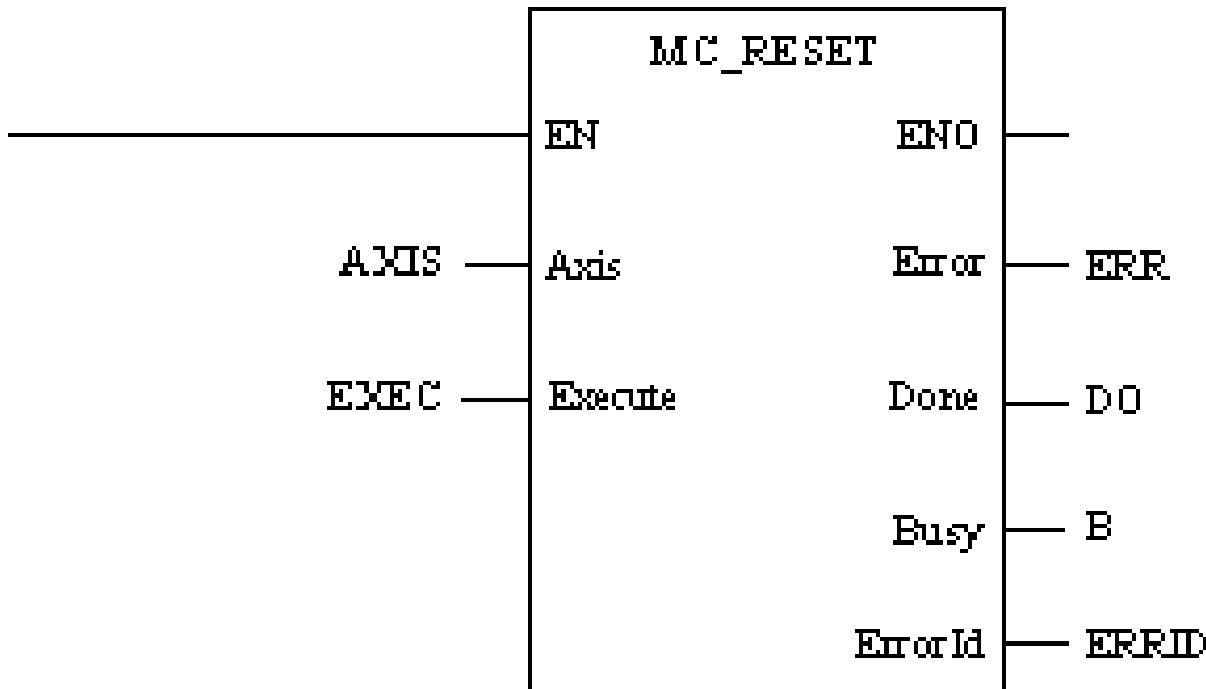
Função dos blocos

- MC_STOP
- Com a função MC_STOP ocorre a parada do inversor. É possível monitorar o ERRID e se o canal está ocupado (BUSY)



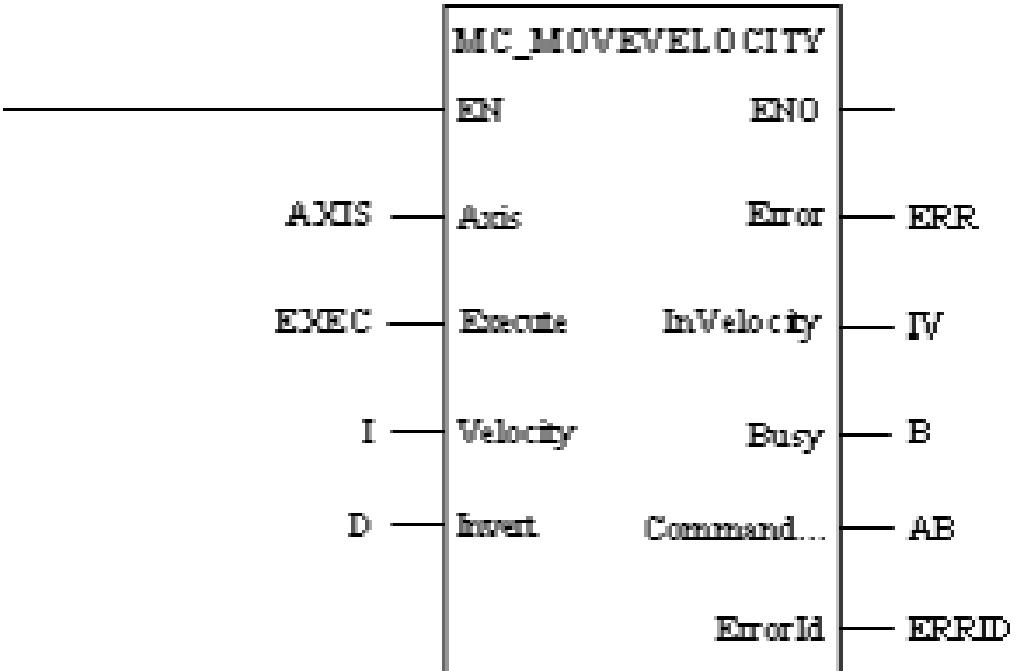
Função dos blocos

- MC_RESET
- A função MC_RESET é usada retornar o sistema a seu estado inicial, sem a exibição dos erros e advertências ocorridos durante a simulação.



Função dos blocos

- MC_MOVEVELOCITY
- A função MC_MOVEVELOCITY é usada para determinar o setpoint de velocidade. A entrada Execute possibilita habilitar o sistema para sentido avanço com o setpoint. Para inversão é necessária habilitar a entrada Invert e em seguida a entrada Execute para habilitar o sentido.



Configurando inversor de frequência CFW300 na rede CANopen

- Para trabalhar com um inversor de frequência CFW300 do fabricante WEG na rede CANopen precisamos de uma interface de comunicação CCAN.
- A interface CAN possui um conector de 5 vias com a seguinte pinagem:



Detalhe do conector CAN

Pinagem do conector CAN

Pino	Nome	Função
25	V-	Pólo negativo da fonte de alimentação
26	CAN_L	Sinal de comunicação CAN_L
27	Shield	Blindagem do cabo
28	CAN_H	Sinal de comunicação CAN_H
29	V+	Pólo positivo da fonte de alimentação

Configuração dos parâmetros do Drive:

Estes parâmetros são utilizados na configuração da fonte de comandos para os modos local e remoto do produto.

Para que o equipamento seja controlado através da interface CANopen, deve-se selecionar uma das opções 'CANopen(CO/DN/DP)' disponíveis nos parâmetros.

P313 – AÇÃO PARA ERRO DE COMUNICAÇÃO

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

Descrição:

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

P700 – PROTOCOLO CAN

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

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.

P701 – ENDEREÇO CAN

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

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 P700:

- P700 = 1 (CANopen): endereços válidos: 1 a 127.
- P700 = 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.

P702 – 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:

CAN

P703 – RESET DE BUS OFF

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

Descrição:

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

Opções para o parâmetro P703

Opção	Descrição
0 = Reset Manual	Caso ocorra bus off, será indicado na HMI o alarme A134/F34, a ação programada no parâmetro P313 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 P313.

P705 – 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.

P706 – CONTADOR DE TELEGRAMAS CAN RECEBIDOS

Faixa de Valores:	0 a 9999	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.

P707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS

Faixa de Valores:	0 a 9999	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.

P708 – CONTADOR DE ERROS DE BUS OFF

Faixa de Valores:	0 a 9999	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.

P709 – CONTADOR DE MENSAGENS CAN PERDIDAS

Faixa de Valores:	0 a 9999	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.

P721 – ESTADO DA COMUNICAÇÃO CANOPEN

Faixa de Valores:	0 = Desabilitado 1 = Reservado 2 = Comunicação habilitada 3 = Controle de erros Habilitado 4 = Erro de Garding 5 = Erro de Heartbeat	Padrão: -
Propriedades:	RO	

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).

P722 – ESTADO DO NÓ CANOPEN

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

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.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Função																
Reservado									Reset de Falhas	Parada Rápida	Utiliza Segunda Rampla	LOC/REM	JOG	Sentido de Giro	Habilita Geral	Gra/Para

P684 – PALAVRA DE CONTROLE

- Palavra de comando do equipamento via interface CANopen.
- Este parâmetro somente pode ser alterado via interface CANopen.
- Para as demais fontes (HMI, etc.) ele se comporta como um parâmetro somente de leitura.
- Para que os comandos escritos neste parâmetro sejam executados, é necessário que o equipamento esteja programado para ser controlado via CANopen.
- Esta programação é feita através dos parâmetros P105 e P220 até P228.
- Cada bit desta palavra representa um comando que pode ser executado no produto.



CANopen®

Função dos bits do parâmetro P684

Bit	Valor/Descrição
Bit 0 Gira/Para	0: para motor por rampa de desaceleração. 1: gira motor de acordo com a rampa de aceleração até atingir o valor da referência.
Bit 1 Habilita Geral	0: desabilita geral o drive, interrompendo a alimentação para o motor. 1: habilita geral o drive, permitindo a operação do motor.
Bit 2 Sentido de Giro	0: sentido de giro do motor oposto ao da referência (sentido reverso). 1: sentido de giro do motor igual ao da referência (sentido direto).
Bit 3 JOG	0: desabilita a função JOG. 1: habilita a função JOG.
Bit 4 LOC/REM	0: drive vai para o modo local. 1: drive vai para o modo remoto.
Bit 5 Utiliza Segunda Rampa	0: drive utiliza como rampa de aceleração e desaceleração do motor os tempos da primeira rampa, programada nos parâmetros P100 e P101. 1: drive utiliza como rampa de aceleração e desaceleração do motor os tempos da segunda rampa, programada nos parâmetros P102 e P103.
Bit 6 Parada Rápida	0: não executa comando de parada rápida 1: executa comando de parada rápida.
Bit 7 Reset de Falhas	0: sem função. 1: se em estado de falha, executa o reset do drive.
Bit 8...15	Reservado.



P685 – REFERÊNCIA DE VELOCIDADE

- Permite programar a referência de velocidade para o motor via interface CANopen.
- Este parâmetro somente pode ser alterado via CANopen.
- Para as demais fontes (HMI, etc.) ele se comporta como um parâmetro somente de leitura.
- Para que a referência escrita neste parâmetro seja utilizada, é necessário que o produto esteja programado para utilizar a referência de velocidade via CANopen.
- Esta programação é feita através dos parâmetros P221 e P222.
- Esta palavra utiliza resolução de 13 bits com sinal para representar a frequência nominal (P403) do motor:
- $P685 = 0000h$ (0 decimal) \rightarrow referência de velocidade = 0 $P685 = 2000h$ (8192 decimal) \rightarrow referência de velocidade = frequência nominal (P403)
- Valores de velocidade intermediários ou superiores podem ser obtidos utilizando esta escala.
- Por exemplo, se $P403 = 60$ Hz de frequência nominal, caso deseje-se uma referência de 30 Hz, deve-se calcular:
 - $60 \text{ Hz} \Rightarrow 8192$, $30 \text{ Hz} \Rightarrow$ Referência em 13 bits
 - Referência em 13 bits = $30 \times 8192 / 60$
 - Referência em 13 bits = 4096 \Rightarrow Valor correspondente a 30 Hz na escala em 13 bits



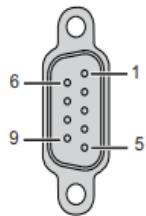
Modicon M258

CANopen®

- O controlador lógico Modicon M258 é um CLP compacto, de alto desempenho e totalmente expansível. Ele faz parte da abordagem de Controle de Máquina Flexível, um componente chave do MachineStruxur TM, que traz flexibilidade e oferece uma solução de controle otimizada.
- Este CLP é projetado para fabricantes de máquinas (OEMs) com foco em aplicações como embalagem, elevação, transporte e armazenamento, têxteis e marcenaria, elevação, etc.
- Oferece soluções de alto desempenho para controle de velocidade, contagem, controle de eixo e funções de comunicação.

Pin Assignment

The following illustration describes the pins of the CAN port:



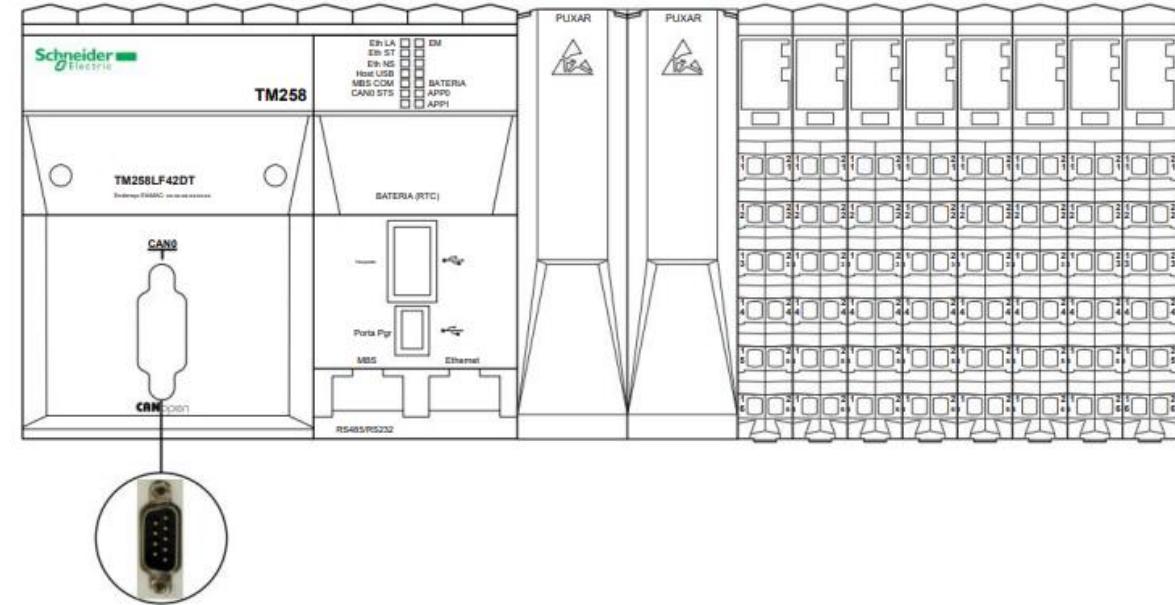
The following table describes the pins of the CAN port:

PIN N°	Signal	Description
1	N.C.	Reserved
2	CAN_L	CAN_L bus Line (Low)
3	CAN_GND	CAN 0 Vdc
4	N.C.	Reserved
5	CAN_SHLD	N.C.
6	GND	0 Vdc
7	CAN_H	CAN_H bus Line (High)
8	N.C.	Reserved
9	N.C.	Reserved

N.C.: No Connection.

The shield is connected to pin 6, the 0 Vdc pin.

NOTE: Pin 9 is not connected internally. The controller does not provide power on CAN_V+.

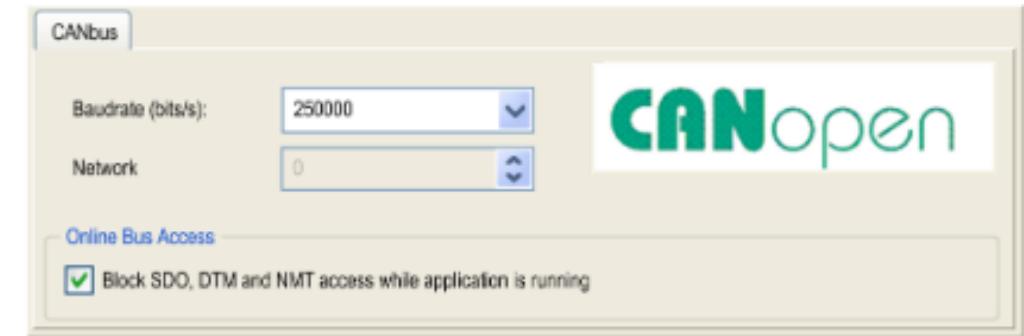


CANopen®

Interface CANopen

CANopen Configuração no SoMachine

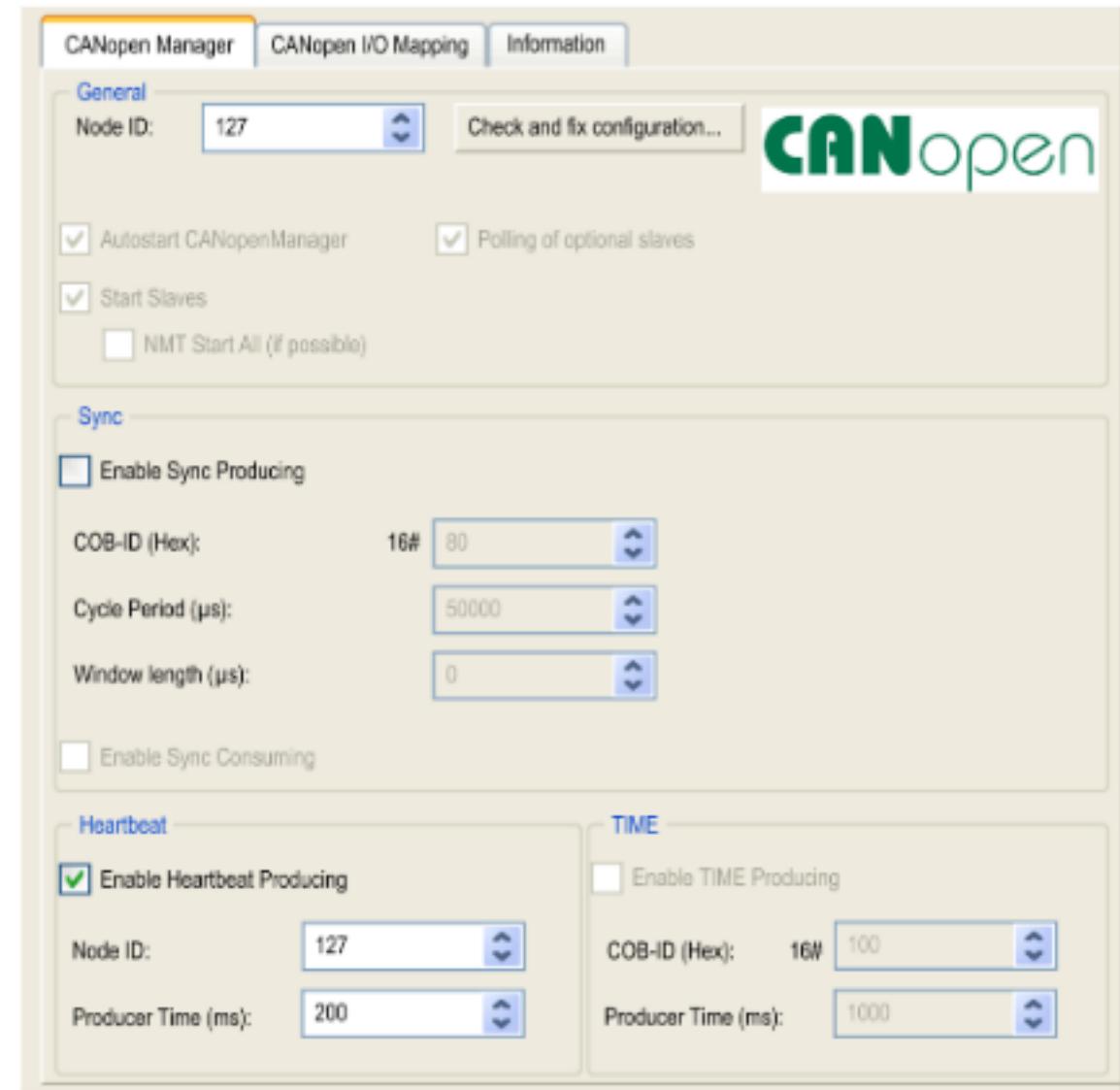
- 1º - Na árvore Dispositivos , clique duas vezes em CAN .
- 2º - Configure a taxa de transmissão (por padrão: 250.000 bits/s):.
- OBS: A opção Online Bus Access permite bloquear o envio de SDO,DTM, e NMT através da tela de status.



canopen

Criação e Configuração do CANopen Manager

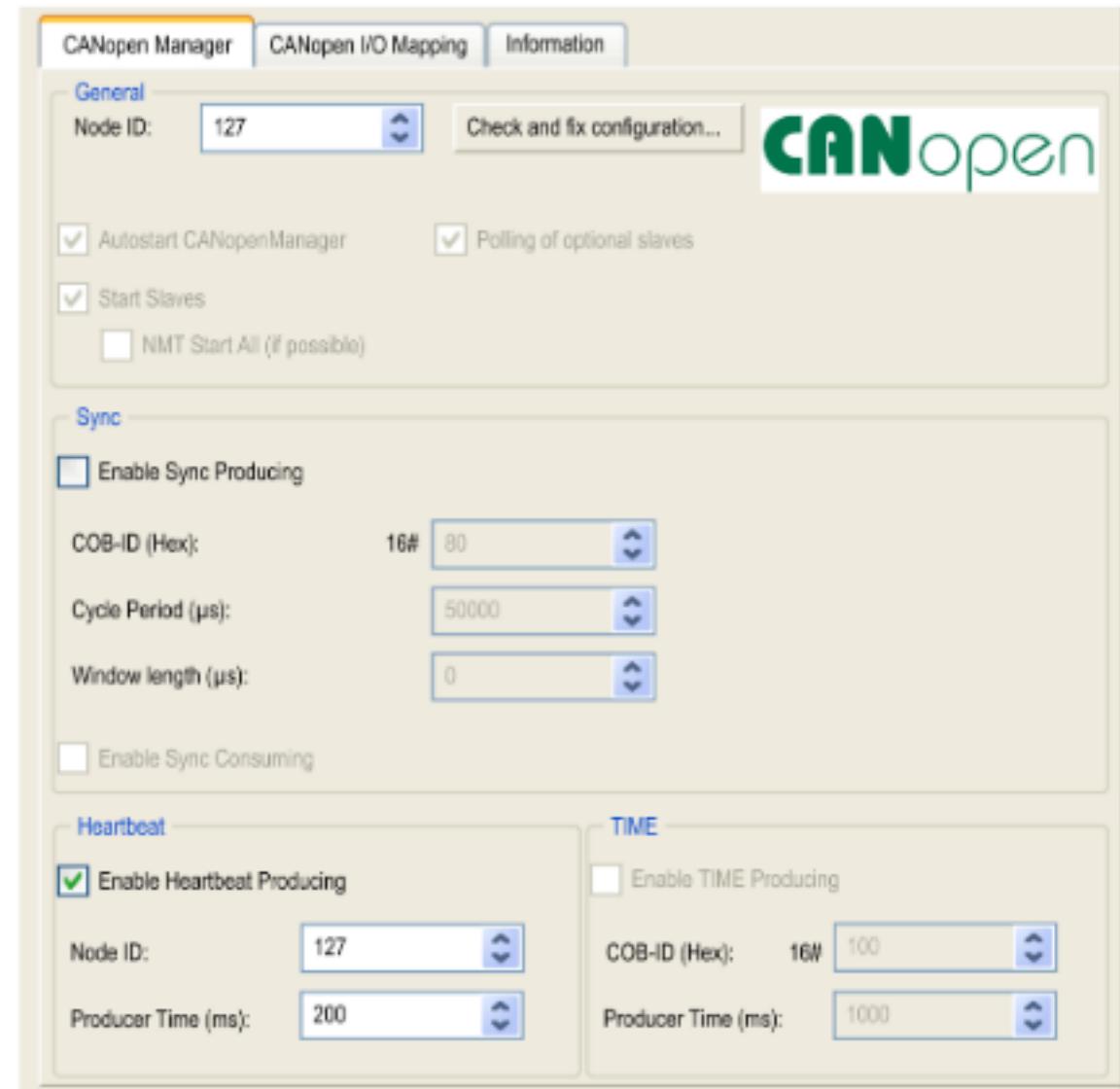
- 1º - Para adicionar um CANopen Otimizado ao seu controlador, clique no botão Mais ao lado do nó CAN na Árvore de Dispositivos .
- Na janela Add Device , selecione CANopen Optimized e clique no botão Add Device ..
- 2º - Clique duas vezes em CANopen_Optimized .
- Resultado: Aparece a janela de configuração do CANopen Manager :



CANopen®

Criação e Configuração do CANopen Manager

- NOTA: Se Ativar Produção de Sincronização estiver marcada, a tarefa CAN_x_Sync será adicionada ao nó Aplicativo > Configuração de Tarefa na guia da árvore Aplicativos .
- Não exclua ou altere os atributos de tipo ou evento externo das tarefas CAN_x_Sync . Se você fizer isso, o SoMachine detectará um erro quando você tentar criar o aplicativo e não poderá baixá-lo para o controlador.
- Se você desmarcar a opção Ativar produção de sincronização na subguia CANopen Manager da guia CANopen_Optimized , a tarefa CAN0_Sync será excluída automaticamente de seu programa.



canopen®

Limites Operacionais CANopen do CLP M258

O Modicon M258 Logic Controller CANopen mestre tem os seguintes limites de operação:

- ✓ Número máximo de dispositivos escravos 63
- ✓
- ✓ Número Máximo de PDO Recebido (RPDO) 126
- ✓
- ✓ Número máximo de PDO transmitido (TPDO) 126



CANopen®

Recomendações ao operar com M258 em rede CANopen

- Não conecte mais de 63 dispositivos escravos CANopen ao controlador.
-
- Programe seu aplicativo para usar 126 ou menos Transmit PDO (TPDO).
-
- Programe seu aplicativo para usar 126 ou menos Receive PDO (RPDO).
-



CANopen®

Funções de gerenciamento do CANopen com a Biblioteca CAA CiA 405

- A interface CANopen e o perfil do dispositivo para controladores programáveis IEC 61131-3 (CiA405) descrevem dois métodos de acesso à rede CANopen a partir do controlador:
 - ✓ network variables
 - ✓ function blocks
- Variáveis de Rede: As variáveis de rede são geralmente mapeadas em PDO para serem recebidas ou transmitidas. No dicionário de objetos, as variáveis IEC 61131-3 são acessíveis em um intervalo definido de índices.
- Blocos de função: O perfil também define alguns blocos de função específicos do CANopen, por exemplo, SDO, NMT e serviços de comunicação de emergência.

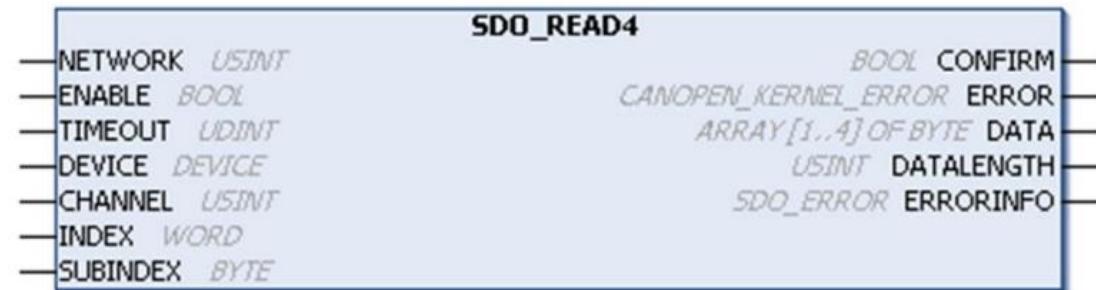


Biblioteca CAA CiA 405

- A biblioteca CAA CiA 405 oferece um conjunto de blocos de função para atender aos requisitos do CiA405 para o acesso à rede CANopen a partir do aplicativo (programa IEC61131-3) do controlador (CANopen mestre).
- A biblioteca é declarada automaticamente no gerenciador de biblioteca do controlador quando um gerenciador CANopen é adicionado a uma interface de barramento CAN do controlador.

Blocos de função de acesso SDO

- CIA405.SDO_READ4: Lê objetos CANopen de até 4 bytes:
- O bloco funcional CIA405.SDO_READ4 é usado para ler um objeto CANopen de até 4 bytes de um dispositivo especificado por meio de uma mensagem SDO.
- Esses parâmetros específicos devem ser passados para o bloco de função:
 - ✓ ID do nó do dispositivo.
 - ✓ Canal cliente/servidor SDO (por padrão, apenas um canal é definido) Índice/subíndice de objeto CANopen.
- O bloco de função retorna o tamanho do objeto lido se a leitura foi concluída com sucesso.
- Os dados estão disponíveis em uma matriz de 4 bytes.



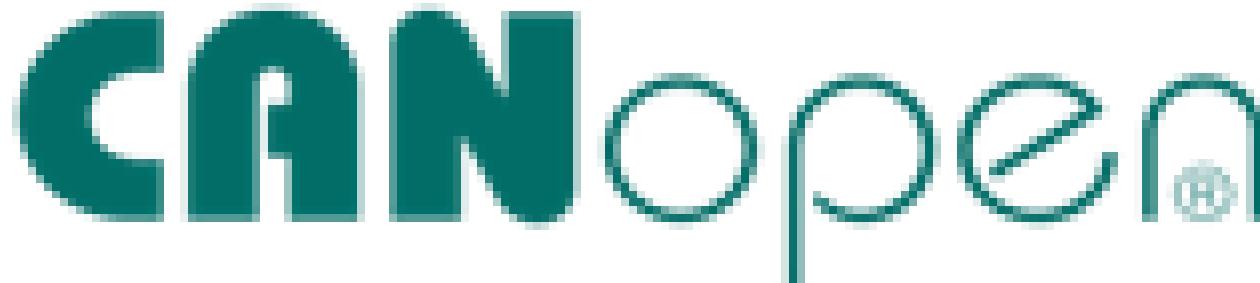
CANopen®

A tabela a seguir mostra o tamanho do objeto em relação ao conteúdo da matriz de DADOS.

Object size Example	DATALENGTH	DATA(1)	DATA(2)	DATA(3)	DATA(4)
1 byte 01 hex	1	01 hex	not significant	not significant	not significant
2 bytes 01 23 hex	2	LSB 23 hex	MSB 01 hex	not significant	not significant
3 bytes 01 23 45 hex	3	LSB 45 hex	23 hex	MSB 01 hex	not significant
4 bytes 01 23 45 67 hex	4	LSB 67 hex	45 hex	23 hex	MSB 01 hex

LSB = low significant byte

MSB = most significant byte



CIA405.SDO_WRITE4: Grava objetos CANopen de até 4 bytes

- O bloco funcional CIA405.SDO_WRITE4 é usado para escrever um objeto CANopen de até 4 bytes de um dispositivo especificado por meio de mensagens SDO.
- Esses parâmetros específicos devem ser passados para o bloco de função:
 - ✓ ID do nó do dispositivo.
 - ✓ Canal cliente/servidor SDO (por padrão, apenas um canal é definido).
 - ✓ Valores de índice/subíndice de objeto CANopen para gravar.
 - ✓ Número de bytes para escrever (tamanho do objeto).



CANopen®

CIA405.NMT: Gerenciamento de estado NMT de dispositivos

- O bloco funcional CIA405.NMT permite o controle do estado NMT dos dispositivos CANopen a partir do aplicativo do controlador.
- Ele executa a solicitação de serviço NMT para um dispositivo de destino CANopen para executar a transição de estado NMT solicitada.



canopen®

CIA405.NMT: Gerenciamento de estado NMT de dispositivos

Descrição das variáveis de entrada específicas

Input	Data Type	Description
DEVICE	DEVICE (USINT)	CANopen targeted device node ID. 0 (default value) = all NMT slave devices 1...127 = targeted device node ID
STATE	TRANSITION_STATE. Refer below for details	Requested NMT-state transition.



CIA405.NMT: Gerenciamento de estado NMT de dispositivos

- CIA405.TRANSITION_STATE ENUM
- O tipo enumerado CIA405.TRANSITION_STATE contém os comandos de transição de estado NMT descritos na tabela.

Enumerator	Value (hex)	Description
STOP_REMOTE_NODE	0004	Switch to <i>Stopped</i> state. (transition B)
START_REMOTE_NODE	0005	Switch to <i>Operational</i> state. (transition A)
RESET_NODE	0006	Switch to <i>Reset Application</i> state. Load saved data of the device profiles and switch automatically to <i>Pre-Operational</i> state through <i>Reset Communication</i> . (transition D)
RESET_COMMUNICATION	0007	Switch to <i>Reset Communication</i> state. Load stored data of the communication profile and switch automatically to <i>Pre-Operational</i> state. (transition E)
ENTER_PRE_OPERATIONAL	007F	Switch to <i>Pre-Operational</i> state. (transition C)
ALL_EXCEPT_NMT_AND_SENDER	0800	Not implemented (invalid parameter)

CIA405.GET_STATE:
Obter estado do
dispositivo CANopen



- O bloco de funções CIA405.GET_STATE retorna o estado NMT atual do dispositivo CANopen especificado se o heartbeat ou o node guarding estiverem ativos.

Descrição das variáveis de entrada e saída específicas

Input	Type	Description
DEVICE	DEVICE (USINT)	Node ID of the CANopen device to be checked. 0 (default value) = local device (controller) 1...127 = device node ID



Output	Type	Description
STATE	DEVICE_STATE (Refer below for details)	NMT-state of the CANopen device.

O tipo enumerado CIA405.DEVICE_STATE contém a lista de estados NMT do dispositivo CANopen.

Enumerator	Value	Description
INIT	0	<i>Initialization state</i>
RESET_COMM	1	<i>Reset Communication state</i>
RESET_APP	2	<i>Reset Application state</i>
PRE_OPERATIONAL	3	<i>Pre-Operational state</i>
STOPPED	4	<i>Stopped state</i>
OPERATIONAL	5	<i>Operational state</i>
UNKNOWN	6	Unknown NMT-state. The node guarding or heartbeat is not active for the selected device or the controller is not the heartbeat consumer.
NOT_AVAIL	7	NMT-state not available. The node guarding or heartbeat is active for the selected device but the device does not report its NMT-state correctly before timeout.