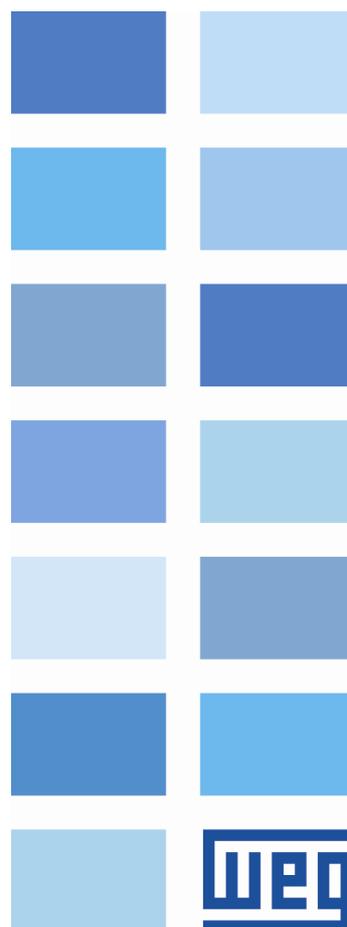


Software

WSCAN

Manual do Usuário





Manual do Usuário

Série: WSCAN V2.0X

Idioma: Português

Data de Publicação: 11/2010

Índice

	0
Parte I Informações Gerais	6
1 Sobre o manual	6
2 O que é o WSCAN	6
3 Abreviações e definições	6
4 Representação numérica	7
5 Documentos	7
6 Principais características WSCAN	7
Parte II Introdução ao protocolo CANopen	7
1 CAN	7
2 Frame de dados	8
3 Frame remoto	8
4 Acesso a rede	8
5 Controle de erros	8
6 CAN e CANopen	9
7 Características da rede CANopen	9
8 Meio físico	9
9 Endereço na rede CANopen	9
10 Acesso aos dados	10
11 Transmissão de dados	10
12 Objetos responsáveis pela comunicação - COBs	10
13 COB-ID	11
14 Arquivo EDS	12
Parte III Software WSCAN	12
1 Executando o software WSCAN	12
2 Janela principal	14
3 Menu principal	14
4 Barra de botões	15
5 Device Tree	16
6 Log Window	16
7 Rede CANopen	17
8 Configurando uma nova rede	17
9 Configurando o mestre	19
Configurando o mestre	19
NMT	20

PRODUTOR SYNC	20
SDO	21
PRODUTOR HEARTBEAT	22
FOLLOW	24
10 Configurando os escravos	25
Configurando os escravos	25
NMT	26
NODE GUARDING	26
PRODUTOR HEARTBEAT	27
EMCY	28
SAVE/RESTORE	31
FOLLOW	32
11 Dicionário de objetos do escravo	33
Dicionário de objetos do escravo	33
Lista de objetos	33
Informações do objeto	34
Alterando a base numérica	35
Alterando o valor de objetos	36
12 Configuração de PDOs do mestre/escravos	36
Configuração de PDOs do mestre/escravos	36
Lista de PDOs	37
Parâmetros de comunicação do PDO	38
Mapeamento de objetos no PDO	40
13 Visualizador/configurador de conexões entre PDOs	41
Visualizador/configurador de conexões entre PDOs	41
Procedimento para efetuar uma conexão entre PDOs	41
Procedimento para efetuar uma conexão entre PDOs.....	41
Selecionando o produtor (TxPDO)	42
Selecionando o consumidor (RxPDO).....	43
Conectando o produtor (TxPDO) ao consumidor (RxPDO).....	45
Conectado objetos de tamanho diferentes.....	46
Listando as conexões	48
14 Configuração da rede	48
Gerar configuração	48
Transmitir configuração	49
Monitoração online	49
15 Interface com a programação ladder	52
Interface com a programação ladder	52
Mapeamento de memória de I/O	53
Palavras de comando e estado da comunicação	53
Palavras de comando e estado da comunicação.....	53
Estado da comunicação CANopen.....	54
Estado do escravo CANopen	55
Controle da comunicação CANopen.....	56
Endereço do escravo destino	56
Bloco ladder (SDO)	56
16 Exemplo de configuração de rede	57
Introdução	57
Configuração mestre (PLC2)	57
Configuração do escravo (SCA-05)	57
PDOs do mestre (PLC2)	57
PDOs do escravo (SCA-05)	58

Conexões PDOs	60
Leitura do status do escravo no ladder do mestre	62
Comando do escravo no ladder do mestre	63
17 Exemplo de configuração follow CANopen	63
Introdução	63
Configuração WSCAN	64
Configuração WLP	67
Parametrização CFW11 e PLC11	67
Índice	68

1 Informações Gerais

1.1 Sobre o manual

Este manual fornece a descrição necessária para a operação do software WSCAN e dos mestres de rede CANopen desenvolvidos pela WEG. Este manual deve ser utilizado em conjunto com manual do software WLP e do cartão mestre de rede CANopen.

Veja também :

O que é o WSCAN	6
Abreviações e definições	6
Representação numérica	7
Documentos	7
Principais características WSCAN	7
CAN	7
Executando o software WSCAN	12
Janela principal	14
Menu principal	14
Barra de botões	15
Device Tree	16
Log window	16
Rede CANopen	17
Configurando uma nova rede	17
Configurando o mestre	19
Configurando os escravos	25
Dicionário de objetos do escravo	33
Configuração de PDOs do mestre/escravos	36
Visualizador/configurador de conexões entre PDOs	41
Gerar configuração	48
Transmitir configuração	49
Monitoração online	49
Interface com a programação ladder	52
Exemplo de configuração de rede	57
Exemplo de configuração follow CANopen	63

1.2 O que é o WSCAN

O WSCAN “Weg Software CANopen Config” é um software para ambiente Windows utilizado para a configuração e programação dos mestres de rede CANopen desenvolvidos pela WEG.

1.3 Abreviações e definições

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

1.4 Representação numérica

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

1.5 Documentos

<i>Documento</i>	<i>Versão</i>	<i>Fonte</i>
CAN Specification	2.0	CiA
CiA DS 301 CANopen Application Layer and Communication Profile	4.02	CiA
CiA DS 302 Framework for CANopen Managers and Programmable CANopen Devices	3.2.1	CiA
CiA DS 405 Interface and Device Profile for IEC 61131-3 Programmable Devices	2.0	CiA
CiA DRP 303-1 Cabling and Connector Pin Assignment	1.1.1	CiA
CiA DSP 306 Electronic Data Sheet Specification for CANopen	1.1	CiA

1.6 Principais características WSCAN

- Montagem da rede através de ambiente gráfico.
- Configuração da rede através de caixas de diálogos.
- Transferência da configuração através da porta serial ou USB do PC.
- Possibilidade de configurar até 25⁽¹⁾ dispositivos escravos da rede CANopen.
- Configuração de até 1600⁽²⁾ pontos digitais de entrada e 1600⁽²⁾ pontos digitais de saída.
- Monitoração on-line do estado da rede e possíveis erros de comunicação.

(1) 8 para PLC2 e 25 para PLC11-01

(2) 512 para PLC2 e 1600 para PLC11-01

2 Introdução ao protocolo CANopen

2.1 CAN

A rede CANopen é uma rede baseada em CAN, ou seja utiliza telegramas CAN para troca de dados na rede.

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

NOTA !

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

Veja também :

- [Frame de dados](#) ⁸
- [Frame remoto](#) ⁸
- [Acesso a rede](#) ⁸
- [Controle de erros](#) ⁸
- [CAN e CANopen](#) ⁹
- [Características da rede CANopen](#) ⁹
- [Meio físico](#) ⁹
- [Endereço na rede CANopen](#) ⁹
- [Acesso aos dados](#) ¹⁰
- [Transmissão de dados](#) ¹⁰
- [Objetos responsáveis pela comunicação - COBs](#) ¹⁰
- [COB-ID](#) ¹¹
- [Arquivo EDS](#) ¹²

2.2 Frame de dados

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

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

NOTA !

A especificação CAN 2.0 define dois tipos de frames de dados: *standard* (11bits) e *extended* (29 bits). Para o protocolo CANopen do mestre WEG de rede, somente frames *standard* são aceitos (11 bits).

2.3 Frame remoto

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

2.4 Acesso a rede

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

2.5 Controle de erros

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

Um dispositivo da rede CAN possui contadores internos que são incrementados toda vez que um erro de transmissão ou recepção for detectado, e decrementado quando um telegrama é enviado ou recebido com sucesso. Caso ocorra uma quantidade considerável de erros, o dispositivo pode ser levado para os

seguintes estados:

- Warning: quando esse contador passa de um determinado limite, o dispositivo entra no estado de *warning*, significando a ocorrência de uma elevada taxa de erros.
- Error Passive: quando este valor ultrapassa um limite maior, ele entra no estado de *error passive*, onde ele para de atuar na rede ao detectar que um outro dispositivo enviou um telegrama com erro.
- Bus Off : por último, temos o estado de *bus off*, no qual o dispositivo não irá mais enviar ou receber telegramas.

2.6 CAN e CANopen

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

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

2.7 Características da rede CANopen

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

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

2.8 Meio físico

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

O mestre de rede CANopen utiliza um circuito de interface com a rede isolado com alimentação externa. O componente responsável pela transmissão e recepção de sinais é denominado transceiver, que obedece ao especificado pela ISO 11898.

2.9 Endereço na rede CANopen

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

Para o mestre de rede CANopen WEG, será possível no máximo 8 escravos sendo endereçados de 1 até 63.

2.10 Acesso aos dados

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

Uma descrição mais detalhada de como o dicionário de objetos está estruturado é fornecida no anexo I.

2.11 Transmissão de dados

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

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

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

2.12 Objetos responsáveis pela comunicação - COBs

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

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

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

2.13 COB-ID

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

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

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

A seguir é apresentada uma tabela com os valores padrão para os diferentes objetos de comunicação disponíveis nos dispositivos. É necessário observar que o valor padrão do objeto depende do endereço do escravo, com exceção dos COB-IDs para NMT e SYNC, que são comuns para todos os elementos da rede. Estes valores também podem ser alterados durante a etapa de configuração do dispositivo.

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

2.14 Arquivo EDS

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

3 Software WSCAN

3.1 Executando o software WSCAN

O software WSCAN pode ser executado de duas maneiras:

1ª - Através do software WLP pelo menu ferramentas opção CANopen ou pelas teclas de atalho Shift+F11.

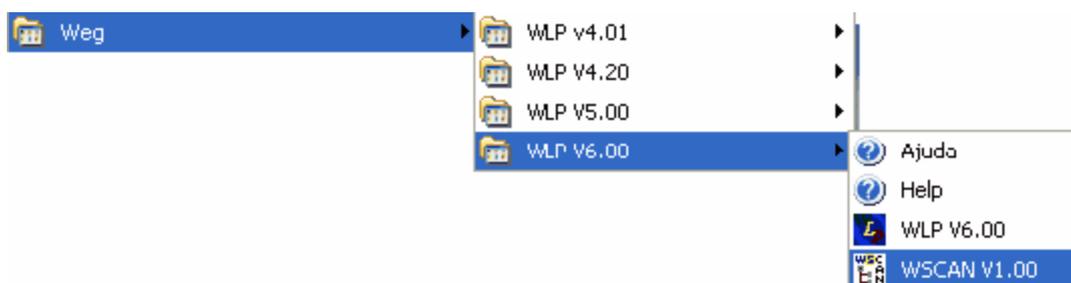
Ferramentas	
Tabela de Parâmetros	F10
Fieldbus	F11
CANopen	Shift+F11

NOTA!

A função CANopen mestre está disponível apenas para o cartão PLC2 versão 1.30 ou superior.

Ao executar o software WSCAN a partir do WLP o mesmo criará/abrirá um arquivo de configuração da rede CANopen com o mesmo nome do projeto corrente do WLP, este arquivo estará armazenado dentro da pasta do projeto em questão.

2ª - Através do atalho criado no menu iniciar do Windows.



Veja também :

[Janela principal](#) ¹⁴

[Menu principal](#) ¹⁴

[Barra de botões](#) ¹⁵

[Device Tree](#) ¹⁶

[Log window](#) ¹⁶

[Rede CANopen](#) ¹⁷

[Configurando uma nova rede](#) ¹⁷

[Configurando o mestre](#) ¹⁹

[Configurando os escravos](#) ²⁵

[Dicionário de objetos do escravo](#) ³³

[Configuração de PDOs do mestre/escravos](#) ³⁶

[Visualizador/configurador de conexões entre PDOs](#) ⁴¹

[Gerar configuração](#) ⁴⁸

[Transmitir configuração](#) ⁴⁹

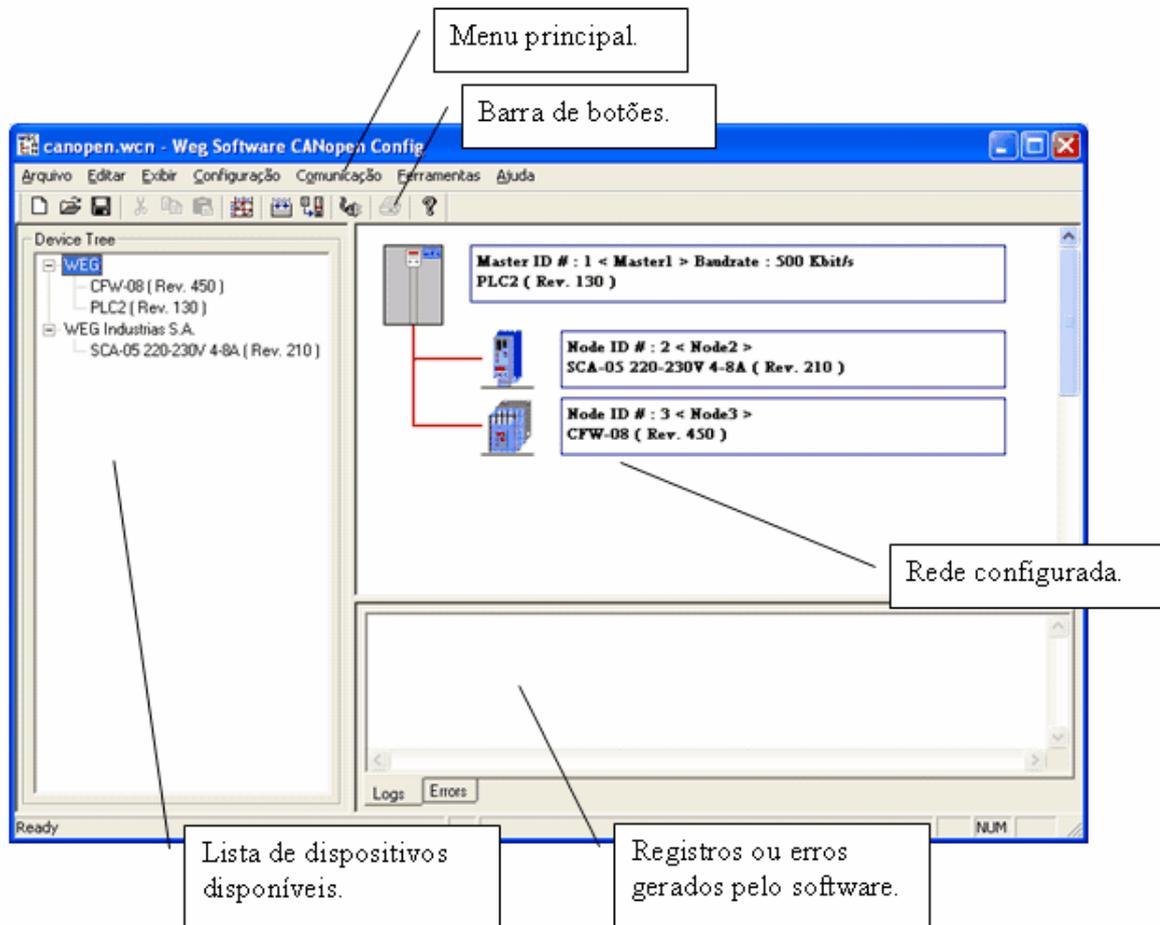
[Monitoração online](#) ⁴⁹

[Interface com a programação ladder](#) ⁵²

[Exemplo de configuração de rede](#) ⁵⁷

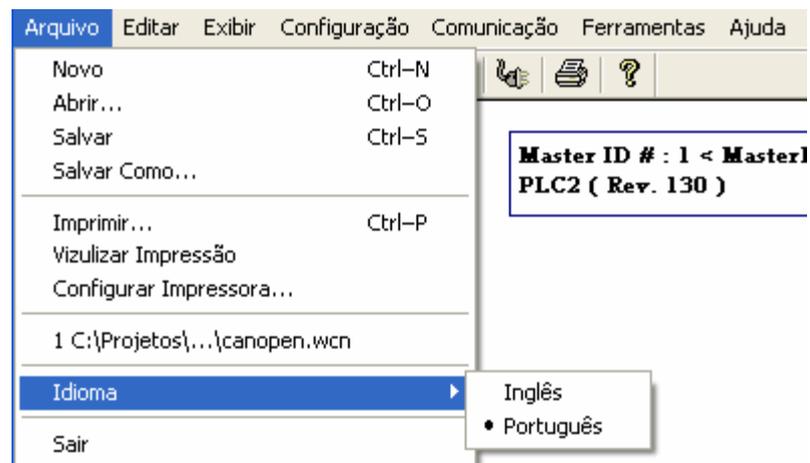
3.2 Janela principal

A janela principal do software WSCAN tem as seguintes características:



3.3 Menu principal

Menu arquivo:



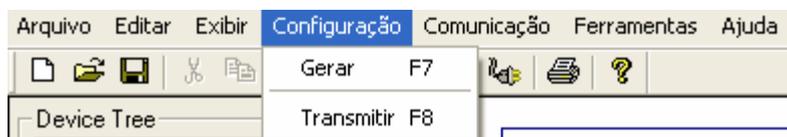
Menu editar:



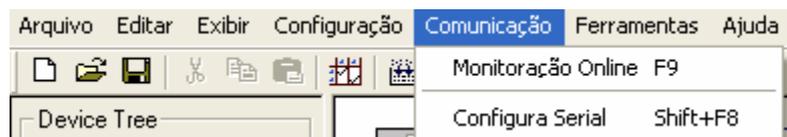
Menu exibir:



Menu configuração:



Menu comunicação:



Menu ferramentas:



Menu ajuda:



3.4 Barra de botões

É composta dos seguintes botões :

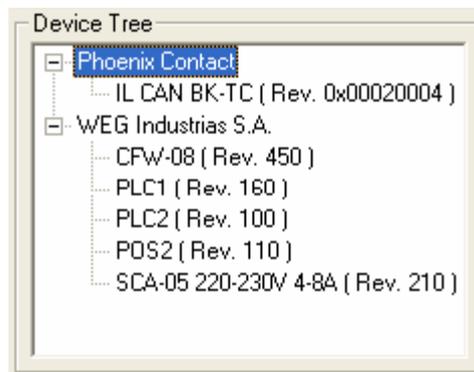
- Novo arquivo de configuração CANopen.
- Abre arquivo de configuração CANopen.

-  - Salva arquivo de configuração CANopen.
-  - Recortar.
-  - Copiar.
-  - Colar.
-  - Visualizador/configurador de conexões PDOs.
-  - Gerar configuração.
-  - Transmitir configuração.
-  - Monitoração on-line.
-  - Imprime.
-  - Sobre o WSCAN

3.5 Device Tree

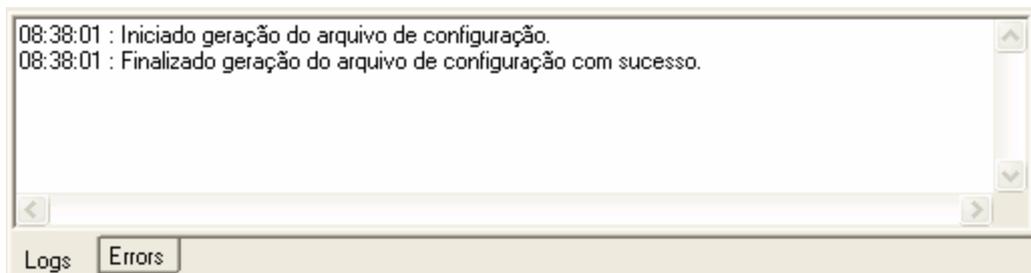
Lista dos dispositivos disponíveis para serem utilizados como escravos da rede CANopen. Essa lista é gerada a partir do conteúdo da pasta “EDS” contida no local de instalação do WSCAN que é o mesmo do WLP, normalmente “C:\Weg\WLP VX.YZ” onde X.YZ é a versão do WLP.

Para adicionar um dispositivo a essa lista basta através do “Windows” copiar o arquivo “EDS - Eletronic Data Sheet” fornecido pelo fabricante para a pasta “EDS” citada anteriormente. É necessário reiniciar o software para que os arquivos adicionados sejam incluídos nesta lista.



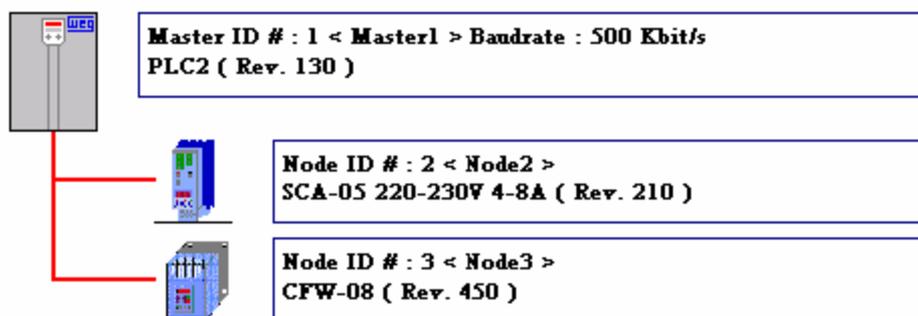
3.6 Log Window

Lista de registros ou erros gerados pelo software. Durante a análise dos arquivos EDS e geração de configuração poderão existir registros ou erros que serão mostrados nessa janela.

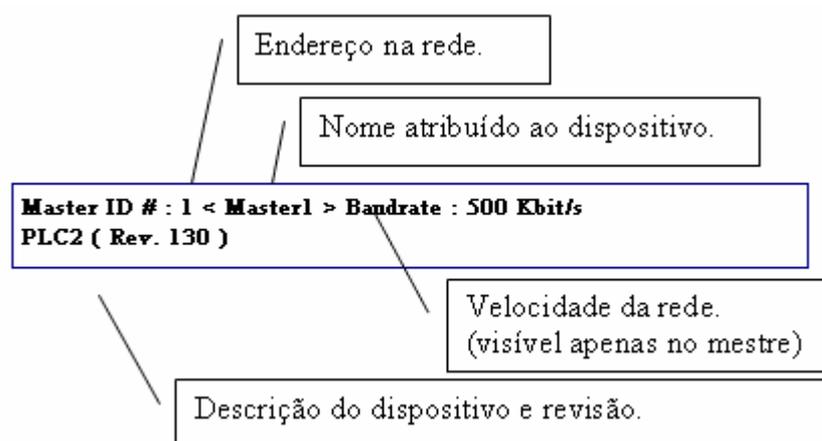


3.7 Rede CANopen

A estrutura da rede sempre será mostrada graficamente no software WSCAN. Nessa representação existirão as principais informações relativas ao mestre e aos escravos.



As informações dos dispositivos estão dispostas da seguinte maneira :



3.8 Configurando uma nova rede

Ao iniciar um novo projeto de rede CANopen nenhum dispositivo estará configurado, inclusive o mestre. Devemos então executar os seguintes procedimentos:

1º - Definir o mestre clicando com o mouse sobre a figura do mestre e selecionando a opção "Propriedades".

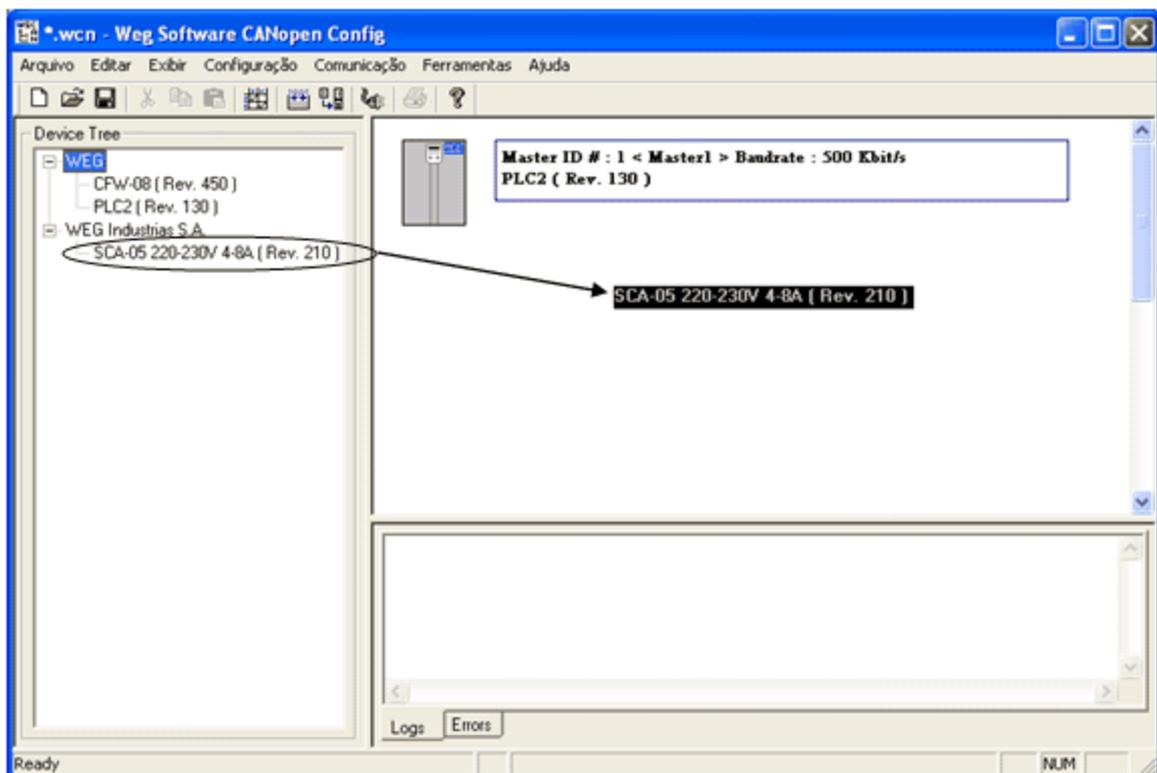


A seguinte caixa de dialogo aparecerá:



Nesta caixa serão definidas as propriedades básicas do mestre, entre elas o próprio dispositivo mestre (neste caso “PLC2”), o endereço do mestre e a taxa de transmissão da rede “Baudrate”.

2º - Adicionar dispositivos escravos clicando com o mouse na lista de dispositivos e arrastando o mesmo com o botão do mouse pressionado para a área da estrutura da rede, soltando o botão do mouse na seqüência.



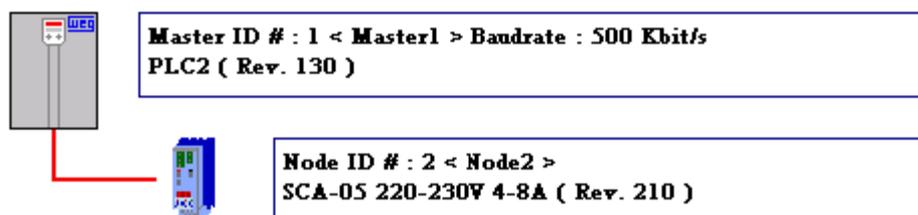
Após soltar o botão do mouse a seguinte caixa de dialogo aparecerá:



Fig. 18 - Propriedades do escravo.

Nesta caixa será definida as propriedades básicas do escravo entre elas o endereço do escravo. Uma vez definido o endereço do escravo e pressionado o botão "OK", esse endereço não poderá ser mais alterado, então, em alguns casos que deseja-se alterar o endereço do escravo e necessário removê-lo e então adicionar o mesmo novamente.

Após esses procedimentos uma rede com um dispositivo escravo estará configurada e será representada da seguinte maneira:



3.9 Configurando o mestre

3.9.1 Configurando o mestre

Para ter acesso a configuração do mestre clique com o mouse sobre a figura do mestre e selecione a opção "Configuração". Após uma caixa de dialogo aparecerá com as opções citadas nos ítems a seguir.

Veja também :

[NMT](#) ^[20]

[PRODUTOR SYNC](#) ^[20]

[SDO](#) ^[21]

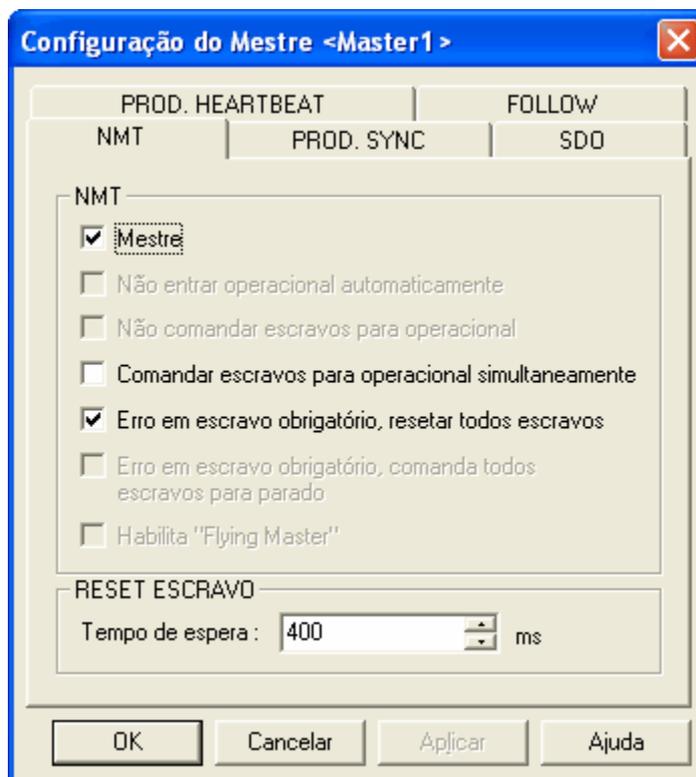
[PRODUTOR HEARTBEAT](#) ^[22]

[FOLLOW](#) ^[24]

3.9.2 NMT

Propriedades de controle e gerenciamento da rede do mestre CANopen.

Define as propriedades básicas do mestre durante a inicialização e gerenciamento da rede:



- Mestre: se esta opção for desabilitada o dispositivo se comportará como um escravo da rede, não sendo possível configurar os outros escravos e não partindo a rede.

- Comandar escravos para modo operacional simultaneamente: permite selecionar se o mestre deve comandar o escravo para operacional após a inicialização de cada escravo, ou então comandar todos os escravos para o modo operacional simultaneamente depois de concluída toda a inicialização da rede.

- Erro em escravo obrigatório, resetar todos os escravos: permite programar se, caso seja detectado um erro de comunicação com algum escravo que foi programado como obrigatório, o mestre deve reiniciar todos os escravos da rede, ou somente o escravo no qual foi detectado o erro. A programação se o escravo é obrigatório ou não é feita nas configurações de cada escravo.

RESET ESCRAVO :

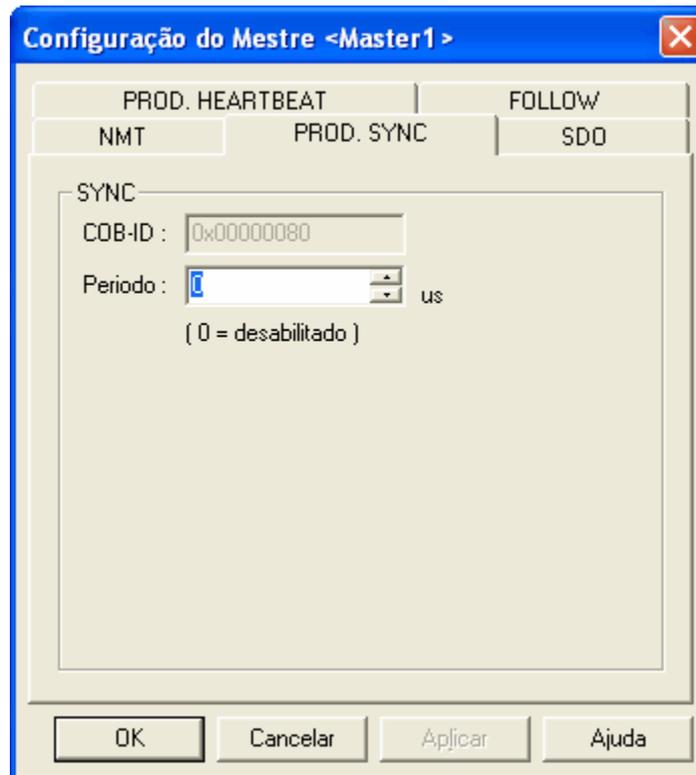
- Tempo de espera ⁽¹⁾₂₀ : tempo de espera após o comando de reset do escravo pelo mestre.

⁽¹⁾ Somente disponível para PLC11-01.

3.9.3 PRODUTOR SYNC

Habilita serviço de sincronismo do mestre CANopen.

Este serviço é utilizado para sincronizar a transmissão de dados do serviço PDO, garantindo que informações sejam produzidas ou consumidas no mesmo instante em dispositivos distintos.



- COB-ID: indica qual o identificador do telegrama SYNC. Alterações neste identificador não são permitidas para o mestre, e, portanto deve-se utilizado o valor padrão para o telegrama SYNC na rede CANopen.

- Período: permite programar qual o período para transmissão do telegrama SYNC pelo mestre da rede. Se o período de transmissão for desabilitado, o mestre irá se comportar como consumidor SYNC.

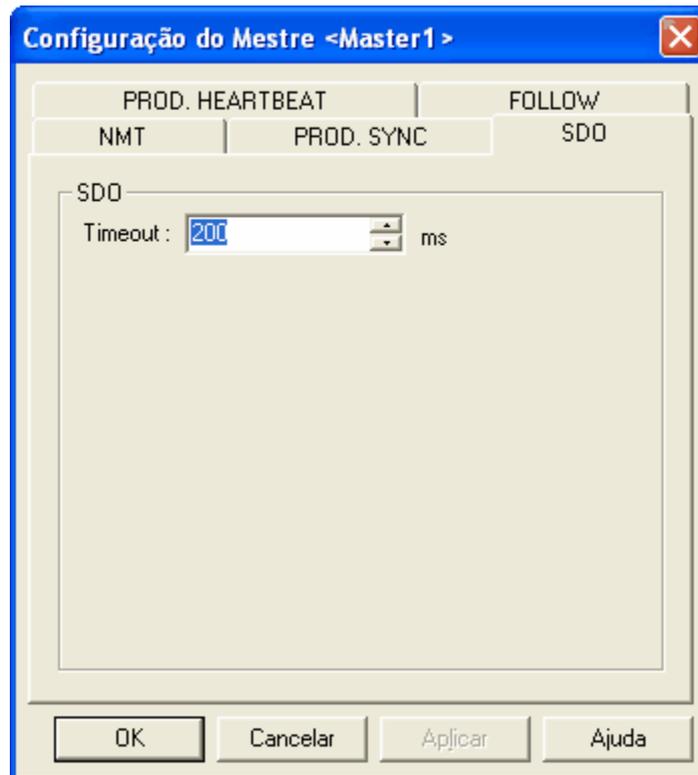
NOTA!

Períodos muito pequenos de transmissão de SYNC irão gerar sobrecarga de telegramas na rede, ocasionando falhas de comunicação dos diversos serviços.

3.9.4 SDO

Propriedades do servidor SDO do mestre CANopen.

Configura propriedades do servidor SDO que é utilizado pelo bloco SDO do ladder do WLP e pela comunicação com os escravos CANopen através da comunicação modbus do mestre de rede CANopen.

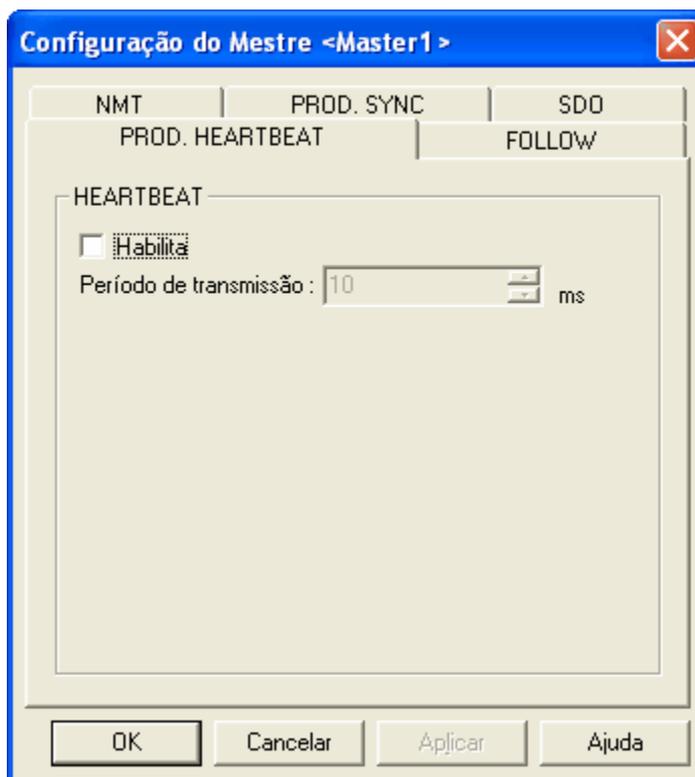


- Timeout: permite programar o tempo que o mestre da rede deverá aguardar por uma resposta, toda vez que o cliente SDO do mestre fizer uma requisição para o servidor SDO de algum escravo da rede. Caso a resposta não seja recebida decorrido o tempo programado, será considerado erro na requisição feita pelo cliente.

3.9.5 PRODUTOR HEARTBEAT

Habilita serviço de produtor heartbeat do mestre CANopen.

Utilizado para monitoração da comunicação do mestre com a rede CANopen por algum escravo que tenha um consumidor heartbeat configurado.



-Habilita: permite habilitar ou desabilitar a produção de mensagens do tipo heartbeat pelo mestre da rede.

-Período de transmissão: caso esteja habilitado, este será o período de transmissão das mensagens heartbeat. Uma vez programado, outros dispositivos da rede com a função de consumidores heartbeat poderão ser programados para monitorar estas mensagens e detectar erros de comunicação.

3.9.6 FOLLOW

Habilita serviço de produtor ou consumidor Follow via CANopen.

Utilizado para fazer sincronismo de velocidade ou posição via bloco Follow do WLP [\(1\)](#) ^[24].



- Desabilita: desabilita função produtor ou consumidor follow.
- Habilita Produtor (Eixo Real): habilita mestre para produzir dados follow através do TPDO1.
- Habilita Produtor (Eixo Virtual): habilita mestre para produzir dados follow através do TPDO1 [\(2\)](#) ^[24].
- Habilita Consumidor: habilita mestre para consumir dados follow através do RXPDO1.
- COB-ID (hexa): identificador do TPDO1 ou RXPDO1 na rede CANopen, para que os dados follow produzidos em um equipamentos sejam consumidos em outro o valor desse parâmetro deve ser o mesmo nos dois equipamentos.
- Fonte de velocidade: seleciona qual velocidade será transmitida pelo produtor follow (velocidade real ou velocidade de referência).

Maiores detalhes ver [exemplo de programação do follow via CANopen](#) ^[63].

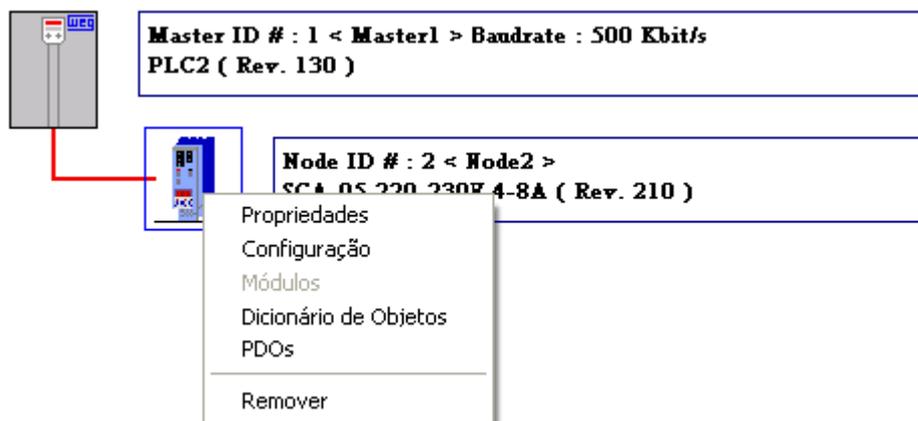
(1) Somente disponível para PLC11-01 e PC11-02 com versão ≥ 1.20 .

(2) Somente disponível para PLC11-01 e PC11-02 com versão ≥ 1.30 .

3.10 Configurando os escravos

3.10.1 Configurando os escravos

Para ter acesso a configuração do escravo clique com o mouse sobre a figura do escravo e selecione a opção “Configuração”. Após uma caixa de diálogo aparecerá com as opções citadas nos ítems a seguir.



Veja também :

[NMT](#) ^[26]

[NODE GUARDING](#) ^[26]

[PRODUTOR HEARTBEAT](#) ^[27]

[EMCY](#) ^[28]

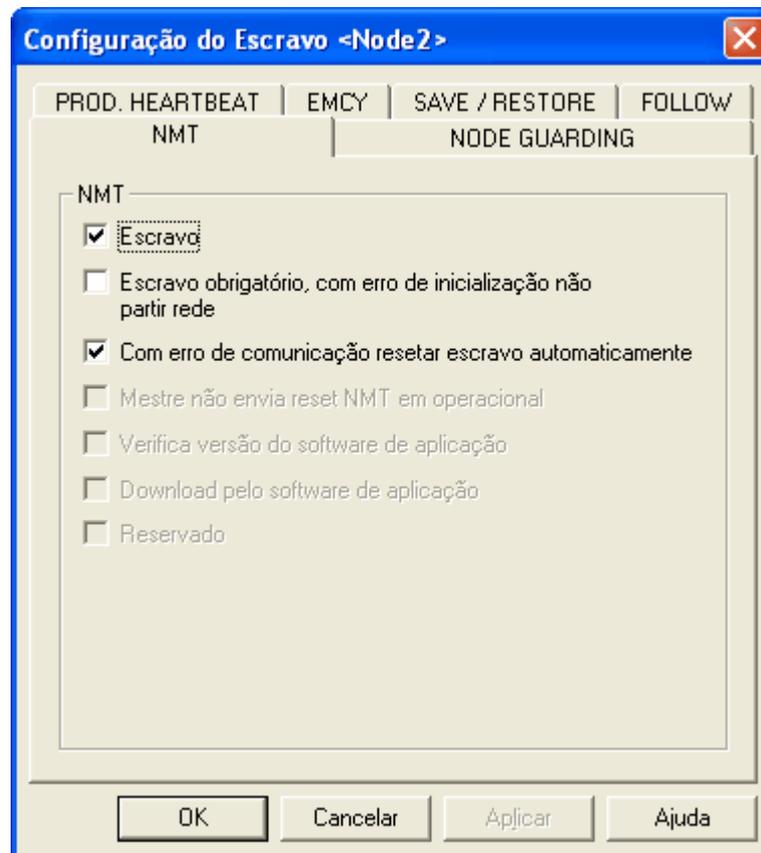
[SAVE/RESTORE](#) ^[31]

[FOLLOW](#) ^[32]

3.10.2 NMT

Propriedades de controle e operação do escravo.

Define as propriedades básicas do escravo durante a inicialização e operação da rede:



- Escravo: se esta opção for desabilitada, o dispositivo não será configurado e nem inicializado pelo mestre da rede.

- Escravo obrigatório, com erro de inicialização não partir rede: permite programar se o escravo é obrigatório ou não para o mestre da rede. Se for programado como obrigatório, caso ocorra erro na sua inicialização, o mestre não tentará inicializar os demais escravos na rede, até que seja possível concluir a inicialização deste escravo.

- Com erro resetar escravo automaticamente: caso seja detectado erro de comunicação com este escravo, indica se o mestre deve resetar o escravo e inicializá-lo novamente, ou então deve apenas fazer a indicação de erro nas palavras de estado, e esperar que o usuário faça este reset e inicialização (que pode ser efetuado utilizando as palavras de controle da comunicação).

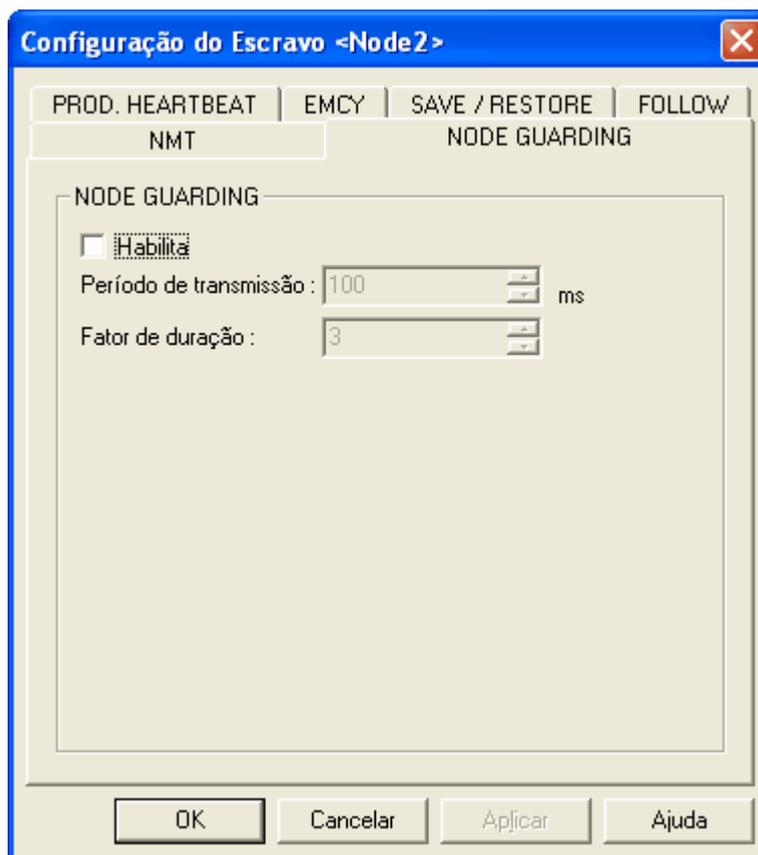
NOTA!

- Com erro resetar escravo automaticamente : é necessário habilitar algum serviço de detecção de falha de comunicação (Node Guarding ou Heartbeat)

3.10.3 NODE GUARDING

Habilita serviço de node guarding do escravo.

Utilizado para monitoração da comunicação do escravo com a rede CANopen, essa monitoração é feita tanto pelo mestre quanto pelo escravo.



- Habilita: permite habilitar ou desabilitar o serviço de controle de erros Node Guarding para este escravo.
- Período de transmissão: permite programar o período no qual o mestre deve enviar mensagens para o escravo, e receber a resposta do escravo.
- Fator de duração: permite programar o número de períodos que o escravo ou o mestre devem esperar até considerar falha na comunicação, no caso de falha na comunicação entre os dispositivos.

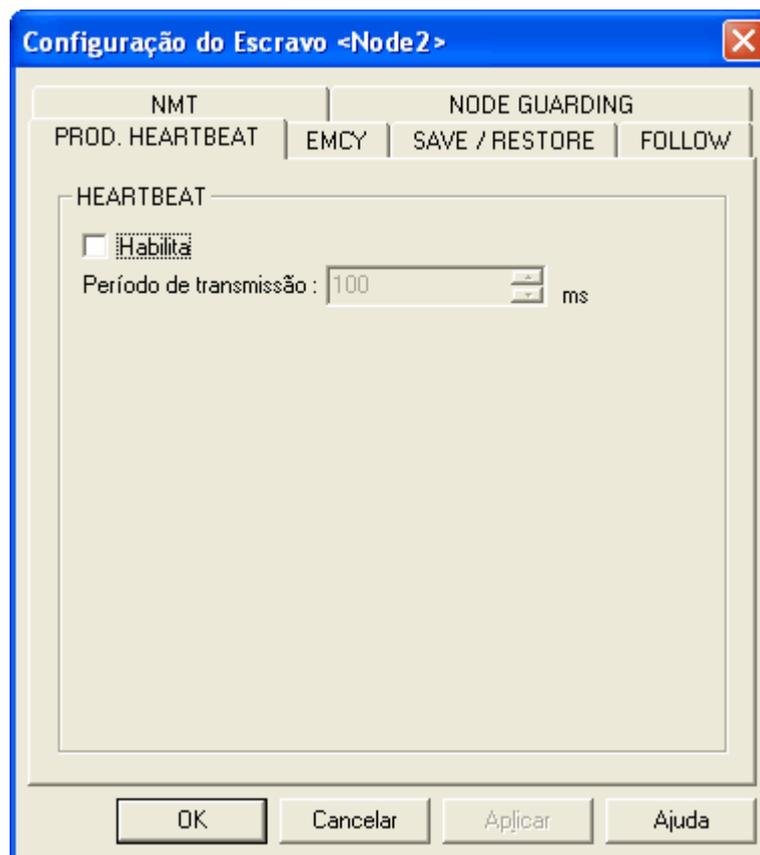
NOTA!

- Este serviço somente estará disponível se o escravo possuir os objetos necessários para a sua execução descritos no arquivo EDS (100Ch e 100Dh).
- Somente um dos serviços - Node Guarding ou Heartbeat pode ser habilitado por escravo.

3.10.4 PRODUTOR HEARTBEAT

Habilita serviço de produtor heartbeat do escravo.

Utilizado para monitoração da comunicação do escravo com a rede CANopen, essa monitoração é feita somente pelo mestre.



- Habilita: permite habilitar ou desabilitar o serviço de controle de erros Heartbeat para este escravo.
- Período de transmissão: permite programar o período no qual o escravo deve enviar mensagens heartbeat para a rede. Estas mensagens serão monitoradas pelo mestre, possibilitando a detecção de erros de comunicação caso este detecte a interrupção no recebimento destas mensagens.

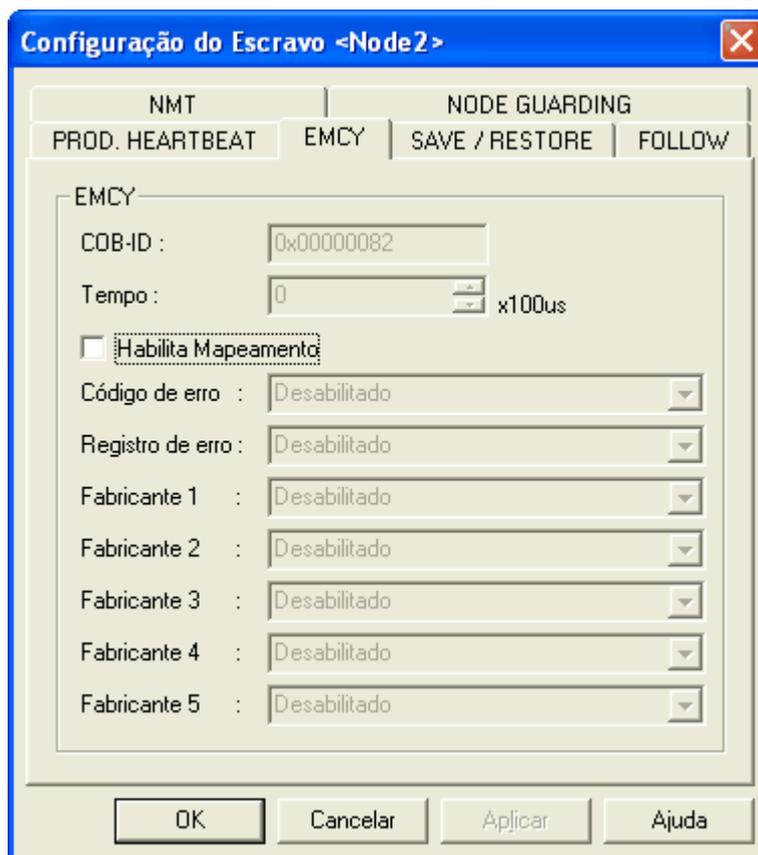
NOTA!

- Este serviço somente estará disponível se o escravo possuir os objetos necessários para a sua execução descritos no arquivo EDS (1017h).
- Somente um dos serviços - Node Guarding ou Heartbeat pode ser habilitado por escravo.

3.10.5 EMCY

Habilita serviço de EMCY do escravo.

Utilizado para sinalizar a ocorrência de um erro no dispositivo, essas mensagens de erro serão armazenadas em até 8 words de status que serão acessadas no ladder do WLP do mestre de rede CANopen.



- Tempo: permite programar no escravo, após enviar uma mensagens EMCY, o tempo de deve ser decorrido para que uma nova mensagen EMCY possa ser transmitida para a rede.
- Habilita mapeamento: permite habilitar ou desabilitar a mapeamento pelo mestre, de mensagens EMCY enviadas pelo escravo.
- Word de status 1 ... 7: permite programar em qual marcador do mestre as informações transmitidas na mensagem EMCY devem ser armazenadas, sendo possível identificar o tipo de erro ocorrido no escravo. Cada word representa um dos 7 campos existentes em uma mensagem EMCY:
 - Word 1 : código do erro (CiA)
 - Word 2 : registro de erro (objeto 1001h)
 - Word 3 a 7 :Campos definidos pelo fabricantes 1 até 5.

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains Voltage
32xx	Voltage inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient Temperature
42xx	Device Temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
62xx	User Software
63xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	recovered from bus off
8150	Transmit COB-ID collision
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

Código de erros (CiA) Word 1 EMCY.

Bit	M/O	Meaning
0	M	generic error
1	O	current
2	O	voltage
3	O	temperature
4	O	communication error (overrun, error state)
5	O	device profile specific
6	O	Reserved (always 0)
7	O	manufacturer specific

Registro de Erro (Objeto 1001h) Word 2 EMCY.

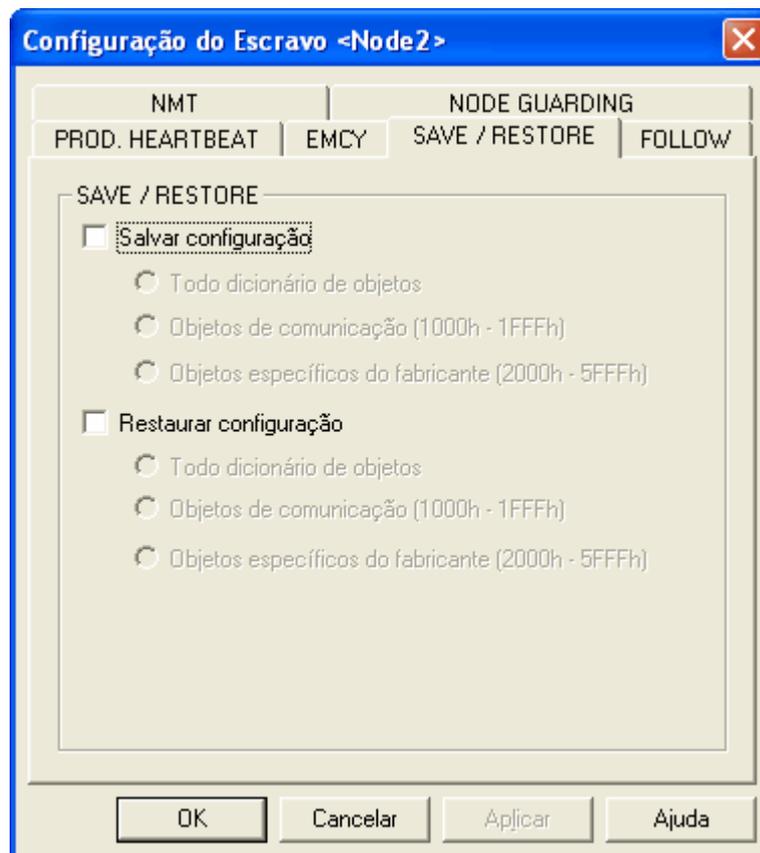
NOTA!

- EMCY : Este serviço somente estará disponível se o escravo possuir os objetos necessários para a sua execução descritos no arquivo EDS (1014h).

- Tempo de inibição: Este serviço somente estará disponível se o escravo possuir os objetos necessários para a sua execução descritos no arquivo EDS (1015h).

3.10.6 SAVE/RESTORE

Utilizado para salvar ou restaurar os valores programados no dicionário de objetos do escravo.



- Salvar configuração : ao final da configuração do escravo o mestre solicita ao escravo que as configurações sejam salvas em uma memória local não volátil.
- Restaurar configuração : antes de realizar a configuração do escravo o mestre solicita que as configurações sejam restauradas para o valor padrão.

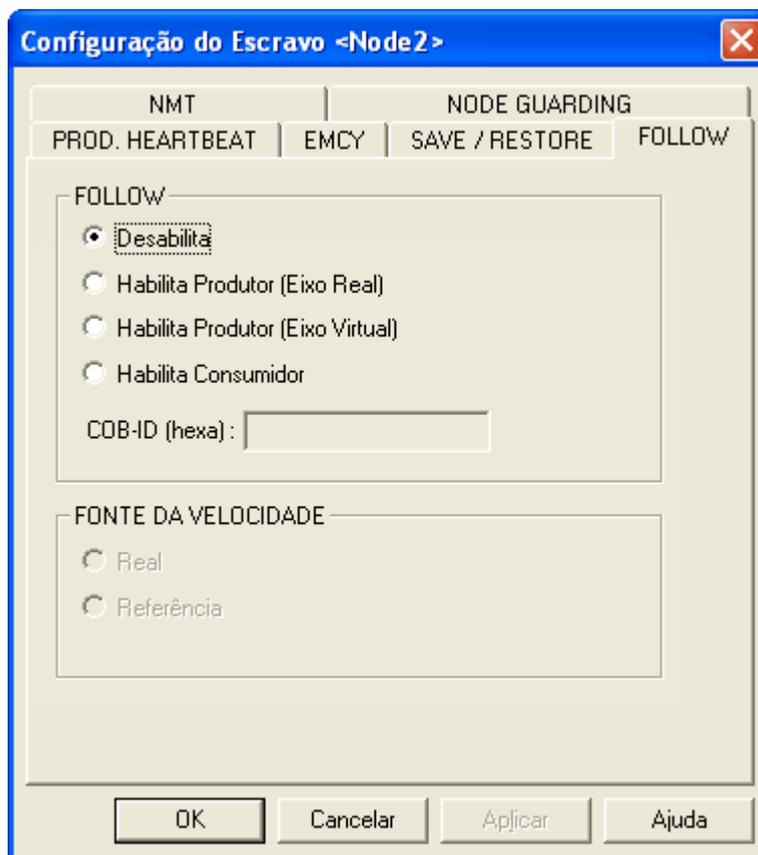
NOTA!

- Este serviço somente estará disponível se o escravo possuir os objetos necessários para a sua execução descritos no arquivo EDS (1010h ou 1011h).
- A utilização desses serviços consome um tempo relativamente grande nos escravos, logo poderão ocorrer erros de comunicação por "timeout", para evitar a ocorrência desses erros é necessário aumentar o valor do "timeout" dos SDOs na configuração do mestre.

3.10.7 FOLLOW

Habilita serviço de produtor ou consumidor Follow via CANopen.

Utilizado para fazer sincronismo de velocidade ou posição via bloco Follow do WLP [\(1\)](#) ³².



- Desabilita: desabilita função produtor ou consumidor follow.
- Habilita Produtor (Eixo Real): habilita mestre para produzir dados follow através do TPDO1.
- Habilita Produtor (Eixo Virtual): habilita mestre para produzir dados follow através do TPDO1 [\(2\)](#) ³².
- Habilita Consumidor: habilita mestre para consumir dados follow através do RXPDO1.
- COB-ID (hexa): identificador do TPDO1 ou RXPDO1 na rede CANopen, para que os dados follow produzidos em um equipamentos sejam consumidos em outro o valor desse parâmetro deve ser o mesmo nos dois equipamentos.
- Fonte de velocidade: seleciona qual velocidade será transmitida pelo produtor follow (velocidade real ou velocidade de referência).

Maiores detalhes ver [exemplo de programação do follow via CANopen](#) ⁶³.

(1) Somente disponível para PLC11-01 e PC11-02 com versão \geq 1.20.

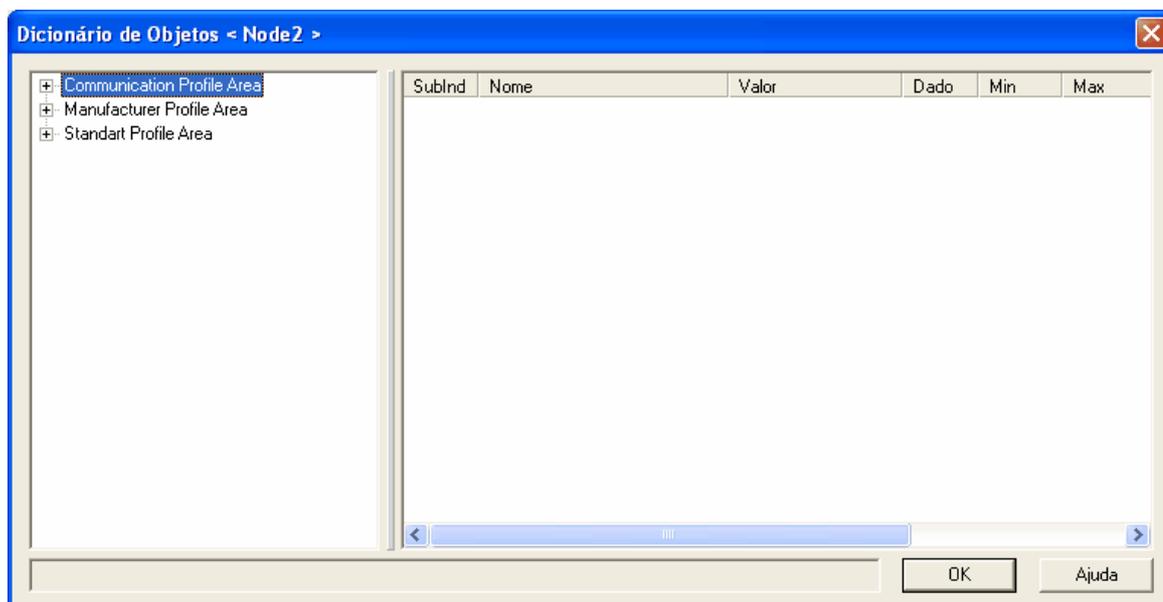
(2) Somente disponível para PLC11-01 e PC11-02 com versão \geq 1.30.

3.11 Dicionário de objetos do escravo

3.11.1 Dicionário de objetos do escravo

Como foi citado anteriormente todos os objetos acessíveis via rede são declarados em um dicionário de objetos dentro do arquivo “EDS”, então pode ser interessante ter acesso ao conteúdo desses objetos em determinadas situações.

Para ter acesso ao dicionário de objetos do escravo clique com o mouse sobre a figura do escravo e selecione a opção “Dicionário de Objetos”. Será mostrada a seguinte caixa de diálogo:



Veja também :

[Lista de objetos](#) ^[33]

[Informações do objeto](#) ^[34]

[Alterando a base numérica](#) ^[35]

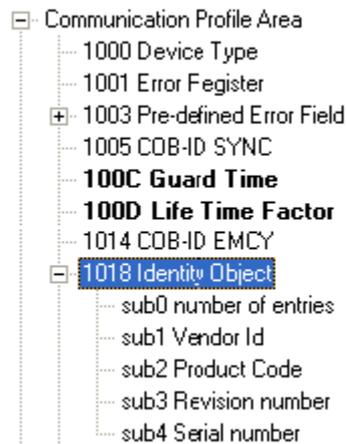
[Alterando o valor de objetos](#) ^[36]

3.11.2 Lista de objetos

O dicionário de objetos está dividido em três grupos, “Communication Profile Area” onde estão os objetos de comunicação, “Manufacturer Profile Area” onde estão objetos definidos pelo fabricante do dispositivo e “Standart Profile Area” onde estão os objetos padrão para o tipo do dispositivo.

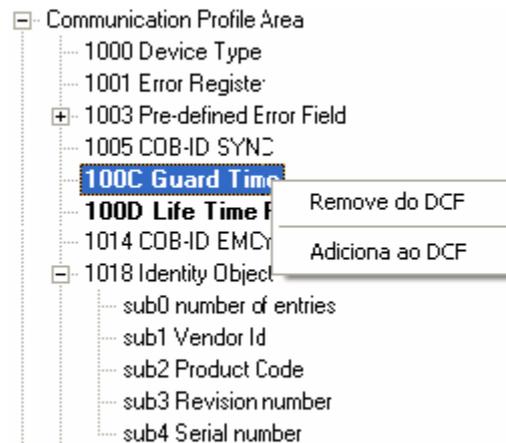


Os objetos sempre são identificados pelo seu índice (“Index”) e sub-índice (“Sub-index”) quando existir, sendo os dois números representados em hexadecimal. Por exemplo: O objeto “Device Type” tem o índice = 1000h e sub-índice = 0h por não ter nenhum, o objeto “Vendor Id” tem o índice = 1018h e sub-índice = 1h.



Nesta lista os objetos destacados em negrito são os que pertencem ao “Arquivo DCF” que é uma lista específica que será utilizada pelo mestre da rede CANopen durante a inicialização do escravo. Dizer que um objeto pertence ao arquivo DCF significa que o valor do mesmo será escrito pelo mestre no escravo durante a inicialização.

É possível adicionar ou remover objetos do arquivo DCF, para isto clique com o botão direito do mouse sobre o objeto que o seguinte menu aparecerá:



Só efetue operações de adicionar/remover objetos do arquivo DCF ao conhecer especificamente a função do objeto em questão.

3.11.3 Informações do objeto

Ao selecionar um objeto da lista de objetos as informações do mesmo serão mostradas na janela ao seu lado. Quando for selecionado um objeto que possua sub-objetos será mostrada a lista de todos os sub-objetos.

SubInd	Nome	Valor	Dado	Mir	Max	Acesso	Padrão	PDO
1	Vendor Id	0x00000123	UNSIGNED32	0xC0000000	0xFFFFFFFF	ro	0x00000123	no

São mostradas as seguintes informações do objeto :

- SubInd: Sub-índice do objeto.
- Nome: Nome do objeto.
- Valor: Valor atual do objeto.

- Dado: Tipo numérico do objeto.
- Min: Valor mínimo do objeto.
- Max: Valor máximo do objeto.
- Acesso: Tipo permitido do acesso ao objeto.
- Padrão: Valor padrão do objeto.
- PDO: Mapeável em PDOs.

Tipos de dados:

- BOOLEAN
- INTEGER8
- INTEGER16
- INTEGER32
- UNSIGNED8
- UNSIGNED16
- UNSIGNED32
- REAL32
- VISIBLE_STRING
- OCTET_STRING
- UNICODE_STRING
- TIME_OF_DAY
- TIME_DIFFERENCE
- DOMAIN
- INTEGER24
- REAL64
- INTEGER40
- INTEGER48
- INTEGER56
- INTEGER64
- UNSIGNED24
- UNSIGNED40
- UNSIGNED48
- UNSIGNED56
- UNSIGNED64

Tipos de acesso :

- rw : leitura e escrita
- wo : somente escrita
- ro : somente leitura
- const : constante somente leitura

3.11.4 Alterando a base numérica

É possível mudar a base numérica das informações do objeto, para isso clique com o botão direito do mouse sobre as informações e escolha a opção “Formato” e então escolha qual a base numérica desejada.

SubInd	Nome	Valor	Dado	Min	Max	Acesso	Padrão	PDO
1	Vendor Id	0x00000123	UNSIGNED32	0x00000000	0xFFFFFFFF	ro	0x00000123	no

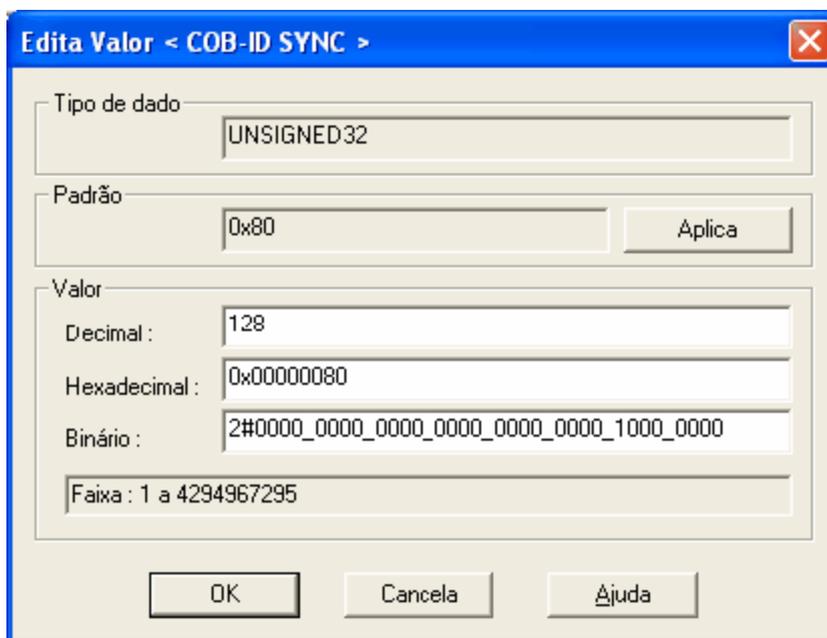
Edita Valor

- Formato
 - Decimal
 - ✓ Hexadecimal
 - Binário

3.11.5 Alterando o valor de objetos

É possível alterar o valor dos objetos com tipo de acesso “rw” e “wo”. Ao alterar o valor desse objeto o mesmo será adicionado ao arquivo DCF.

Para alterar o valor do objeto, dê um clique duplo do mouse sobre o objeto, ou clique com o botão direito do mouse sobre as informações e escolha a opção “Edita Valor” do objeto escolhido. Será mostrada a seguinte caixa de diálogo:



A caixa de diálogo "Edita Valor < COB-ID SYNC >" possui os seguintes campos e botões:

- Tipo de dado: UNSIGNED32
- Padrão: 0x80 (com botão "Aplica" ao lado)
- Valor:
 - Decimal: 128
 - Hexadecimal: 0x00000080
 - Binário: 2#0000_0000_0000_0000_0000_1000_0000
 - Faixa: 1 a 4294967295
- Botões: OK, Cancela, Ajuda

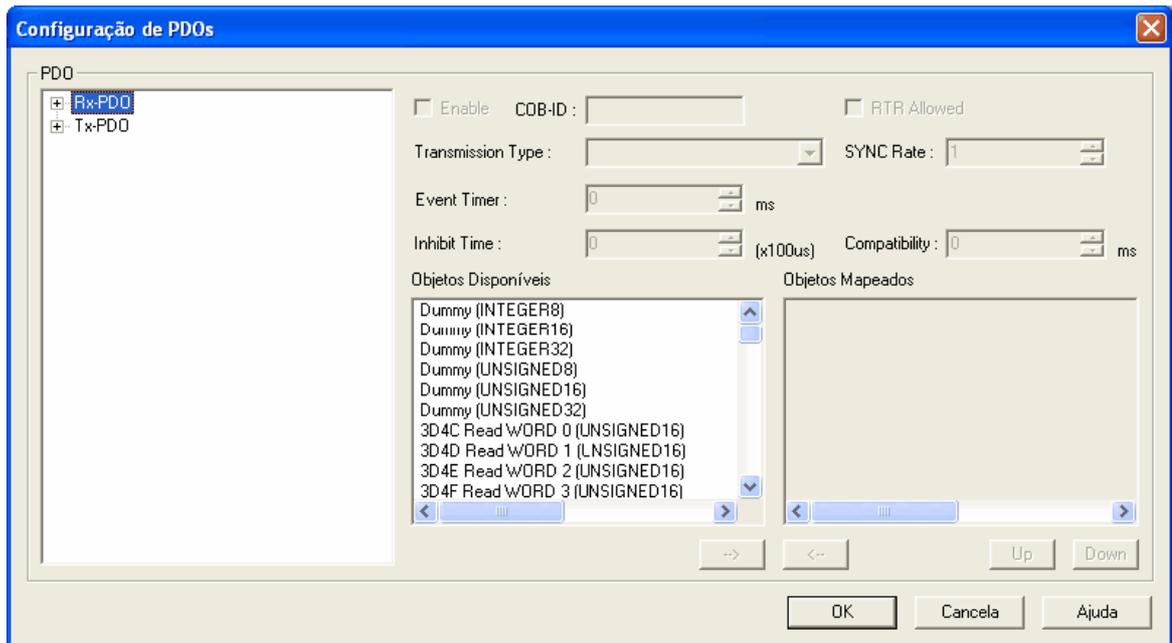
Nesse dialogo é possível alterar o valor do objeto em três bases numéricas, sendo que a conversão é automática, e aplicar o valor padrão. Ao aplicar o padrão ao objeto o mesmo será removido do arquivo DCF>

3.12 Configuração de PDOs do mestre/escravos

3.12.1 Configuração de PDOs do mestre/escravos

Conforme citado anteriormente, os PDOs usualmente são utilizados para transmissão e recepção de dados utilizados durante a operação do dispositivo, e por isso é necessário configurá-los.

Para ter acesso a configuração dos PDO do mestre/escravo clique com o mouse sobre a figura do mestre/escravo e selecione a opção “PDOs”. Após a seguinte caixa de dialogo aparecerá.



Veja também :

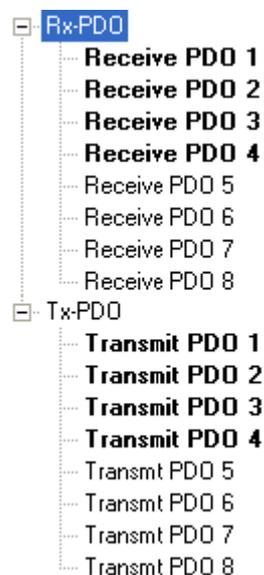
[Lista de PDOs](#) ³⁷

[Parâmetros de comunicação do PDO](#) ³⁸

[Mapeamento de objetos no PDO](#) ⁴⁰

3.12.2 Lista de PDOs

Os PDOs são divididos em dois grupos “RxPDO” e “TxPDO”, sendo que “RxPDO” são os PDOs de leitura ou recepção e os “TxPDO” são os PDOs de escrita ou transmissão.



Nesta lista os PDOs destacados em negrito são os que encontram-se habilitados.

3.12.3 Parâmetros de comunicação do PDO

Ao selecionar um PDO na lista de PDOs, os parâmetros de comunicação do PDO serão mostrados no diálogo conforme figura a seguir:

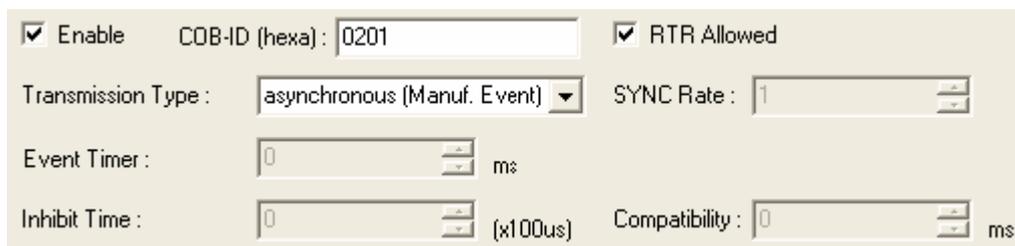


Fig. 41 - Parâmetros de comunicação do RxPDO.

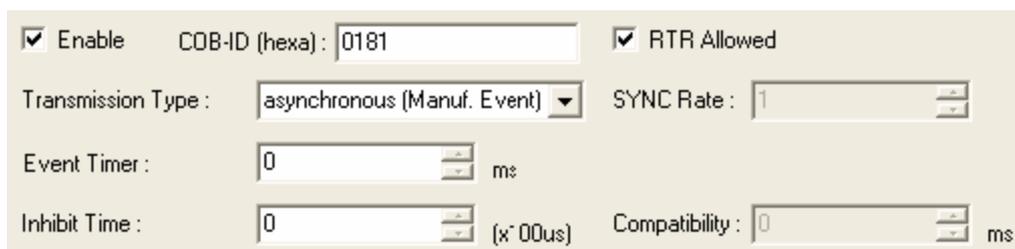


Fig. 42 - Parâmetros de comunicação do TxPDO.

Descrição dos parâmetros de comunicação :

- Enable: habilita/desabilita PDO.
- COB-ID (hexa): número de identificação do PDO na rede CANopen, sua faixa é 0181h a 057Fh . Através desse número os TxPDOs e RxPDOs serão ligados, ou seja, se queremos que um dado produzido em um TxPDO seja consumido em um RxPDO os dois deverão ter o mesmo COB-ID.
- RTR Allowed: permite/não permite RTR (frame remoto).
- Transmission Type: Tipo da recepção/transmissão.

Os seguintes tipos de transmissão são suportados :

Descrição	RxPDO	TxPDO
Synchronous acyclic	Após receber o PDO, o dispositivo somente atualizará seus dados mapeados no próximo telegrama SYNC detectado rede.	A transmissão do PDO será após detectar um telegrama SYNC, mas não periodicamente. A transmissão vai depender também de algum evento, dependente da configuração do dispositivo.
Synchronous cyclic	Após receber o PDO, o dispositivo somente atualizará seus dados mapeados no próximo telegrama SYNC detectado rede.	A transmissão do PDO dependerá apenas da detecção do telegrama SYNC. O valor programado em "SYNC Rate" entre 1 a 240 corresponderá ao número de telegramas SYNC que deverão ser detectados para que o PDO seja transmitido.
Synchronous RTR only	Valor não permitido para RxPDOs.	Após receber um frame remoto, os dados mapeados no PDO de transmissão serão atualizados, mas a transmissão do PDO será feita apenas no próximo telegrama SYNC detectado.
Asynchronous RTR only	Valor não permitido para RxPDOs.	O PDO é transmitido imediatamente após receber um frame remoto.
Asynchronous manif. event	Os dados mapeados são atualizados imediatamente após a recepção do PDO.	A transmissão do PDO é dependente de algum evento especificado pelo fabricante do dispositivo. Por exemplo, o dispositivo pode ser transmitido de acordo com o timer programado "Event Timer".
Asynchronous profile event	Os dados mapeados são atualizados imediatamente após a recepção do PDO.	A transmissão do PDO é dependente de algum evento especificado pelo perfil (<i>profile</i>) do dispositivo. Também pode ser programado para ser transmitido de acordo com o timer programado "Event Timer".

- Sync Rate: Utilizado para definir o número de telegramas SYNC para a transmissão do TxPDOs tipo "Synchronous cyclic".
- Event Timer: Utilizado para definir um evento temporizado, o mesmo pode ou não ser utilizado como um evento para os TxPDO de acordo com a configuração do dispositivo.
- Inhibit time: tempo de inibição dos TxPDOs, ou seja, esse é o tempo que o dispositivo não transmitirá um PDO após a sua transmissão.
- Compatibility: reservado.

NOTA!

No mestre, a transmissão de PDOs assíncronos, somente pode ser feita programando o timer "Event Timer".

3.12.4 Mapeamento de objetos no PDO

Todos os PDOs podem transmitir ou receber até 64bits de informação, essa informação deverá ser configurada através do mapeamento dos PDOs que basicamente consiste em atribuir quais objetos serão transmitidos via TxPDOs e quais objetos receberão via RxPDOs. Para cada dispositivo existirá uma lista de objetos que podem ser mapeados, bem como um número máximo de objetos mapeados por TxPDO ou RxPDO.

Como descrito anteriormente para ligarmos um TxPDO a um RxPDO ambos devem ter o mesmo COB-ID. Da mesma maneira, o tamanho total de bits mapeados nesses PDOs deve ser o mesmo, podendo o TxPDO transmitir mais informação que o RxPDO está programado para receber e nunca o contrário.



Para o mapeamento o dialogo dispõem de uma lista de objetos disponíveis para o mapeamento no RxPDO ou TxPDO, uma lista dos objetos mapeados para o PDO em questão, um botão para inserir um objeto no mapeamento "?", um botão para remover um objeto no mapeamento "?", um botão para subir o objeto no mapeamento "Up" e um botão para descer o objeto no mapeamento "Down".

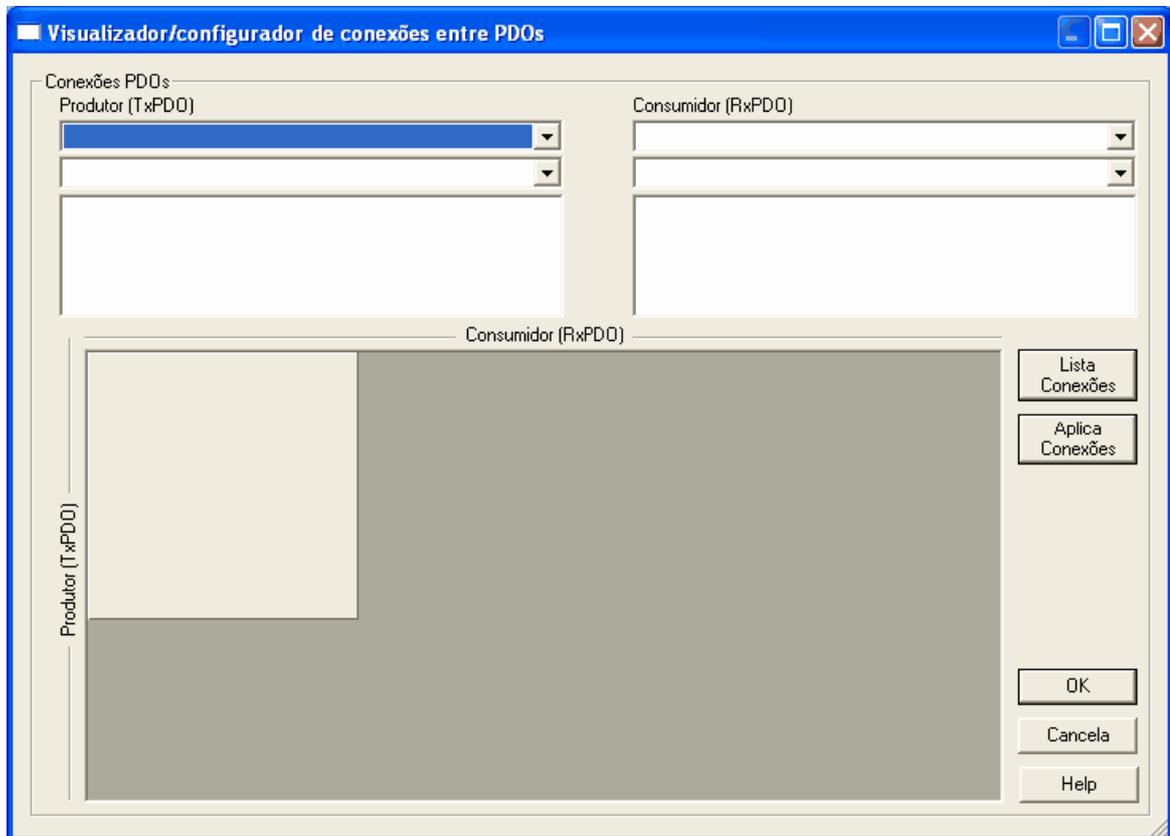
NOTA!

- Para RxPDO estarão disponíveis os objetos com acesso "rw" e "wo" e para TxPDO os objetos com acesso "rw" e "ro".

3.13 Visualizador/configurador de conexões entre PDOs

3.13.1 Visualizador/configurador de conexões entre PDOs

Esta janela é utilizada para efetuar conexões entre PDOs de transmissão (TxPDO) e PDOs de recepção (RxPDO) de dispositivos distintos.



Veja também :

[Procedimento para efetuar uma conexão entre PDOs](#) ⁴¹

[Listando as conexões](#) ⁴⁸

3.13.2 Procedimento para efetuar uma conexão entre PDOs

3.13.2.1 Procedimento para efetuar uma conexão entre PDOs

O procedimento de efetuar uma conexão basicamente consiste em selecionar um dispositivo que produza determinada informação (produtor) e um dispositivo que consuma essa informação produzida para a rede (consumidor).

Veja também :

[Selecionando o produtor \(TxPDO\)](#) ⁴²

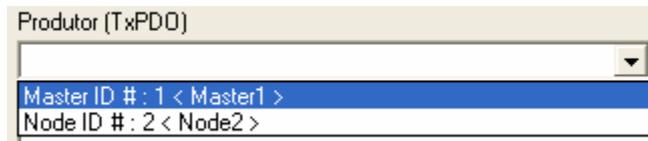
[Selecionando o consumidor \(RxPDO\)](#) ⁴³

[Conectando o produtor \(TxPDO\) ao consumidor \(RxPDO\)](#) ⁴⁵

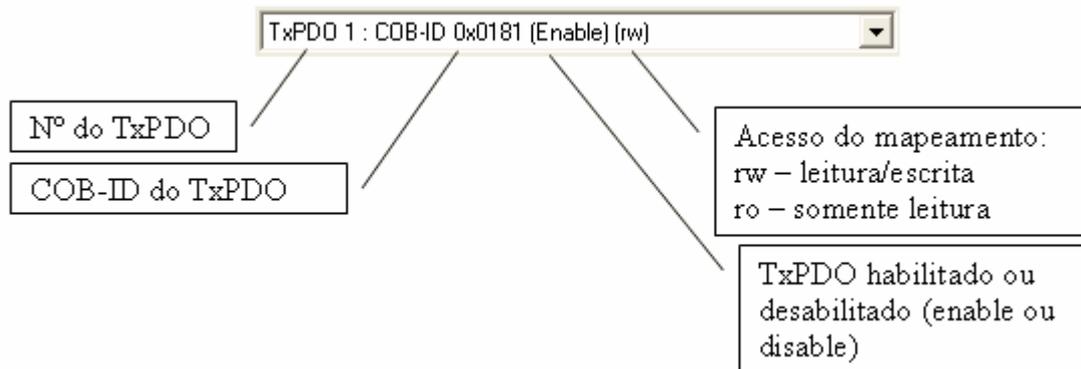
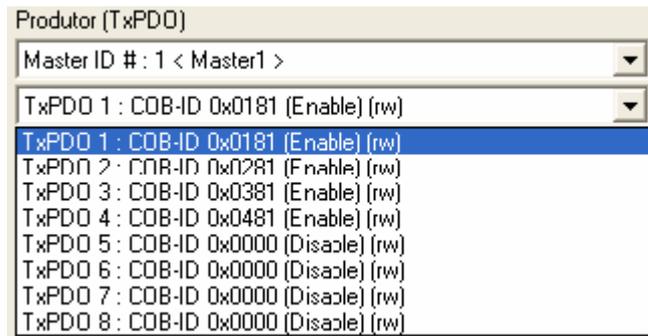
[Conectado objetos de tamanho diferentes](#) ⁴⁶

3.13.2.2 Selecionando o produtor (TxPDO)

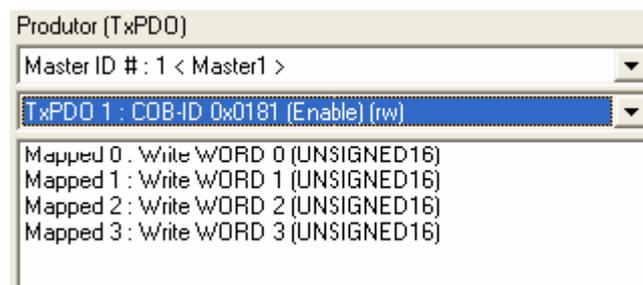
O produtor será o dispositivo que enviará os dados para a rede. Para selecioná-lo utilize a caixa de seleção do mesmo.



Após selecionar o dispositivo produtor uma lista será preenchida indicando quais os TxPDOs disponíveis e suas informações básicas.



Após selecionar um TxPDO uma lista será preenchida indicando os mapeamentos atuais do TxPDO e suas respectivas conexões.



Ao efetuar um duplo clique com o mouse sobre as informações do mapeamento a tabela do produtor será deslocada para a posição do primeiro objeto mapeado.

Quando existirem conexões relacionadas ao TxPDO será indicado informações relativas a essa

conexão conforme exemplo a seguir :

```

Connected to Node ID #: 2 < Node2 > RxPDO 1
Mapped 0 : Write WORD 0 (UNSIGNED16)
Mapped 1 : Write WORD 1 (UNSIGNED16)
Mapped 2 : Write WORD 2 (UNSIGNED16)
Mapped 3 : Write WORD 3 (UNSIGNED16)
    
```

Nessa figura é informado que o TxPDO está conectado ao RxPDO 1 do Node ID 2. Ao efetuar um duplo clique com o mouse sobre essa informação da conexão a janela selecionará o dispositivo e seu respectivo RxPDO na configuração do consumidor.

Se não existirem conexões informadas para o TxPDO e o mesmo está habilitado (Enable) significará que o mesmo está configurado para produzir uma informação para a rede e nenhum outro dispositivo está configurado para consumir.

3.13.2.3 Selecionando o consumidor (RxPDO)

O consumidor será o dispositivo que receberá os dados da rede. Para seleciona-lo utilize a caixa de seleção do mesmo.

Após selecionar o dispositivo consumidor uma lista será preenchida indicando quais os RxPDOs disponíveis e suas informações básicas.

Após selecionar um RxPDO uma lista será preenchida indicando os mapeamentos atuais do RxPDO e suas respectivas conexões.

Os objetos destacados com a cor verde correspondem aos objetos mapeados atualmente, para TxPDOs com acesso do tipo “rw” eles poderão ser modificados e com acesso do tipo “ro” não poderão ser modificados.

Os objetos destacados com a cor laranja correspondem aos objetos mapeados atualmente, para RxPDOs com acesso do tipo “rw” eles poderão ser modificados e com acesso do tipo “ro” não poderão ser modificados.

3.13.2.4 Conectando o produtor (TxPDO) ao consumidor (RxPDO)

O procedimento de conexão consiste basicamente em marcar através das caixas de check o objeto produzido e seu respectivo objeto consumidor, conforme figura a seguir :

Produtor (TxPDO)		Consumidor (RxPDO)										
		P099 Enable (INTEGER16)	P100 Acceleration Time (INTEGER16)	P101 Deceleration Time (INTEGER16)	P111 Direction of Rotation (INTEGER16)	P117 Position Reference (INTEGER16)	P119 Current Reference (TORQUE)	P121 Speed Reference (INTEGER16)	P202 Operation Mode (INTEGER16)	P432 Starts STOP Function (INTEGER16)	P435 Starts MOVE Function (INTEGER16)	Controlword (UNSIGNED16)
	Write WORD 0 (UNSIGNED16)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Após efetuado todas as conexões necessárias entre os objetos produzidos e respectivos objetos consumidores, basta clicar no botão “Aplica Conexões” para que seja criada a conexão em questão.

Durante o processo de criação da conexão desencadeado pelo botão “Aplica Conexões” vários procedimentos de verificação serão executados, em função do número de objetos do produtor e consumidor esse tempo poderá variar e chegar até a alguns segundos. Após esse procedimento a janela de conexões conterá as seguintes informações:

Produtor (TxPDO)	Consumidor (RxPDO)
Master ID #: 1 < Master1 >	Node ID #: 2 < Node2 >
TxPDO 1 : COB-ID 0x0181 (Enable) (rw)	RxPDO 1 : COB-ID 0x0181 (Enable) (rw)
Connected to Node ID #: 2 < Node2 > TxPDO 1 Mapped 0 : Write WORD 0 (UNSIGNED16)	Connected to Master ID #: 1 < Master1 > TxPDO 1 Mapped 0 : Controlword (UNSIGNED16)

Podemos observar que agora o TxPDO e o RxPDO possuem o mesmo COB-ID o que significa que estão conectados, também existe a informação que tanto o TxPDO quanto o RxPDO estão conectados.

O procedimento de “Aplica Conexões” não salva diretamente nos dicionário de objetos do projeto as novas configurações do PDO, para isso é necessário clicar no botão “OK” da janela de modo a confirmar que todas as novas configurações são válidas.

Neste mesmo exemplo em determinadas situações é necessário preservar o mapeamento do TxPDO para isso utilizaremos objetos “Dummy” no consumidor que são objetos virtuais utilizados para consumir informações não necessárias para o dispositivo.

Nesse exemplo os objetos produzidos “Write WORD 1” e “Write WORD 2” que tem tamanho de 16 bits cada serão consumidos pelo objeto “Target Velocity” que tem tamanho de 32 bits, ou seja, 16 bits + 16 bits = 32 bits.

Após aplicar as conexões os mapeamentos ficarão da seguinte forma.

Produtor (TxPDO)	Consumidor (RxPDO)
Master ID #: 1 < Master1 >	Node ID #: 2 < Node2 >
TxPDO 1: COB-ID 0x0181 (Enable) (rw)	RxPDO 1: COB-ID 0x0181 (Enable) (rw)
Connected to Node ID #: 2 < Node2 > TxPDO 1 Mapped 0: Write WORD 0 (UNSIGNED16) Mapped 1: Write WORD 1 (UNSIGNED16) Mapped 2: Write WORD 2 (UNSIGNED16)	Connected to Master ID #: 1 < Master1 > TxPDO 1 Mapped 0: Controlword (UNSIGNED16) Mapped 1: Target Velocity (INTEGER32)

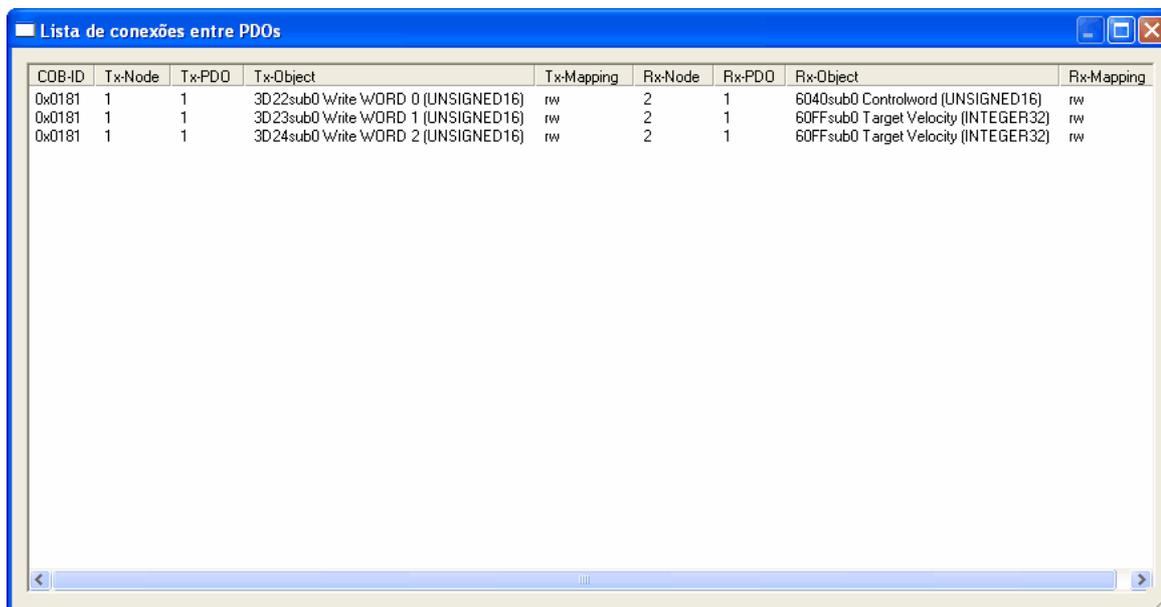
Outro exemplo :

Consumidor (RxPDO)	
Produtor (TxPDO)	Read WORD 30 (UNSIGNED16)
	Read WORD 31 (UNSIGNED16)
	Read BYTE 0 (UNSIGNED8)
	Read BYTE 1 (UNSIGNED8)
	Read BYTE 2 (UNSIGNED8)
	Read BYTE 3 (UNSIGNED8)
	Read BYTE 4 (UNSIGNED8)
	Read BYTE 5 (UNSIGNED8)
	Read BYTE 6 (UNSIGNED8)
	Read BYTE 7 (UNSIGNED8)
	Read BYTE 8 (UNSIGNED8)
	Read BYTE 9 (UNSIGNED8)
	Read BYTE 10 (UNSIGNED8)
	Read BYTE 11 (UNSIGNED8)
Read BYTE 12 (UNSIGNED8)	
Read BYTE 13 (UNSIGNED8)	
P053 Position (number of revolutions)	<input type="checkbox"/>
Statusword (UNSIGNED16)	<input checked="" type="checkbox"/>
Modes of Operation Display (INTEGER32)	<input type="checkbox"/>
Position Actual Value (INTEGER32)	<input type="checkbox"/>

Produtor (TxPDO)	Consumidor (RxPDO)
Node ID #: 2 < Node2 >	Master ID #: 1 < Master1 >
TxPDO 1: COB-ID 0x0182 (Enable) (rw)	RxPDO 1: COB-ID 0x0182 (Enable) (rw)
Connected to Master ID #: 1 < Master1 > TxPDO 1 Mapped 0: Statusword (UNSIGNED16)	Connected to Node ID #: 2 < Node2 > TxPDO 1 Mapped 0: Read BYTE 0 (UNSIGNED8) Mapped 1: Read BYTE 1 (UNSIGNED8)

3.13.3 Listando as conexões

Para listar as conexões basta clicar no botão “Lista Conexão” que a seguinte janela aparecerá:



COB-ID	Tx-Node	Tx-PDO	Tx-Object	Tx-Mapping	Rx-Node	Rx-PDO	Rx-Object	Rx-Mapping
0x0181	1	1	3D22sub0 Write WORD 0 (UNSIGNED16)	rw	2	1	6040sub0 Controlword (UNSIGNED16)	rw
0x0181	1	1	3D23sub0 Write WORD 1 (UNSIGNED16)	rw	2	1	60FFsub0 Target Velocity (INTEGER32)	rw
0x0181	1	1	3D24sub0 Write WORD 2 (UNSIGNED16)	rw	2	1	60FFsub0 Target Velocity (INTEGER32)	rw

Nessa janela aparecerá todas as conexões listadas por objeto.

São mostradas as seguintes informações da conexão:

- COB-ID: Identificador do TxPDO e RxPDO.
- Tx-Node: Nó do TxPDO.
- Tx-PDO: Número do TxPDO.
- Tx-Object: Objeto mapeado no TxPDO.
- Tx-Mapping: Acesso do mapeamento do TxPDO (“ro” ou “rw”).
- Rx-Node: Nó do RxPDO.
- Rx-PDO: Número do RxPDO.
- Rx-Object: Objeto mapeado no RxPDO.
- Rx-Mapping: Acesso do mapeamento do TxPDO (“ro” ou “rw”).

3.14 Configuração da rede

3.14.1 Gerar configuração

Para gerar a configuração do mestre basta clicar no menu “Configuração” opção “Gerar”, ou tecla F7 ou botão .

Uma vez gerada a configuração a mesma pode ser transmitida.

Veja também :

[Transmitir configuração](#) ⁴⁹

[Monitoração online](#) ⁴⁹

3.14.2 Transmitir configuração

Para transmitir a configuração do mestre basta clicar no menu “Configuração” opção “Transmitir”, ou tecla F8 ou botão .

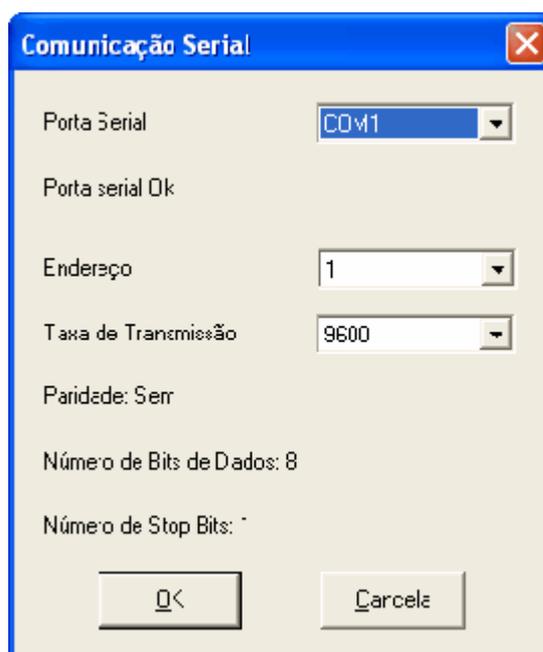
Durante a transmissão de configuração do mestre tanto o programa do usuário em ladder quanto a comunicação CANopen serão interrompidos, após a transmissão o programa em ladder será executado e o mestre inicializará a rede CANopen executando todos os serviços de inicialização inclusive a configuração dos escravos.

Caso não seja possível a comunicação com o mestre CANopen via porta modbus a seguinte janela aparecerá :



Neste caso é necessário verificar a correta conexão do cabo serial entre a porta serial do computador e o conector serial do cartão, também é necessário verificar a configuração serial utilizada pelo software WSCAN e a configuração setada no cartão em questão.

Para acessar a configuração da serial basta clicar no menu “Comunicação” opção “Configura Serial”, ou tecla Shift+F8 que a seguinte janela aparecerá :



3.14.3 Monitoração online

Após transmitir a configuração do mestre e necessário monitorar se todos os dispositivos configurados estão conectados a rede e efetivamente comunicando para isso basta clicar no menu “Comunicação” opção “Monitoração Online”, ou tecla F9 ou botão .

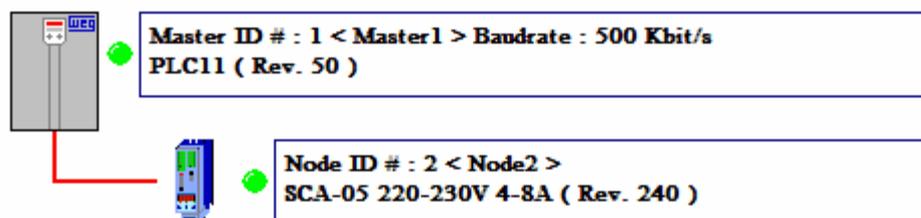
Neste momento o WSCAN tentará estabelecer a comunicação com o cartão testando a

comunicação com a mesma, se a comunicação estiver OK a seguinte mensagem aparecerá no status bar que está na parte inferior do WSCAN conforme figura a seguir:

Porta USB aberta com sucesso.

Nesta mesma barra existirá um indicador tipo LED da cor azul que ficará piscando indicando que a comunicação está operando .

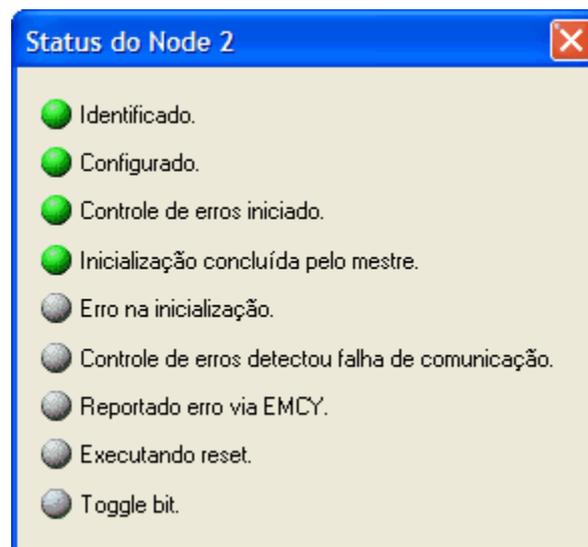
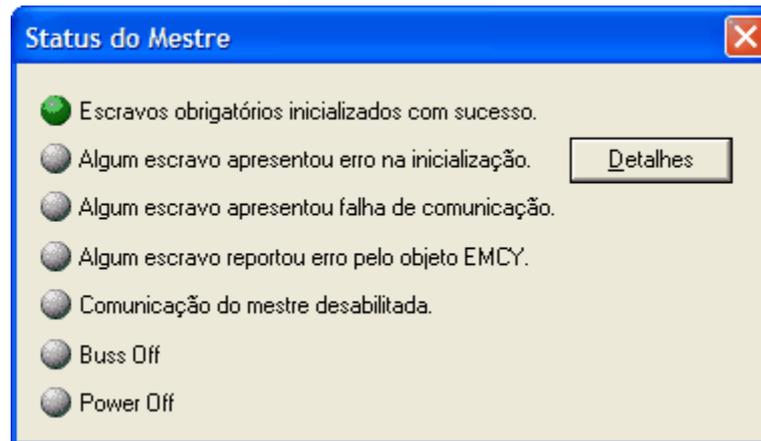
Após a efetiva comunicação em monitoração online a representação da rede aparecerá da seguinte maneira :



Cada dispositivo terá um LED indicando seu status conforme tabela a seguir:

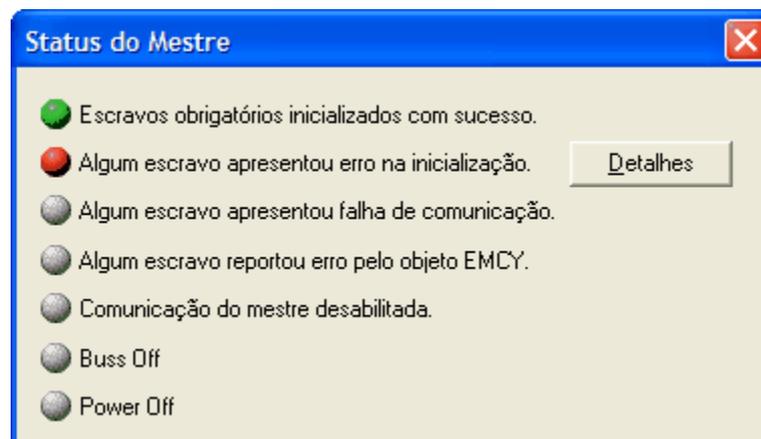
Cor do LED	Condição Mestre	Condição Escravo
Verde	<ul style="list-style-type: none"> - Comunicação habilitada. - Todos escravos inicializados com sucesso. - Nenhuma falha de comunicação detectada em algum escravo. - Nenhum escravo reportou falha EMCY. 	<ul style="list-style-type: none"> - Identificado. - Configurado. - Controle de erros inicializado caso necessário. - Inicialização concluída pelo mestre. - Sem erro de inicialização. - Sem erro de comunicação. - Não foi reportado nenhuma falha EMCY. - Não está executando reset.
Amarelo (piscante)	<ul style="list-style-type: none"> - Detectado falha na inicialização em algum escravo - Detectado falha de comunicação em algum escravo - Algum escravo reportou falha EMCY 	<ul style="list-style-type: none"> - Reportou falha EMCY. - Executando reset. - Controle de erros não inicializado quando necessário.
Vermelho (piscante)	<ul style="list-style-type: none"> - Comunicação desabilitada. 	<ul style="list-style-type: none"> - Não inicializado. - Detectou falha de comunicação.

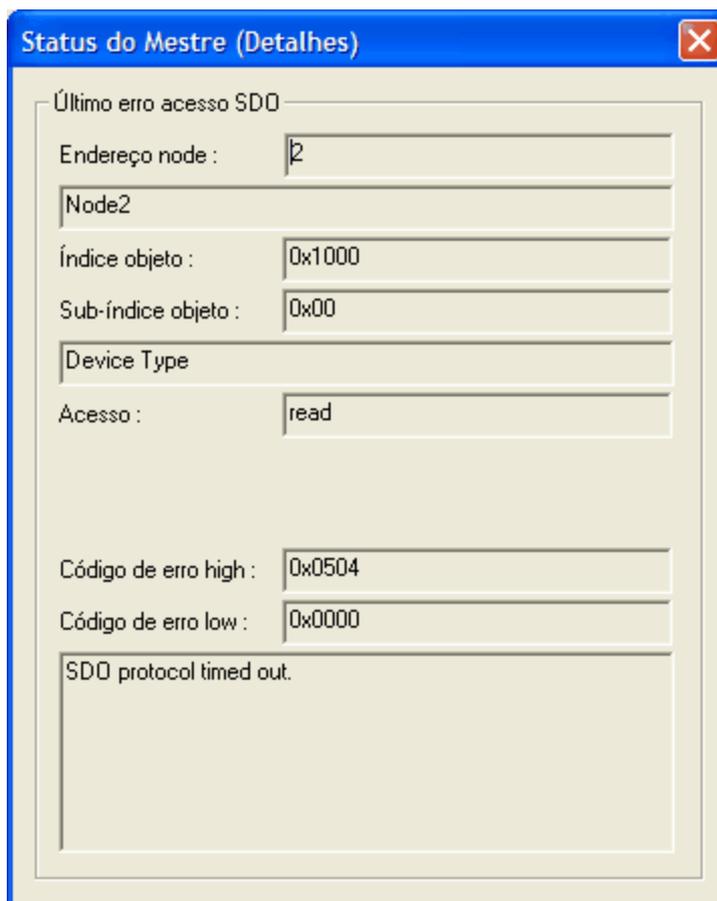
Também é possível ter um status detalhado de cada dispositivo, para isso durante a monitoração online clique sobre o dispositivo que as seguintes caixas de monitoração aparecerão :



(1) Indicações "Bus Off" e "Power Off" somente disponíveis para PLC11-01.

Quando for detectado falha de inicialização em algum escravo é possível identificar em qual etapa da inicialização aconteceu a falha, para isso clique no botão "Detalhes" do status do mestre conforme figuras a seguir:





No exemplo acima quando o mestre tentou escrever o objeto 1A03h sub-índice 1 o escravo endereço 2 reportou erro que é detalhado na caixa de texto do dialogo em questão. Para a solução do problema reportado nesse dialogo é necessário verificar o objeto em questão no “Dicionário de objetos” e também no manual CANopen do dispositivo em questão.

A informação desse dialogo só é válida quando ocorrer erro de inicialização em algum escravo.

3.15 Interface com a programação ladder

3.15.1 Interface com a programação ladder

Uma vez configurado, o mestre da rede fica responsável por gerenciar toda a comunicação. No entanto, deve ser possível acessar os dados recebidos e transmitidos, e programar a lógica de operação desejada em ladder. Para isto, foram definidos dados disponibilizados na rede, blocos em linguagem ladder e informações de status e controle da comunicação.

Veja também :

[Mapeamento de memória de I/O](#) ⁵³

[Palavras de comando e estado da comunicação](#) ⁵³

[Bloco ladder \(SDO\)](#) ⁵⁶

3.15.2 Mapeamento de memória de I/O

Para o mapeamento dos dados em rede CANopen, são disponibilizados words de leitura, words de escrita, bytes de leitura e bytes de escrita. Os mesmos estão organizados da seguinte maneira:

- PLC2 :

Endereço	Nome	Acesso Via Ladder	Mapeável
%RW0 ... %RW31	Words de Leitura	Leitura (Input: escravo → mestre)	RPDOs do mestre
%WW0 ... %WW31	Words de Escrita	Escrita (Output: mestre → escravo)	TPDOs do mestre
%RB0 ... %RB31	Bytes de Leitura	Leitura (Input: escravo → mestre)	RPDOs do mestre
%WB0 ... %WB31	Bytes de Escrita	Escrita (Output: mestre → escravo)	TPDOs do mestre

- PLC11-01 :

Endereço	Nome	Acesso Via Ladder	Mapeável
%RW4200 ... %RW4299	Words de Leitura	Leitura (Input: escravo → mestre)	RPDOs do mestre
%WW4600 ... %WW4699	Words de Escrita	Escrita (Output: mestre → escravo)	TPDOs do mestre
%RB4400 ... %RB4499	Bytes de Leitura	Leitura (Input: escravo → mestre)	RPDOs do mestre
%WB4800 ... %WB4899	Bytes de Escrita	Escrita (Output: mestre → escravo)	TPDOs do mestre

Estes marcadores são voláteis. Nestes marcadores que todos os dados enviados e recebidos via PDO e EMCY serão armazenados.

3.15.3 Palavras de comando e estado da comunicação

3.15.3.1 Palavras de comando e estado da comunicação

Para indicação do estado e controle da comunicação da rede CANopen, são disponibilizadas palavras de estado e controle. Os mesmos estão organizados da seguinte maneira:

- PLC2 :

Endereço	Nome	Acesso Via Ladder
%RS0	Estado da comunicação CANopen	Leitura (Input: escravo → mestre)
%RS1 ... %RS63	Estado do escravo CANopen	Leitura (Input: escravo → mestre)
%WC0	Controle Comunicação CANopen	Escrita (Output: mestre → escravo)
%WC1	Endereço do escravo destino	Escrita (Output: mestre → escravo)

- PLC11-01 :

Endereço	Nome	Acesso Via Ladder
%RS4000	Estado da comunicação CANopen	Leitura (Input: escravo → mestre)
%RS4001 ... %RS4063	Estado do escravo CANopen	Leitura (Input: escravo → mestre)
%WC4136	Controle Comunicação CANopen	Escrita (Output: mestre → escravo)
%WC4137	Endereço do escravo destino	Escrita (Output: mestre → escravo)

Estas palavras terão o significado conforme descrito a seguir.

Veja também :

[Estado da comunicação CANopen](#) ⁵⁴

[Estado do escravo CANopen](#) ⁵⁵

[Controle da comunicação CANopen](#) ⁵⁶

[Endereço do escravo destino](#) ⁵⁶

3.15.3.2 Estado da comunicação CANopen

WORD que indica o estado da comunicação CANopen do mestre. Cada bit possui um significado:

Bit	Descrição	Valores
0...2	Reservado	-
3	Inicialização dos escravos	0: Procedimento de inicialização da rede CANopen não está concluído. 1: Todos os escravos configurados como obrigatórios foram inicializados com sucesso.
4	Falha na inicialização dos escravos	0: Nenhum escravo possui erro de inicialização. 1: Ocorreu erro na inicialização de algum escravo. Caso o escravo seja obrigatório, o procedimento de inicialização não poderá ser concluído até que o mestre consiga configurar este escravo.
5	Falha no controle de erros do escravo	0: Todos os escravos que possuem mecanismos para controle de erros (Node Guarding ou Heartbeat) estão operando corretamente. 1: Ocorreu falha no mecanismo de controle de erros de pelo menos um escravo.
6	Escravo enviou EMCY	0: Nenhuma ocorrência do objeto Emergency foi detectada. 1: Pelo menos um escravo reportou a ocorrência de erro através do objeto Emergency.
7	Reservado	-
8	Retorno do toggle bit	Indica o valor do toggle bit que foi escrito na palavra de comando. Sempre que um novo comando for enviado para um escravo e o valor do toggle bit for comando for alterado, após executar o comando este bit assumirá o mesmo valor do bit de comando, fornecendo um feedback para o programador que a ação foi executada.
9...14	Reservado	-

13	Bus Off ⁽¹⁾	0: Mestre normal. 1: Mestre não está comunicando por erro de Bus Off.
14	Power Off ⁽¹⁾	0: Mestre com alimentação normal. 1: Mestre sem alimentação na interface CAN.
15	Comunicação desabilitada	0: Comunicação operando normalmente. 1: Mestre recebeu o comando de desabilitação da comunicação através da palavra de comando, e a comunicação foi desabilitada.

(1) Indicações "Bus Off" e "Power Off" somente disponíveis para PLC11-01.

3.15.3.3 Estado do escravo CANopen

Array com 63 WORDs, onde cada posição representa o endereço do escravo na rede, indicando o status deste escravo. Cada bit de cada escravo possui a seguinte função:

Bit	Descrição	Valores
0	Escravo identificado	0: Identificação do escravo não realizada. 1: Identificação do escravo (através da leitura do tipo de dispositivo) realizada com sucesso.
1	Escravo configurado	0: Configuração do dispositivo não realizada 1: Procedimento de configuração do escravo realizado com sucesso.
2	Controle de erros iniciado	0: Serviço de controle de erros não iniciado. 1: Serviço de controle de erros para o escravo (guarding ou heartbeat) iniciado.
3	Procedimento de inicialização concluído	0: Escravo não foi inicializado pelo mestre. 1: Mestre concluiu a inicialização do escravo.
4	Erro na inicialização	0: Escravo não apresenta falha na sua inicialização 1: Escravo apresentou falha na sua inicialização. O procedimento de inicialização não pode continuar até que este escravo consiga ser inicializado.
5	Serviço de controle de erro (guarding ou heartbeat) detectou falha	0: Serviço de controle de erros não apresenta falha. 1: Serviço de controle de erros para o escravo (guarding ou heartbeat) reportou uma falha.
6	Escravo enviou objeto EMCY reportando a ocorrência de erro	0: Escravo não reportou a ocorrência de erros via EMCY. 1: Escravo reportou a ocorrência de erros via EMCY.
7	Reset do escravo	0: Nenhum operação sendo realizada sobre o escravo. 1: Executando reset do escravo, devido à falha na comunicação ou pelo comando executado sobre o escravo na palavra de controle.
8...15	Reservado	-

NOTA!

Apesar de a rede CANopen permitir endereços de 1 até 127, o valor máximo de endereços permitidos para os escravos é 63, pois as demais posições não possuem palavras de estado correspondentes.

3.15.3.4 Controle da comunicação CANopen

WORD que permite realizar algumas funções com relação à comunicação CANopen em um dispositivo determinado. Cada bit possui um significado:

Bit	Descrição	Valores
0...7	Comando para o escravo	Comando de gerenciamento do estado do escravo. Os seguintes valores podem ser escritos nestes bits, de forma que o mestre possa executar os comandos nos escravos presentes na rede: 1: Start Node 2: Stop Node 128: Entre pre-operational 129: Reset Node 130: Reset Communication O comando programado nestes bits será enviado para o escravo indicado na palavra de endereço do escravo destino, sempre que o valor do <i>toggle bit</i> for alterado.
8	<i>Toggle bit</i>	Bit que indica quando um novo comando pode ser enviado para o escravo. Sempre que o valor deste bit for alterado (0 ? 1 ou 1 ? 0), o mestre irá enviar o comando programado nos bits 0 ... 7 para o escravo indicado na palavra de endereço do escravo destino.
9 ... 14	Reservado	-
15	Desabilita comunicação CANopen	0: Mestre opera normalmente 1: Interrompe a comunicação e tratamento do mestre CANopen. Ao colocar novamente o bit para 0 (zero), a interface CANopen é reiniciada e o mestre repete o procedimento de inicialização de todos os escravos.

3.15.3.5 Endereço do escravo destino

WORD que permite programar o endereço do escravo para o qual se deseja enviar um comando de gerenciamento para o escravo na palavra de controle da comunicação. Os valores permitidos para esta WORD são:

- 0: Mestre envia um comando do tipo *broadcast* (todos os escravos da rede devem executar o comando).
- 1 ... 127: Endereço específico de um escravo, para o qual o comando deve ser enviado.

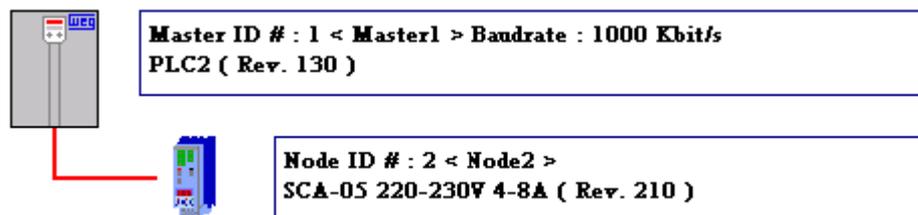
3.15.4 Bloco ladder (SDO)

Para leitura/escrita em objetos dos escravos foi criado o bloco SDO que é utilizado no programa ladder. Para maiores detalhes sobre o mesmo favor verificar no manual do WLP.

3.16 Exemplo de configuração de rede

3.16.1 Introdução

Mestre : PLC2 Rev. 130, Endereço: 1, Baudrate: 1000 Kbit/s
 Escravo : SCA-05 Rev. 210, Endereço: 2



Veja também :

[Configuração mestre \(PLC2\)](#) ^[57]

[Configuração do escravo \(SCA-05\)](#) ^[57]

[PDOs do mestre \(PLC2\)](#) ^[57]

[PDOs do escravo \(SCA-05\)](#) ^[58]

[Conexões PDOs](#) ^[60]

[Leitura do status do escravo no ladder do mestre](#) ^[62]

[Comando do escravo no ladder do mestre](#) ^[63]

3.16.2 Configuração mestre (PLC2)

Padrão, nenhum parâmetro modificado.

3.16.3 Configuração do escravo (SCA-05)

Padrão, nenhum parâmetro modificado.

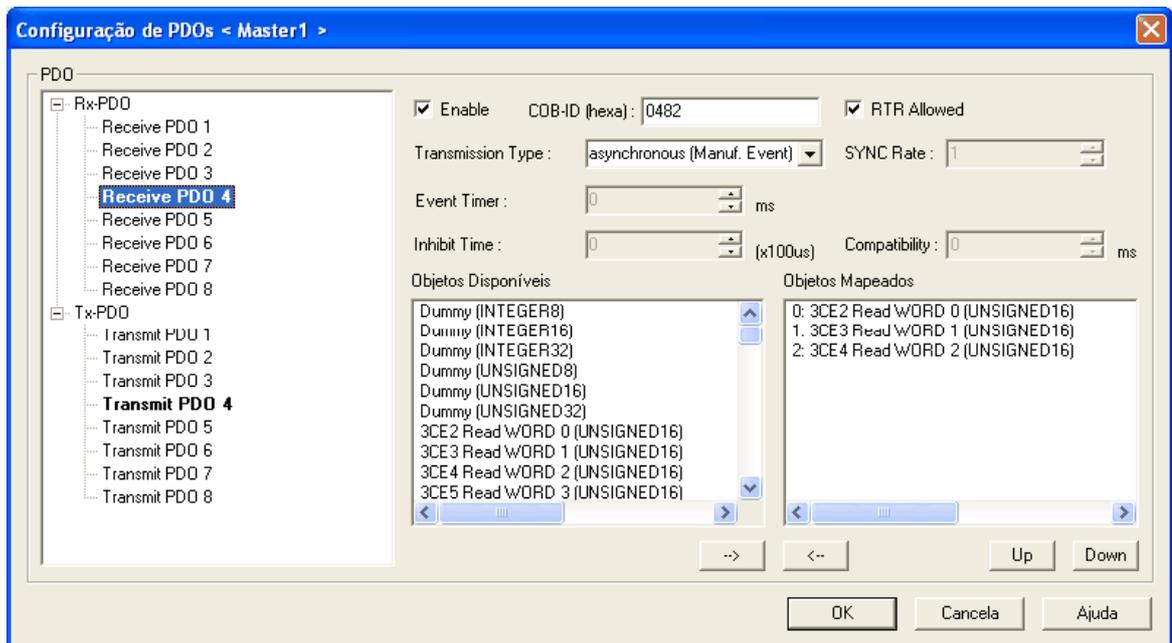
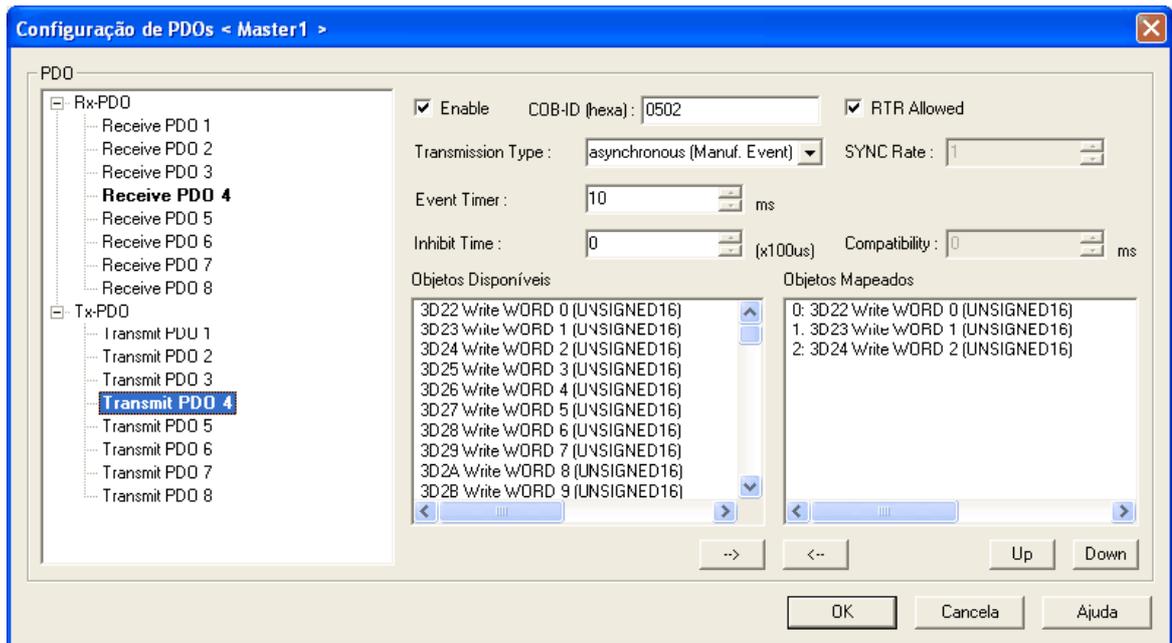
3.16.4 PDOs do mestre (PLC2)

No mestre foi criado um PDO de transmissão “TxPDO” com COB-ID 0502h (mesmo COB-ID do RxPDO do escravo) transmitindo três UNSIGNED16, sendo eles Write Word 0, 1 e 2.

Também foi criado um PDO de recepção “RxPDO” com COB-ID 0482h (mesmo COB-ID do TxPDO do escravo) recebendo três UNSIGNED16, sendo eles Read Word 0, 1 e 2.

Pode-se notar que nos dois PDOs foram ligados dados INTEGER32 do escravo a dois UNSIGNED16 do mestre. Isto é uma situação prevista pela rede CANopen que interpreta o número de bits total mapeado nos PDOs. Isto também é necessário porque o mestre da rede CANopen não possui objetos INTEGER32 para o mapeamento, e também sabemos que o SCA-05 os objetos “Target Velocity” e “Velocity Actual Value” só utilizam a parte baixa da mesma, ou seja, os primeiros 16bits.

Somente é mostrado os PDOs habilitados, os demais foram desabilitados.



3.16.5 PDOs do escravo (SCA-05)

No escravo foi criado um PDO de recepção “RxPDO” com COB-ID 0502h recebendo um UNSIGNED16 para sua “Controlword” e um INTEGER32 para sua “Target Velocity”.

Também foi criado um PDO de transmissão “TxPDO” com COB-ID 0482h transmitindo um UNSIGNED16 com sua “Statusword” e um INTEGER32 com sua “Velocity Actual Value”.

Somente é mostrado os PDOs habilitados, os demais foram desabilitados.

Configuração de PDOs < Node2 >

PDO

- Rx-PDO
 - Receive PDO 1
 - Receive PDO 2
 - Receive PDO 3
 - Receive PDO 4**
- Tx-PDO
 - Transmit PDO 1
 - Transmit PDO 2
 - Transmit PDO 3
 - Transmit PDO 4**

Enable COB-ID (hexa): 0502 RTR Allowed

Transmission Type: asynchronous (Manuf. Event) SYNC Rate: 1

Event Timer: 0 ms

Inhibit Time: 0 (x100us) Compatibility: 0 ms

Objetos Disponíveis

- Dummy (INTEGER8)
- Dummy (INTEGER16)
- Dummy (INTEGER32)
- Dummy (UNSIGNED8)
- Dummy (UNSIGNED16)
- Dummy (UNSIGNED32)
- 2063 P099 Enable (INTEGER16)
- 2064 P100 Acceleration Time (INTEGER16)
- 2065 P101 Deceleration Time (INTEGER16)
- 206F P111 Direction of Rotation (INTEGER16)

Objetos Mapeados

- 0: 6040 Controlword (UNSIGNED16)
- 1: 60FF Target Velocity (INTEGER32)

OK Cancela Ajuda

Configuração de PDOs < Node2 >

PDO

- Rx-PDO
 - Receive PDO 1
 - Receive PDO 2
 - Receive PDO 3
 - Receive PDO 4**
- Tx-PDO
 - Transmit PDO 1
 - Transmit PDO 2
 - Transmit PDO 3
 - Transmit PDO 4**

Enable COB-ID (hexa): 0482 RTR Allowed

Transmission Type: asynchronous (Manuf. Event) SYNC Rate: 1

Event Timer: 10 ms

Inhibit Time: 0 (x100us) Compatibility: 3 ms

Objetos Disponíveis

- 2002 P002 Motor Speed (INTEGER16)
- 2003 P003 Motor Current (INTEGER16)
- 2004 P004 DC Voltage (INTEGER16)
- 200C P012 Digital Input Status (INTEGER16)
- 200E P014 Last Fault (INTEGER16)
- 2012 P018 AI1 Value (INTEGER16)
- 2013 P019 AI2 Value (INTEGER16)
- 2032 P050 Shaft Position (INTEGER16)
- 2034 P052 Position (fraction of revolution) (IN
- 2035 P053 Position (number of revolutions) (IN

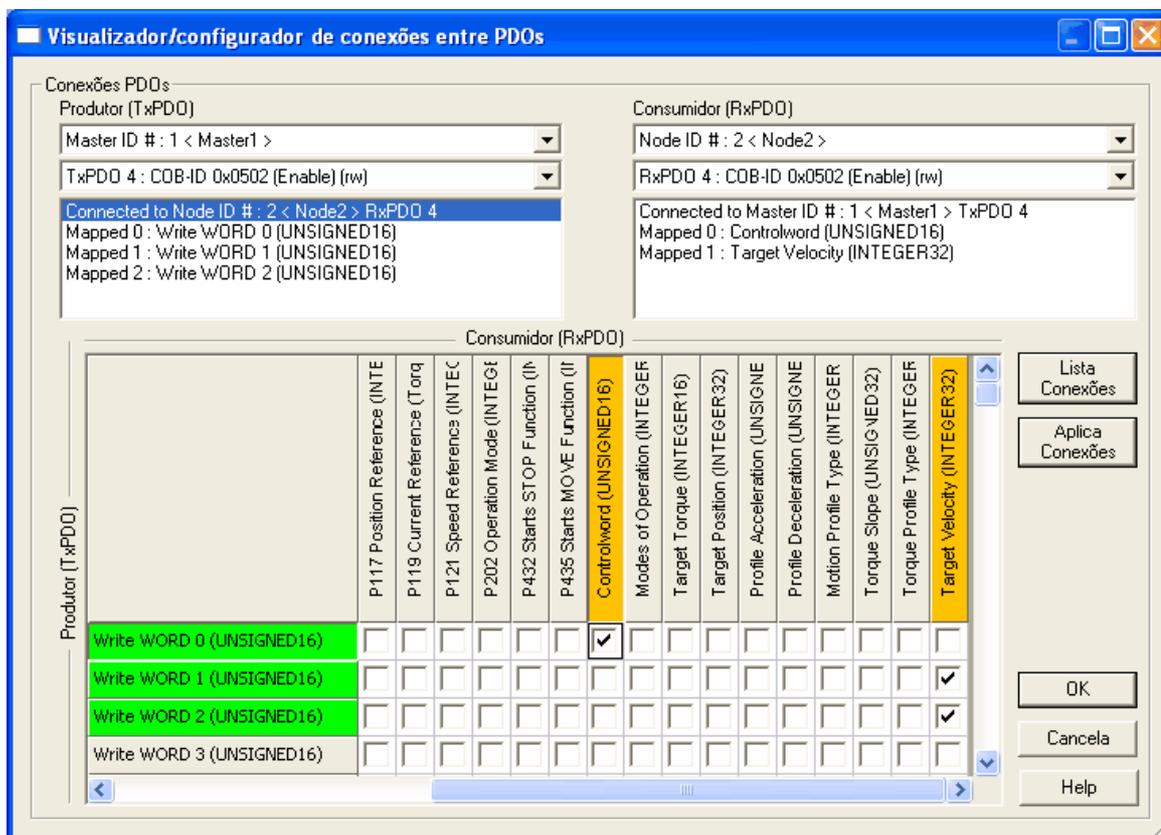
Objetos Mapeados

- 0: 6041 Statusword (UNSIGNED16)
- 1: 606C Velocity Actual Value (INTEGER32)

OK Cancela Ajuda

3.16.6 Conexões PDOs

Neste exemplo demonstro a configuração dos PDOs individualmente mais é possível configurar diretamente na janela de “Conexões PDOs”, conforme figuras a seguir:



Visualizador/configurador de conexões entre PDOs

Conexões PDOs

Produtor (TxPDO)

Node ID #: 2 < Node2 >

TxPDO 4: COB-ID 0x0482 (Enable) (rw)

Connected to Master ID #: 1 < Master1 > RxPDO 4

Mapped 0: Statusword (UNSIGNED16)

Mapped 1: Velocity Actual Value (INTEGER32)

Consumidor (RxPDO)

Master ID #: 1 < Master1 >

RxPDO 4: COB-ID 0x0482 (Enable) (rw)

Connected to Node ID #: 2 < Node2 > TxPDO 4

Mapped 0: Read WORD 0 (UNSIGNED16)

Mapped 1: Read WORD 1 (UNSIGNED16)

Mapped 2: Read WORD 2 (UNSIGNED16)

Consumidor (RxPDO)

Produtor (TxPDO)	Read WORD 0 (UNSIGNED16)	Read WORD 1 (UNSIGNED16)	Read WORD 2 (UNSIGNED16)	Read WORD 3 (UNSIGNED16)	Read WORD 4 (UNSIGNED16)	Read WORD 5 (UNSIGNED16)	Read WORD 6 (UNSIGNED16)	Read WORD 7 (UNSIGNED16)	Read WORD 8 (UNSIGNED16)	Read WORD 9 (UNSIGNED16)	Read WORD 10 (UNSIGNED16)	Read WORD 11 (UNSIGNED16)	Read WORD 12 (UNSIGNED16)	Read WORD 13 (UNSIGNED16)	Read WORD 14 (UNSIGNED16)	Read WORD 15 (UNSIGNED16)
Statusword (UNSIGNED16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modes of Operation Display (INTEGER32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Position Actual Value (INTEGER32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Position Actual Value in User Unit (INTEGER32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Velocity Sensor Actual Value (INTEGER32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Velocity Demand Value (INTEGER32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Velocity Actual Value (INTEGER32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							

Lista Conexões

Aplica Conexões

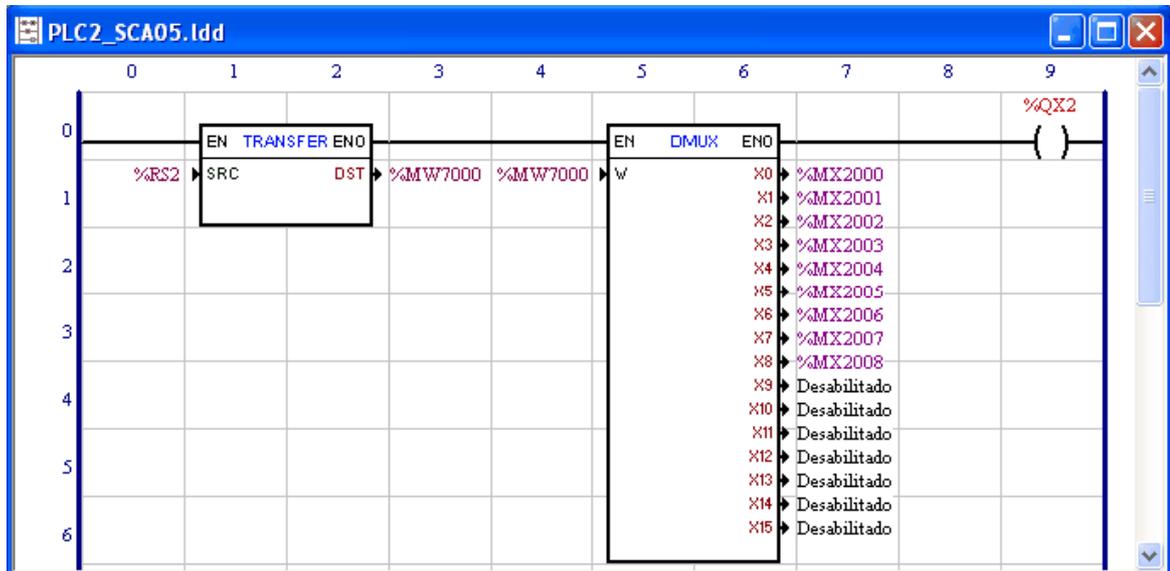
OK

Cancela

Help

3.16.7 Leitura do status do escravo no ladder do mestre

A leitura do status do Node 2 é feita pelo “Word Leitura Estado 2” %RS2 e transformada em marcadores de bit através do bloco “DMUX”.



Veja também :

[Configuração WSCAN](#) ⁶⁴

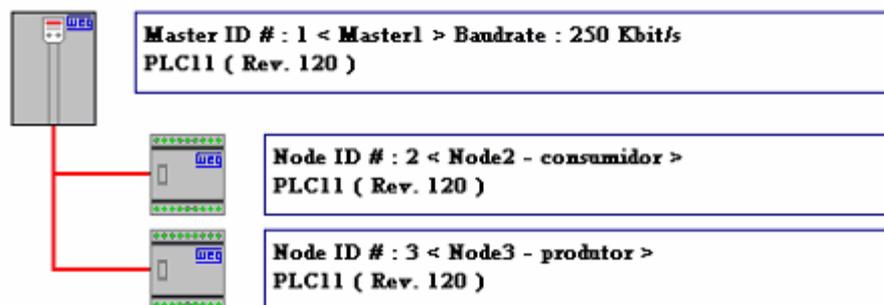
[Configuração WLP](#) ⁶⁷

[Parametrização CFW11 e PLC11](#) ⁶⁷

3.17.2 Configuração WSCAN

No WSCAN é montada toda a configuração de comunicação e definido quem é o produtor e o consumidor Follow.

A figura a seguir ilustra um exemplo de configuração da rede CANopen com três equipamentos, um mestre e dois escravos. O escravo 2 está configurado como consumidor Follow e o escravo 3 como produto Follow.



Na janela de Configuração, na aba FOLLOW, é realizada toda a configuração dos PDOs para a função Follow.

Inicialmente deve-se definir o equipamento que será o produtor, informando o CobId do PDO. É necessário definir se a velocidade é teórica ou real, como ilustra a figura a seguir.

Configuração do Escravo <Node3 - produtor>

NMT NODE GUARDING

PROD. HEARTBEAT EMCY FOLLOW

FOLLOW

Desabilita

Habilita Produtor

Habilita Consumidor

COB-ID (hexa):

FONTE DA VELOCIDADE

Real

Referência

OK Cancelar Aplicar Ajuda

Próximo passo é a configuração do consumidor conforme figura a seguir. O Cobld do consumidor deve ser igual ao Cobld definido no produtor.

Configuração do Escravo <Node2 - consumidor>

NMT NODE GUARDING

PROD. HEARTBEAT EMCY FOLLOW

FOLLOW

Desabilita

Habilita Produtor

Habilita Consumidor

COB-ID (hexa):

FONTE DA VELOCIDADE

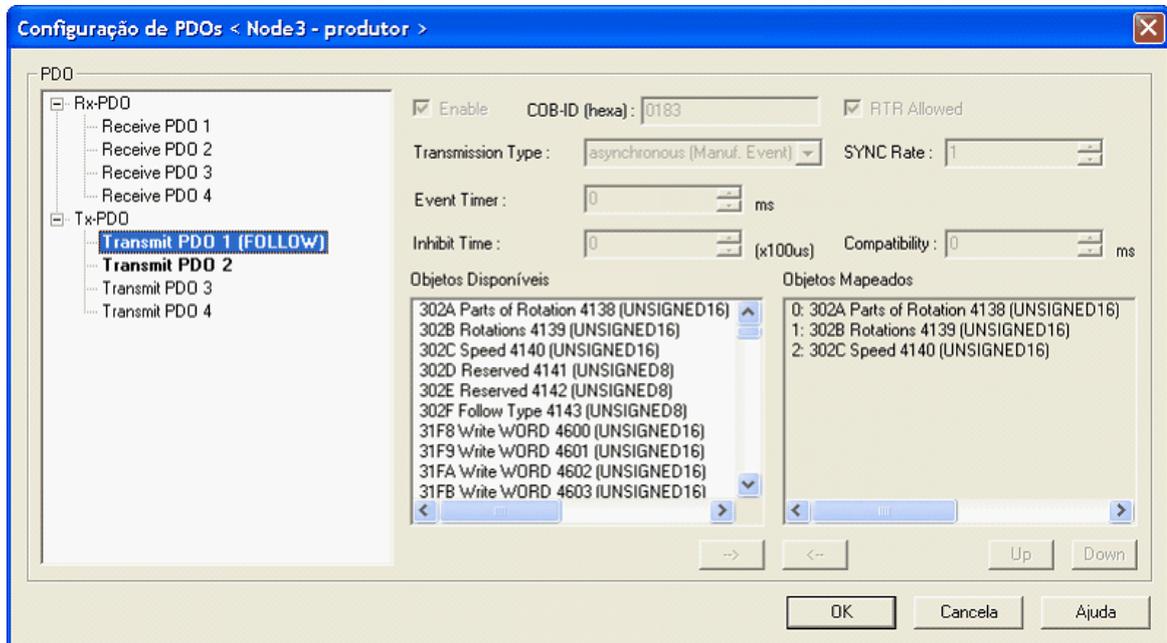
Real

Referência

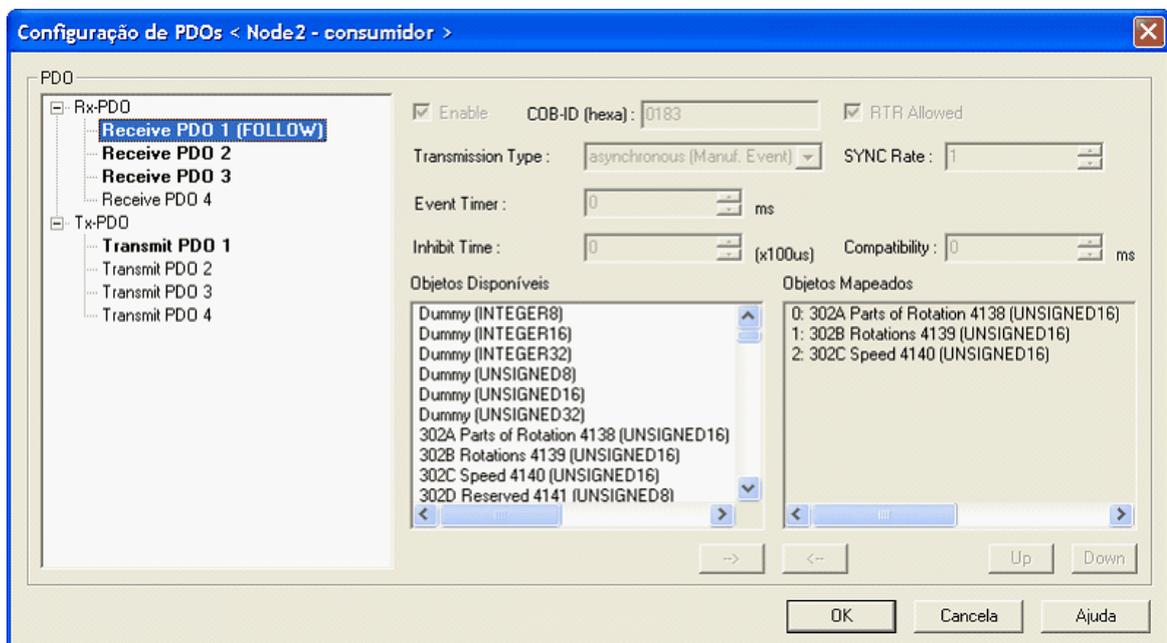
OK Cancelar Aplicar Ajuda

As definições realizadas através da janela de Configuração Follow podem ser observadas na janela Configuração de PDOs.

A figura a seguir ilustra a configuração do PDO de transmissão (TPDO 01) no produtor Follow. Todas as opções de configuração do TPDO 01 são travadas, pois a função Follow está habilitada.



A figura a seguir mostra a configuração do PDO de recepção (RPDO 01) no consumidor Follow. Todas as opções de configuração do RPDO 01 são travadas, pois a função Follow está habilitada.

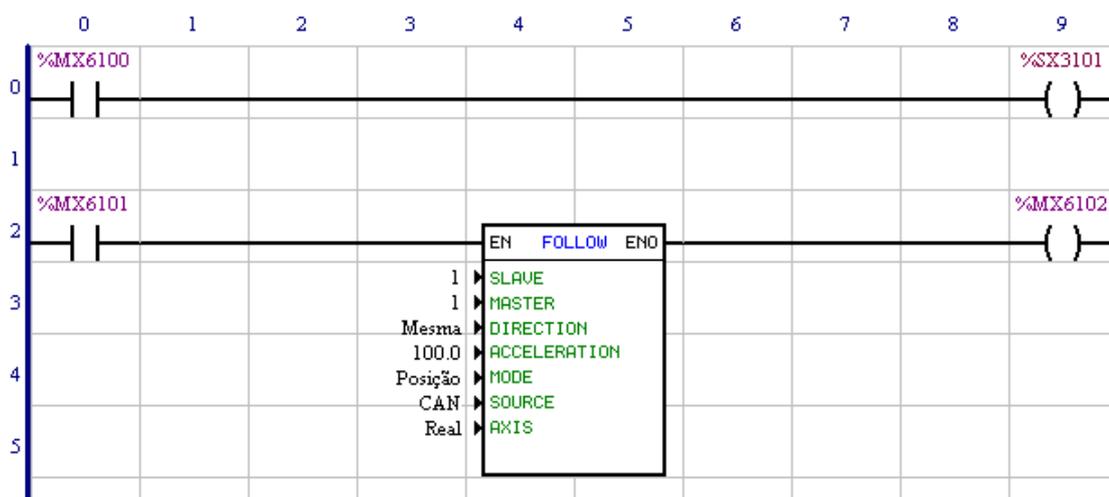


A taxa de comunicação e o número de telegramas na rede CANopen podem influenciar na precisão do posicionamento. Recomenda-se utilizar a função Follow com no mínimo 250bps de taxa de comunicação.

Quando a função Follow for habilitada em uma placa PLC11 e esta estiver operando como mestre CANopen, a mesma deverá ser inicializada toda vez que a função Follow for modificada, de Produtor para Consumidor ou Consumidor para Produtor.

3.17.3 Configuração WLP

O Bloco Follow é utilizado para configurar o sincronismo no consumidor. Para maiores detalhes consultar o Help do WLP.



3.17.4 Parametrização CFW11 e PLC11

CFW11:

P0202 – selecionar controle por encoder (4); (Produtor e Consumidor)

P0221 ou P0222 – parametrizar um dos parâmetros para PLC11 como referência; (Produtor e Consumidor)

PLC11:

P1262 – informar a resolução do encoder principal; (Produtor e Consumidor)

P1285 – Habilitar protocolo CANopen (1); (Produtor e Consumidor)

P1286 – Endereço CAN; (Produtor e Consumidor)

P1287 – Taxa comunicação; (Produtor e Consumidor)

Índice

- A -

Abreviações e definições 6
Acesso a rede 8
Acesso aos dados 10
Alterando a base numérica 35
Alterando o valor de objetos 36
Arquivo EDS 12

- B -

Barra de botões 15
Bloco ladder (SDO) 56

- C -

CAN 7
CAN e CANopen 9
Características da rede CANopen 9
COB-ID 11
Comando do escravo no ladder do mestre 63
Conectando o produtor (TxPDO) ao consumidor (RxPDO) 45, 46
Conexões PDOs 60
Configuração de PDOs do mestre/escravos 36
Configuração do escravo (SCA-05) 57
Configuração mestre (PLC2) 57
Configurando o mestre 19
Configurando os escravos 25
Configurando uma nova rede 17
Controle da comunicação CANopen 56
Controle de erros 8

- D -

Device Tree 16
Dicionário de objetos do escravo 33
Documentos 7

- E -

EMCY 28
Endereço do escravo destino 56
Endereço na rede CANopen 9
Estado da comunicação CANopen 54
Estado do escravo CANopen 55
Executando o software WSCAN 12
Exemplo de configuração de rede 57

- F -

FOLLOW 24, 32
Frame de dados 8
Frame remoto 8

- G -

Gerar configuração 48

- I -

Informações do objeto 34
Interface com a programação ladder 52

- J -

Janela principal 14

- L -

Leitura do status do escravo no ladder do mestre 62
Lista de objetos 33
Lista de PDOs 37
Listando as conexões 48
Log Window 16

- M -

Mapeamento de memória de I/O 53
Mapeamento de objetos no PDO 40
Meio físico 9
Menu principal 14
Monitoração online 49

- N -

NMT 20, 26
NODE GUARDING 26

- O -

O que é o WSCAN 6
Objetos responsáveis pela comunicação - COBs
10

- P -

Palavras de comando e estado da comunicação
53
Parâmetros de comunicação do PDO 38
PDOs do escravo (SCA-05) 58
PDOs do mestre (PLC2) 57
Principais características WSCAN 7
Procedimento para efetuar uma conexão entre
PDOs 41
PRODUTOR HEARTBEAT 22, 27

- R -

Rede CANopen 17
Representação numérica 7

- S -

SAVE/RESTORE 31
SDO 21
Selecionando o consumidor (RxPDO) 43
Selecionando o produtor (TxPDO) 42
Sobre o manual 6
SYNC 20

- T -

Transmissão de dados 10
Transmitir configuração 49

- V -

Visualizador/configurador de conexões entre PDOs
41