CANopen - Mestre de Rede PLC500

Nota de Aplicação







Mestre CANopen - Nota de Aplicação

Série: PLC500 Idioma: Português Documento: 10010165216 / 00

Data de publicação: 10/2022



A informação abaixo descreve as revisões ocorridas neste manual.

Versão	Revisão	Descrição
-	R00	Primeira edição



Sumário

1	INT	IRODUÇÃO	5
	1.1	COMPONENTES DA REDE CANOPEN	5
	1.2		5
	1.3	ARQUITETURA DE REDE	5
2	СС	NFIGURAÇÃO CANOPEN MESTRE	7
	2.1	INICIANDO O PROJETO NO CODESYS	7
3	СС	NFIGURAÇÃO CANOPEN	10
	3.1	CONFIGURAÇÃO DA REDE	10
	3.2	CONFIGURAÇÃO E/S	11
	3.3	DECLARAÇÃO DE ENDEREÇOS	11
	3.4	RESISTORES DE TERMINAÇÃO	12
4	MC	ONITORAÇÃO	13
	4.1		13
	4.2	ERROS DE COMUNICAÇÃO	13
	4.3		14

1 INTRODUÇÃO

Esta nota de aplicação destina-se a fornecer uma descrição de como operar o PLC500 como mestre de uma rede CANopen. Para as configurações do controlador programável PLC500, recomenda-se seguir as etapas descritas neste documento para configurar a rede CANopen de maneira apropriada.



ATENÇÃO!

Esta nota de aplicação é direcionado para profissionais treinados em redes industriais. A instalação e configuração dos dispositivos deve ser feita de acordo com o manual do fabricante.

1.1 COMPONENTES DA REDE CANOPEN

Para os componentes passivos da rede - cabos, conectores, resistores de terminação, fonte de alimentação - recomenda-se utilizar somente componentes certificados para aplicações industriais. Consulte a documentação dos produtos para obter informações sobre a instalação adequada da rede CANopen.

Para uma descrição mais profunda e detalhada sobre a operação da rede CANopen e as suas configurações, é aconselhável acessar a ajuda online do **Codesys** em: *https://help.codesys.com*.

1.2 INTERFACE CANOPEN

A Figura 1.1 apresenta o PLC500 com a interface CANopen em evidência. A indicação dos pinos da interface CANopen é descrita na Tabela 1.1.



Figura 1.1: Indicação dos pinos do conector CAN no PLC500.

1	
I -	Comum da rede CAN(ligado ao polo negativo da rede CANopen)
L L	Sinal de comunicação CAN_L
.D S	Blindagem do Cabo
н н	Sinal de comunicação CAN_H
+	Não conectado (pode receber o polo positivo da rede CANopen)
	_L L _D S _H H +

Tabela 1.1: Descrição interface CANopen.

1.3 ARQUITETURA DE REDE

A Figura 1.2 apresenta os componentes e a arquitetura da rede CANopen. Neste exemplo é mostrada a conexão de um PLC500 (mestre de rede) e uma RUW100 (escravo de rede). O computador realiza a programação dos dispositivos, sendo o CODESYS no PLC500 e o WPS para RUW100.



Figura 1.2: Componentes e arquitetura da rede CANopen.

NOTA!

Este manual de aplicação é direcionado ao PLC500 e a ferramenta de programação **Codesys**, recomendamos utilizar o **Codesys V3.5 SP18 ou superior**. Caso necessite de mais informações sobre o protocolo de comunicação CANopen, consulte os manuais do mesmo.



2 CONFIGURAÇÃO CANOPEN MESTRE

2.1 INICIANDO O PROJETO NO CODESYS

Para as configurações e definições de uma rede CANopen deve-se inicialmente criar o projeto e incluir o controlador programável PLC500. No software Codesys, crie um novo projeto, escolha o diretório e o nome da aplicação. Depois, selecione o dispositivo PLC500-WEG e a linguagem de programação desejada, conforme a Figura 2.1.

New Project	×	
Categories Iemplates		
A project containing one device, one application, and an empty implementation for PLC_PRG	i	You are about to create a new standard project. This wizard will create the following objects within this project: One programmable device as specified below One programmable device as specified below One programme PLC_PRG One are PLC_PRG One are programmable to the standard library currently installed.
Name Example Location C: 'Users'user'Documents/CODESYS OK Cancel		Device PLC500-WEG (WEG Drives & Controls) PLC_PRG in Ladder Logic Diagram (LD) OK Cancel

Figura 2.1: Configuração do projeto no Codesys.

NOTA!

Caso o dispositivo PLC500 ainda não estar disponível nas opções do Codesys, deve-se baixar e instalar o arquivo de configuração, confira o **Manual do Produto** para encontrar os passos e configurações necessárias.

Com o dispositivo PLC500 selecionado, resultará em um projeto com as interfaces de redes disponíveis já pré-configuradas, como indicado na Figura 2.2.



Figura 2.2: Interfaces PLC500.

No próximo passo deve-se adicionar o dispositivo CANopen_Manager, como indicado na Figura 2.3.

Devices		- ₽ X	A III	dd Device				×
🗏 🎒 Example		•	_					
🖮 📶 Device (PLC500-WEG	5)		Name	e CANopen Manager				
🗉 📳 PLC Logic			Act	ion				
🔤 Setup (Setup)			ALL		Diversion of a second		a data da star	
I_Os (I/Os)				Append device O Insert device O) Plug devic		pdate device	
🗄 🛄 Expansions (Expa	ansio	ns)	Stri	ng for a full text search	V	endor	<all vendors=""></all>	~
ETH1 (ETH1)							Child Chadrey	
ETH2 (ETH2)			Na	ame		Vend	lor	Versio ^
CAN (CAN)	ж	Cut		Fieldbuses				
👚 😭 RS485 (RS485)	Ba	Conv		CANopen				
		Dasta		CANopenManager				
		Paste		CANopen_Manager		3S - S	Smart Software Solutions GmbH	3.5.17
	~	Delete		CANopen_Manager_Si	IL2	3S - S	Smart Software Solutions GmbH	3.5.17
		Refactoring	•	CANopen_Manager_S	oftMotion	3S - 5	Smart Software Solutions GmbH	3.5.17
	e.	Properties		GIII Local Device				J
	4:2	Topenesii	<	m - 242 11939				>
	1.0.0	Add Object		Group by category Display all ver	sions (for e	experts o	nly) Display outdated versi	ons
		Add Folder				· · ·		
		Add Device		Name: CANopen_Manager Vendor: 35 - Smart Software Solut	tions GmbH		^	
		Disable Device		Categories: CANopenManager				-
		Update Device		Version: 3.5.17.0 Order Number:				
	fî"	Edit Object		Description: CANopen Manager			v ~)
	_	Edit Object With						
		-	App	pend selected device as last child o	of			
		Edit IO mapping	CAI	N				
		Import mappings from CSV	···· 0	(You can select another target node	e in the nav	rigator w	hile this window is open.)	
		Export mappings to CSV						
<	_	>					Add Device	Close
🛫 Devices 📋 POUs								

Figura 2.3: Configuração CANopen Etapa 1.

Em **CANopen_Manager**, adiciona-se o dispositivo escravo de rede CANopen. A Figura 2.4 apresenta os passos descritos acima para esta configuração. Neste caso, seleciona-se a RUW100 como exemplo de escravo de rede.

Devices -	ą 🗙							
Example	-	•	_					
🖻 🔟 Device (PLC500-WEG)			1	Add Device				\times
PLC Logic								
E Setup (Setup)			Nar	me RUW01				
I_Os (I/Os)			A	Action				
Expansions (Expansions)				Append device O Insert device O Plug		Indate device		
ETH1 (ETH1)						pulle derice		
ETH2 (ETH2)			St	tring for a full text search	Vendor	<all vendors=""></all>		\sim
CAN (CAN)					_			
CANopen_Manager (CANopen_Manager)	X	Cut		Name			Vendor	
	80	Conv		LexiumSD3_SoftMotion			Schneider Electric	
		Сору		nanotec CANopen DS401 PD4	C_SoftMotio	n	Nanotec Electroni	c
		Paste		III RUW01			WEG	
	\mathbf{X}	Delete		SMCI47S_SoftMotion			nanotec	
		Refactoring +		SVMCAN CMZ CANopen node	_SoftMotion		CMZ Sistemi Elettr	°C I
				SVMCAN CM2 CANopen node	_Softmotion i	Encoder	CM2 Sistemi Elettr	C III
	ЦШ.	Properties		ucan sphDrive SoftMotion			Brunner Elektronik	· •
	122	Add Object		c			>	
	\bigcirc	Add Folder		Group by category Display all versions (for experts o	nly) 🗌 Display out	tdated versions	
		Add Device						
		Insert Device	l l "	Vendor: WEG		^		
		Scan for Devices		Categories: Remote Device Version: Revision = 16 #00000078 FileVer	sion=1.0			
		Disable Device		Order Number: contact WEG			S	
		Update Device		Description: Imported from CO_RUW01_	V12X.eds	×		
	E°.	Edit Object						-1
		Edit Object With	A	ppend selected device as last child of				
			0	ANopen_Manager				
		Edit IO mapping	6	 (You can select another target node in the 	navigator w	hile this window is o	pen.)	
		Import mappings from CSV						
<		Export mappings to CSV				Add Dev	rice Clos	e
Revices POUs			-		_			_

Figura 2.4: Configuração CANopen Etapa 2.

Na primeira vez que algum equipamento é adicionado como dispositivo escravo de rede CANopen, deve-se obter o arquivo EDS respectivo do produto e importá-lo no Codesys. A instalação do arquivo EDS deve ser feita no Codesys em: **Tools -> Device Repository -> Install** e selecionar o arquivo EDS desejado. Todos os recursos podem ser encontrados na página do produto em https://www.weg.net.





NOTA! Electro

Electronic Device Description (EDS) é o arquivo necessário para descrever o dispositivo escravo para o mestre de rede CANopen, o fabricante deve fornecer este arquivo para permitir a correta configuração deste dispositivo quando disponível a opção de escravo da rede.

Neste momento a interface CAN deve possuir os itens indicados na Figura 2.5.



Figura 2.5: Interface CAN atuando como mestre CANopen.

3 CONFIGURAÇÃO CANOPEN

3.1 CONFIGURAÇÃO DA REDE

A rede deve ser configurada no mestre para atender os endereços definidos nos escravos de rede. Para isso, deve-se definir os parâmetros e endereços que serão transmitidos a partir do mestre de rede. Cada dispositivo escravo deve ser configurado com um endereço único e na mesma taxa de transmissão definida, caso contrário, a rede apresentará problemas. Na Figura 3.1, são definidas as configurações do dispositivo mestre.

CAN X M CANopen_Manager	RUW01		
General	General		
Log	Network	0	CAN
CANbus IEC Objects	Baud rate (kbit/s)	250 ~	
Status			
Information			

Figura 3.1: Configuração Mestre CANopen.

A configuração de rede para o dispositivo escravo está indicada na Figura 3.2. Cada escravo também deve possuir um endereço próprio e estar dentro do intervalo de endereços disponíveis para os escravos. Desta forma, as configurações para iniciar a comunicação entre os dispositivos mestre e escravo estão completas.

General	General	
PDOs	Node-ID 2 SDO Chann	rels (1/1 Active)
SDOs	Enable expert settings Optional device	e
Log	Enable SYNC producing No initialization	n Reset node 🗸
CANopen I/O Mapping	▲ Guarding	
	Enable nodeguarding	Enable heartbeat producing
CANopen IEC Objects	Guard time (ms) 0	Producer time (ms) 200
Status	Life time factor 0	✓ Heartbeat consuming (1/4 active)
Information	▲ Emergency (EMCY)	I TIME
	Enable emergency (EMCY)	Enable TIME producing
	COB-ID 0	COB-ID (Hex) 16# 100 🐥
		Enable TIME consuming
	▲ Checks at Startup	
	Check vendor ID Check product num	ber 🔲 Check revision number

Figura 3.2: Configuração CANopen do dispositivo escravo.

Como este exemplo utiliza o dispositivo RUW100, configura-se o endereço e a taxa diretamente nas chaves DIP da parte exterior do produto.

NOTA! O méto

O método de configuração pode ser diferente para cada equipamento, variando com o tipo de produto e o software de programação. Configure o dispositivo escravo de acordo com o manual disponibilizado pelo fabricante.

3.2 CONFIGURAÇÃO E/S

Variáveis e endereços configurados para um escravo devem ser programados levando em consideração o mestre daquela rede CANopen. Como indicado na Figura 3.3, após a importação do escravo de rede CANopen no Codesys, o conjunto de PDOs (Process Data Object) ficará disponível de acordo com cada dispositivo.

eneral	Receive PDOs (Master => Slave)	Transmit PDOs (Slave => Master)
	🕂 Add PDO 🕂 Add Mapping 💉 Edit 🗙 Delete 🛧 Move Up	🖡 Add PDO 🕂 Add Mapping 💉 Edit 🗙 Delete 🛧 Move Up
DOs	Name	Name
00s	16#1400: Receive PDO Communication Parameter 1	✓ 16#1800: Transmit PDO Communication Parameter 1
	Write Output 1 to 8	Read Input 1 to 8
og	Write Output 9 to 16	Read Input 9 to 16
	✓ 16#1401: Receive PDO Communication Parameter 2	✓ 16#1801: Transmit PDO Communication Parameter 2
ANopen I/O Mapping	Write Output 1 to 16	Read Input 1 to 16
	✓ 16#1402: Receive PDO Communication Parameter 3	✓ 16#1802: Transmit PDO Communication Parameter 3
ANopen IEC Objects	Write Analogue Output 1	Read Analogue Input 1
The base	Write Analogue Output 2	Read Analogue Input 2
status	Write Analogue Output 3	Read Analogue Input 3
nformation	Write Analogue Output 4	Read Analogue Input 4
	16#1403: Receive PDO Communication Parameter 4	✓ 16#1803: Transmit PDO Communication Parameter 4
	Write Analogue Output 5	Read Analogue Input 5
	Write Analogue Output 6	Read Analogue Input 6
	Write Analogue Output 7	Read Analogue Input 7
	Write Analogue Output 8	Read Analogue Input 8
	16#1404: Receive PDO Communication Parameter 5	16#1804: Transmit PDO Communication Parameter 5
	16#1405: Receive PDO Communication Parameter 6	V 16#1805: Transmit PDO Communication Parameter 6

Figura 3.3: Lista dos PDOs disponíveis no dispositivo escravo.

Para o equipamento RUW100 temos 32 PDOs para envio e outros 32 PDOs para recepção de dados, sendo que podem ser configurados individualmente para transmitir até 8 bytes de dados. Estes itens estão descritos no **Manual do Usuário** da RUW100.



NOTA!

Para cada aplicação podem surgir configurações e endereços próprios. Dessa forma, o fabricante deve informar a disponibilidade dos PDOs de acordo com o equipamento. Recomenda-se verificar o manual do produto específico para mais detalhes.

3.3 DECLARAÇÃO DE ENDEREÇOS

Pode-se definir com endereços de transmissão quais dados são relevantes para a aplicação. No exemplo da Figura 3.4 escrevemos algum valor nas saídas e realizamos uma leitura logo em seguida.



3.4 RESISTORES DE TERMINAÇÃO

Para a rede CANopen deve-se utilizar resistores nas suas terminações, sendo essenciais para o correto funcionamento. A configuração deste item é feita apenas através do **Codesys** para o PLC500, deve-se seguir para **Setup** e dentro de **Setup Parameters** onde estão os campos de **Termination Resistors**, como indicado na Figura 3.5.

🔣 Setup 🗙						
Setup Parameters	Parameter	Туре	Value	Default Value	Unit	Description
	🖃 🗀 Firmware					
PLC Setup IEC Objects	Firmware version	STRING	'Not connected'	'Not connected'		PLC's firmware version
Satup I/O Mapping	🖤 🖗 Update available	STRING	'Not connected'	'Not connected'		Firmware version available in PLC for update
Secup t/o happing	🛛 🕸 Update	Enumeration of BYTE	No	No		Select "Yes" to update PLC's firmware with t
	🖃 🚞 Date and Time					
	🔷 🕸 Date	STRING	'2021-12-31'	'2021-12-31'		Read PLC date value, use the format '2021-
	🖉 Time	STRING	'00:00:00'	'00:00:00'		Read PLC time value, use the format '12:59:
	🗐 🗀 Termination Resistors					
	🔷 🖗 RS485	Enumeration of BYTE	Not Connected	Not Connected		RS485 termination tesistor configuration
	🖗 CAN	Enumeration of BYTE	Not Connected	Not Connected		CAN termination resistor configuration

Figura 3.5: Página de configuração dos resistores de terminação.

MONITORAÇÃO 4

MONITORAÇÃO DE VARIÁVEIS 4.1

A Figura 4.1 mostra a tela de visualização de variáveis do Codesys para o escravo de rede utilizado.

General	Find		Silver Channell					
			ziner show all			- 🕂 Add FB for IO Cha	annel 🔭 Go to Instance	
	Variable	Mapping	Channel	Address	Туре	Current Value	Prepared Value Unit	Description
PDOS		×.	Write Output 1 to 8	%OB2	USINT	1		
SDOr	8-50		Write Output 9 to 16	%OB3	USINT			
3003	8-50		Write Output 1 to 16	%QW2	UINT			
Log	· · · · ·		Write Analogue Output 1	%QW3	INT			
	······		Write Analogue Output 2	%QW4	INT			
CANopen I/O Mapping	·····		Write Analogue Output 3	%QW5	INT			
	B- 50		Write Analogue Output 4	%QW6	INT			
CANopen IEC Objects	iii- * ≱		Write Analogue Output 5	%QW7	INT			
a	iii- * ø		Write Analogue Output 6	%QW8	INT			
Status	10 - 5 0		Write Analogue Output 7	%QW9	INT			
Information	i⊞ * ø		Write Analogue Output 8	%QW10	INT			
	B ∛≱ INPUT	N 10	Read Input 1 to 8	%IB34	USINT	0		
	18 *9		Read Input 9 to 16	%IB35	USINT			
	· · · · · ·		Read Input 1 to 16	%IW18	UINT			
			Read Analogue Input 1	%IW19	INT			
	18 - Mp		Read Analogue Input 2	%IW20	INT			
	18 *9		Read Analogue Input 3	%IW21	INT			
	iii - ¥≱		Read Analogue Input 4	%IW22	INT			
			Read Analogue Input 5	%IW23	INT			
	18- 3 9		Read Analogue Input 6	%IW24	INT			
	i≣- * ≱		Read Analogue Input 7	%IW25	INT			
	i ⊞ - 🍫		Read Analogue Input 8	%IW26	INT			
11	1.5							
	SDOs Log CANopen I/O Mapping CANopen I/C Objects Status Information	SDOs * * • Log * * • CANopen I/O Mapping * • CANopen I/O Mapping * • Sature * • Jadromation * • • • • • •	SDOs Log CANopen I/C Objects Status Information Information	SD0s Log UM Applied CANopen I/CO Objects Status Information Inform	SDOs Mitte Output 18 to 16 %(R03) Log Witte Output 18 to 16 %(R03) CANopen I/C Objects Witte Output 18 to 16 %(R03) CANopen I/C Objects Witte Output 18 to 16 %(R03) Status Witte Output 18 to 16 %(R04) If ormation Witte Andopac Output 18 '%(V04) %(V04) Witte Andopac Output 18 '%(V04) %(V04) %(V04) Status Witte Andopac Output 18 '%(V04) %(V04) If ormation Witte Andopac Output 18 '%(V04) %(V04) Witte Andopac Output 18 '%(V04) %(V04) <	SD0s Corror Wite Output 9 to 16 %CR3 USINT Log Wite Output 1 to 16 %CR3 USINT CANopen I//O Mapping Wite Analogue Output 3 %CW3 BitT CANopen I//O Mapping Wite Analogue Output 3 %CW3 BitT CANopen I//O Mapping Wite Analogue Output 3 %CW3 BitT CANopen I//O Mapping Wite Analogue Output 3 %CW3 BitT Status Wite Analogue Output 3 %CW3 BitT Information Wite Analogue Output 4 %CW3 BitT Wite Analogue Output 3 %CW3 BitT %CW3 BitT Information % Wite Analogue Output 4 %CW3 BitT % Wite Analogue Output 5 %CW3 BitT %CW3 BitT % Wite Analogue Output 7 %CW3 BitT %CW3 BitT %CW3 BitT % BitT %Ead Invit 1 to 16 %CW3 BitT %CW3 BitT % BitT Read Analogue Input 1	SDGs Mithe Jougue 100 Mithe Jougue 300 Mithe Jougue 300	SDGs Interception Interception Interception Log White Output 19 is 16 Weige Output 19 USBNT Net updated Log White Output 19 is 16 Weige Output 19 USBNT Net updated CANopen IJO Mapping White Analogue Output 1 Status White Analogue Output 1 Weige Analogue Output 1 Net updated CANopen IJC Objects White Analogue Output 2 WOVE Analogue Output 3 Status Net updated Interception Satus White Analogue Output 4 WOVE Analogue Output 5 Net updated Interception Information White Analogue Output 5 NEQVID Net updated Interception Vertex Analogue Output 5 NEQVID 1 Net updated Interception Interception Information Read Provid 10 8 Weite Analogue Output 5 NEQVID 1 Net updated Interception Read Provid 10 16 Weite Analogue Dutput 5 NEQVID 2 Net updated Interception Read Provid 10 16 NetWite Analogue Dutput 5 NEQVID 2 Net updated Interceptin Read Provid 1

Figura 4.1: Monitorando variáveis no Codesys.

Pode-se monitorar as variáveis declaradas na rede através de dois métodos, primeira possibilidade seria adicionar no programa do Codesys e monitorar online os valores das variáveis, no segundo método requer apenas que habilite Always Update Variables no rodapé da página anterior, como mostrado na seta vermelha da Figura 4.2.

	General	Find		Filter Show all			•	Add FB for IC	O Channel → Go to Instance	
l	PDOs	Variable	Mapping	Channel	Address	Туре	Unit	Description		
	1003	😟 🍢		1. Digital Output 8-Bit	%QB26	USINT				
	SDOs	🕮 - 🍢		2. Digital Output 8-Bit	%QB27	USINT				
		±*		1. Digital Output 16-Bit	%QW14	UINT				
	Log	🕸 - 🍫		1. Digital Input 8-Bit	%IB34	USINT				
				2. Digital Input 8-Bit	%IB35	USINT				
	CANopen I/O Mapping			1. Digital Input 16-Bit	%IW18	UINT				
	CANopen IEC Objects									
	Status									
	Status									
	Status Information						_			
	Status							! Ŀ		
	Status									
	Status						5			
	Status			Reset Ma	apping A	lways upda	atevariab	les Enabled 1 ((use bus cycle task if not used in any	v task)
	Status			Reset Ma	apping A	lways upda	atevariab	les Enabled 1 ((use bus cycle task if not used in any	/ ta

Figura 4.2: Monitorando variáveis no Codesys.

4.2 ERROS DE COMUNICAÇÃO

O monitoramento dos estados das redes pode ser feito em Devices, indicando o estado de cada uma das etapas de comunicação e reportando o estado (Status). Ao encontrar problemas de conexão, como mostrado na Figura 4.3, verifique se os cabos estão devidamente conectados, qual o estado do LED CAN no PLC500, e então revise as configurações realizadas.



Figura 4.3: Indicação do erro de comunicação com o dispositivo escravo.

Conectado ao PLC500, acesse as abas **Status** e **Log** indicadas na Figura 4.4. Nesse local o Codesys informará qual problema está encontrando para realizar a comunicação. Quando as configurações estiverem corretas, e os dispositivos identificarem um ao outro, devemos observar todos os demais itens em verde.

Devices 👻 🕂 🗙	RUW01 X	
Example	Conorol	
Device (PLC500-WEG)	General	CANopen :
国一国山 PLC Logic	PDOs	Last diagnostic message
I Os (I/Os)		Diag Christer
Expansions (Expansions)	SDOs	Diag String:
	Log	
ETH2 (ETH2)		
CAN (CAN)	CANopen I/O Mapping	
RUW01 (RUW01)	CANopen IEC Objects	
RS485 (RS485)		
	Status	
	Information	
< >>		
Sevices POUs		

Figura 4.4: Páginas de Status e Log contidas em cada um dos itens das interfaces de comunicação.

4.3 LED CAN INDICATIVO

O controlador programável PLC500 possui LEDs na parte frontal do produto que são utilizados para indicação das interfaces. A indicação de cores do LED indicado na Figura 4.5 é descrito na Tabela 4.1.



Figura 4.5: LEDs indicadores do PLC500.

LED CAN	STATUS	DESCRIÇÃO
APAGADO	Sem configuração.	Não existe uma configuração de co- municação CANopen na aplicação.
VERDE	Comunicação CANopen sem falha.	Comunicação CANopen confi- gurada na aplicação e todos os elementos rodando sem erro/falha.
LARANJA	Comunicação CANopen com falha parcial.	Comunicação CANopen configu- rada na aplicação e com alguns elementos com erro/falha.
VERMELHO	Comunicação CANopen com falha total.	Comunicação CANopen confi- gurada na aplicação e todos os elementos com erro/falha.

Tabela 4.1: Funcionamento do LED indicativo da interface CAN.



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