Modbus/TCP - Escravo de Rede PLC500

Nota de Aplicação







Escravo Modbus/TCP - Nota de Aplicação

Série: PLC500 Idioma: Português Documento: 10010165938 / 00

Data de publicação: 10/2022



A informação abaixo descreve as revisões ocorridas neste manual.

Versão	Revisão	Descrição
-	R00	Primeira edição

Sumário

1	INTRODUÇÃO	5
	1.1 COMPONENTES DE REDE MODBUS/TCP	. 5
	1.2 INTERFACE ETHERNET	. 5
	1.3 ARQUITETURA DE REDE	. 5
2	CONFIGURAÇÃO MODBUS/TCP	. 7
	2.1 CRIANDO PROJETO NO CODESYS	. 7
3	CONFIGURAÇÃO MODBUS/TCP	9
	3.1 CONFIGURAÇÃO DA REDE	. 9
	3.2 CONFIGURAÇÃO E/S	. 9
4	MONITORAÇÃO	12
	4.1 MONITORAMENTO DE VARIÁVEIS	. 12
	4.2 ERROS DE COMUNICAÇÃO	. 12

1 INTRODUÇÃO

Esta nota de aplicação destina-se a fornecer uma descrição simplificada de como operar o PLC500 para comunicar em rede Modbus/TCP, aqui demonstraremos a configuração para **escravo de rede**. Para as configurações do produto PLC500, deve-se seguir as etapas descritas neste documento para configurar a rede Modbus/TCP 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 DE REDE MODBUS/TCP

Para os componentes passivos de rede - cabos, conectores, fonte de alimentação, switchs, hubs - recomendamos utilizar somente componentes certificados para aplicações industriais. Consulte a documentação dos produtos para obter informações sobre a instalação adequada da rede Ethernet.

Para uma descrição mais profunda e detalhada sobre a operação da rede Modbus/TCP e as suas configurações, é aconselhável acessar a página web provida pela Codesys em: https://help.codesys.com/.

1.2 INTERFACE ETHERNET

Como indicado na Figura 1.1, utiliza-se conexões Ethernet para a comunicação Modbus/TCP. Inicialmente cada uma das portas possui o endereço IPv4 indicado na Tabela 1.1, podendo ser alterado a qualquer momento pelo *software Codesys* ou ainda através da página Web.



Figura 1.1: Conector para interface Ethernet.

Tabela 1.1: Endereço inicial para cada porta Ethernet.

Conexão	Endereço de IPv4 padrão
ETH1	192.168.1.10
ETH1	192.168.2.10

1.3 ARQUITETURA DE REDE

Na Figura 1.2 os dispositivos são programados pelo computador e comunicam entre si através do comutador de rede Ethernet. Para as configurações do produto PLC500, deve-se seguir as etapas descritas neste documento para configurar a rede Modbus/TCP de maneira apropriada.



Figura 1.2: Network components.



Codesys:

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 Modbus/TCP, consulte os manuais do mesmo.



2 CONFIGURAÇÃO MODBUS/TCP

2.1 CRIANDO PROJETO NO CODESYS

No *software* Codesys, crie um novo projeto e escolha o diretório e nome da aplicação, Depois, selecione o dispositivo PLC500-WEG e linguagem de programação desejada, conforme a Figura 2.1.

🖹 New Pro	ject					×				
Categories	raries ojects	Templates	HMI project	Standard project	Standard project w					
A project co	ontaining one device, one ap	plication, and an e	empty implemen	tation for PLC.	_PRG		Standard I	Project You are abou objects withi - One progra - A program - A cyclic tas - A reference	ut to create a new standard project. This wizard will create the following in this project: mmable device as specified below PLC_PRG in the language specified below k which calls PLC_PRG to the newest version of the Standard library currently installed.	×
<u>N</u> ame <u>L</u> ocation	Example C:\Users\user\Documents	(CODESYS			~		-	<u>D</u> evice <u>P</u> LC_PRG in	PLC500-WEG (WEG Drives & Controls) Ladder Logic Diagram (LD)	> >
				ОК	Cancel				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.

Selecionando o dispositivo PLC500, deve resultar em um projeto com as redes disponíveis já pré-configuradas, como indicado na Figura 2.2



Figura 2.2: Interfaces de rede para o PLC500.

Na interface ETH1 ou ETH2 adicione "Modbus TCP Slave", como indicado na Figura 2.3.

×

De

Close

Version





Figura 2.3: Codesys - Configuração Modbus/TCP Etapa 1.

Neste momento a interface ETH escolhida deve possuir os itens indicados na Figura 2.4.

<



Figura 2.4: Codesys - Configuração Modbus/TCP Etapa 2.

3 CONFIGURAÇÃO MODBUS/TCP

3.1 CONFIGURAÇÃO DA REDE

Cada dispositivo escravo deve se configurar com endereço único e selecionar a faixa de endereços IPv4 permitida, caso contrário, a rede apresentará problemas. Na Figura 3.1 são definidas as configurações do dispositivo mestre, selecionando a interface para configurar o endereço IPv4 e o restante necessário.

ModbusSlave.project* - CODESYS									-		×
File Edit View Project Build Online Debug 🖺 🗃 💭 🕼 🕼 🗠 🖉 🍓 🏠	Tools Window Help 실 비 위계계(哈吉)는 습니	📆 Application	[Device: PLC Lo	ogic] • 😋 C	8 -	ÇI GI dI d	≡ \$ ¢ p	5 ½			6
Devices - 4 2	Device PLC_PRG	Mail Ethernet	×								- >
- ModbusSlave	-										77
😑 👔 Device [connected] (PLC500-WEG)	General	Network inter	face eth1			Browse					
DLC Logic DLC Logic Application	Log	IP address	192 .	168 . 1 .	10						
Ubrary Manager DLC_PRG (PRG)	Status	Subnet mask	255 .	255 . 255 .	0						
🖻 🧱 Task Configuration 🗟 🌚 MainTask	Ethernet Device I/O Mapping	Adjust op	erating system:	ettings	0						
B PLC_PRG	Ethernet Device IEC Objects		Network Ada	pters						×	
Setup (Setup)	Information		Interfaces								
📊 I_Os (I/Os)	· · · · · · · · · · · · · · · · · · ·		Name	Description	IP address						
Expansions (Expansions)			reserv1		127.0.0.1						
- iii Ethernet (Ethernet)			eth1		192.168.1.10						
ModbustCP_Slave_Device (ModbustCP Slave L	Messages - Total 4 error(s), 0 warning(s). 36 message(s)	eth2		192.168.2.10					a	×
	Download	-,,	reserv2		0.0.0.0						
			usb2		192.168.234.23	4					۰.
	Description									- 1.	
	 A core dump created on the 14/02/ 	2019 08:35:00 for	IP address	192 .	168 . 1 . 1	0					
			Subnet mask	255	255 . 255 .	0					
			Default gate	vay 0.	0.0.	0					
			MAC addres	s 00:01:	C0:28:53:13						
C Set Devices	>								OK Cancel		
- searce roos				(S Lacthui	H- O O O O	Precomole 🔿	0- 64	Project users (exhedu)		0

Figura 3.1: Configuração de rede para o dispositivo programado no Codesys.

Realizada esta etapa, configura-se a rede para o dispositivo escravo. Cada um deve possuir seu endereço próprio e estar dentro do intervalo de endereços IPv4 disponíveis, o restante das configurações como taxa de transmissão serão gerenciadas automaticamente pela rede.

3.2 CONFIGURAÇÃO E/S

Antes de declarar as variáveis de rede, deve-se selecionar o valor de **Offset** dos endereços. Os endereços podem ser definidos de maneira independente para cada função, como pode ser visto em **Start Addresses** dentro do retângulo vermelho na Figura 3.2.

ModbusTCP_Slave_Device 🗙			
General	Configured Parameters		
	Watchdog	500 🚖 (ms) 🗹 close TCP socket	
Serial Gateway	Slave port	502 🖨 🗌 Bind to adapter	
Modbus TCP Slave Device I/O Mapping	Holding registers	10 (%QW) Vriteable	
Modbus TCP Slave Device IEC Objects	Input registers	10 🔶 (%QW)	
Status	Discrete Bit Areas		
Status	Coils	0 (%QX)	
Information	Discrete Inputs	0 (%QX)	
	Data Model		
	Startadoresses		
	Cons		
	Discrete inputs		
	Holding register	20000	
	Input register	0	
	Holding- and input reg	gister data areas overlay	

Figura 3.2: Declaração das variáveis transmitidas pelo dispositivo escravo de rede.

Nesta etapa declaramos os endereços de memória reservados para as variáveis de transmissão, fazendo a leitura de um valor no registrador do escravo de rede e escrita em outro. Na Figura 3.3 abaixo estão as funções Modbus para leitura e escrita destes dados no dispositivo escravo.



Figura 3.3: Configurações dos Canais de Transmissão.

O número de registradores e bobinas declaradas da Figura 3.3 estão indicadas na página na Figura 3.4. Levar em consideração a diferença entre os **Registers** e **Coils**, as bobinas ficam em endereços separados e devem ser habilitadas com **Discrete Bit Areas**.

neral	Find		Filter Show all		▼ ቍ Add	FB for IC) Channel 🕇	Go to In
rial Cateway	Variable	Mapping	Channel	Address	Туре	Unit	Description	
lai Galeway	B- *		Holding Registers	%QW13	ARRAY [09] OF WORD			
ibus TCP Slave Device I/O	🕀 🦘 Input	*	Holding Registers[0]	%QW13	WORD			
ping	😐 🐨 🐪		Holding Registers[1]	%QW14	WORD			
ous TCP Slave Device IEC	🕀 🍫		Holding Registers[2]	%QW15	WORD			
	😐 ··· 🦦		Holding Registers[3]	%QW16	WORD			
5	😟 - 🍫		Holding Registers[4]	%QW17	WORD			
	😐 ··· 🦦		Holding Registers[5]	%QW18	WORD			
rmation	😟 - 🍫		Holding Registers[6]	%QW19	WORD		Description	
	😟 ··· 🦦		Holding Registers[7]	%QW20	WORD			
	😟 - 🦦		Holding Registers[8]	%QW21	WORD			
	- · · · · · · · · · · · · · · · · · · ·		Holding Registers[9]	%QW22	WORD			
	🚊 - 🍫		Input Registers	%QW23	ARRAY [09] OF WORD			
	🖻 🐶 Output	*	Input Registers[0]	%QW23	WORD			
	😟 - *		Input Registers[1]	%QW24	WORD			
	📺 🍫		Input Registers[2]	%QW25	WORD			
	🗎 - *		Input Registers[3]	%QW26	WORD			
	📺 🍫		Input Registers[4]	%QW27	WORD			
	🖻 - * ø		Input Registers[5]	%QW28	WORD			
	Image: Second	~∳ = Ma	ap to existing variable	Reset Map	ping Always updatevar	iables (Jse parent device	setting

Figura 3.4: Declaração dos endereços transmitidos pelo dispositivo escravo de rede.

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 indicado pela seta vermelha na Figura 3.5.

File Edit View Project Bu	ild Online Debug To 🖹 🗙 🏘 🎲 🐴 🌿 🎚	ols Window He 비위체체(음	‼p ‱	oplication [[Device: PLC Logic] 👻 🕵	¢ğ ↓	■ 🖋 〔≣ ਓ≣ ⁵≣ *:	■ 왕 ♥ 麗 ☴	₹ ∛/	10
ModbusTCP_Slave_Devic	e X									•
General	Find		Filter Show all		- ⊕	Add FB f	for IO Channel → 🗍 Go t	to Instance		
Serial Gateway Modbus TCP Slave Device I/O	Variable	Mapping	Channel Holding Registers Input Registers	Address %IW17 %QW13	Type ARRAY [09] OF WORD ARRAY [09] OF WORD	Unit	Description			
Modbus TCP Slave Device IEC Objects							-			
Status						4	5			
Status	= Create new variat	ole 🍫 = Ma	p to existing variable	Reset N	apping Always update	evariable	s Enabled 1 (use bus cycl Use parent device setti Enabled 1 (use bus cycl Enabled 1 (use bus cycl Enabled 2 (always in bu	le task if not used in an ng e task if not used in an is cycle task)	y task) - ~ y task)	

Figura 3.5: Opção para mostrar valores atualizados das variáveis a todos os instantes.

4 MONITORAÇÃO

4.1 MONITORAMENTO DE VARIÁVEIS

Após a configuração da rede Modbus/TCP e declarados os endereços de transmissão, podemos monitorar e controlar os dados transmitidos pelos dispositivos. Por fim, devemos declarar as variáveis que desejamos atribuir aos endereços, no modo *online*, na aba **Mapping**, é possível verificar e escrever os valores nas variáveis de rede.

Moducative Find Price Second Edd Edd Cold Web Cold Edd Edd Cold Web Price of the Cold Web Pric of the Cold Web Pric of the Cold Web <th>ices 👻 🖣 🗙</th> <th>Device PLC_PRG</th> <th>Ethernet Mo</th> <th>lbusTCP_Slave</th> <th>Device X</th> <th></th> <th></th> <th></th> <th></th> <th></th>	ices 👻 🖣 🗙	Device PLC_PRG	Ethernet Mo	lbusTCP_Slave	Device X					
Brit Cage Status Image: Status Image: Statu	ModbusSlave	General	Find		Filter Show all		🔹 🖶 Add	FB for IO Channel	+ Go to Instance	
	PLC Logic Application [run]	Serial Gateway	Variable	Mapping	Channel Holding Registers	Address	Type	Current Value	Prepared Value Unit	De
Image: Second	Library Manager	Modbus TCP Slave Device I/O	B- No Innut	***	Holding Registers[0]	%IW17	WORD	40128		
* @ Take Configuration Modes TCP Stars Device IEC * * * Holder Spragetra(1) W010 Not coloned • @ Statut_EDP, Taket - @ Statut_EDP, Taket * * * Holder Spragetra(1) W010 Not coloned • @ Statut_EDP, Taket - @ Statut_EDP, Taket * * * Holder Spragetra(1) W010 Not coloned • @ Statut_EDP, Taket - @ Statut_EDP, Taket * * * Holder Spragetra(1) W010 Not coloned • @ Statut_EDP, Taket - @ Holder Spragetra(1) W010 Not coloned Not coloned • @ Statut_EDP, Taket - @ Holder Spragetra(1) W010 Not coloned Not coloned • @ Downsort EDP, Taket - @ Holder Spragetra(1) W010 Not coloned Not coloned • @ Output Holder Spragetra(1) W010 Not coloned Not coloned • @ Downsort EDP, Storter (MindowsTOP Store Device Modes MICP Store Device Workswarts) W010 Not coloned Not coloned • @ Downsort EDP, Storter (MindowsTOP Store Device Modes MICP Store Device Workswarts) W010 Not coloned Not coloned • @ Downsort EDP, Storter (Workswarts) W010 Not coloned Not coloned Not coloned <t< td=""><td>PLC_PRG (PRG)</td><td>Mapping</td><td>8-10</td><td></td><td>Holding Registers[1]</td><td>%IW18</td><td>WORD</td><td></td><td></td><td></td></t<>	PLC_PRG (PRG)	Mapping	8-10		Holding Registers[1]	%IW18	WORD			
Get Natriak Get Natr	🖹 🎉 Task Configuration	Modbus TCP Slave Device IEC	8-10		Holding Registers[2]	%IW19	WORD			
Image: Statut Etc., rake Image: Statut	🖹 🥝 🍪 MainTask	objects	8- %		Holding Registers[3]	%IW20	WORD			
- O - O	DIC_PRG	Status	· · · ·		Holding Registers[4]	%IW21	WORD			
Image: Seture (Seture) Image: Seture (Seture) NUN22 VORD Nun2 colored Image: Seture (Seture) NUN23 VORD Nun2 colored Image: Seture (Seture) Nun24	😳 🚱 StatusLEDs_Task		B- 🍫		Holding Registers[5]	%IW22	WORD			
• ③ Expension (Spatients) • ▲ ● Hodde Registers[] • ₩ ● Hodde Registers[] • ♥ ● Hodde Registers[] • ♥ ● Hodde Registers[] • ♥ ♥ Hodde Registers[- 😏 🔟 Setup (Setup)	Information	8-10		Holding Registers[6]	%IW23	WORD			
 Generation() Generation() Generation() Generation() Generation() Modual(OP_Stave_Device (Modual(OP_Stave_Device (Mod	🔂 🔟 I_Os (I/Os)		8-19		Holding Registers[7]	%IW24	WORD			
	Cypansions (Expansions)		⊕-¥p		Holding Registers[8]	%IW25	WORD			
Image: Space Device (Notburl/Or Save Device) Imax Registers NQVII AREA (0.5) (0 m/ODD) Mole control Imax Registers NQVII AREA (0.5) (0 m/ODD) NOTE Not control Imax Registers NQVII WORD NOTE Not control Imax Registers NQVII WORD Not control	= 😏 🔟 Ethernet (Ethernet)		· · · · · · · · · · · · · · · · · · ·		Holding Registers[9]	%IW26	WORD			
* Output "ipput Register[1] %QVII WORD A1239 * * ipput Register[2] %QVII WORD Not schedel * * > ipput Register[2] %QVII WORD Not schedel * * > ipput Register[2] %QVII WORD Not schedel * * > ipput Register[4] %QVII WORD Not schedel * * > ipput Register[4] %QVIII WORD Not schedel * * > ipput Register[4] %QVIII WORD Not schedel * * > ipput Register[4] %QVIII WORD Not schedel * * > ipput Register[4] %QVIII WORD Not schedel	ModbusTCP_Slave_Device (ModbusTCP Slave Device)		B- 🍬		Input Registers	%QW1	ARRAY [09] OF WORD			
Provide Register [1] 40/012 WORD Introduced Provide Register [2] 40/012 WORD Introduced Provide Register [2] 40/014 WORD Introduced			8- V Output	×.	Input Registers[0]	%QW1	WORD	40129		
Provid Register (2) %QV13 WORD Not codeted Provid Register (2) %QV14 WORD Not codeted Provid Register (2) %QV14 WORD Not codeted Provid Register (2) %QV15 WORD Not codeted Provid Register (3) %QV17 WORD Not codeted			8 · 🌘		Input Registers[1]	%QW2	WORD			
Implementation Implementation Implementation Implementation Implementation Implementation Implementation Implementation </td <td></td> <td></td> <td>B- 🔅</td> <td></td> <td>Input Registers[2]</td> <td>%QW3</td> <td>WORD</td> <td></td> <td></td> <td></td>			B- 🔅		Input Registers[2]	%QW3	WORD			
* Input Register[4] %QVIS WORD Intruction * * Input Register[3] %QVIS WORD Not updated * * Input Register[3] %QVIS WORD Not updated * * Input Register[3] %QVIS WORD Not updated * * Input Register[4] %QVIS WORD Not updated * * Input Register[4] %QVIS WORD Not updated			÷-*•		Input Registers[3]	%QW4	WORD			
* *			8-10		Input Registers[4]	%QW5	WORD			
* Input Register(6) %QV/V WORD Net updated * * Input Register(7) %QV8 WORD Net updated * * Input Register(6) %QV9 WORD Net updated			- *		Input Registers[5]	%QW6	WORD			
# Input Register(T) %(VM) WORD Not sequence # # Input Register(B) %(VM) WORD Not sequence			8-10		Input Registers[6]	%QW7	WORD			
Input Registers[8] %QW9 WORD Not updated			8		Input Registers[7]	%QW8	WORD			
			B- 0		Input Registers[8]	%QW9	WORD			
B→T∳ Input Registers[9] %QW10 WORD Not updated					Input Registers[9]	%QW10	WORD			
Rest: Mapping: Always update variables Use parent device setting			<			Reset Mapp	ing Always update varia	ables Use parent de	vice setting	

Figura 4.1: Declaração das variáveis transmitidas pelo dispositivo escravo de rede.

4.2 ERROS DE COMUNICAÇÃO

Monitorar o estado das redes no *software Codesys* PLC500 pode ser feito em **Devices** também, indicando o estado de cada uma das etapas de comunicação e reportando o estado (**Status**). Caso encontre problemas de conexão, conecte-se com o PLC500 e acesse a aba de **Status** e **Log** dentro dos itens criados, o Codesys informará qual problema está encontrando para fazer a comunicação.



Figura 4.2: Indicação de erro na comunicação.



NOTA!

Caso encontrar problemas, verifique se os cabos estão devidamente conectados e se o LED respectivo da rede se encontra acesso.

Quando as configurações estiverem corretas e os dispositivos identificarem um ao outro, a aplicação deve se apresentar como mostrado na Figura 4.3.







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