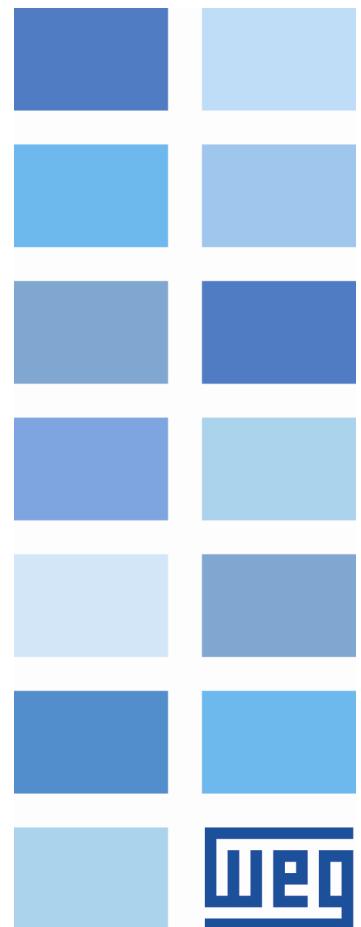


Software

WPS V1.30

Manual do Usuário





Manual do Usuário

Série: WPS

Idioma: Português

Nº do Documento: 10001027753 / R01

Data de Publicação: 05/2011

Sumário

1	Antes de Iniciar	9
1	Bem Vindo	9
2	Identificação	9
3	Aviso	9
4	Informações Legais	9
5	Visão Geral	9
6	Novidades e Correções do WPS	10
7	Contato	11
8	Suporte Técnico	11
9	Download	11
2	Instalação	12
1	Antes de Instalar	12
2	Requisitos Mínimos	12
3	Equipamentos Suportados	13
4	Java Runtime Environment	13
5	Instalando	14
6	Desinstalando	15
7	Gerenciador de Comunicação	15
3	Iniciando	16
1	Ambiente	16
2	Guia de Inicialização Rápida	17
4	Gerenciador de Comunicação	25
1	Executando	25
2	Configuração	25
3	Adicionar/Remover Conexões	25
4	Hardware	26
	Cabos	26
	Conversor USB/Serial	28
5	Driver USB	28
	Instalando Driver USB	28
	Verificando a Instalação do Driver USB	29
	Desinstalando Driver USB	29
	Atualizando Driver USB	30
5	Ladder	31

1	Conceitos	31
	Introdução	31
	Lógica com Contatos	33
	Terminologia	34
	Tipo de Dados	36
	Representação Direta	36
2	Editor	37
	Área de Trabalho	37
	Menu Ladder	38
	Rungs	39
	Rungs - Visão Geral	39
	Editando	40
	Título e Comentário	40
	Inserindo Elementos	41
	Inserindo Elementos - Visão Geral	41
	Em Série	42
	Em Paralelo	42
	Navegando	43
	Com o Teclado	43
	Com o Mouse	45
	Copiar / Colar	46
	Variáveis	49
	Variáveis - Visão Geral	49
	Campos	51
	Editando na Rung	53
	Literais na Rung	55
	Arrays na Rung	56
	Instâncias e Estruturas na Rung	57
	Compilar	58
	Transferir	60
	Monitoração Online	62
6	PLC300	67
	1 Descrição	67
	2 Novidades e Correções do PLC300	67
	3 I/O's	70
	4 Marcadores de Sistema	73
	5 Editor de Telas	87
	Configuração das Telas	87
	Teclas de Função	90
	Gerenciamento de Telas	90
	Manipulação de Bits	91
	Senhas	93
	Programação	95
	Visão Geral	95
	Componente Numeric Input	99
	Componente Numeric Output	100
	Componente Bargraph	101
	Componente Message	102
	Componente Text	103
6	Alarmes	103

	Visão Geral	103
	Configuração dos Alarmes	104
7	Tarefas	109
	Configuração de Tarefas	109
8	Log de Eventos	116
	Configuração de Log de Eventos	116
9	Estruturas	126
	Configuração de Estruturas	126
10	Receitas	130
	Configuração de Receitas	130
11	Setup	133
	Configuração de Setup	133
	Janelas Configuração de Setup	135
	Display	135
	Analógicas	136
	Encoder	137
	RS232	137
	RS485	138
	CAN	138
	LAN	139
	Modbus TCP	140
	Programa do Usuário	141
	Comandos Online	141
	Ajustes do Relógio	143
12	SD Card	144
	Gerenciador de arquivos modbus	144
13	Componentes e Blocos	147
	Contact	147
	NOCONTACT	147
	NCCONTACT	147
	PTSCONTACT	148
	NTSCONTACT	149
	Coil	150
	DIRECTCOIL	150
	INVERTEDCOIL	150
	SETCOIL	151
	RESETCOIL	152
	TOGGLECOIL	153
	IMMEDIATECOIL	153
	Logic Bit	154
	RESETBIT	154
	SETBIT	155
	TESTBIT	156
	Logic Boolean	157
	NOT	157
	AND	158
	OR	159
	XOR	160
	XNOR	161
	Logic Shift	162
	ASHL	162
	ASHR	163

SHL	164
SHR	165
Logic Rotate	166
ROL	166
ROR	167
Math Basic	168
ABS	168
NEG	168
ADD	169
SUB	170
MUL	172
DIV	173
MOD	174
Math Extended	175
LN	175
EXP	176
LOG10	176
ALOG10	177
POW	178
SQRT	179
Math Trigonometry	180
SIN	180
COS	181
TAN	181
ASIN	182
ACOS	183
ATAN	184
ATAN2	184
Math Util	185
MAX	185
MIN	186
SAT	187
Comparação	189
COMPEQ	189
COMPGE	190
COMPGT	191
COMPLE	192
COMPLT	193
COMPNE	194
Conversão	195
BCD	195
BCD_TO_WORD	195
WORD_TO_BCD	195
BOOL	196
BYTE_TO_BOOL	196
REAL_TO_BOOL	197
DWORD_TO_BOOL	198
WORD_TO_BOOL	198
BYTE	199
BOOL_TO_BYTE	199
REAL_TO_BYTE	200
DWORD_TO_BYTE	201
WORD_TO_BYTE	202
WORD	202

BOOL_TO_WORD	202
BYTE_TO_WORD	203
DWORD_TO_WORD	204
REAL_TO_WORD	205
REAL	205
BOOL_TO_REAL	205
DWORD_TO_REAL	206
WORD_TO_REAL	207
BYTE_TO_REAL	208
DWORD	208
BOOL_TO_DWORD	208
REAL_TO_DWORD	209
WORD_TO_DWORD	210
BYTE_TO_DWORD	211
Rad-Deg	211
RAD_TO_DEG	211
DEG_TO_RAD	212
Data Transfer	213
STORE	213
ILOADBOOL	214
ILOAD	215
ISTOREBOOL	216
ISTORE	217
MUX	218
DEMUX	220
SEL	222
SWAP	223
ReadRecipe	224
WriteRecipe	225
Counter	227
CTD	227
CTU	228
CTUD	229
Timer	231
TON	231
TOF	233
TP	234
Control	235
PID	235
Filter	239
LOWPASS	239
Hardware	240
READENC	240
READENC2	241
READENC3	242
PWM	243
IMMEDIATE_INPUT	243
IMMEDIATE_OUTPUT	245
RTC	245
INTIME	245
INWEEKDAY	247
Screen	249
SETSCREEN	249
Rede de Comunicação	250

CANopen	250
Visão Geral do CANopen	250
CO_SDORead	252
CO_SDOWrite	253
CO_MasterControlStatus	255
CO_SlaveStatus	257
Modbus RTU	259
Visão Geral do Modbus RTU	259
MB_ReadBinary	260
MB_ReadRegister	262
MB_WriteBinary	264
MB_WriteRegister	266
MB_MasterControlStatus	268
MB_SlaveStatus	270
Modbus TCP	271
Visão Geral do Modbus TCP	271
MBTCP_ReadBinary	272
MBTCP_ReadRegister	274
MBTCP_WriteBinary	276
MBTCP_WriteRegister	278
MBTCP_ServerStatus	281
MBTCP_ClientControlStatus	283

1 Antes de Iniciar

1.1 Bem Vindo

Bem vindo ao Software!

Esta documentação fornece uma visão geral do WPS V1.30 e contém toda a documentação disponível para o WPS V1.30.

Use o índice à esquerda para navegar pelos tópicos.

1.2 Identificação

WPS V1.30 - Manual do Usuário
Data de Publicação: 05/2011

1.3 Aviso



Perigo! Uso deste software pode alterar a operação ou desempenho do drive. O usuário é responsável pela adoção de todas as precauções necessárias para garantir a segurança do equipamento e do pessoal envolvido. Antes de aplicar este Software, leia atentamente as Instruções da Ajuda Online. A não observação destas instruções pode causar sérios danos no equipamento e resultar em graves lesões corporais.

1.4 Informações Legais



Nota! Este programa de computador está protegido pelas leis dos direitos autorais e tratados internacionais. Sua reprodução ou distribuição, parcial ou total, sem autorização prévia pode resultar em severas penalidades civis e criminais, sujeito às sanções previstas em lei.

1.5 Visão Geral

O Software ou WPS V1.30 é a mais nova geração de software da WEG, concebida para ser uma ferramenta integrada, agregando os conceitos de:

- Multi-Produtos, visando atender ampla gama de produtos da WEG; inicialmente estará atendendo o equipamento PLC300,
- Multi-Utilização visando permitir:

- Configuração dos equipamentos,
- Programação dos equipamentos em Linguagem Ladder,
- Monitoração dos equipamentos, e
- Assistente de criação e configuração de aplicações (futuro).

O ambiente de programação em Linguagem Ladder atende aos requisitos da NORMA IEC61131-3.

É uma ferramenta que proporciona flexibilidade ao usuário, pois lhe permite a programação de um ou mais recursos, dentro da mesma configuração.

1.6 Novidades e Correções do WPS

WPS V1.30

Funcionalidades:

- [Atualização Editor de Variáveis](#)
- Configuração da Ethernet no Setup
- Configuração de Estruturas
- Configuração de Receitas
- Configuração de Log de Eventos
- Gerenciador de Arquivos Modbus

Editor ladder:

- [Mudanças na Área de Trabalho](#)
- [Utilização de Literais](#)
- [Utilização de Arrays](#)
- [Utilização de Instâncias e Estruturas](#)
- [Argumentos Opcionais para Blocos Funcionais](#)
- [Novas Funcionalidades Copiar/Colar](#)
- [Novas Funcionalidades Monitoração Online](#)

Correções:

- Otimização do editor ladder de modo a garantir arquivos ladder com até 200 rungs e consumo de memória e cpu moderados, na versão 1.20 do WPS e nessa situação ocorriam lentidão e travamentos

PLC300:

- Novidades e Correções do PLC300.

WPS V1.20

- Comunicação com equipamento PLC300 V1.2X,
- Programação de tarefas,
- Alteração no endereços de marcadores para compatibilidade com o Modbus,
- Ferramenta para importar/exportar projetos,
- Verificação da versão de firmware do PLC300 no download,
- Parar/executar programa como uma ação global,
- Persistência do ambiente ao sair,

- Melhorias na comunicação,
- Opções de download:
 - Inicializar variáveis retentivas,
 - Limpar histórico de alarmes,
 - Download do código fonte,
- Compatível com Windows 7 x64, Windows 7 x86, e
- Compatível com Windows Vista x64, Windows Vista x86.

1.7 Contato

Website:  <http://www.weg.net/>

1.8 Suporte Técnico

Contate uma filial ou representante.

1.9 Download

Faça o download do WPS V1.30 da página de downloads em <http://www.weg.net/>.

Selecione o(s) arquivo(s) para fazer o download e os armazene em uma pasta no seu disco rígido.

Descompacte os arquivos.

Na pasta onde você descompactou os arquivos você encontrará o instalador. Dê um duplo clique neste ícone para iniciar a instalação.

Para mais detalhes veja item Instalação > [Instalando](#).

2 Instalação

2.1 Antes de Instalar

Verifique os seguintes itens antes de instalar o WPS V1.30:

- Se o microcomputador possui os [Requisitos Mínimos](#),
- Se a versão do WPS V1.30 é compatível com seu equipamento, veja [Equipamentos Suportados](#), e
- O [Java Runtime Environment](#) V1.6.0 ou superior precisa estar instalado no seu computador. Instale o JRE antes de instalar o WPS V1.30.

2.2 Requisitos Mínimos

Item	Descrição
Processador	PC com processador mínimo Pentium IV Intel(R) Core(TM)2 DUO ou superior é recomendado
Plataforma	Windows XP x86 SP3 Windows XP x64 SP1 Windows Vista x86 SP1 Windows Vista x64 SP1 Windows 7 x86 Windows 7 x64
	Você deve ter privilégios de administrador para instalar o software
Memória	512 MB para Windows XP, 1 GB ou mais é recomendado 1 GB para Windows Vista, 2 GB ou mais é recomendado 1 GB para Windows 7, 2 GB ou mais é recomendado
Cor do Monitor	High color (16 bits) ou melhor
Resolução do Monitor	1024x768 pixels ou resolução superior é recomendado
Espaço em Disco	100 MB de espaço disponível no disco para WPS 150 MB de espaço disponível no disco para Java Runtime Environment (depende da versão)
Unidade de Disco	Unidade CD-ROM ou DVD-ROM
Comunicação	Interface serial RS232 (ponto-a-ponto) Comunicação USB (ponto-a-ponto)
Dispositivo Apontador	Mouse ou dispositivo apontador compatível
Java Runtime Environment	Java Runtime Environment (JRE) 6.0 ou superior
Navegador	Mozilla Firefox 3.6 ou superior Opera 10 ou superior Internet Explorer 7 ou superior JavaScript habilitado Navegador deve suportar Cascading Style Sheets (CSS1) e JavaScript

2.3 Equipamentos Suportados

A seguinte tabela apresenta os equipamentos e versões de firmware que cada base de dados fornecida com o WPS suporta.

Versão do WPS	PLC300
WPS V1.30	V1.30 ou inferior
WPS V1.20	V1.20
WPS V1.01	V1.00 * (Versão obsoleta)
WPS V1.00	V1.00 * (Versão obsoleta)

* Recomendamos fazer upgrade do firmware do PLC300 para V1.20 ou superior.

2.4 Java Runtime Environment

O Java Runtime Environment (JRE) V1.6.0 ou superior precisa ser instalado no seu computador.

Para obter a última versão do JRE gratuitamente, vá para <http://www.java.com>.

Para verificar se você tem o JRE instalado ou qual versão que você possa já ter, veja abaixo.

Verifique a versão do JRE no seu Computador Usando o Prompt de Comando

Para verificar se você tem o JRE correto digite o seguinte comando no prompt de comando ou num terminal.

```
java -version
```

Se você receber uma mensagem de erro indicando 'java' como um comando desconhecido então você precisa fazer o download do JRE.

Se você receber uma resposta ao comando que indica que o número da versão do Java é menor que 1.6.0, tal como 1.2, 1.3, 1.4.1, 1.4.2, 1.5.0, etc, então você precisa fazer o download de uma versão mais nova do JRE.

Verifique a versão do JRE no seu Computador Usando o Sítio Web Java.com

Opção 1:

Existe uma outra opção para verificar a instalação do software Java.

No endereço <http://www.java.com/> há um link chamado [Tenho o Java?](#) onde você pode encontrar instruções para verificar o JRE instalado no seu computador.

Quando este manual foi escrito o link era http://www.java.com/pt_BR/download/installer.jsp.

Siga as instruções nesta página.

Opção 2:

Você também pode utilizar o seguinte link (página em inglês) para verificar a versão do java: <http://www.java.com/en/download/help/testvm.xml> .

Instruções de Instalação

Siga as intruções em <http://www.java.com/>.

Mais Informações

Mais detalhes sobre Java Runtime Environment podem ser encontrados em:

- <http://www.java.com/>

2.5 Instalando

Ao instalar uma versão mais nova do WPS é recomendado primeiro [remover](#) a versão anterior e reiniciar o sistema para ter certeza que a nova instalação será executada corretamente.

Fechar todos os programas abertos atualmente em execução para evitar interferência com o processo de instalação.

Para instalar o WPS V1.30, por favor seguir os passos abaixo.

Instalação

- Dar um duplo clique no ícone do instalador para iniciar o instalador. O programa de instalação iniciará e uma janela de boas vindas aparecerá.
- No assistente de instalação, responder à Licença de Uso, então pressionar botão Avançar.
- Especificar uma pasta vazia dentro da qual será instalada o WPS V1.30. Certificar-se que a localização da instalação está correta e que existe espaço suficiente em disco para a instalação.
- Confirmar para criar uma pasta vazia.
- Selecionar o tipo de instalação que desejar.
- Selecionar o grupo de programa para criar os atalhos do WPS V1.30.
- Selecionar os atalhos de programas adicionais: criar um atalho no desktop e criar um atalho para inicialização rápida.

- Por favor revisar ou alterar qualquer uma das configurações. Clicar no botão Instalar para iniciar a instalação.
- Clicar no botão Concluir para concluir o setup.

2.6 Desinstalando

Se necessário, remover o WPS V1.30 usando os procedimentos abaixo.

Desinstalação



Nota! Conforme indicado abaixo, usar o utilitário Adicionar ou Remover Programas para remover o WPS V1.30. Não apagar arquivos e pastas manualmente.

- Do menu Iniciar, selecionar Painel de Controle.
- Dar um duplo clique no ícone Adicionar ou Remover Programas.
- Selecionar WPS V1.30 da lista e clicar no botão Remover.
- Seguir as instruções para remover o software.

2.7 Gerenciador de Comunicação

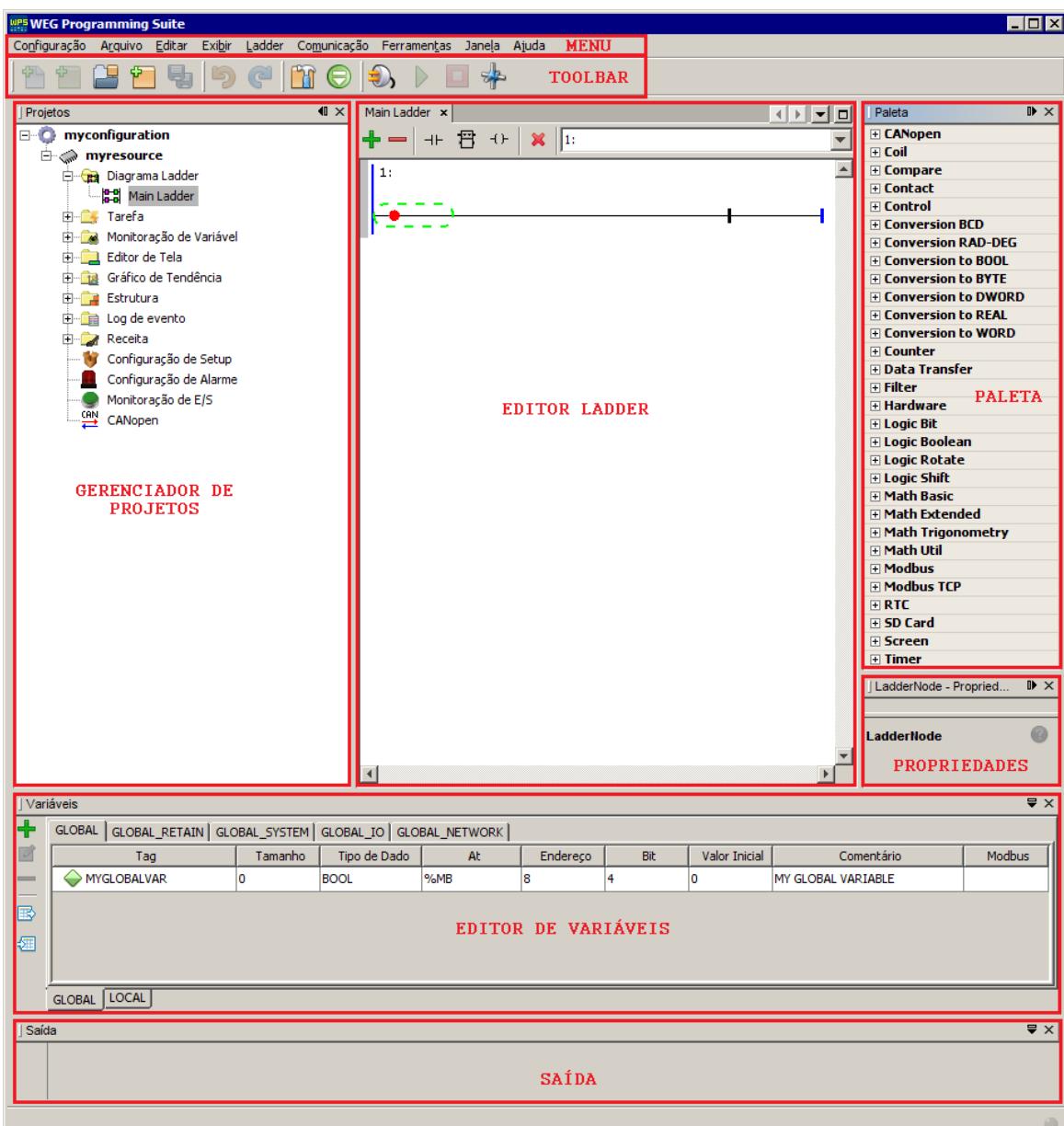
O Gerenciador de Comunicação é instalado automaticamente quando o WPS V1.30 é instalado.

3 Iniciando

3.1 Ambiente

O Software é dividido em 6 campos principais:

- Menus - Exibe as diversas opções de edição, visualização, comunicação e ajuda do ambiente de desenvolvimento;
- Toolbar - Exibe os principais comandos do software;
- Janela de Projetos - Exibe a árvore contendo a estrutura da Configuração aberta;
- Janela do Editor - Parte principal do ambiente de desenvolvimento, onde ocorre a edição dos componentes que irão formar o arquivo fonte;
- Janela da Paleta - Contem os componentes a serem inseridos no editor através do drag'n drop;
- Janela de Propriedades - Exibe as propriedades do componente selecionado;
- Janela de Saída - Exibe mensagens de compilação e download.

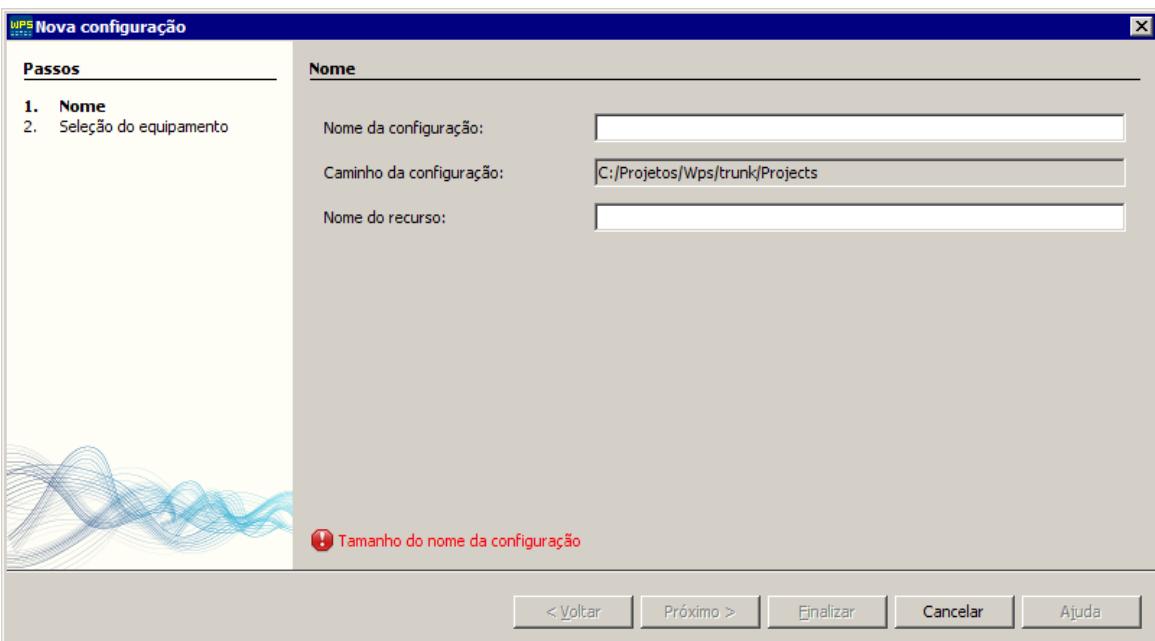


(Figura 1: Ambiente WPS)

3.2 Guia de Inicialização Rápida

Criação da Configuração

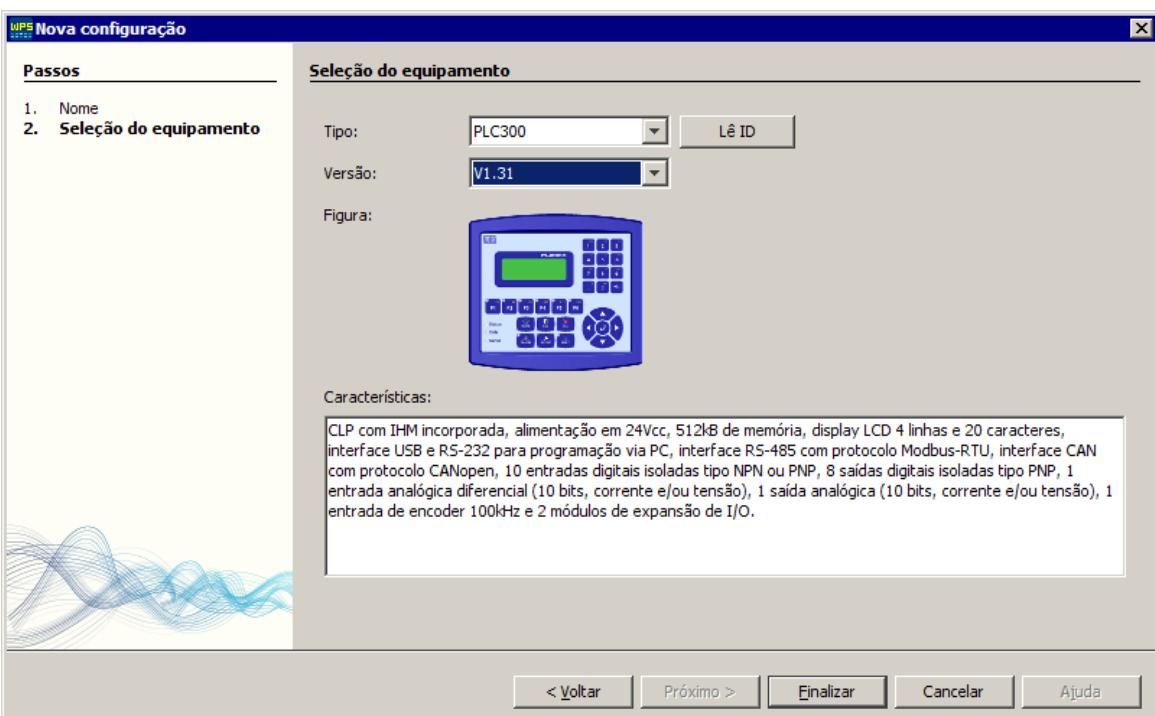
Para criar uma configuração é preciso selecionar a opção "Projeto > Nova Configuração" no menu ou pressionar as teclas de atalho "Ctrl+Shift+C". Uma janela para configurar o nome da Configuração (Projeto ou Planta) e do Recurso (Equipamento) irá abrir.



(Figura 1: Nome da Configuração e do Recurso)

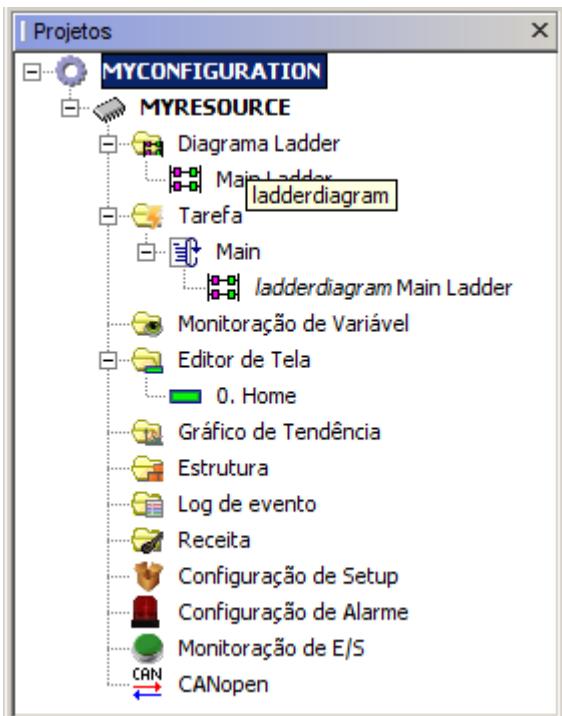
Insira nomes válidos e selecione a botão próximo.

Uma janela contendo os equipamentos suportados pelo ambiente de programação irá aparecer. Selecione o equipamento e a versão desejados e selecione o botão finalizar.



(Figura 2: Seleção do Equipamento e sua Versão)

O projeto criado deverá conter a seguinte estrutura:

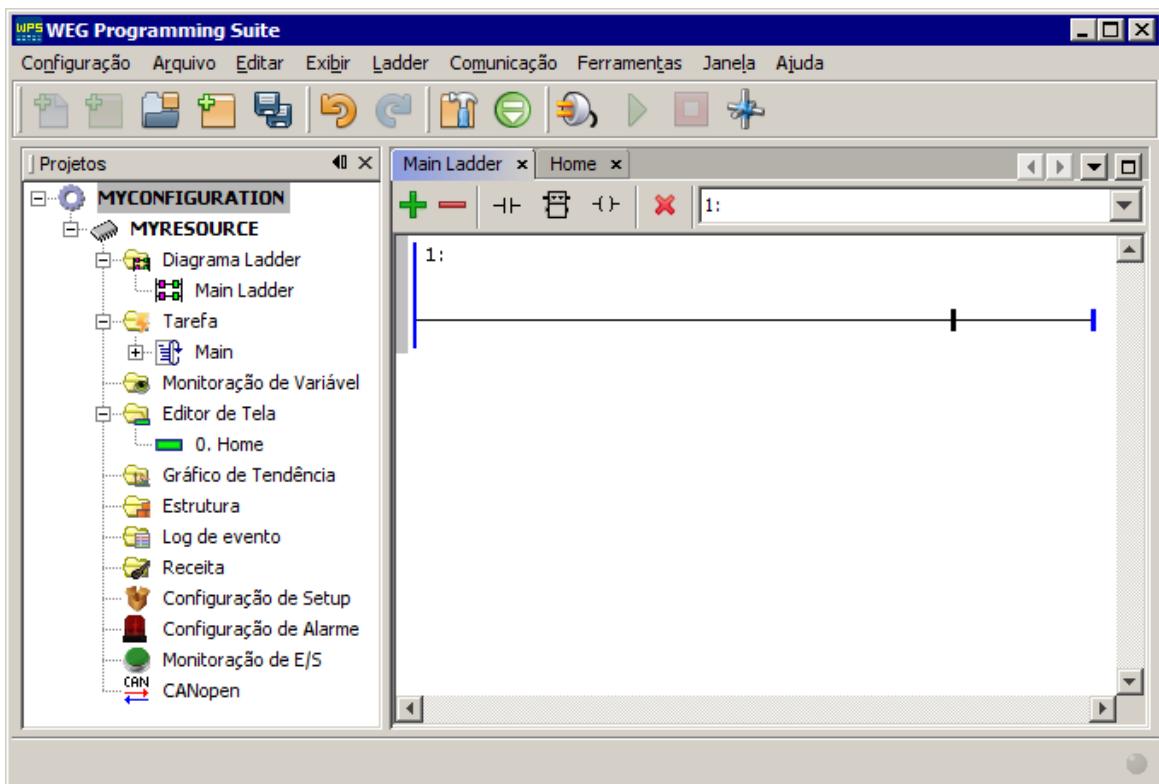


(Figura 3: Gerenciamento de Projetos)

Edição do Programa

Após a criação da configuração e do recurso o usuário deve editar programa conforme a necessidade da aplicação.

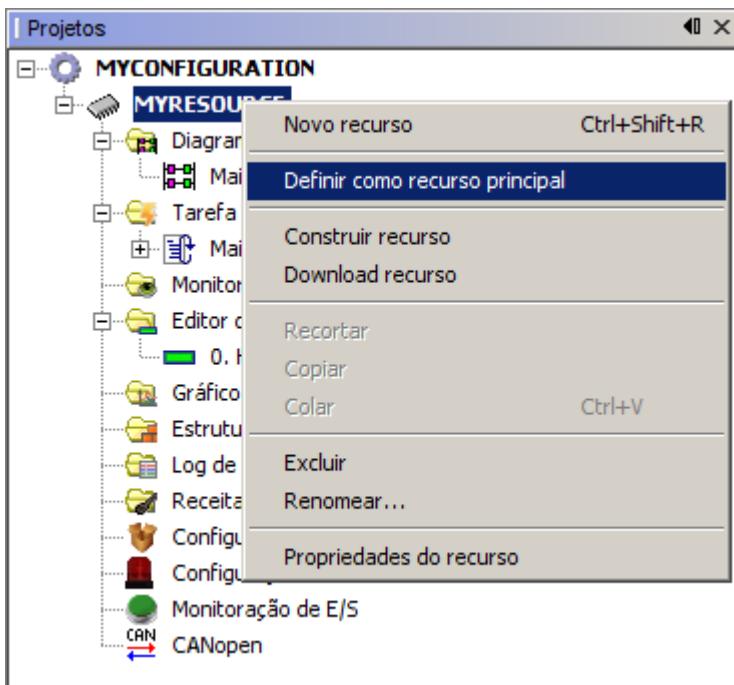
Para maiores detalhes consulte [Editor Ladder](#) e [Editor de Telas](#).



(Figura 4: WPS com Programa Ladder aberto)

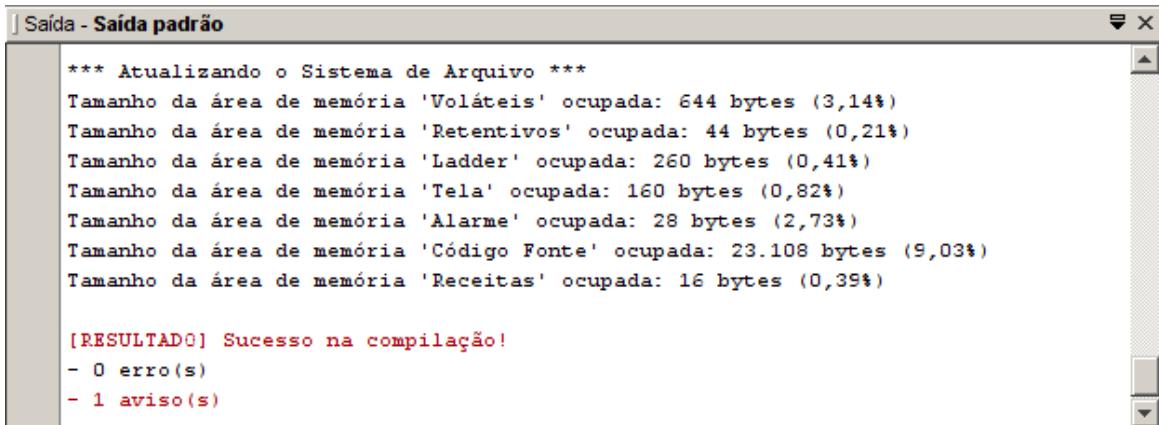
Compilar Recurso

Para compilar o recurso, o usuário deverá selecionar a opção "Construir>Construir recurso principal" no menu ou pressionar a tecla de atalho "F4". Para isso, certifique-se de que o recurso a ser compilado é o recurso selecionado na janela de projetos. O recurso selecionado tem sua formatação em negrito. Caso o recurso principal não seja o recurso desejado deve-se clicar com o botão direito no recurso e selecionar a opção "Selecionar recurso principal".



(Figura 5: Seleção do recurso principal)

O recurso será compilado e no final deverá aparecer a mensagem: "[RESULTADO] Sucesso na compilação!" na janela de Saída, caso contrário verifique as mensagens de erro geradas e faça as correções necessárias no editor.



(Figura 6: Resultado da Compilação na Saída Padrão do WPS)

Execução do Gerenciador de Comunicação (Local)

Antes de iniciar o download é necessário que o gerenciador de comunicação seja ativado.

Para ativar o gerenciador de comunicação o usuário deve selecionar a opção "Online>Gerenciador de comunicação local".

Uma tela com as configurações do gerenciador irá abrir. Configure a conexão com o

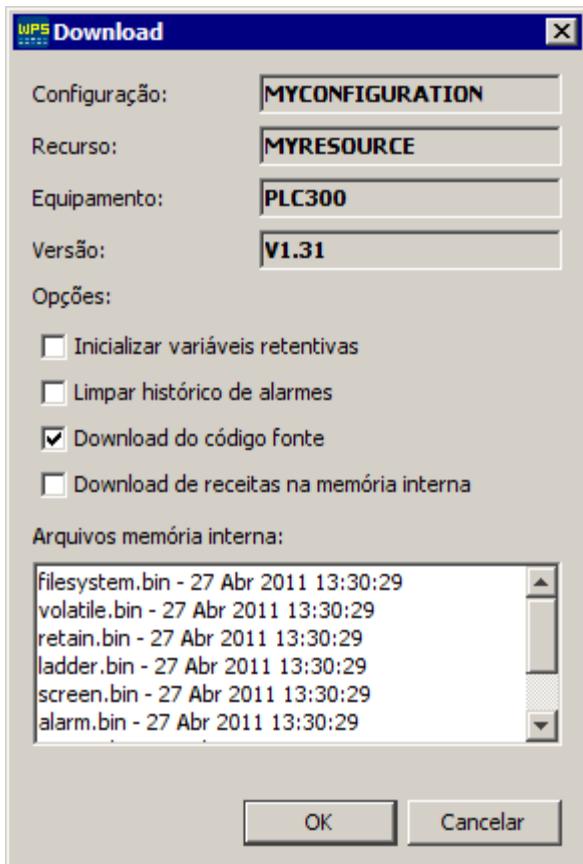
equipamento, maiores detalhes ver [Gerenciador de Comunicação](#).



(Figura 7: Gerenciador de Comunicação do WPS)

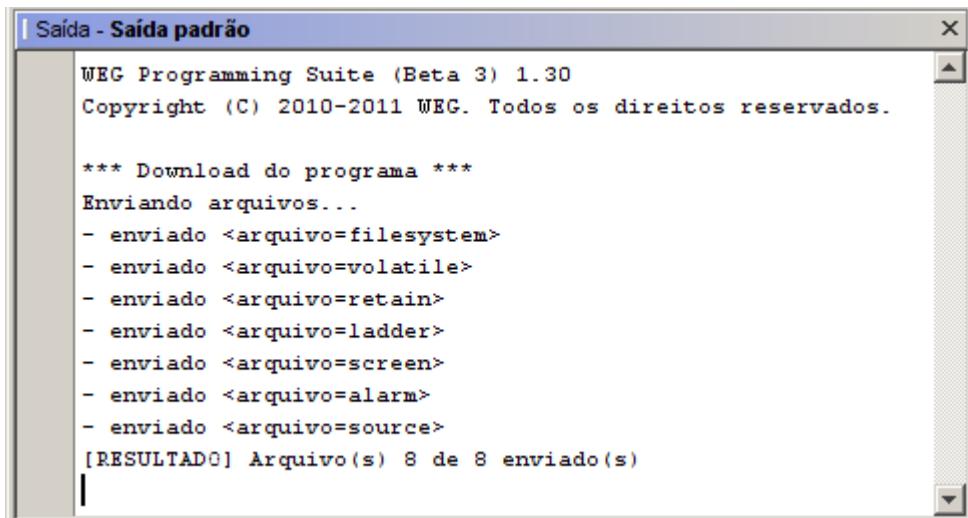
Download do Recurso

Para realizar o download do recurso o usuário deve selecionar a opção "Online>Download do recurso principal". Uma janela contendo as informações do download deverá aparecer.



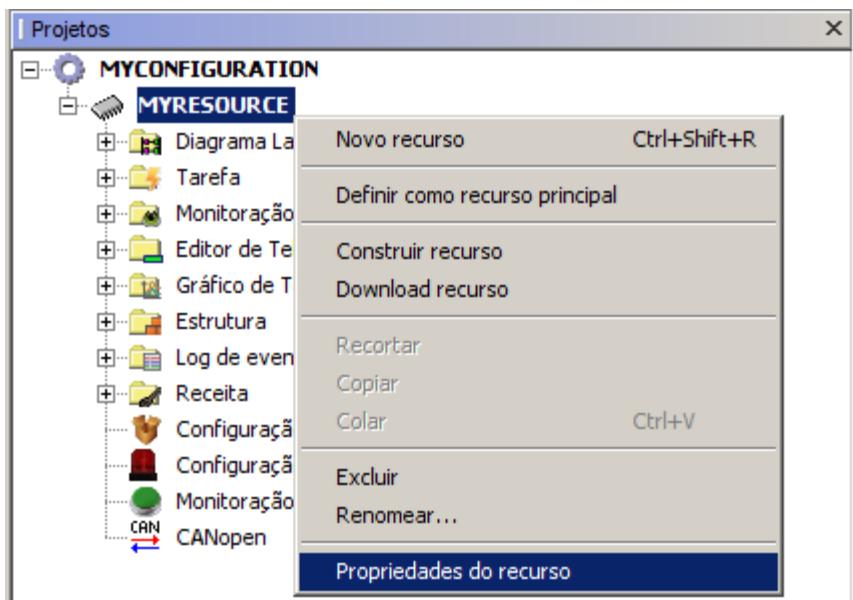
(Figura 8: Seleção das Opções de Download)

Verifique se as informações correspondem ao equipamento conectado. A ação de download irá parar a execução do programa do usuário e reiniciá-lo logo após a conclusão do download. No final do download deverá aparecer uma mensagem "[RESULTADO] Arquivo(s) 7 de 7 enviado(s)". Caso tenha acontecido alguma falha durante o envio, verifique as propriedades de comunicação.



(Figura 9: Resultado após o Download)

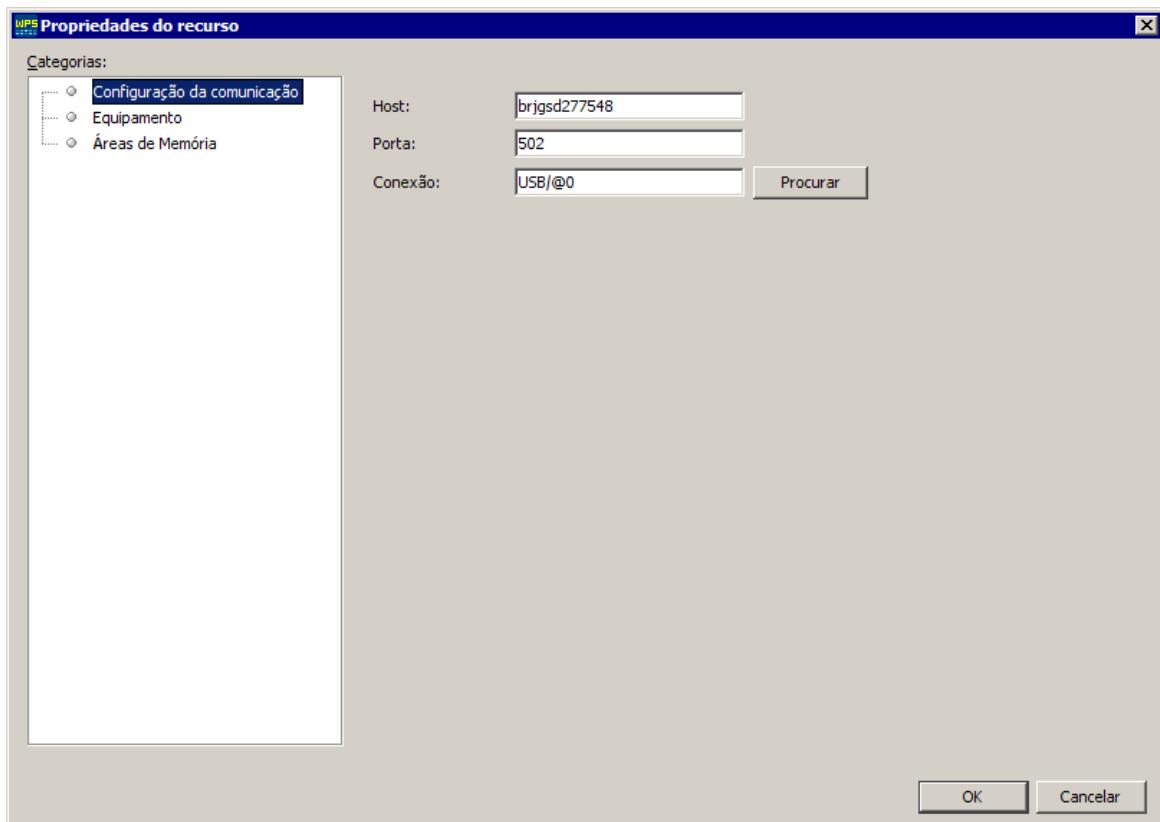
No Software estas propriedades podem ser visualizadas clicando com o botão direito sobre o recurso e selecionando a opção "Propriedades do recurso".



(Figura 10: Acesso as Propriedades do Recurso)

Uma janela contendo as propriedades do recurso irá aparecer. Selecione a opção

"Configuração da comunicação". Verifique se as opções estão configuradas corretamente.



(Figura 11: Configuração da Comunicação)

4 Gerenciador de Comunicação

4.1 Executando

Para executar o gerenciador de comunicação na máquina local basta clicar no ícone  disponível no Toolbar.

O comando para executar o gerenciador de comunicação também está disponível no menu "Online" opção "Gerenciador de comunicação local".

4.2 Configuração

Configurações para o Gerenciador de Comunicação

- Ativo: ativa/desativa a comunicação do Gerenciador de Comunicação com as aplicações que se comunicam com os equipamentos através dele,
- Número da Porta: número da porta de comunicação TCP/IP que o Gerenciador de Comunicação utiliza para comunicação com as aplicações que se comunicam com os equipamentos através dele,
- Autenticação de IP: lista com os endereços IPs que podem acessar o Gerenciador de Comunicação. Uma lista vazia indica que qualquer endereço IP pode acessar o Gerenciador de Comunicação.

4.3 Adicionar/Remover Conexões

Para adicionar uma nova conexão, selecione na árvore a interface a qual o equipamento está conectado.

Interface Portas de Comunicação

- Expanda o item da porta COM desejada.
- Altere as configurações da porta COM conforme as utilizadas pelo equipamento.
- Selecione o protocolo de comunicação (Modbus-RTU).
- Altere as configurações do protocolo conforme o desejado.
 - Atraso de transmissão (ms): aguarda este tempo antes de enviar o próximo telegrama.
Faixa de valores: 0 ... 20000
Padrão: 0
 - Atraso na resposta (ms): aguarda este tempo antes de tentar receber a resposta ao telegrama enviado.
Faixa de valores: 0 ... 20000
Padrão: 0
 - Timeout (ms): aguarda este tempo antes de indicar que a resposta não foi recebida.

Faixa de valores: 100 ... 20000

Padrão: 5000

- Clique em nova conexão...
 - Selecione o endereço serial para a nova conexão conforme o configurado no equipamento.

Interface USB

- Selecione o item USB.
 - Altere as configurações conforme o desejado.
 - Atraso de transmissão (ms): aguarda este tempo antes de enviar o próximo telegrama.
Faixa de valores: 0 ... 20000
Padrão: 0
 - Atraso na resposta (ms): aguarda este tempo antes de tentar receber a resposta ao telegrama enviado.
Faixa de valores: 0 ... 20000
Padrão: 0
 - Timeout (ms): aguarda este tempo antes de indicar que a resposta não foi recebida.
Faixa de valores: 100 ... 20000
Padrão: 5000
- Clique em nova conexão...

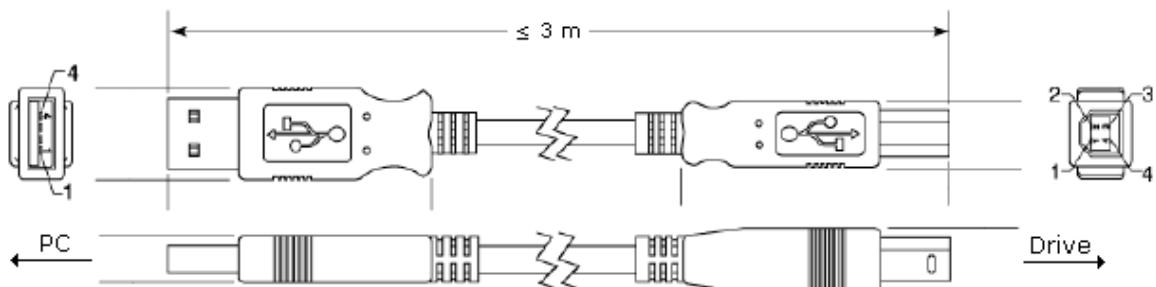
4.4 Hardware

4.4.1 Cabos

Consulte o manual do usuário do PLC300.

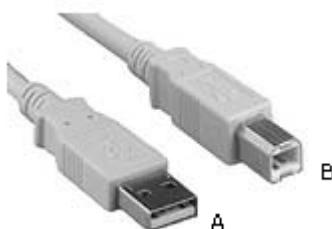
Cabo USB

A figura abaixo apresenta o detalhamento do cabo para a conexão USB ponto-a-ponto.



(Figura 1: Cabos USB)

A figura abaixo apresenta os conectores.



(Figura 2: Conectores USB)

A = Conector Tipo A

B = Conector Tipo B

Tamanho máximo do cabo: 3 metros.



Nota! Utilize sempre cabo de interconexão USB blindado, "standard host/device shielded USB cable". Cabos sem blindagem podem provocar erros de comunicação.



Nota! A conexão USB é isolada galvânicamente da rede elétrica de alimentação e de outras tensões elevadas internas ao drive. A conexão USB, porém, não é isolada do terra de proteção (PE). Usar laptop isolado para ligação ao conector USB ou desktop com conexão ao mesmo terra de proteção (PE) do drive.

Sugestões de Compra

Fabricante:

- Samtec, Inc: <http://www.samtec.com/>

Se você desejar comprar um cabo USB diretamente da Samtec, por favor veja abaixo.

Descrição	Item
Cabo USB blindado de alta velocidade Revisão 2.0, 1 m, Samtec	USBC-AM-MB-B-B-S-1
Cabo USB blindado de alta velocidade Revisão 2.0, 2 m, Samtec	USBC-AM-MB-B-B-S-2
Cabo USB blindado de alta velocidade Revisão 2.0, 3 m, Samtec	USBC-AM-MB-B-B-S-3

No momento em que este manual foi escrito, a especificação era encontrada em <http://www.samtec.com/ftppub/cpdf/USBC-AM-BM-B-B-S-X-MKT.pdf>

4.4.2 Conversor USB/Serial

O conversor USB/SERIAL é a melhor solução para quem deseja conectar equipamentos seriais (RS232) em portas USB, isto é, para conectar a saída USB do microcomputador a um equipamento serial RS-232.

É uma solução de baixo custo que resolve a necessidade de instalar novas portas seriais em microcomputadores que possuam todos os barramentos ocupados ou em equipamentos (notebooks) que não possuem portas RS232.

O conversor USB/SERIAL permite a conexão plug & play com seu microcomputador, deixando livre a porta serial existente.

A figura abaixo apresenta exemplos deste conversor.



(Figura 1: Conversor USB/Serial)

Sugestões de Compra no Brasil

Fabricante: Leadership Comércio e Importação Ltda - código do produto 8430.

Fabricante: Clone - código do produto 05087.

Fabricante: GoldShip Computer Accessories - código do produto 0470.

Para maiores informações sobre a instalação, consulte o manual do conversor USB/SERIAL que acompanha o produto.

4.5 Driver USB

4.5.1 Instalando Driver USB

Instalação

Utilize o instalador disponível do driver USB.

Feche todas as aplicações no seu PC. Se você está usando um software anti-virus ou firewall, feche-os (ou desabilite suas funções).

Inicie o instalador e siga os menus.

O instalador realiza uma pré-instalação do software do driver. Durante este processo, o driver se torna conhecido do sistema. No Windows XP o arquivo INF é

copiado para a pasta INF e um arquivo INF pré-compilado é criado.

No Windows Vista e no Windows 7 o pacote do driver é armazenado no local de armazenamento de drivers. Quando um dispositivo PnP é conectado ao sistema, ele encontra o driver relacionado e instalá-lo.

No final da instalação é solicitado ao usuário para desconectar e conectar o equipamento.

Durante a instalação do driver uma entrada no painel de controle de programas do windows é criado. Esta entrada pode ser usado para desinstalar o software do driver.

Verificação da Instalação

Você pode verificar se a instalação foi bem sucedida no tópico [Verificando a Instalação do Driver USB.](#)

4.5.2 Verificando a Instalação do Driver USB

Você pode verificar se a instalação foi bem sucedida no gerenciador de dispositivos (o equipamento precisa estar conectado ao PC).

Para executar o gerenciador de dispositivos, clique Iniciar , clique Executar, digite devmgmt.msc, e então clique OK. O gerenciador de dispositivos também pode ser acessado por Configurações > Painel de Controle > Sistema > Hardware > Gerenciador de Dispositivos,

No gerenciador de dispositivos, próximo ao final da lista você deve encontrar a entrada WEG USB devices contendo WEG USBIO V6.00. Isto indica que a instalação foi realizada com sucesso.

Verifique com o fabricante se existe algum driver USB mais novo.

4.5.3 Desinstalando Driver USB

Se você precisa desinstalar o driver USB do PC, siga o procedimento descrito abaixo.

Desinstalação

O desinstalador remove a instalação do driver realizado anteriormente. Durante o processo de remoção o driver é desinstalado e as entradas do registro correspondente e pré-instalação do software de driver são removidos. Após esta etapa o sistema se comporta de tal maneira como se o software do driver nunca tivesse sido instalado.

4.5.4 Atualizando Driver USB

Se você precisa atualizar o driver USB do PC, siga o procedimento descrito abaixo.

Atualização

A atualização de driver é uma combinação de a remoção do software de driver e uma instalação pela primeira vez. O instalador detecta automaticamente se uma atualização ou uma instalação pela primeira vez é necessário.

Utilize sempre o instalador de driver disponível.

Verificação da Instalação

Você pode verificar se a instalação foi bem sucedida no tópico [Verificando a Instalação do Driver USB.](#)

5 Ladder

5.1 Conceitos

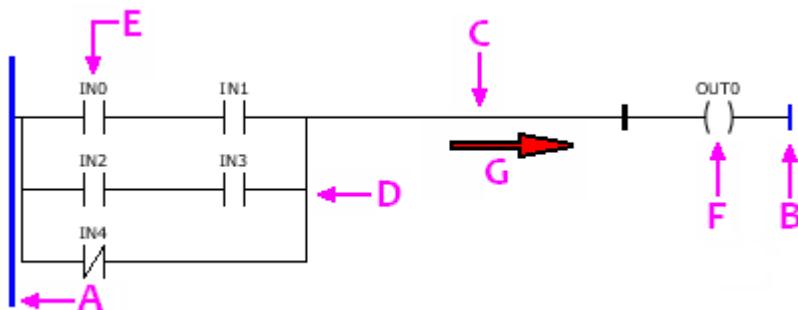
5.1.1 Introdução

Programação Ladder é a representação gráfica de equações booleanas, combinando contatos (argumentos de entradas) com bobinas (resultados de saída).

O programa em Ladder possibilita testar e modificar dados por símbolos gráficos padrões. Estes símbolos são posicionados no diagrama ladder de maneira semelhante a uma linha de um diagrama lógico com relés. O diagrama Ladder é delimitado na esquerda e na direita por linhas de barramento.

Componentes Gráficos

Os componentes gráficos básicos de um diagrama Ladder são mostrados abaixo.



(Figura 1: Fluxo de Funcionamento do Ladder)

- A - Barramento esquerdo
- B - Barramento direito
- C - Ligação horizontal
- D - Ligação vertical
- E - Contato
- F - Bobina
- G - Fluxo de potência

Barramentos

O editor é delimitado na esquerda por uma linha vertical conhecida como barramento esquerdo, e na direita por uma linha vertical conhecida como barramento direito.

Elementos de Ligação e Estados

Os elementos de ligação podem ser horizontal ou vertical. O estado dos elementos de ligação podem ser denotados por 1 ou 0, correspondendo ao valor Booleano

literal 1 ou 0, respectivamente. O termo estado da ligação tem que ser sinônimo do termo fluxo da potência.

O estado do barramento esquerdo pode ser considerado sempre 1. Nenhum estado é definido no barramento direito.

Um elemento de ligação horizontal tem que ser indicado por uma linha horizontal. Um elemento de ligação horizontal transmite o estado do elemento imediatamente a esquerda para o elemento imediatamente a direita.

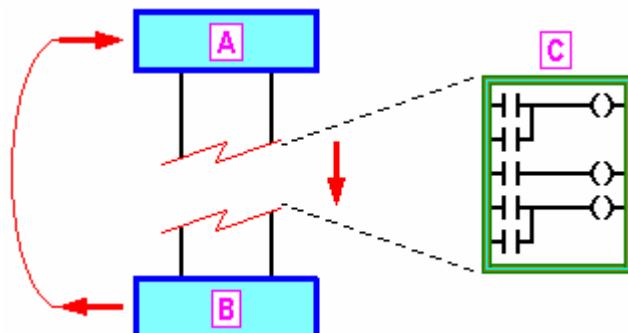
Um elemento de ligação vertical tem que consistir de linhas verticais intersectadas por uma ou mais ligações horizontais em cada lado. O estado da ligação vertical deverá representar o OU lógico dos estados 1 das ligações horizontais no lado esquerdo, isto é, o estado das ligações verticais deverá ser:

- 0 se o estado de todas as ligações horizontais incluídas na sua esquerda são 0,
- 1 se o estado de uma ou mais ligações horizontais incluídas na sua esquerda são 1.

O estado das ligações verticais tem que ser copiados para todas as ligações horizontais associados à sua direita. O estado das ligações verticais não pode ser copiado para as ligações horizontais associadas à sua esquerda.

Controle de Execução

A figura a seguir mostra como o programa em Ladder é executado. O cartão executa continuamente um ciclo de Varredura. O ciclo começa com o Sistema de E/S do hardware, compilando os últimos valores de todos os sinais de entrada e gravando seus valores em regiões fixas da memória.



(Figura 2: Controle de Execução)

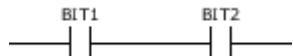
- A - Entradas lidas para a memória
B - Memória escrita nas Saídas
C - Varredura das linhas do ladder

As linhas do programa ladder são então executados num ordem fixa, iniciando com a primeira linha. Durante a varredura do programa, novos valores das saídas físicas., como determinadas a partir da lógica das várias linhas do ladder, são inicialmente inscritos numa região da memória de saída. Finalmente, quando o programa ladder

terminou a execução, todos os valores de saída retidos na memória são inscritos nas saídas físicas pelo hardware numa única operação.

5.1.2 Lógica com Contatos

LÓGICA AND - Contatos em Série



(Figura 1: Contatos em Série)

A figura acima executa uma Lógica AND entre os dois últimos elementos carregados no [STACK](#), baixa um nível do [STACK](#) e joga o resultado para o topo do [STACK](#). Isto significa que é executada a seguinte operação Booleana: topo do [STACK](#) = BIT1. BIT2.

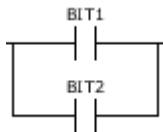
Em linguagem IL (Lista de Instruções) fica:

```
LD  BIT1 (* carrega o valor da variável BIT1 para o STACK = | BIT1 | *)
LD  BIT2 (* carrega o valor da variável BIT2 para o STACK = | BIT2 | BIT1 | *)
AND (* Executa a Lógica AND entre o BIT1 e BIT2 através do STACK = |
      BIT1.BIT2 | *)
```

Tabela Verdade

BIT1	BIT2	STACK
0	0	0
0	1	0
1	0	0
1	1	1

LÓGICA OR - Contatos em Paralelo



(Figura 2: Contatos em Paralelo)

A figura acima executa uma Lógica OR entre os dois últimos elementos carregados no [STACK](#), baixa um nível do [STACK](#) e joga o resultado para o topo do [STACK](#). Isto significa que é executada a seguinte operação Booleana: topo do [STACK](#) = BIT1 + BIT2.

Em linguagem IL (Lista de Instruções) fica:

```

LD  BIT1 (* carrega o valor da variável BIT1 para o STACK = | BIT1 | *)
LD  BIT2 (* carrega o valor da variável BIT2 para o STACK = | BIT 2 | BIT1 | *)
OR   (* Executa a Lógica OR entre o BIT1 e BIT2 através do STACK = | BIT1
      +BIT2| *)

```

Tabela Verdade

BIT1	BIT2	STACK
0	0	0
0	1	1
1	0	1
1	1	1

5.1.3 Terminologia

AT: Representação direta de uma variável,

BLOCO FUNCIONAL (FUNCTION BLOCK): Consiste numa Função que necessita de uma Instância.

CONFIGURAÇÃO (CONFIGURATION): Consiste na organização de um software em um nível mais elevado. Pode conter vários Recursos dentro de uma Configuração.

FUNÇÃO (FUNCTION): É um bloco responsável por executar um determinado comportamento ou uma ação, baseado em possíveis parâmetros (VAR_IN, VAR_IN_OUT, VAR_OUT).

INSTÂNCIA (INSTANCE): Área de memória ocupada conforme o Bloco Funcional.

LD (Ladder Diagram): Linguagem gráfica de programação baseada nos diagramas elétricos (contatos e bobinas interconectadas), conforme o fluxo de potência entre os elementos.

OVERFLOW: Ocorre quando o resultado de um cálculo matemático, ultrapassa os limites permitidos para o tipo de dado do resultado.

POU: Unidade de Organização de Programas. Pode ser: Programa, Bloco Funcional ou Função.

PROGRAMA (PROGRAM): É um agrupamento lógico de todos os elementos de programação e construções necessários ao processamento de sinais requeridos para o controle de uma máquina ou um processo..

RECURSO (RESOURCE): Consistem em qualquer elemento com capacidade de processamento, responsável pela execução dos programas.

SCAN: Ciclo de varredura de um programa.

STACK: Pilha do programa Ladder. É a área de memória utilizada para realizar as Lógicas dos Programas.

TAG: Nome da Variável.

TAREFA (TASK): Responsável pelo controle de execução de programas, de forma periódica ou engatilhada, com disparo por eventos.

TIPO DE DADO (DATATYPE): Informa ao compilador o espaço ocupado por uma variável e seu respectivo formato (aos Blocos).

VARIÁVEL: Consiste numa posição da memória capaz de reter e representar um valor ou expressão. Pode ter escopo:

- Local: cuja posição automática, é calculada pelo compilador.
- Global: localizada numa área de memória determinada, como entradas e saídas digitais, podendo ser acessada em qualquer ponto da Configuração.

A Variável pode ser:

- Retentiva: armazena o valor após a desenergização do equipamento;
- Volátil: inicia com o valor contido no campo de valor inicial, após a energização do equipamento;
- Constante: não permite alteração do seu conteúdo.

VAR: Variáveis de uso interno de um Bloco do Usuário (USERFB). Equivalente a uma variável de escopo Local.

VAR_IN: Argumento de entrada de um Bloco do Usuário (USERFB). As variáveis configuradas neste campo, apenas serão lidas no USERFB.

VAR_IN_OUT: Argumento de entrada e de saída de um Bloco do Usuário (USERFB). Não permite variáveis CONSTANTES, ENTRADAS FÍSICAS (%I_), ENTRADAS DE REDE (%I_) ou MARCADORES DE SISTEMA DE LEITURA (%S_), pois elas serão lidas no início do USERFB e serão atualizadas com novos valores (escritas) no final do USERFB.

VAR_OUT: Argumento de saída de um Bloco do Usuário (USERFB). Não permite variáveis CONSTANTES, ENTRADAS FÍSICAS (%I_), ENTRADAS DE REDE (%I_)

ou MARCADORES DE SISTEMA DE LEITURA (%S_), pois elas serão atualizadas com novos valores (escritas) no final do USERFB.

WATCHDOG: É um meio proporcionado pelo fabricante para dempenhar ações específicas, se a integridade do sistema for violado.

5.1.4 Tipo de Dados

Tipo de Dado	Tamanho	Sinal	Faixa
BOOL	Aloca 1 bit de memória.		0 ou 1
BYTE	Aloca 8 bits de memória (1 byte).		0 a 255
USINT	Aloca 8 bits de memória (1 byte).		0 a 255
SINT	Aloca 8 bits de memória (1 byte).	Sim	-128 a 127
WORD	Aloca 16 bits de memória (2 bytes).		0 a 65535
UINT	Aloca 16 bits de memória (2 bytes).		0 a 65535
INT	Aloca 16 bits de memória (2 bytes).	Sim	-32768 a 32767
DWORD	Aloca 32 bits de memória (4 bytes).		0 a ($2^{32} - 1$)
UDINT	Aloca 32 bits de memória (4 bytes).		0 a ($2^{32} - 1$)
DINT	Aloca 32 bits de memória (4 bytes).	Sim	-2^{31} a ($2^{31} - 1$)
LWORD	Aloca 32 bits de memória (8 bytes).		0 a ($2^{64} - 1$)
ULINT	Aloca 32 bits de memória (8 bytes).		0 a ($2^{64} - 1$)
LINT	Aloca 32 bits de memória (8 bytes).	Sim	-2^{63} a ($2^{63} - 1$)
REAL	Dado em ponto flutuante (com ponto decimal). Aloca 32 bits de memória, conforme padrão IEEE 559.	Sim	$+/-10^{+/-38}$; Precisão = 1 / (2^{23})
LREAL	Dado em ponto flutuante (com ponto decimal). Aloca 64 bits de memória, conforme padrão IEEE 559.	Sim	$+/-10^{+/-308}$; Precisão = 1 / (2^{52})

5.1.5 Representação Direta

Utilizado para definir a posição de memória de uma Variável Global.

Sintaxe: %<Formato><Tamanho>

Primeira Letra:

At - Format o	Descrição
I	Entradas (Input): Recebe os valores das variáveis analógicas e discretas, ou variáveis de rede de entrada.
Q	Saída (Output): Armazena os valores a serem escritos nas saídas

	análogicas e discretas, ou variáveis de rede de saída.
M	Marcador interno (em RAM).
S	Marcador de Sistema de Leitura.
C	Marcador de Sistema de Comando.

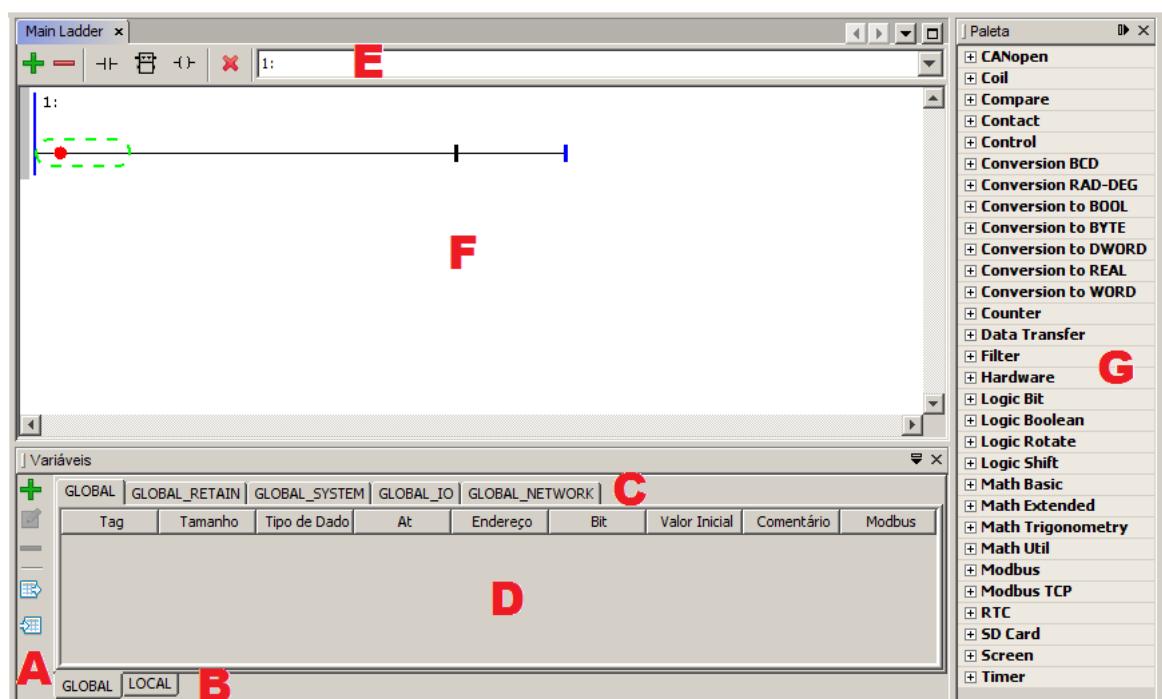
Segunda Letra:

At - Tamanho	Descrição
X	Bit
B	Byte (8 bits)
W	Word (16 bits)
D	Double word (32 bits)
L	Long word (64 bits)

5.2 Editor

5.2.1 Área de Trabalho

Sempre que um arquivo ladder for aberto através do seu atalho na árvore de projetos o editor variáveis/ladder aparecerá conforme figura a seguir.



(Figura 1: Área de Trabalho do Editor Ladder do WPS)

O editor variáveis/ladder possui os seguintes componentes:

- A - Barra de ferramentas para edição de variáveis
- B - Escopo das variáveis
- C - Grupo de variáveis
- D - [Editor/lista de variáveis](#)
- E - Barra de ferramentas de edição de programas
- F - [Editor ladder/rungs](#)
- G - Paleta de componentes ladder*

* Caso não esteja visível ativar pelo menu Janela opção Paleta

5.2.2 Menu Ladder

Ao editar um arquivo ladder do seu recurso o menu ladder estará ativo conforme figura a seguir.

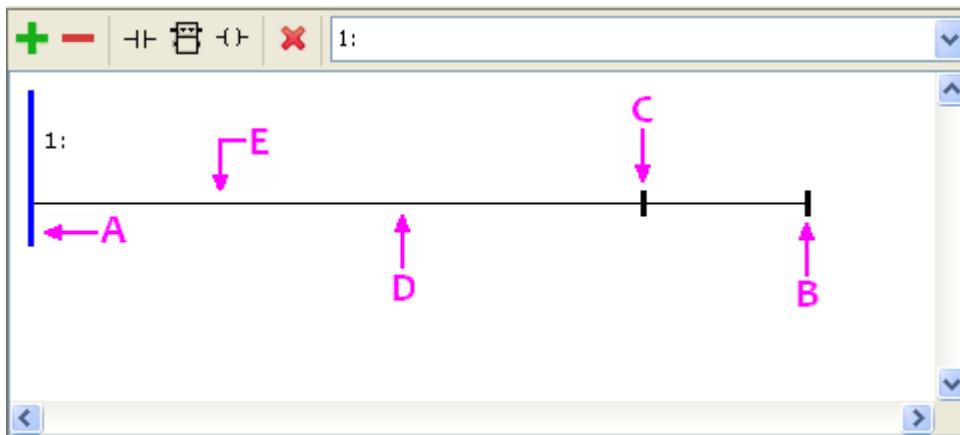
Ladder	
Comunicação	Ferramentas
Janela	Ajuda
Insere nova rung no final	Ctrl+Alt+N
Insere nova rung abaixo	Ctrl+Alt+B
Insere nova rung acima	Ctrl+Alt+A
Seleciona rung abaixo	Ctrl+Abaixo
Seleciona rung acima	Ctrl+Acima
Apaga rung selecionada	Ctrl+Excluir
Seleciona próximo elemento	Direita
Seleciona elemento anterior	Esquerda
Seleciona elemento abaixo	Abaixo
Seleciona elemento acima	Acima
Insere contato	C
Insere bloco (Função/Funcional)	F
Insere saída	O
Seleciona grupo de elementos	G
Apaga elemento	
Edita argumentos	F2
Edita título da rung	Shift+F2

Através desse menu é possível fazer todas as operações relativas ao ladder bem como conhecer os atalhos de teclado para essas operações.

5.2.3 Rungs

5.2.3.1 Rungs - Visão Geral

O programa ladder é editado através de um editor gráfico que organiza o mesmo através de rungs. Cada rung do programa ladder corresponde a um intertravamento que relaciona elementos de entrada a elementos de saída. Esse editor é mostrado conforme figura a seguir.



(Figura 1: Elementos da Rung)

Os elementos principais de uma rung são:

- A - Barramento esquerdo
- B - Barramento direito
- C - Barramento de saídas(*)
- D - Ligações entre elementos
- E - Título e comentário da rung

(*) As saídas sempre estarão conectados a direita do barramento de saídas

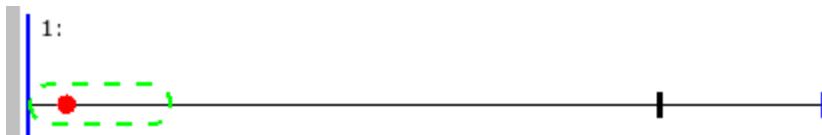
Maiores detalhes da programação ladder [clique aqui](#).

Através da barra de ferramentas do editor ladder/runs é possível executar as seguintes operações:

- Adicionar uma nova rung
- Remover a rung selecionada
- Inserir um contato na rung
- Inserir um bloco funcional na rung
- Inserir um bobina de saída na rung
- Apagar um elemento da rung (contato, bloco funcional ou bobina)
- Selecionar a rung para edição

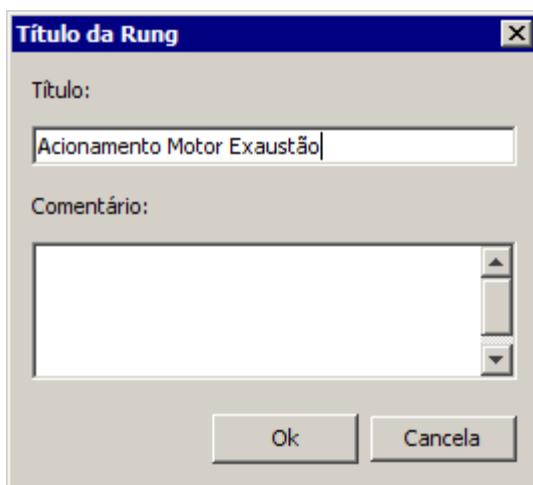
5.2.3.2 Editando

Para editar um rung a mesma deve estar selecionada. Para fazer a seleção da rung você pode clicar com o mouse sobre a mesma ou utilizar o controle de seleção de rungs na barra de ferramentas do editor ladder. Sempre que a rung estiver selecionada existirá uma barra cinza na parte esquerda da rung conforme figura a seguir.



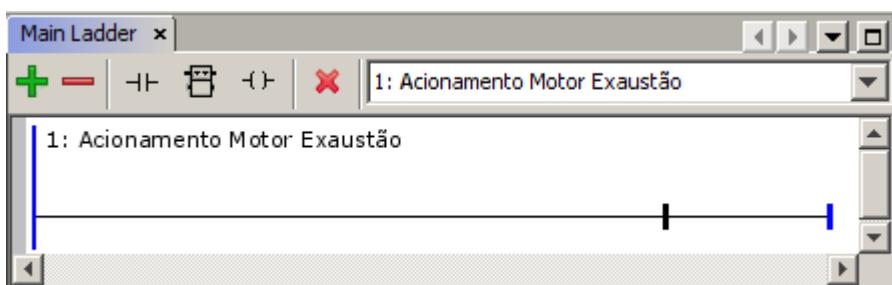
5.2.3.3 Título e Comentário

Para editar o título e comentário da rung basta dar um duplo clique do mouse sobre a região do título e comentário ou pressionar as teclas shift+F2. O seguinte diálogo aparecerá.



(Figura 1: Editando o Título da Rung)

Após editar o título e comentário o editor ficará da seguinte maneira.

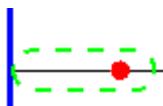


(Figura 2: Rung com o seu Título)

5.2.3.4 Inserindo Elementos

5.2.3.4.1 Inserindo Elementos - Visão Geral

Para inserir um elemento ladder na rung é necessário que um elemento já existente do ladder esteja selecionado, conforme figura a seguir.



O elemento selecionado estará envolvido por um retângulo verde e riscado. Existirá também um ponto vermelho indicando aonde o novo elemento será inserido (ponto de inserção).

A inserção de elementos ladder pode ser feita através de três maneiras distintas.

1 - Pelo teclado através dos seguintes atalhos

tecla **C** - inserir contato

tecla **F** - inserir bloco funcional

tecla **O** - inserir bobina de saída

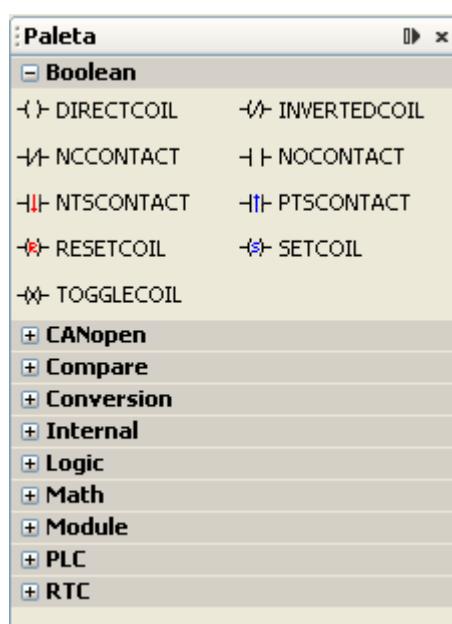
2 - Pelo mouse pela barra de ferramentas do editor ladder/rungs

- Inserir um contato na rung

- Inserir um bloco funcional na rung

- Inserir um bobina de saída na rung

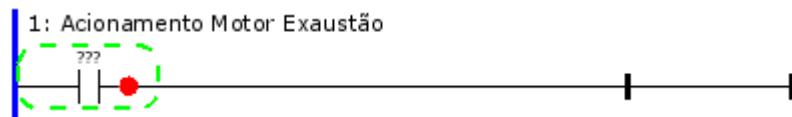
3 - Pelo mouse pela paleta de elementos ladder*



Para inserir elementos pela paleta deve-se utilizar a operação arrastar e soltar do mouse. Para isso clique sobre o elemento da paleta, mantenha o mouse pressionado, mova o mouse até o ponto de inserção no ladder e solte o botão do mouse.

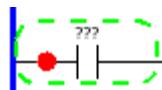
* Caso a paleta não esteja visível ativar pelo menu Janela opção Paleta

Após inserir um elemento, por exemplo um contato, a rung ficará da seguinte maneira.

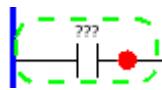


5.2.3.4.2 Em Série

Através do ponto de inserção dos elementos é possível inserir um elementos em série. Para isso utilize os seguintes pontos de inserção:



- Insere elemento em série antes do elemento selecionado



- Insere elemento em série depois do elemento selecionado

Inserir elementos em série corresponde a lógica booleana AND conforme explicação a seguir:

5.2.3.4.3 Em Paralelo

Através do ponto de inserção dos elementos é possível inserir um elementos em paralelo. Para isso utilize os seguintes pontos de inserção:



- Insere elemento em paralelo acima do elemento selecionado



- Insere elemento em paralelo abaixo do elemento selecionado

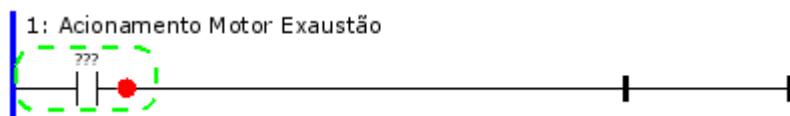
Inserir elementos em série corresponde a lógica booleana OR conforme explicação a seguir:

5.2.3.5 Navegando

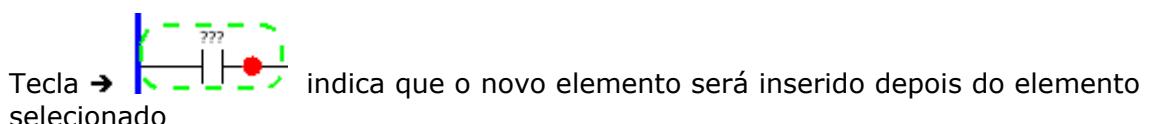
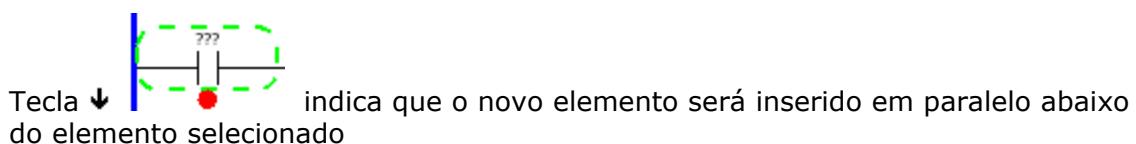
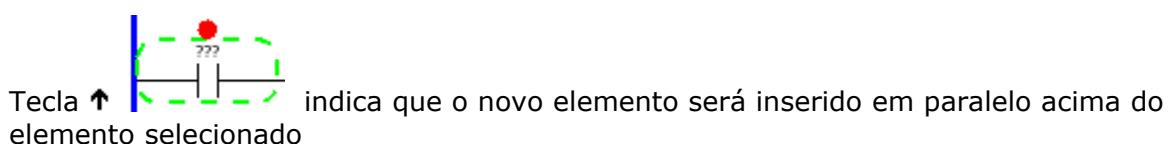
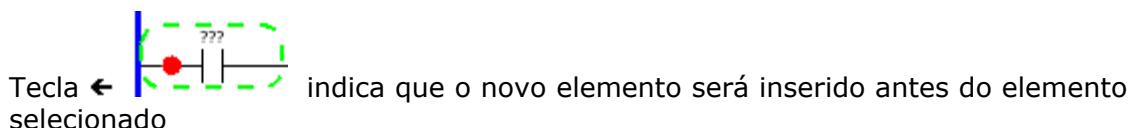
5.2.3.5.1 Com o Teclado

A navegação pelo teclado na rung é feito pelas teclas **←, →, ↑, ↓**. Através dessas teclas é possível selecionar os diversos elementos que estão dentro da rung e também definir o ponto de inserção dos novos elementos.

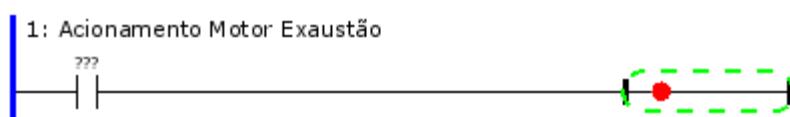
Conforme mostrado no item anterior ao [inserir um elemento](#) do tipo contato na rung a mesma ficará da seguinte maneira.



A pressionar as teclas de navegação podemos modificar o ponto de inserção de um novo elemento conforme figuras a seguir:



Estando o ponto de inserção já selecionado e ao clicar a mesma tecla da direção correspondente a seleção passará para o próximo elemento conforme figura a seguir.



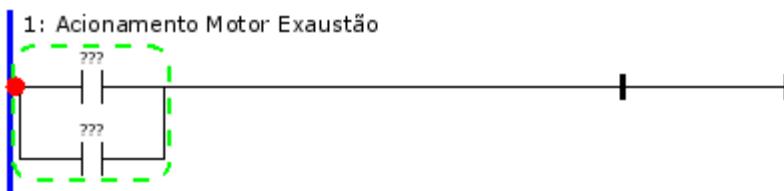
Ao inserir elementos em paralelo ao navegar pelos elementos navegamos por cada elemento individualmente conforme exemplo a seguir.

1: Acionamento Motor Exaustão

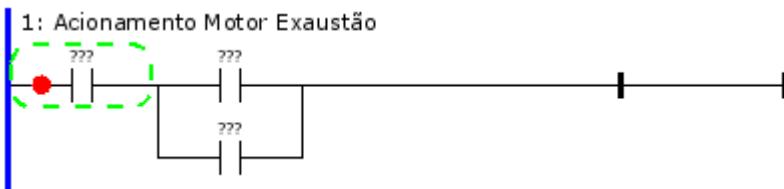


Nessa situação ao inserirmos um elemento estaremos fazendo essa inserção relacionada a esse elemento. Caso seja necessário inserir um elemento relacionado ao paralelo que é um grupo de elementos devemos utilizar a tecla **G** que selecionara o grupo e então poderemos definir o ponto de inserção e fazer a inserção conforme figuras a seguir.

- Pressionado a tecla **G**:

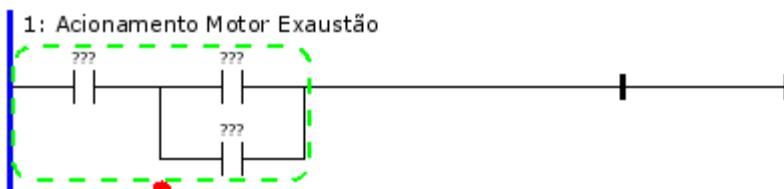


- Inserido contato antes do paralelo:

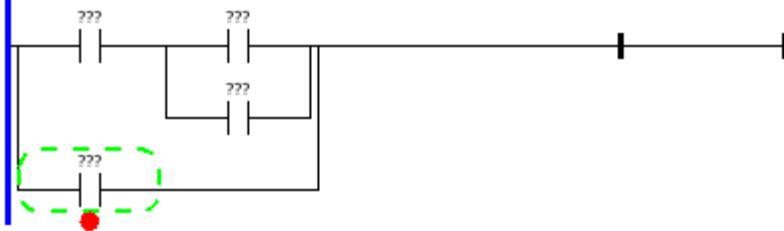


Ao cada toque da tecla **G** selecionamos o grupo imediatamente superior ao elemento/grupo selecionado e ao final voltamos para o elemento original, então em algumas situações é necessário mais que um toque da tecla **G**. Através da tecla **G** é possível fazer diversas operações de inserção conforme exemplos a seguir.

- Inserir um elemento em paralelo ao grupo de elementos da rung

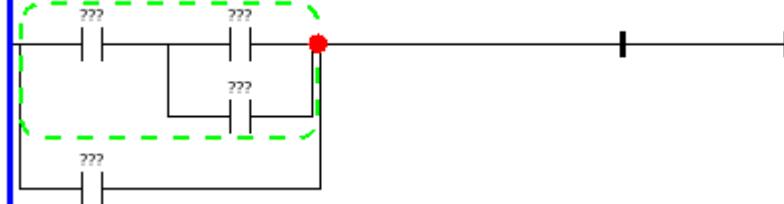


1: Acionamento Motor Exaustão

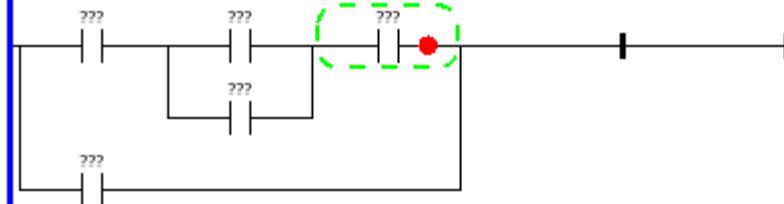


- Inserir um elemento em serie ao grupo formado por elementos internos a um paralelo

1: Acionamento Motor Exaustão



1: Acionamento Motor Exaustão

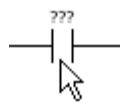
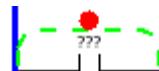


5.2.3.5.2 Com o Mouse

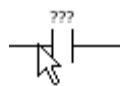
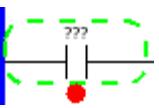
Todas as funcionalidades de navegação pelo teclado na rung também estão disponíveis no mouse. Diretamente pelo clique do mouse é possível selecionar um elemento e o respectivo ponto de inserção.



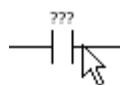
- Seleciona elemento e ponto de inserção superior



- Seleciona elemento e ponto de inserção inferior



- Seleciona elemento e ponto de inserção anterior

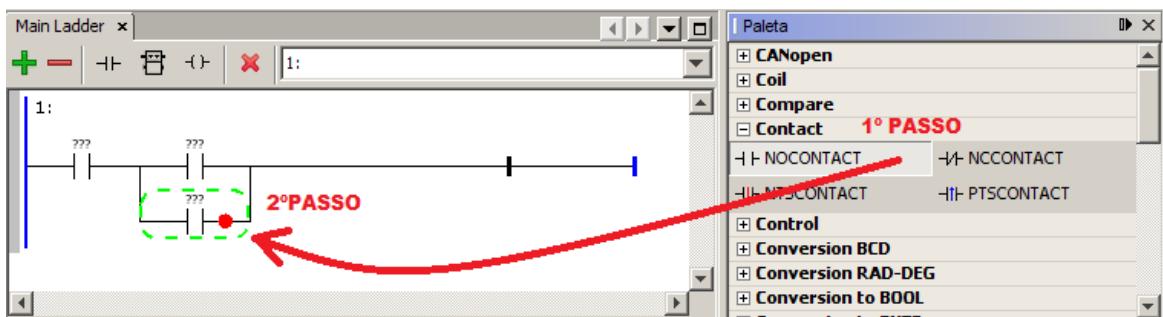


- Seleciona elemento e ponto de inserção posterior



Durante a operação de inserção de elemento via mouse, através do arrastar e soltar

do mouse da paleta, a medida que arrastamos os elementos em cima da rung a seleção e o ponto de inserção acompanha o cursor do mouse de modo a determinar o ponto de inserção. A figura a seguir exemplifica a inserção de elementos via mouse.

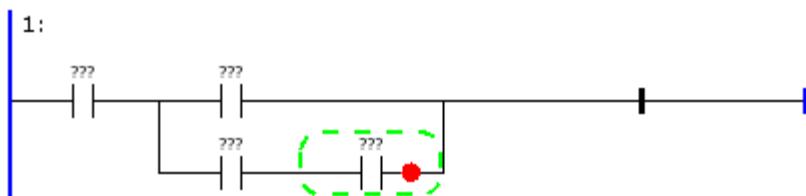


(Figura 1: Operação de arrastar componente com o Mouse)

Sequência de inserção de elementos via mouse:

- 1 - Selecione o elemento da paleta que deseja inserir clicando com o mouse sobre o mesmo e mantendo o botão do mouse pressionado
- 2 - Arraste esse elemento sobre a rung
- 3 - Selecione o ponto de inserção e solte o botão do mouse sobre ele

O resultado da inserção nesse exemplo será os seguinte.



(Figura 2: Resultado após a operação de arraste)

Durante a inserção via mouse também é possível selecionar o grupo de elementos de maneira análoga a tecla G. Para isso durante o arraste do elemento previamente selecionado pressione e mantenha pressionada a tecla modificador Ctrl, enquanto a tecla Ctrl estiver pressionada o mouse irá selecionar pontos de inserção relacionados a grupos.

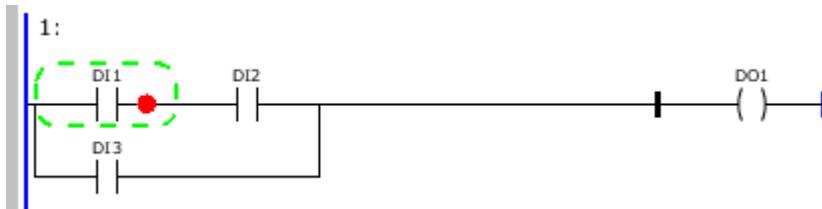
5.2.3.6 Copiar / Colar

Visão Geral

Todas funcionalidades de copiar, recortar e colar estão disponíveis no editor ladder através do menu editar ou dos atalhos de teclado correspondentes. Para executar essas operações é necessário ter rung e/ou elemento selecionado.

Copiar / colar um elemento

A seguir um exemplo de copiar colar de um elemento.



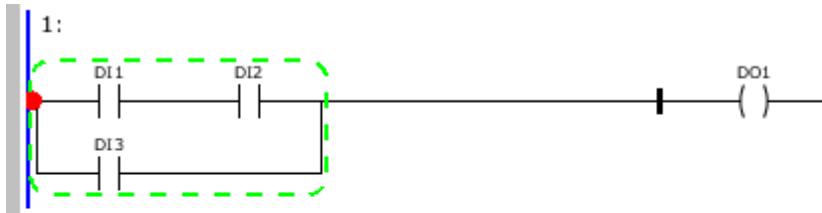
(Figura 1: Antes, elemento selecionado e pressionado Ctrl+C)



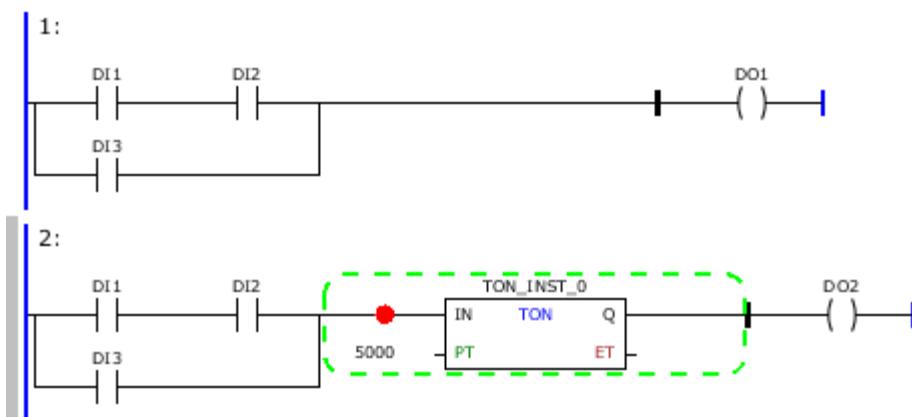
(Figura 2: Depois, contato selecionado com ponto de inserção depois e pressionado Ctrl+V)

Copiar / colar de múltiplos elementos

Também é possível copiar, recortar e colar múltiplos elementos selecionados ou pelo mouse ou pela tecla G conforme mencionado anteriormente. A seguir um exemplo de copiar colar de múltiplos elementos.



(Figura 3: Antes, grupo selecionado e pressionado Ctrl+C)

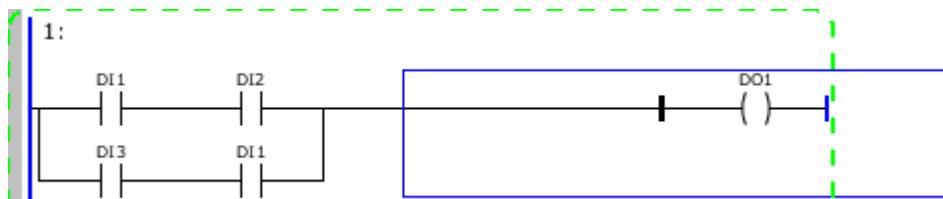


(Figura 4: Depois, temporizador selecionado com ponto de inserção antes e pressionado Ctrl+V)

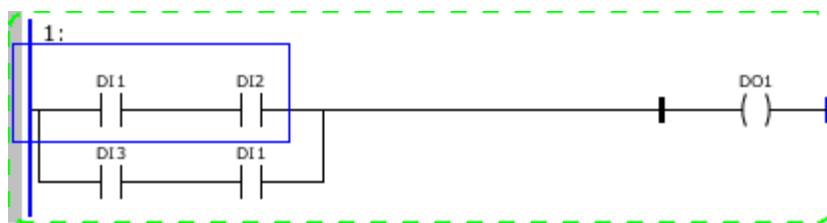
* Poderão haver situações que o colar não será possível e então não haverá alteração na rung após o comando.

Copiar / colar de rungs

A seleção de rungs para as operações de copiar, recortar e colar é feita somente através do mouse conforme figuras a seguir.



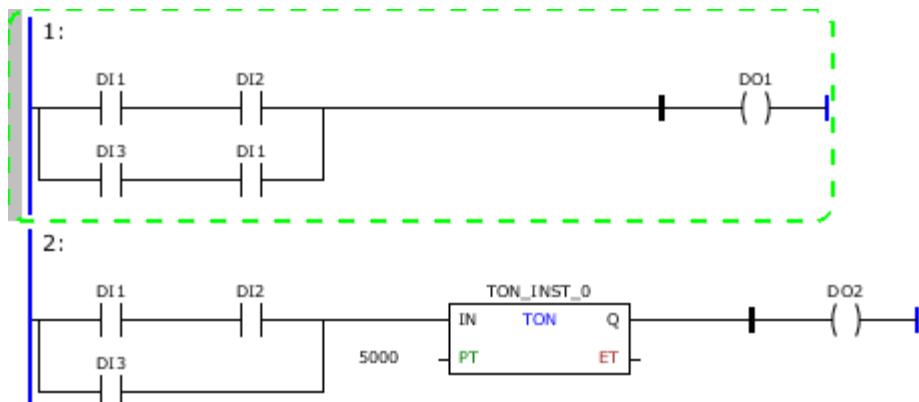
(Figura 5: Através do mouse a partir do lado de fora direito da rung)



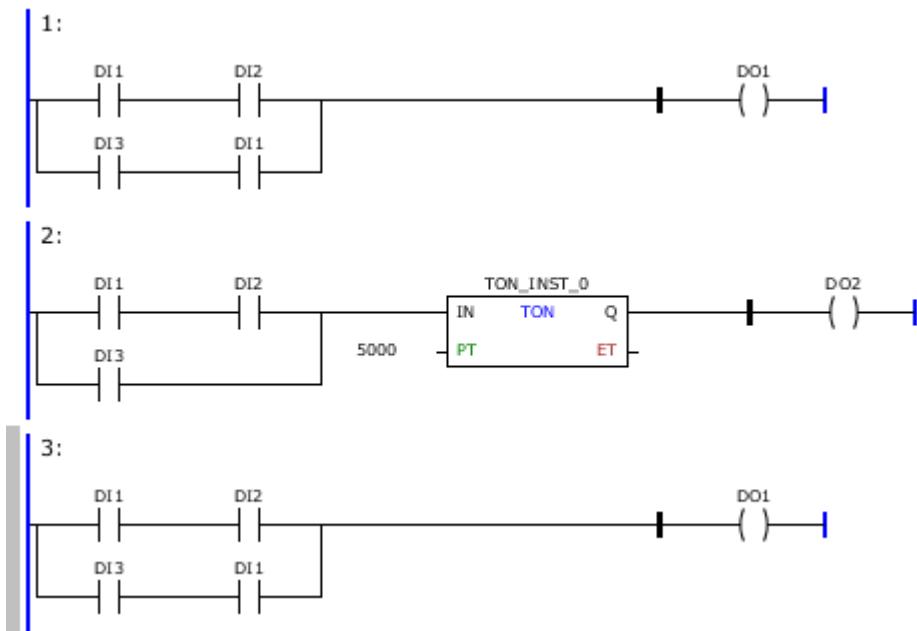
(Figura 6: Através do mouse a partir do lado esquerdo da rung)

* Também é possível selecionar múltiplas rungs para essas operações.

A seguir exemplo de copiar / colar de rung.



(Figura 7: Antes, rung selecionada e pressionado Ctrl+C)



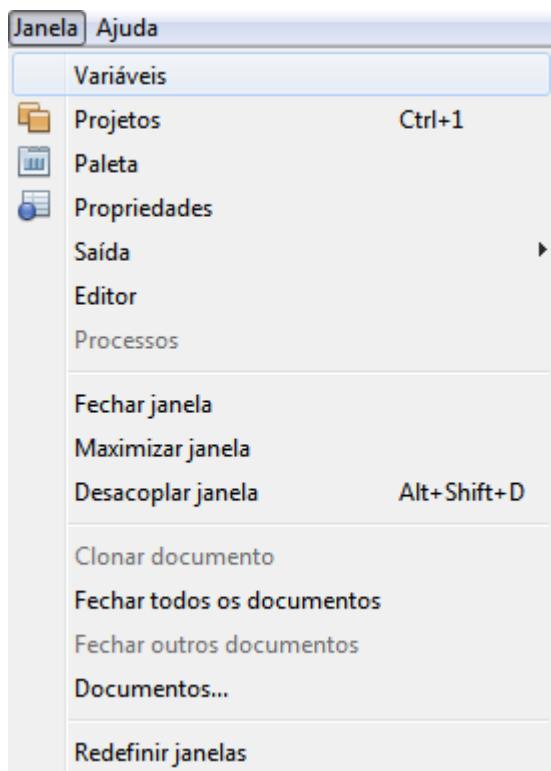
(Figura 8: Depois, Selecionado rung 2 e pressionado Ctrl+V, foi criado rung 3 e colado o conteúdo copiado anteriormente)

5.2.4 Variáveis

5.2.4.1 Variáveis - Visão Geral

Durante a edição do ladder é necessário definir as variáveis utilizadas nos componentes ladder, essa definição pode ser feita diretamente no ladder e/ou no editor/lista de variáveis.

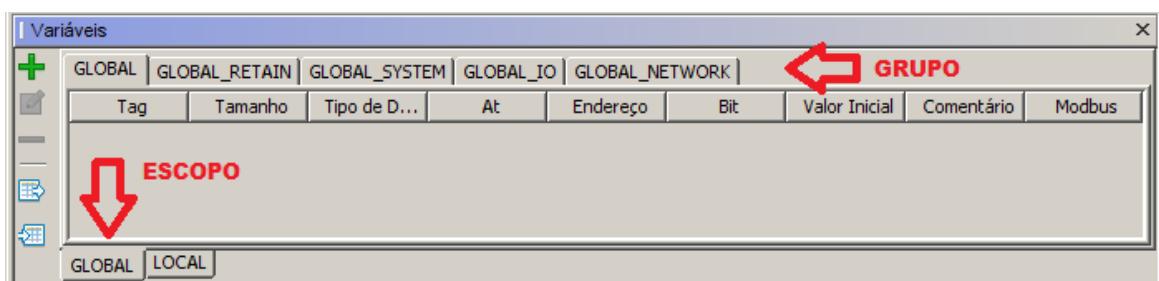
A tabela de variáveis deve ser ativada através do comando "Menu-Janela-Variáveis" conforme figura a seguir.



Na janela de edição de variáveis estão disponíveis os seguintes comandos:

- Adicionar uma nova variável
- Edita a variável selecionada
- Remover a variável selecionada
- Exporta variáveis deste grupo para um arquivo
- Importa variáveis para este grupo de um arquivo

As variáveis do escopo "GLOBAL" apresentam os grupos conforme com a figura 1.



(Figura 1: Editor de Variáveis para o Escopo "GLOBAL")

Nesta área de variáveis globais, estão disponíveis os seguintes grupos:

- Constante: variáveis que armazenam valores constantes
- Global: variáveis acessíveis por todos os arquivos ladder e por todas as telas de IHM
- Global Retentivo: semelhante ao grupo global mais com memória retentiva (*)

- Global de Sistema: variáveis previamente definidas com funções do sistema
- Global de I/O: entrada e saídas físicas do equipamento
- Global de Rede: variáveis previamente definidas para comunicação em rede

As variáveis do escopo "LOCAL" apresentam os grupos conforme a figura 2.



(Figura 2: Editor de Variáveis para o Escopo "LOCAL")

Nesta área de variáveis locais, estão disponíveis os seguintes grupos:

Local: variáveis acessíveis somente pelo ladder que está sendo editado

Local Retentivo: semelhante ao grupo local mais com memória retentiva (*)

(*) Uma memória retentiva mantém seu valor mesmo com o equipamento desligado.

5.2.4.2 Campos

Ao definir uma variável através do editor/lista de variáveis alguns dados deverão ser definidos para as variáveis. Nos itens a seguir serão apresentados esses dados de acordo com o grupo que a variável pertence.

- Local e Local Retentivo:

Tag	Tamanho	Tipo de Dado	Valor Inicial	Comentário
VARIABLE0	0	BOOL	0	

- Tag: identificação da variável
- Tamanho: número de elementos do array(*) relacionado a variável
- Tipo de Dado: tipo numérico da variável
- Valor Inicial: valor que será carregado para variável durante a inicialização do equipamento
- Comentário: comentário da variável no idioma selecionado

- Constante:

Tag	Tipo de Dado	Valor	Comentário
VARIABLE1	BOOL	0	

- Tag: identificação da variável

- Tipo de Dado: tipo numérico da variável
- Valor: valor constante da variável
- Comentário: comentário da variável no idioma selecionado

- Global e Global Retentivo:

Tag	Tamanho	Tipo de Dado	At	Endereço	Bit	Valor Inicial	Comentário
VARIABLE2	0	BOOL	%MB	0	0	0	

- Tag: identificação da variável
- Tamanho: número de elementos do array(*) relacionado a variável
- Tipo de Dado: tipo numérico da variável
- At: define qual área de memória global a variável acessa
- Endereço: endereço relacionado a área de memória global. Caso não seja configurado (vazio), o compilador irá definir automaticamente o seu endereço.
- Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
- Valor Inicial: valor que será carregado para variável durante a inicialização do equipamento
- Comentário: comentário da variável no idioma selecionado

- Instância (removido a partir do WPS V1.30, pois ficou integrado com os demais tipos de dados):

Tag	Instância	Comentário
INSTO	TON	

- Tag: identificação da variável
- Instância: tipo da estrutura de dados relacionada ao bloco funcional que a mesma será utilizada
- Comentário: comentário da variável no idioma selecionado

- Sistema:

Tag	Tipo de Dado	At	Endereço	Bit	Comentário
FIRMWARE	WORD	%SW	3000		Versão do firmware

- Tag: identificação da variável
- Tipo de Dado: tipo numérico da variável
- At: define qual área de memória global a variável acessa
- Endereço: endereço relacionado a área de memória global
- Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
- Comentário: comentário da variável no idioma selecionado

- I/O:

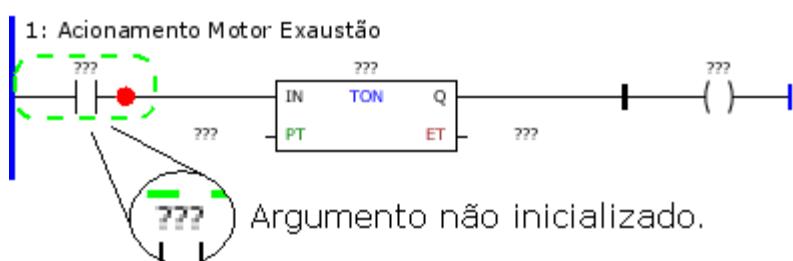
Tag	Tamanho	Tipo de Dado	At	Endereço	Bit	Comentário
DI1	0	BOOL	%IB	0	0	Entrada digital 1

- Tag: identificação da variável
- Tamanho: número de elementos do array(*) relacionado a variável
- Tipo de Dado: tipo numérico da variável
- At: define qual área de memória global a variável acessa
- Endereço: endereço relacionado a área de memória global
- Bit: para tipo de dados booleanos e necessário definir o bit que a mesma acessa (0...7)
- Comentário: comentário da variável no idioma selecionado

(*) Quando tamanho > 0 as variáveis são acessadas no ladder através do seu índice de array

5.2.4.3 Editando na Rung

Os elementos ladder inseridos na rung necessitam que variáveis sejam definidas para cada argumento. Veja figura a seguir.



(Figura 1: Variáveis sem declaração nos elementos e blocos)

Para definir uma variável para o argumento é necessário entrar no modo de edição do argumento que é feito através de duas maneiras.

Atenção! A partir da versão 1.30 do WPS os argumentos de saída dos blocos funcionais podem ser opcionais.
Os argumentos em questão não serão inicializados com a declaração "???" e poderão ser omitidos caso não necessitem serem utilizados na lógica ladder.

1 - Pelo mouse:

Executando um duplo clique do mouse diretamente no argumento

2 - Pelo teclado:

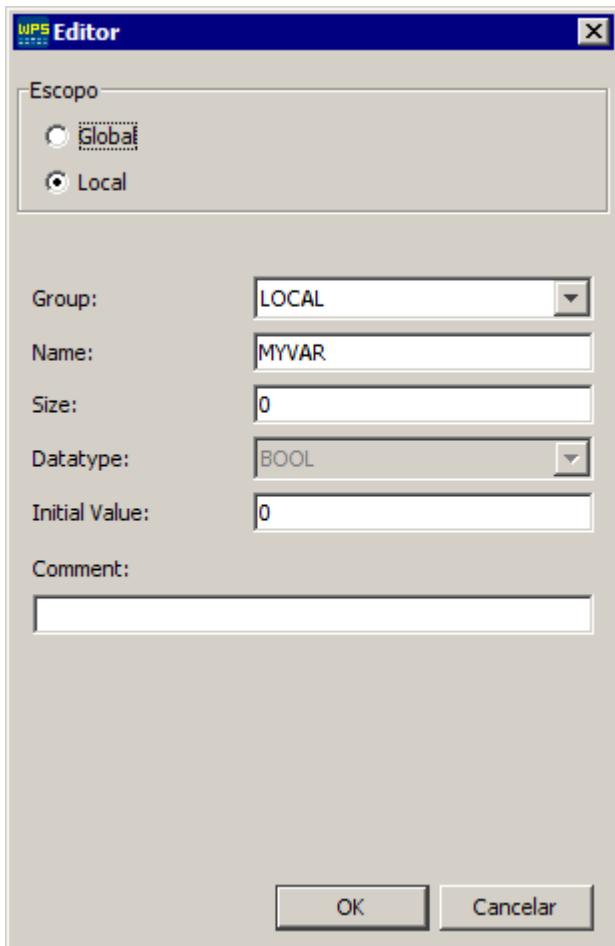
Pressionando a tecla F2. Para elementos com 1 argumento entra no modo de edição de argumento diretamente, nos outros elementos com mais de 1 argumento é necessário selecionar o argumento através das teclas direcionais e então pressionar novamente a tecla F2.

Ao entrar no modo de edição o elemento aparecerá similar a seguinte figura.



(Figura 2: Atribuindo as variáveis)

Neste momento uma caixa de edição estará habilitada para que você entre com o nome da variável. Ao pressionar o botão "Edita" uma caixa para criar a nova variável será habilitada.

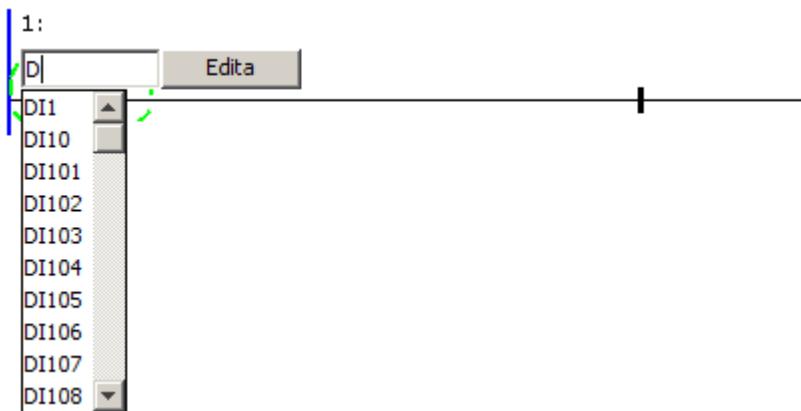


(Figura 3: Criando a variável se o seu tag não está definido)

Nesta caixa você deverá definir as seguintes opções para a variável:

- Escopo: se pertence ao grupo Global ou Local
- Tag: identificação da variável
- Tamanho: número de elementos do array* relacionado a variável
- Tipo de Dado: tipo numérico da variável
- Grupo: Grupo a qual a variável pertence

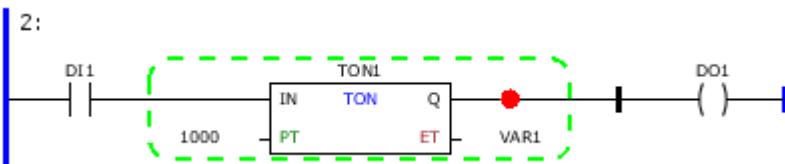
Caso já existam variáveis definidas para o tipo compatível com o elemento ladder uma caixa de seleção com essas variáveis aparecerá em conjunto com a caixa de edição. Para selecionar a variável desejada aperte a tecla direcional para baixo e após estar com a variável selecionada pressione enter. A figura a seguir mostra essa funcionalidade.



(Figura 4: Selecionando variáveis ao digitar)

5.2.4.4 Literais na Rung

Nos blocos funcionais também é possível a entrada de valores literais conforme figura a seguir.



(Figura 1: Exemplo de Programa)

Nesse exemplo foi configurado a entra PT do bloco TON com o valor 1000 que é um literal.

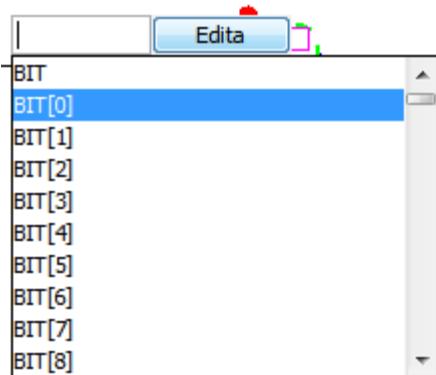
Para a entrada de literais deve ser usado a seguintes convenções:

- Número inteiro não tem ponto.
Ex. 12, 1000, 1555
- Número reais em ponto flutuante devem obrigatoriamente ter ponto.
Ex. 1.5, 2.25, 3.0
- Números representados em hexadecimal obrigatoriamente devem definir o tipo do dado.
Ex. BYTE#16#7F, WORD#16#3CF0, DWORD#16#00FF0088
- Números representados em binário obrigatoriamente devem definir o tipo do dado.
Ex. BYTE#2#1010_0000, WORD#0111_0000_0000_0001

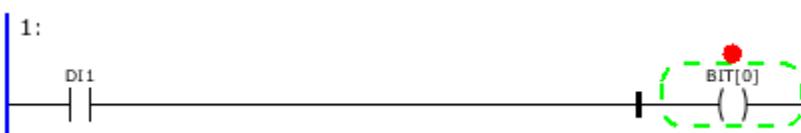
 Atenção! Em alguns blocos por questões de verificação de consistência de dados será necessário definir o tipo do dado do literal através de notação específica que terá as seguintes opções: BOOL#, BYTE#, INT#, UINT#, DINT#, UDINT#, WORD#, DWORD# e REAL#. Exemplo: WORD#17321

5.2.4.5 Arrays na Rung

Nos contatos, bobinas e blocos funcionais também é possível o acesso de variáveis do tipo array conforme figuras a seguir.



(Figura 1: Acesso de array)



(Figura 2: Array na Rung)

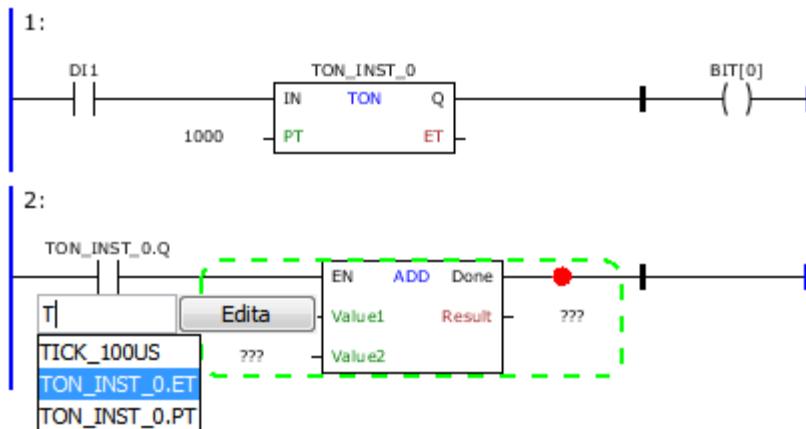
Para visualizar o índices de uma variável do tipo array expanda a variável na janela de variáveis conforme figura a seguir.

Variáveis				
+	LOCAL	LOCAL_RETAIN		
	Tag	Tamanho	Tipo de Dado	Valor Inicial
-	BIT	10	BOOL	0
+	BIT[0]	0	BOOL	
+	BIT[1]	0	BOOL	
+	BIT[2]	0	BOOL	
+	BIT[3]	0	BOOL	
+	BIT[4]	0	BOOL	
+	BIT[5]	0	BOOL	
+	BIT[6]	0	BOOL	
+	BIT[7]	0	BOOL	
+	BIT[8]	0	BOOL	
+	BIT[9]	0	BOOL	
+	TON_INST_0	0	TON	
+	DWO	0	DWORD	0

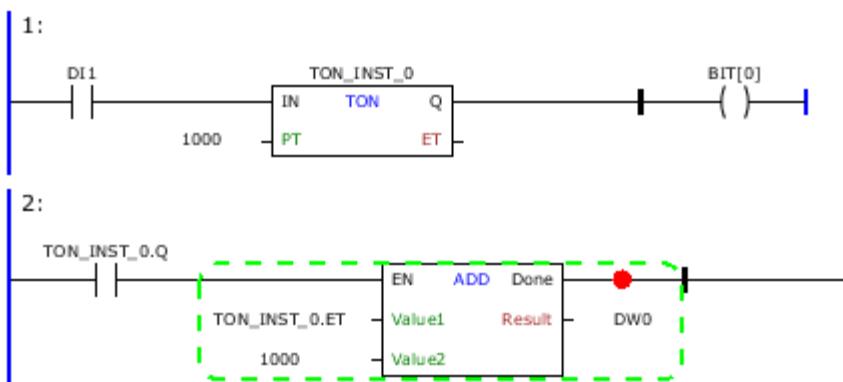
(Figura 3: Exibindo índices de uma variável do tipo array)

5.2.4.6 Instâncias e Estruturas na Rung

Nos contatos, bobinas e blocos funcionais também é possível o acesso as variáveis interna de instâncias e estruturas conforme figuras a seguir.



(Figura 1: Acesso de variável interna de instância ou estrutura)



(Figura 2: Variável interna de instância ou estrutura na Rung)

Para visualizar as variáveis internas das instâncias e estrutura expanda a variável na janela de variáveis conforme figura a seguir.

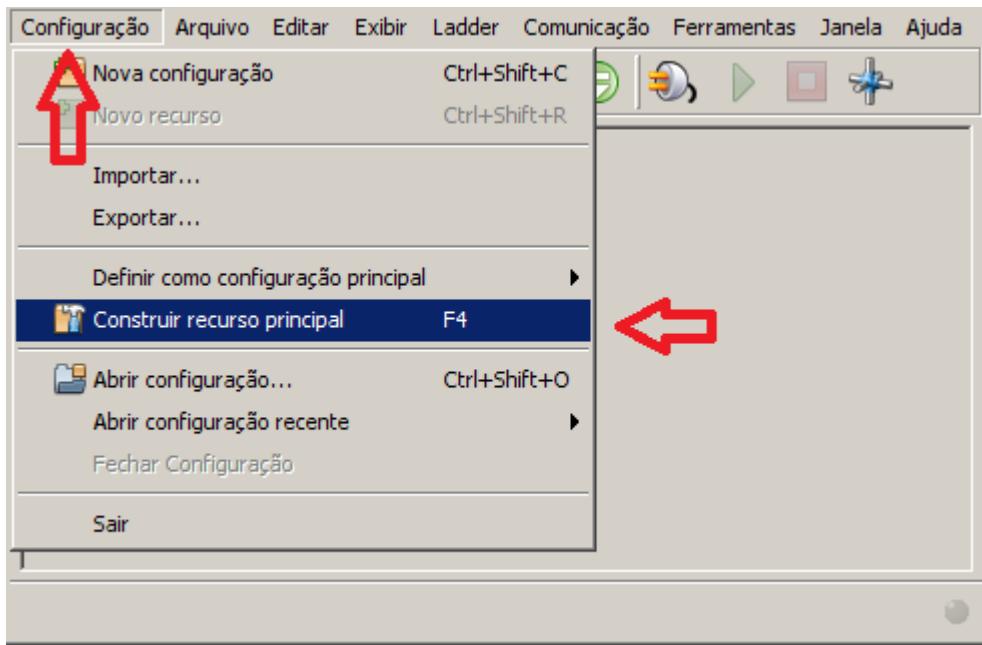
Variáveis				
	LOCAL	LOCAL_RETAIN		
+	BIT	10	BOOL	0
+	TON_INST_0	0	TON	
+	TON_INST_0.PT	0	DWORD	
+	TON_INST_0.ET	0	DWORD	
+	TON_INST_0.IN	0	BOOL	
+	TON_INST_0.Q	0	BOOL	
+	DWO	0	DWORD	0

(Figura 3: Exibindo variáveis internas de instância ou estrutura)

5.2.5 Compilar

Para compilar um Programa (POU), há 4 opções:

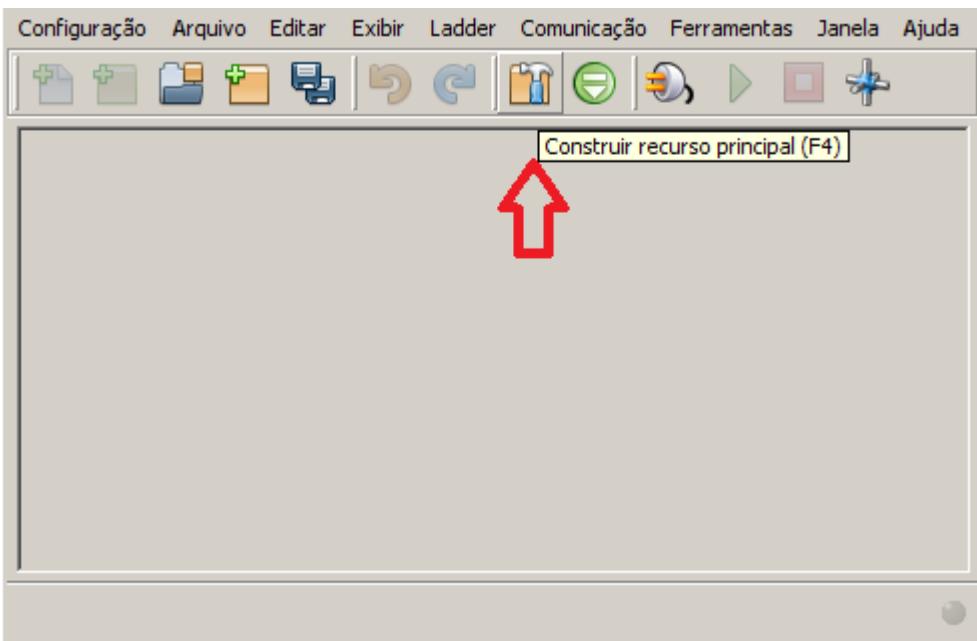
- 1) Através do Menu-Configuração-Construir recurso principal (F4):



(Figura 1: Compilar a partir do Menu)

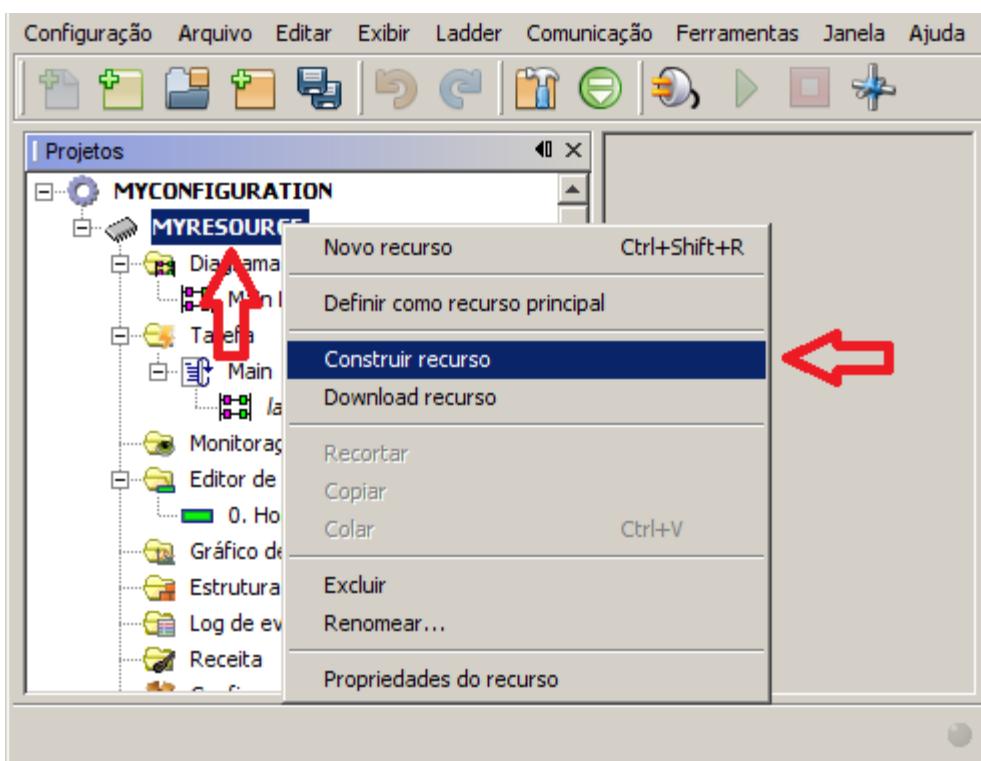
- 2) Através da tecla de atalho F4:

- 3) Através do botão na Barra de Ferramentas:



(Figura 2: Compilar a partir da Barra de Ferramentas)

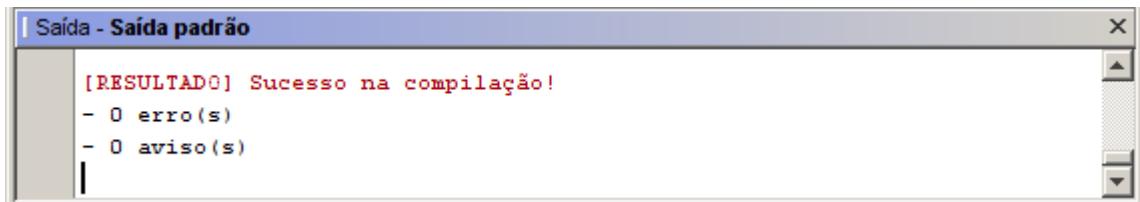
4) Clicando-se com o botão direito do mouse sobre o nome do recurso:



(Figura 3: Botão direito do mouse sobre o recurso)

Os resultados da compilação, indicando os erros e warnings, podem ser visualizadas

através da janela "Saída - Saída Padrão":

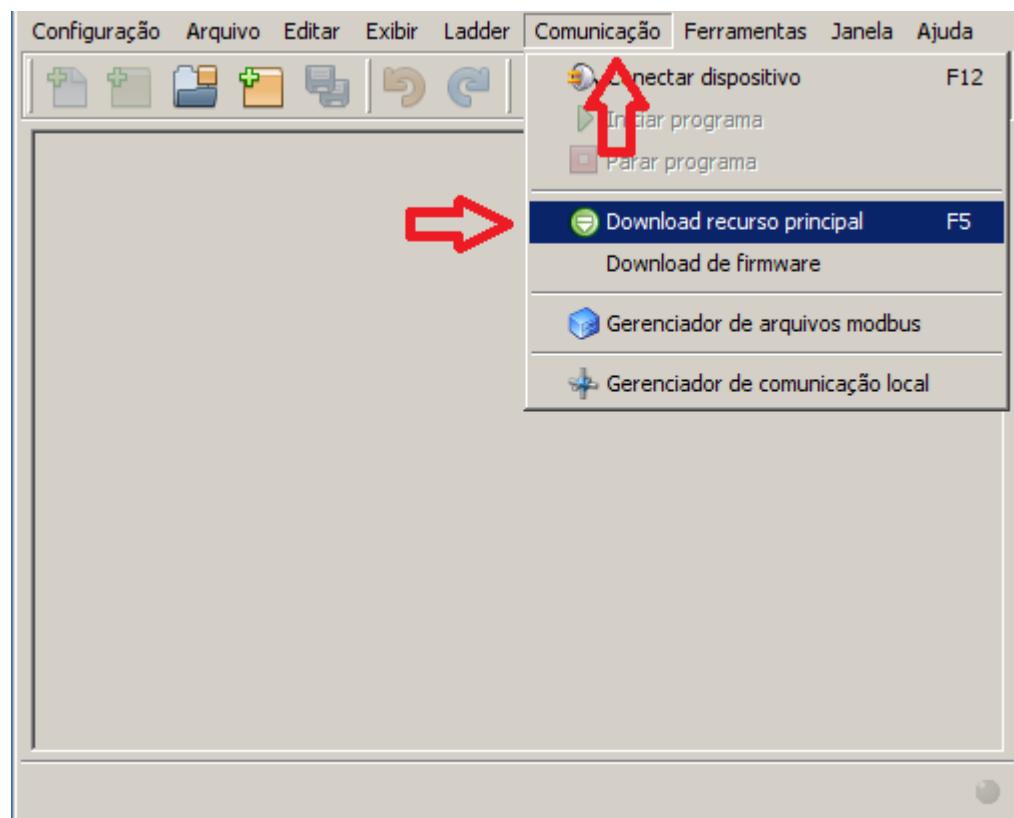


(Figura 3: Resultados da compilação)

5.2.6 Transferir

Para fazer o Download de um Recurso, há 4 opções:

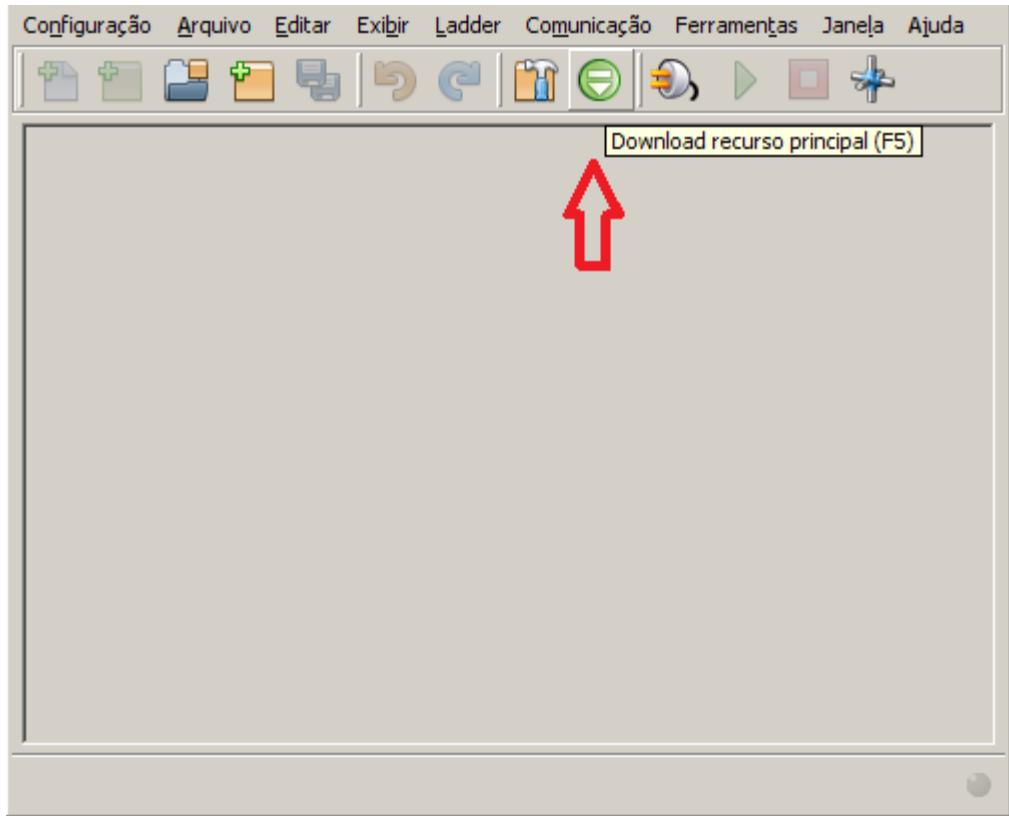
- 1) Através do Menu-Comunicação-Download recurso principal (F5):



(Figura 1: Download a partir do Menu)

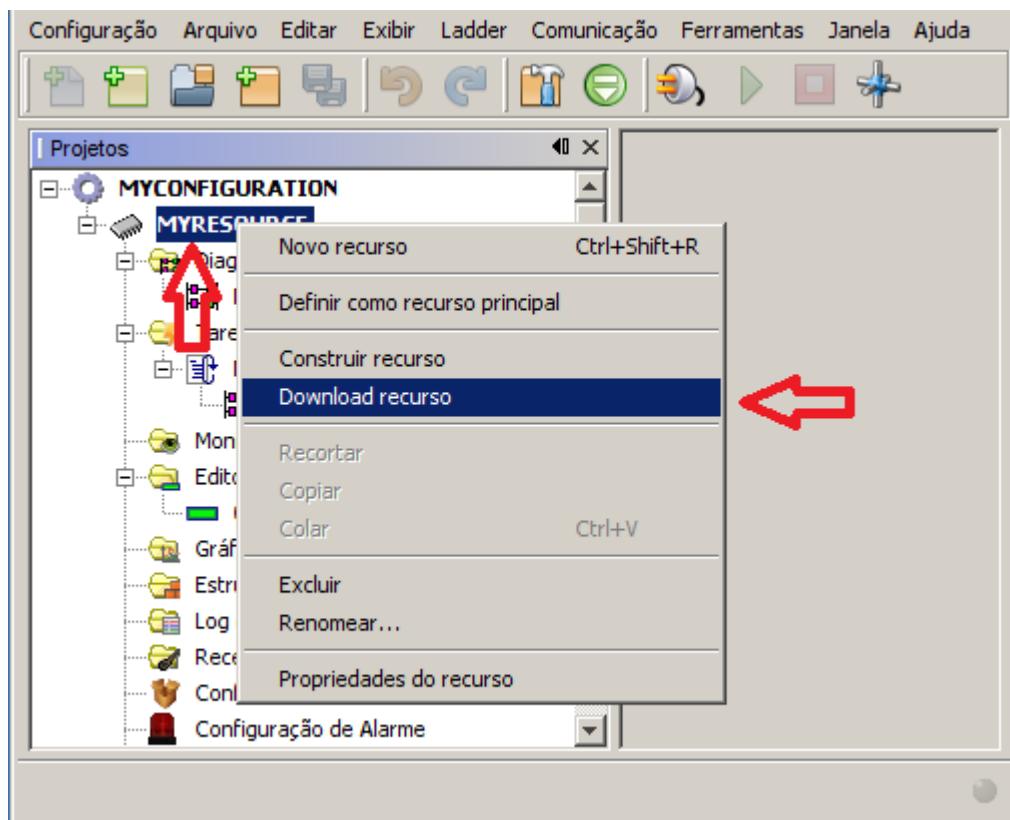
- 2) Através da tecla de atalho F5:

- 3) Através do botão na Barra de Ferramentas:



(Figura 1: Download a partir da Barra de Ferramentas)

4) Clicando-se com o botão direito do mouse sobre o nome do recurso:



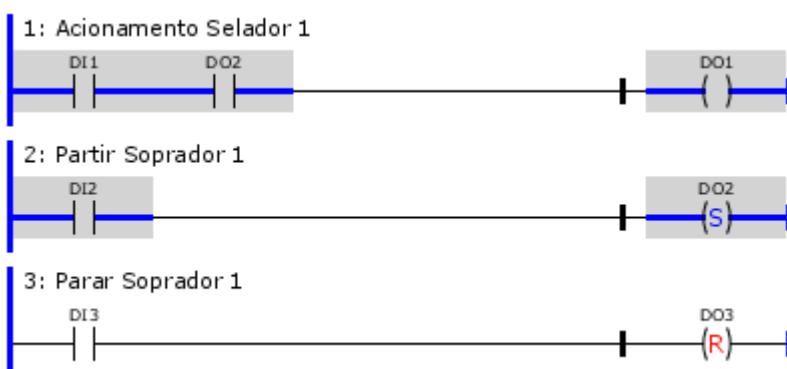
5.2.7 Monitoração Online

Visão Geral

Após o programa ladder ser compilado e carregado na equipamento é possível monitorar o ladder pressionando o botão de monitoração online .

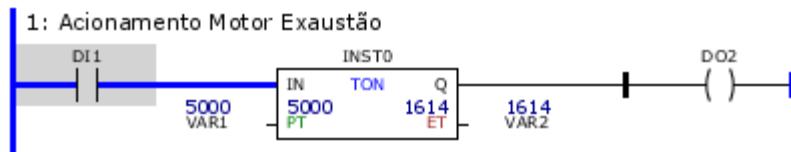
Neste momento o WPS V1.30 tentará estabelecer a comunicação com o equipamento testando a comunicação com o mesmo.

A monitoração online representará graficamente o estado lógico do programa ladder. Um exemplo de monitoração online pode ser vista na figura a seguir.



(Figura 1: Exemplo de monitoração online)

Para blocos funcionais os valores serão apresentados conforme figura a seguir:



(Figura 2: Valores de monitoração dos blocos funcionais)

Os valores das variáveis são mostrados em cima de suas respectivas variáveis, os valores internos da instância são mostrados em cima do nome do argumento respectivo.

Escrita de variáveis

Para escrita de variáveis basta dar um clique duplo sobre a variável que deseja-se escrever e então uma caixa de escrita de valores abrirá conforme figura a seguir.



(Figura 3: Escrita de variáveis)

Nesta caixa deve-se digitar o valor desejado e pressionar o botão "Write" para escrever o valor. O botão "Toogle" serve para alternar o valor escrito de 0 para o valor atual e vice-versa.

Monitoração de instâncias

Para monitorar basta dar um clique duplo sobre a variável da instância e então uma caixa relativa a monitoração de instância abrirá conforme figura a seguir.

Element	Value
PT	5000
ET	5000
IN	TRUE
Q	TRUE

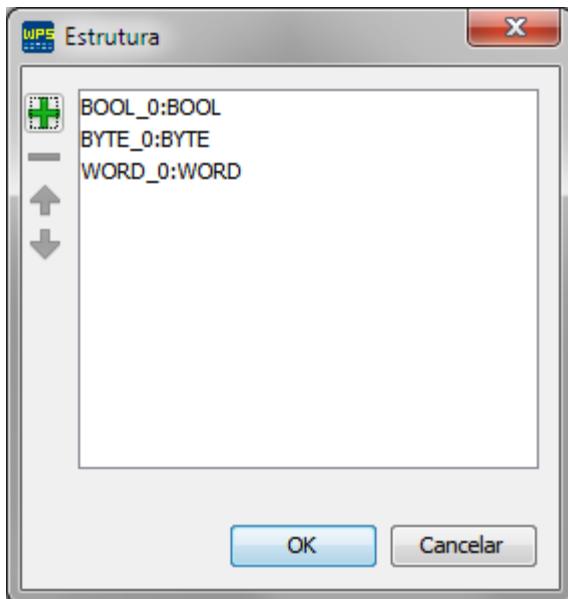
(Figura 4: Monitoração de instâncias)

Para escrever em variáveis internas da instância basta clicar na linha

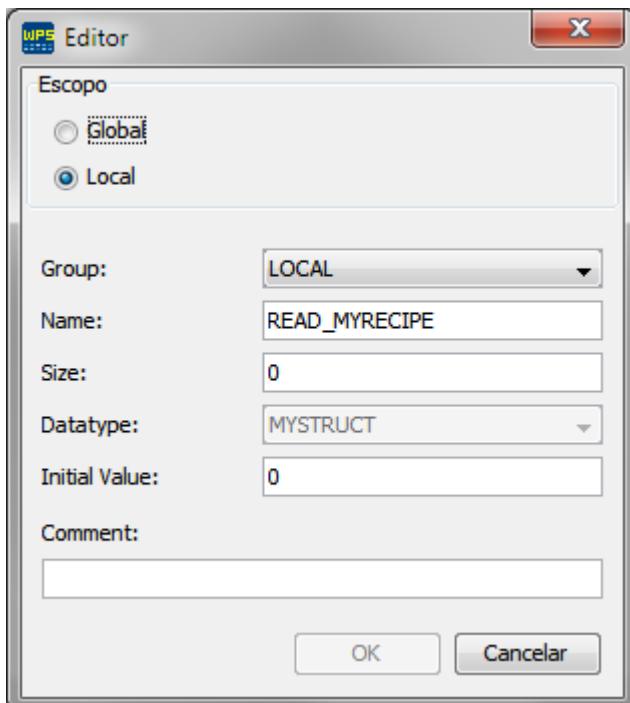
correspondente utilizar a caixa de edição de valores e os botões "Write" e "Toogle" conforme já mencionado no item anterior.

Monitoração de estruturas

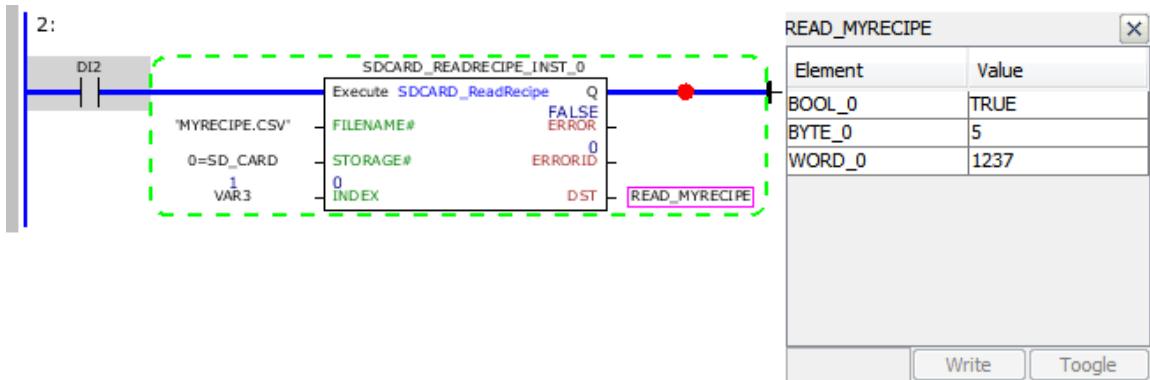
Para as variáveis criadas a partir de estruturas definidas no recurso basta clicar sobre a variável correspondente que um caixa similar a monitoração de instâncias abrirá. Veja o exemplo a seguir.



(Figura 5: Estrutura definida no recurso)



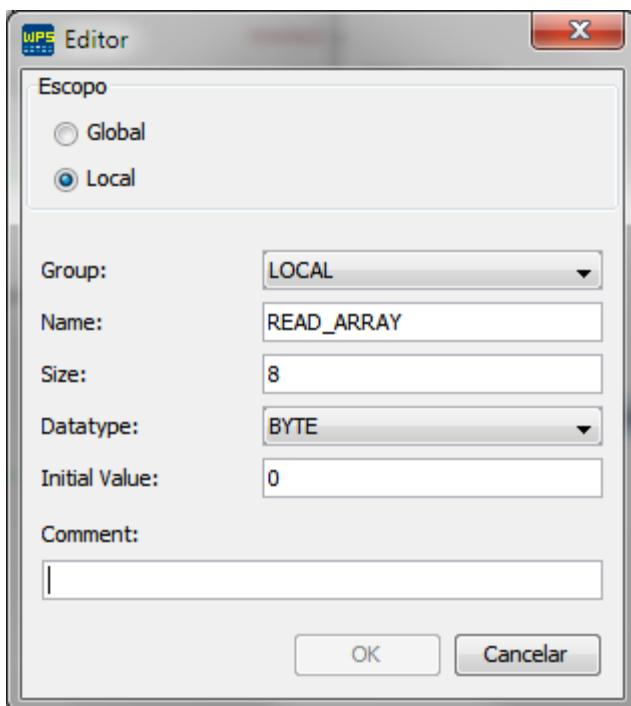
(Figura 6: Variável criada com tipo de dado da estrutura definida no recurso)



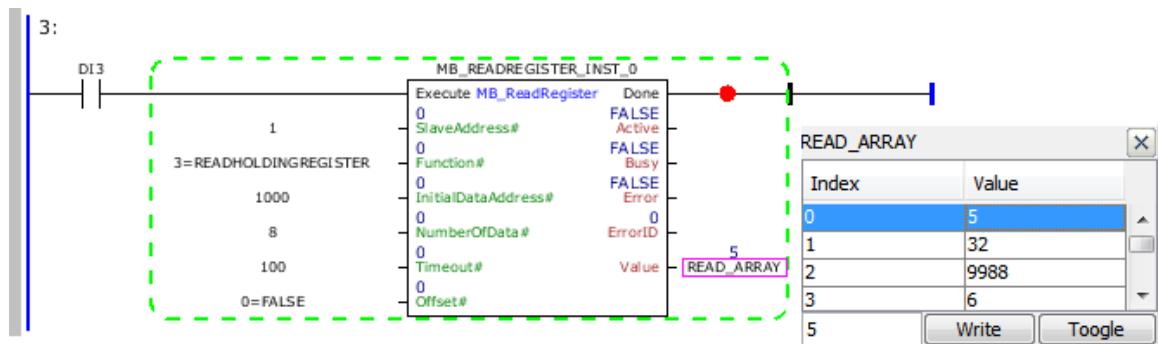
(Figura 7: Monitoração da variável utilizado no bloco SDCARD_ReadRecipe)

Monitoração de Arrays

Para as variáveis criadas com size maior que zero é possível monitorar todos os dados do seu array, para isso, basta clicar sobre a variável correspondente que um caixa de monitoração abrirá. Veja exemplo a seguir.



(Figura 8: Variável criada com size maior que zero, array)



(Figura 9: Monitoração da variável utilizado no bloco MB_ReadRegister)

6 PLC300

6.1 Descrição

Descrição dos Modelos

O PLC300 está disponível em seis modelos diferentes. Com ou sem HMI; standard ou plus e uma versão com HMI, porém sem membrana, em que o usuário poderá personalizar a aparência do PLC, seguindo especificações presentes no CD do produto.

A identificação do PLC300 é feita por um sufixo de duas ou três letras.

As letras têm o seguinte significado:

- H: HMI
- B: Blind (sem HMI)
- P: Plus
- S: Standard
- C: Custom (sem membrana)

Especificações	PLC300HP	PLC300BP	PLC300HS	PLC300BS	PLC300HPC	PLC300HSC
Cód. WEG	11279219	11279283	11477962	11477953	11490903	11491069
IHM	X	--	X	--	X	X
Membrana	X	--	X	--	-	--
Ethernet	X	X	--	--	X	--
Encoder	X	X	--	--	X	--
Expansão	X	X	X	--	X	--
SD Card	X	X	X	X	X	X
CAN	X	X	X	X	X	X
RS485	X	X	X	X	X	X
RS232	X	X	X	X	X	X
USB	X	X	X	X	X	X
Saída PWM	X	X	X	X	X	X
I/O digital	X	X	X	X	X	X
I/O analógico	X	X	X	X	X	X

Para maiores detalhes sobre o produto consulte o manual do usuário do equipamento PLC300.

6.2 Novidades e Correções do PLC300

PLC300 V1.30

Novas Funcionalidades:

1. Marcadores de Sistema de Status (%S_):

- BOOTLOADER
- INTERVAL_TASK9_WATCHDOG ... INTERVAL_TASK16_WATCHDOG
- SINGLE_TASK9_WATCHDOG ... SINGLE_TASK16_WATCHDOG
- COUNT_TASK9_WATCHDOG ... COUNT_TASK16_WATCHDOG
- STS_SD_INVALID

2. Marcadores de Sistema de Comando (%C_):

- INTERVAL_TASK9_DISABLE ... INTERVAL_TASK16_DISABLE
- SINGLE_TASK9_DISABLE ... SINGLE_TASK16_DISABLE
- COUNT_TASK9_DISABLE ... COUNT_TASK9_DISABLE

3. Blocos Modbus TCP

- MBTCP_ReadBinary
- MBTCP_WriteBinary
- MBTCP_ReadRegister
- MBTCP_WriteRegister
- MBTCP_ServerStatus
- MBTCP_ClientControlStatus

4. Novos Blocos Data Transfer (Receitas)

- ReadRecipe
- WriteRecipe

5. Log de

- Alarmes
- Eventos

6. Backup no SD Card do

- Firmware
- Recurso
- Setup

7. Ethernet

- Conexão do gateway do WPS com o PLC300
-

PLC300 V1.20

Novas Funcionalidades:

1. Tarefas

- INTERVAL
- SINGLE
- EXTERN EVENT (DI9, DI10 e pulso Z)
- COUNT (DI9, DI10, pulsos A, B, Z e quadratura AB)
- SYSTEM (start e stop)

2. Marcadores de Sistema de Status (%S_)

- TICK_100US

- INTERVAL_TASK1_WATCHDOG...INTERVAL_TASK8_WATCHDOG
- SINGLE_TASK1_WATCHDOG...SINGLE_TASK8_WATCHDOG
- EXT_EVENT_TASK1_WATCHDOG...EXT_EVENT_TASK3_WATCHDOG
- COUNT_TASK1_WATCHDOG...COUNT_TASK8_WATCHDOG
- MAIN_TASK_WATCHDOG
- START_TASK_WATCHDOG
- STOP_TASK_WATCHDOG

3. Marcadores de Sistema de Comando (%C_)

- INTERVAL_TASK1_DISABLE...INTERVAL_TASK8_DISABLE
- SINGLE_TASK1_DISABLE...SINGLE_TASK8_DISABLE
- EXT_EVENT_TASK1_WATCHDOG...EXT_EVENT_TASK3_WATCHDOG
- COUNT_TASK1_DISABLE...COUNT_TASK8_DISABLE

4. Blocos Hardware

- IMMEDIATEINPUT
- IMMEDIATEOUTPUT
- READENC3

5. Bloco Coil

- IMMEDIATECOIL

Modificações de Funções Existentes

1. Opção de inicializar ou não variáveis retentivas e histórico de alarmes no download.
2. Bloco PWM - permitido valor de frequência de 0 Hz
3. Blocos ReadEnc e ReadEnc2 - Tipo de dado de Value pode ser DINT quando os pulso contados forem Quadratura_AB, permitindo valores negativos conforme o sentido de giro do encoder.

Correções de Desvio Funcional

1. Bloco MB_WriteBinary - em versões anteriores, o bloco sempre escrevia o valor 1.
2. Alterados os endereços das variáveis para acesso via Modbus.

PLC300 Versões Anteriores a V1.20

Recomendamos fazer atualização do firmware.

PLC300 V1.10

Nova Funcionalidade:

1. Protocolo ASCII RS232

PLC300 V1.00

Versão inicial.

6.3 I/O's

Nas variáveis abaixo, contidas no Grupo I/O da Tabela de Variáveis, o Tag pode ser alterado.

Informações de Hardware, podem ser encontradas no Manual da PLC300 através do site www.weg.net.

Entradas Digitais

Endereço	Bit	Modbus	Tag	Descrição
%IB0	0	16000	DI1	Entrada digital 1
%IB0	1	16001	DI2	Entrada digital 2
%IB0	2	16002	DI3	Entrada digital 3
%IB0	3	16003	DI4	Entrada digital 4
%IB0	4	16004	DI5	Entrada digital 5
%IB0	5	16005	DI6	Entrada digital 6
%IB0	6	16006	DI7	Entrada digital 7
%IB0	7	16007	DI8	Entrada digital 8
%IB1	0	16008	DI9	Entrada digital 9
%IB1	1	16009	DI10	Entrada digital 10
%IB2	0	16016	DI101	Entrada digital 1 - Slot 1
%IB2	1	16017	DI102	Entrada digital 2 - Slot 1
%IB2	2	16018	DI103	Entrada digital 3 - Slot 1
%IB2	3	16019	DI104	Entrada digital 4 - Slot 1
%IB2	4	16020	DI105	Entrada digital 5 - Slot 1
%IB2	5	16021	DI106	Entrada digital 6 - Slot 1
%IB2	6	16022	DI107	Entrada digital 7 - Slot 1
%IB2	7	16023	DI108	Entrada digital 8 - Slot 1
%IB3	0	16024	DI109	Entrada digital 9 - Slot 1
%IB3	1	16025	DI110	Entrada digital 10 - Slot 1

%IB3	2	16026	DI111	Entrada digital 11 - Slot 1
%IB3	3	16027	DI112	Entrada digital 12 - Slot 1
%IB3	4	16028	DI113	Entrada digital 13 - Slot 1
%IB3	5	16029	DI114	Entrada digital 14 - Slot 1
%IB3	6	16030	DI115	Entrada digital 15 - Slot 1
%IB3	7	16031	DI116	Entrada digital 16 - Slot 1
%IB4	0	16032	DI201	Entrada digital 1 - Slot 2
%IB4	1	16033	DI202	Entrada digital 2 - Slot 2
%IB4	2	16034	DI203	Entrada digital 3 - Slot 2
%IB4	3	16035	DI204	Entrada digital 4 - Slot 2
%IB4	4	16036	DI205	Entrada digital 5 - Slot 2
%IB4	5	16037	DI206	Entrada digital 6 - Slot 2
%IB4	6	16038	DI207	Entrada digital 7 - Slot 2
%IB4	7	16039	DI208	Entrada digital 8 - Slot 2
%IB5	0	16040	DI209	Entrada digital 9 - Slot 2
%IB5	1	16041	DI210	Entrada digital 10 - Slot 2
%IB5	2	16042	DI211	Entrada digital 11 - Slot 2
%IB5	3	16043	DI212	Entrada digital 12 - Slot 2
%IB5	4	16044	DI213	Entrada digital 13 - Slot 2
%IB5	5	16045	DI214	Entrada digital 14 - Slot 2
%IB5	6	16046	DI215	Entrada digital 15 - Slot 2
%IB5	7	16047	DI216	Entrada digital 16 - Slot 2

Entradas Analógicas

Endereço	Bit	Modbus	Tag	Descrição
%IW6	--	5003	AI1	Entrada analógica 1
%IW8	--	5004	AI101	Entrada analógica 1 - Slot 1
%IW10	--	5005	AI102	Entrada analógica 2 - Slot 1
%IW12	--	5006	AI103	Entrada analógica 3 - Slot 1
%IW14	--	5007	AI104	Entrada analógica 4 - Slot 1
%IW16	--	5008	AI105	Entrada analógica 5 - Slot 1
%IW18	--	5009	AI201	Entrada analógica 1 - Slot 2
%IW20	--	5010	AI202	Entrada analógica 2 - Slot 2
%IW22	--	5011	AI203	Entrada analógica 3 - Slot 2

%IW24	--	5012	AI204	Entrada analógica 4 - Slot 2
%IW26	--	5013	AI205	Entrada analógica 5 - Slot 2

Saídas Digitais

Endereço	Bit	Modbus	Tag	Descrição
%QB0	0	16000	DO1	Saída digital 1
%QB0	1	16001	DO2	Saída digital 2
%QB0	2	16002	DO3	Saída digital 3
%QB0	3	16003	DO4	Saída digital 4
%QB0	4	16004	DO5	Saída digital 5
%QB0	5	16005	DO6	Saída digital 6
%QB0	6	16006	DO7	Saída digital 7
%QB0	7	16007	DO8	Saída digital 8
%QB1	0	16008	DO9	Saída digital 9
%QB2	0	16016	DO101	Saída digital 1 - Slot 1
%QB2	1	16017	DO102	Saída digital 2 - Slot 1
%QB2	2	16018	DO103	Saída digital 3 - Slot 1
%QB2	3	16019	DO104	Saída digital 4 - Slot 1
%QB2	4	16020	DO105	Saída digital 5 - Slot 1
%QB2	5	16021	DO106	Saída digital 6 - Slot 1
%QB2	6	16022	DO107	Saída digital 7 - Slot 1
%QB2	7	16023	DO108	Saída digital 8 - Slot 1
%QB3	0	16024	DO109	Saída digital 9 - Slot 1
%QB3	1	16025	DO110	Saída digital 10 - Slot 1
%QB3	2	16026	DO111	Saída digital 11 - Slot 1
%QB3	3	16027	DO112	Saída digital 12 - Slot 1
%QB3	4	16028	DO113	Saída digital 13 - Slot 1
%QB3	5	16029	DO114	Saída digital 14 - Slot 1
%QB3	6	16030	DO115	Saída digital 15 - Slot 1
%QB3	7	16031	DO116	Saída digital 16 - Slot 1
%QB4	0	16032	DO201	Saída digital 1 - Slot 2
%QB4	1	16033	DO202	Saída digital 2 - Slot 2
%QB4	2	16034	DO203	Saída digital 3 - Slot 2
%QB4	3	16035	DO204	Saída digital 4 - Slot 2

%QB4	4	16036	DO205	Saída digital 5 - Slot 2
%QB4	5	16037	DO206	Saída digital 6 - Slot 2
%QB4	6	16038	DO207	Saída digital 7 - Slot 2
%QB4	7	16039	DO208	Saída digital 8 - Slot 2
%QB5	0	16040	DO209	Saída digital 9 - Slot 2
%QB5	1	16041	DO210	Saída digital 10 - Slot 2
%QB5	2	16042	DO211	Saída digital 11 - Slot 2
%QB5	3	16043	DO212	Saída digital 12 - Slot 2
%QB5	4	16044	DOI13	Saída digital 13 - Slot 2
%QB5	5	16045	DO214	Saída digital 14 - Slot 2
%QB5	6	16046	DO215	Saída digital 15 - Slot 2
%QB5	7	16047	DO216	Saída digital 16 - Slot 2

Saídas Analógicas

Endereço	Bit	Modbus	Tag	Descrição
%QW6	--	5003	AO1	Saída analógica 1
%QW8	--	5004	AO101	Saída analógica 1 - Slot 1
%QW10	--	5005	AO102	Saída analógica 2 - Slot 1
%QW12	--	5006	AO201	Saída analógica 1 - Slot 2
%QW14	--	5007	AO202	Saída analógica 2 - Slot 2

6.4 Marcadores de Sistema

As variáveis abaixo, contidas no Grupo Sistema da Tabela de Variáveis, possuem o Tag fixo.

Alguns marcadores do tipo BYTE (%SB ou %CB), estão alocados no mesmo endereço modbus para registradores. Neste caso:

- (L): indica que é o byte menos significativo da WORD;
- (H): indica que é o byte mais significativo da WORD.

Os Marcadores que tiverem em sua Descrição:

- ($\geq V1.20$): são marcadores que foram inseridos a partir da Versão 1.20 do PLC300;
- ($\geq V1.30$): são marcadores que foram inseridos a partir da Versão 1.30 do PLC300.

Marcadores de Sistema de Leitura (Status)

Ender eço	Bit	Mo dbus	Tag	Descrição
%SW3 000	--	300 0	FIRMWARE	Versão de Firmware do PLC300
%SW3 002	--	300 1	SCAN_CICLE	Tempo do ciclo de scan em ms com uma casa decimal (ex.: 123 = 12.3ms)
%SB3 004	0	32	FALSE	O seu valor sempre permanecerá 0 (zero)
%SB3 004	1	33	TRUE	O seu valor sempre permanecerá 1 (um)
%SB3 004	2	34	FREQ_2HZ	Oscila a 2 Hz
%SB3 004	3	35	PULSE_1SCAN	Setado por um ciclo de scan na transição de STOP para RUN do programa do usuário
%SB3 004	4	36	ENC_DIR	Sentido de giro do encoder: 0 - Horário e 1 - Anti-horário
%SB3 006	0	48	STS_BAT	Alarme ativo - Bateria Fraca - Trocar bateria
%SB3 006	1	49	STS_DOS	Alarme ativo - Falha DOS - Indica sobre temperatura nas saídas digitais
%SB3 006	2	50	STS_ENC	Alarme Ativo - Falha Encoder - Indica que algum sinal do encoder está faltando
%SB3 006	3	51	STS_AI1	Alarme Ativo - Fio Partido AI1 - Corrente abaixo de 2mA na entrada analógica AI1, no modo 4 a 20mA
%SB3 006	4	52	STS_SD_INVALID	SD card inexistente ou inválido.
%SD3 008	--	300 4	ENC_FREQ	Frequência do Encoder - Frequência em Hz do Encoder
%SD3 012	--	300 6	TICK_100US	Marcador incrementado a cada 100 microsegundos (\geq V1.20)
%SW3 016	--	300 8	BOOTLOADER	Versão do módulo de atualização de firmware (\geq V1.30)
%SB3 020	--	301 0	SCREEN	Tela - Mostra o número da tela atual na IHM
%SW3 030	--	301 5	HOUR	Hora do RTC
%SW3 032	--	301 6	MINUTE	Minuto do RTC
%SW3 034	--	301 7	SECOND	Segundo do RTC

%SW3 036	--	301 8	DAY	Dia do RTC
%SW3 038	--	301 9	MONTH	Mês do RTC
%SW3 040	--	302 0	YEAR	Ano do RTC
%SW3 042	--	302 1	WEEKDAY	Dia da semana - 0 = domingo, 1 = segunda ... 6 = sábado
%SB3 050	0	400	INTERVAL_TASK1_ WATCHDOG	Watchdog tarefa 1 por tempo (\geq V1.20)
%SB3 050	1	401	INTERVAL_TASK2_ WATCHDOG	Watchdog tarefa 2 por tempo (\geq V1.20)
%SB3 050	2	402	INTERVAL_TASK3_ WATCHDOG	Watchdog tarefa 3 por tempo (\geq V1.20)
%SB3 050	3	403	INTERVAL_TASK4_ WATCHDOG	Watchdog tarefa 4 por tempo (\geq V1.20)
%SB3 050	4	404	INTERVAL_TASK5_ WATCHDOG	Watchdog tarefa 5 por tempo (\geq V1.20)
%SB3 050	5	405	INTERVAL_TASK6_ WATCHDOG	Watchdog tarefa 6 por tempo (\geq V1.20)
%SB3 050	6	406	INTERVAL_TASK7_ WATCHDOG	Watchdog tarefa 7 por tempo (\geq V1.20)
%SB3 050	7	407	INTERVAL_TASK8_ WATCHDOG	Watchdog tarefa 8 por tempo (\geq V1.20)
%SB3 051	0	408	INTERVAL_TASK9_ WATCHDOG	Watchdog tarefa 9 por tempo (\geq V1.30)
%SB3 051	1	409	INTERVAL_TASK10_ WATCHDOG	Watchdog tarefa 10 por tempo (\geq V1.30)
%SB3 051	2	410	INTERVAL_TASK11_ WATCHDOG	Watchdog tarefa 11 por tempo (\geq V1.30)
%SB3 051	3	411	INTERVAL_TASK12_ WATCHDOG	Watchdog tarefa 12 por tempo (\geq V1.30)
%SB3 051	4	412	INTERVAL_TASK13_ WATCHDOG	Watchdog tarefa 13 por tempo (\geq V1.30)
%SB3 051	5	413	INTERVAL_TASK14_ WATCHDOG	Watchdog tarefa 14 por tempo (\geq V1.30)
%SB3 051	6	414	INTERVAL_TASK15_ WATCHDOG	Watchdog tarefa 15 por tempo (\geq V1.30)
%SB3 051	7	415	INTERVAL_TASK16_ WATCHDOG	Watchdog tarefa 16 por tempo (\geq V1.30)
%SB3 052	0	416	SINGLE_TASK1_WAT CHDOG	Watchdog tarefa 1 por evento (\geq V1.20)

%SB3 052	1	417	SINGLE_TASK2_WAT CHDOG	Watchdog tarefa 2 por evento (\geq V1.20)
%SB3 052	2	418	SINGLE_TASK3_WAT CHDOG	Watchdog tarefa 3 por evento (\geq V1.20)
%SB3 052	3	419	SINGLE_TASK4_WAT CHDOG	Watchdog tarefa 4 por evento (\geq V1.20)
%SB3 052	4	420	SINGLE_TASK5_WAT CHDOG	Watchdog tarefa 5 por evento (\geq V1.20)
%SB3 052	5	421	SINGLE_TASK6_WAT CHDOG	Watchdog tarefa 6 por evento (\geq V1.20)
%SB3 052	6	422	SINGLE_TASK7_WAT CHDOG	Watchdog tarefa 7 por evento (\geq V1.20)
%SB3 052	7	423	SINGLE_TASK8_WAT CHDOG	Watchdog tarefa 8 por evento (\geq V1.20)
%SB3 053	0	424	SINGLE_TASK9_WAT CHDOG	Watchdog tarefa 9 por evento (\geq V1.30)
%SB3 053	1	425	SINGLE_TASK10_WA TCHDOG	Watchdog tarefa10 por evento (\geq V1.30)
%SB3 053	2	426	SINGLE_TASK11_WA TCHDOG	Watchdog tarefa 11 por evento (\geq V1.30)
%SB3 053	3	427	SINGLE_TASK12_WA TCHDOG	Watchdog tarefa 12 por evento (\geq V1.30)
%SB3 053	4	428	SINGLE_TASK13_WA TCHDOG	Watchdog tarefa 13 por evento (\geq V1.30)
%SB3 053	5	429	SINGLE_TASK14_WA TCHDOG	Watchdog tarefa 14 por evento (\geq V1.30)
%SB3 053	6	430	SINGLE_TASK15_WA TCHDOG	Watchdog tarefa15 por evento (\geq V1.30)
%SB3 053	7	431	SINGLE_TASK16_WA TCHDOG	Watchdog tarefa 16 por evento (\geq V1.30)
%SB3 054	0	302 7	EXT_EVENT_TASK1_ WATCHDOG	Watchdog tarefa 1 por evento externo (\geq V1.20)
%SB3 054	1	302 7	EXT_EVENT_TASK2_ WATCHDOG	Watchdog tarefa 2 por evento externo (\geq V1.20)
%SB3 054	2	302 7	EXT_EVENT_TASK3_ WATCHDOG	Watchdog tarefa 3 por evento externo (\geq V1.20)
%SB3 056	0	302 8	COUNT_TASK1_WAT CHDOG	Watchdog tarefa 1 por contagem (\geq V1.20)
%SB3 056	1	302 8	COUNT_TASK2_WAT CHDOG	Watchdog tarefa 2 por contagem (\geq V1.20)
%SB3 056	2	302 8	COUNT_TASK3_WAT CHDOG	Watchdog tarefa 3 por contagem (\geq V1.20)

%SB3 056	3	302 8	COUNT_TASK4_WAT CHDOG	Watchdog tarefa 4 por contagem (\geq V1.20)
%SB3 056	4	302 8	COUNT_TASK5_WAT CHDOG	Watchdog tarefa 5 por contagem (\geq V1.20)
%SB3 056	5	302 8	COUNT_TASK6_WAT CHDOG	Watchdog tarefa 6 por contagem (\geq V1.20)
%SB3 056	6	302 8	COUNT_TASK7_WAT CHDOG	Watchdog tarefa 7 por contagem (\geq V1.20)
%SB3 056	7	302 8	COUNT_TASK8_WAT CHDOG	Watchdog tarefa 8 por contagem (\geq V1.20)
%SB3 057	0	302 8	COUNT_TASK9_WAT CHDOG	Watchdog tarefa 9 por contagem (\geq V1.30)
%SB3 057	1	302 8	COUNT_TASK10_WA TCHDOG	Watchdog tarefa 10 por contagem (\geq V1.30)
%SB3 057	2	302 8	COUNT_TASK11_WA TCHDOG	Watchdog tarefa 11 por contagem (\geq V1.30)
%SB3 057	3	302 8	COUNT_TASK12_WA TCHDOG	Watchdog tarefa 12 por contagem (\geq V1.30)
%SB3 057	4	302 8	COUNT_TASK13_WA TCHDOG	Watchdog tarefa 13 por contagem (\geq V1.30)
%SB3 057	5	302 8	COUNT_TASK14_WA TCHDOG	Watchdog tarefa 14 por contagem (\geq V1.30)
%SB3 057	6	302 8	COUNT_TASK15_WA TCHDOG	Watchdog tarefa 15 por contagem (\geq V1.30)
%SB3 057	7	302 8	COUNT_TASK16_WA TCHDOG	Watchdog tarefa 16 por contagem (\geq V1.30)
%SB3 058	0	302 9	MAIN_TASK_WATCH DOG	Watchdog tarefa MAIN (\geq V1.20)
%SB3 058	1	302 9	START_TASK_WATC HDOG	Watchdog tarefa START (\geq V1.20)
%SB3 058	2	302 9	STOP_TASK_WATCH DOG	Watchdog tarefa STOP (\geq V1.20)
%SB3 100	--	305 0	MBUS_INTERFACE_ DISABLED	Interface do mestre Modbus RTU desabilitada.
%SW3 102	--	305 1	MBUS_REQUEST_CO UNT	Contador de requisições feitas pelo mestre Modbus RTU
%SW3 104	--	305 2	MBUS_RESPONSE_C OUNT	Contador de respostas recebidas com sucesso pelo mestre Modbus RTU
%SW3 106	--	305 3	MBUS_NO_ANSWER_ COUNT	Contador de requisições sem resposta recebida pelo mestre Modbus RTU (timeout)
%SW3 108	--	305 4	MBUS_RESP_ERROR_ COUNT	Contador de respostas de erro recebidas pelo mestre Modbus RTU

%SB3 110	--	305 5 (L)	MBUS_LAST_ERROR _ADDR	Último erro detectado: endereço do escravo Modbus RTU
%SB3 111	--	305 5 (H)	MBUS_LAST_ERROR _TYPE	Último erro detectado: tipo de erro
%SB3 112	--	305 6	MBUS_LAST_ERROR _CODE	Último erro detectado: código do erro
%SB3 114	0	912	RS232_RX_FINISHED	Recepção ASCII concluída
%SW3 120	--	306 0	MBUS_SLAVE_REQU EST_COUNT	Contador de telegramas recebidos com sucesso pelo PLC300 como escravo Modbus RTU
%SW3 122	--	306 1	MBUS_SLAVE_RESP ONSE_COUNT	Contador de respostas transmitidas com sucesso pelo PLC300 como escravo Modbus RTU
%SB3 150	--	307 5	CAN_STATUS	Estado da interface CAN
%SB3 151	0	120 8	CAN_BUS_POWER	Alimentação da interface CAN
%SW3 152	--	307 6	CAN_RX_COUNTER	Contador de telegramas CAN recebidos
%SW3 154	--	307 7	CAN_TX_COUNTER	Contador de telegramas CAN transmitidos
%SW3 156	--	307 8	CAN_BUS_OFF_COU NTER	Contador de erros de bus off detectados
%SW3 158	--	307 9	CAN_OVERRUN_COU NTER	Contador de retelegramas CAN perdidos (overrun)
%SB3 180	--	309 0 (L)	CO_STATUS	Estado da comunicação CANopen
%SB3 181	--	309 0 (H)	CO_NODE_STATE	Estado do nó CANopen (pré-operacional, operacional, parado)
%SB3 200	0	160 0	CO_STS_MASTER_C ONTACTED	Mestre CANopen contactou todos os escravos
%SB3 200	1	160 1	CO_STS_MASTER_C ONFIG_OK	Mestre CANopen fez download das configurações dos escravos
%SB3 200	2	160 2	CO_STS_MASTER_ER ROR_CTRL_OK	Protocolo de controle de erros (node guarding/heartbeat) iniciado com os escravos
%SB3 200	3	160 3	CO_STS_MASTER_IN IT_FINISHED	Inicialização de todos os escravos realizada pelo mestre CANopen
%SB3 200	4	160 4	CO_STS_MASTER_IN IT_ERROR	Algum escravo apresentou erro de inicialização
%SB3	5	160	CO_STS_MASTER_ER	Detectado falha em algum escravo no protocolo de

200		5	ROR_CTRL	detecção de erros pelo mestre CANopen
%SB3_200	6	1606	CO_STS_MASTER_EMCY	Algum escravo reportou EMCY
%SB3_201	0	1608	CO_STS_MASTER_NMT_TOGGLE	Feedback do toggle bit do comando NMT
%SB3_206	5	1613	CO_STS_MASTER_BUS_OFF	Mestre CANopen está em bus off
%SB3_201	6	1614	CO_STS_MASTER_POWER_OFF	Mestre CANopen não possui alimentação na interface CAN
%SB3_201	7	1615	CO_STS_MASTER_COMM_DISABLED	Comunicação do mestre CANopen está desabilitada
%SB3_202	0	1616	CO_STS_SLAVE1_CONTACTED	Escravo no endereço indicado foi contactado com sucesso pelo mestre CANopen
%SB3_202	1	1617	CO_STS_SLAVE1_CONFIG_OK	Configuração do escravo foi feita com sucesso pelo mestre CANopen
%SB3_202	2	1618	CO_STS_SLAVE1_ERROR_ROR_CTRL_OK	Protocolo de controle de erros (node guarding/heartbeat) iniciado com o escravo
%SB3_202	3	1619	CO_STS_SLAVE1_INIT_FINISH	Inicialização do escravo concluída
%SB3_202	4	1620	CO_STS_SLAVE1_INIT_ERROR	Erro na inicialização do escravo no endereço indicado
%SB3_202	5	1621	CO_STS_SLAVE1_ERROR_ROR_CTRL_FAIL	Detectado falha em algum escravo no protocolo de detecção de erro pelo mestre CANopen
%SB3_202	6	1632	CO_STS_SLAVE1_EMCY	Escravo no endereço indicado reportou erro por EMCY
%SB3_204	0	1633	CO_STS_SLAVE2_CONTACTED	Escravo no endereço indicado foi contactado com sucesso pelo mestre CANopen
%SB3_204	1	1634	CO_STS_SLAVE2_CONFIG_OK	Configuração do escravo foi feita com sucesso pelo mestre CANopen
%SB3_204	2	1635	CO_STS_SLAVE2_ERROR_ROR_CTRL_OK	Protocolo de controle de erros (node guarding/heartbeat) iniciado com o escravo
%SB3_204	3	1636	CO_STS_SLAVE2_INIT_FINISH	Inicialização do escravo concluída
%SB3_204	4	1637	CO_STS_SLAVE2_INIT_ERROR	Erro na inicialização do escravo no endereço indicado
%SB3_204	5	1638	CO_STS_SLAVE2_ERROR_ROR_CTRL_FAIL	Detectado falha em algum escravo no protocolo de detecção de erro pelo mestre CANopen
%SB3_204	6	1639	CO_STS_SLAVE2_EMCY	Escravo no endereço indicado reportou erro por EMCY
...
%SB3_454	0	3632	CO_STS_SLAVE127_CONTACTED	Escravo no endereço indicado foi contactado com sucesso pelo mestre CANopen

%SB3 454	1	363 3	CO_STS_SLAVE127_ CONFIG_OK	Configuração do escravo foi feita com sucesso pelo mestre CANopen
%SB3 454	2	363 4	CO_STS_SLAVE127_E RROR_CTRL_OK	Protocolo de controle de erros (node guarding/heartbeat) iniciado com o escravo
%SB3 454	3	363 5	CO_STS_SLAVE127_I NIT_FINISHED	Inicialização do escravo concluída
%SB3 454	4	363 6	CO_STS_SLAVE127_I NIT_ERROR	Erro na inicialização do escravo no endereço indicado
%SB3 454	5	363 7	CO_STS_SLAVE127_E RROR_CTRL_FAIL	Detectado falha em algum escravo no protocolo de detecção de erro pelo mestre CANopen
%SB3 454	6	363 8	CO_STS_SLAVE127_E MCY	Escravo no endereço indicado reportou erro por EMCY
%SW3 460	--	323 0	CO_SDO_ERROR_NO DE_ID	Erro de SDO: endereço do escravo do último erro de SDO detectado
%SW3 462	--	323 1	CO_SDO_ERROR_OBJ ECT_INDEX	Erro de SDO: índice do objeto
%SW3 464	--	323 2	CO_SDO_ERROR_OBJ ECT_SUBINDEX	Erro de SDO: sub-índice do objeto
%SW3 466	--	323 3	CO_SDO_ERROR_FU NCTION	Erro de SDO: função (leitura/escrita)
%SW3 468	--	323 4	CO_SDO_ERROR_VAL UE	Erro de SDO: valor
%SW3 472	--	323 6	CO_SDO_ERROR_CO DE	Erro de SDO: código do erro
%SW3 480	--	324 0	CO_EMCY_SLAVE_I D	Último EMCY reportado: endereço do escravo
%SW3 482	--	324 1	CO_EMCY_DATA	"Último EMCY reportado: dados do objeto
%SB3 492	--	324 6	ETH_MAC[6]	Endereço físico (\geq V1.30)
%SB3 498	--	324 9	ETH_STS_SPD_DUP	Modo de comunicação Ethernet (\geq V1.30)
% SD350 0	--	325 0	ETH_STS_IP	Endereço IP (\geq V1.30)
% SD350 4	--	325 2	ETH_STS_MASK	Máscara de sub-rede (\geq V1.30)
% SD350 8	--	325 4	ETH_STS_GW	Gateway padrão (\geq V1.30)
% SW35	--	325 6	MBTCP_SERVER_REQUEST_COUNT	Contador de telegramas recebidos com sucesso pelo PLC300 como servidor Modbus TCP (\geq V1.30)

12				
% SW35 14	--	325 7	MBTCP_SERVER_RESPONSE_COUNT	Contador de respostas transmitidas com sucesso pelo PLC300 como servidor Modbus TCP (\geq V1.30)
% SW35 16	--	325 8	MBTCP_SERVER_CNXNS	Número de conexões do servidor Modbus TCP ativas (\geq V1.30)
% SB352 0	--	326 0	MBTCP_CLIENT_DISABLED	Cliente Modbus TCP desabilitado (\geq V1.30)
% SW35 22	--	326 1	MBTCP_REQUEST_COUNT	Contador de requisições feitas pelo cliente Modbus TCP (\geq V1.30)
% SW35 24	--	326 2	MBTCP_RESPONSE_COUNT	Contador de respostas recebidas com sucesso pelo cliente Modbus TCP (\geq V1.30)
% SW35 26	--	326 3	MBTCP_NO_ANSWER_COUNT	Contador de requisições sem resposta recebida pelo cliente Modbus TCP (Timeout) (\geq V1.30)
% SW35 28	--	326 4	MBTCP_RESP_ERROR_COUNT	Contador de respostas de erro recebidas pelo cliente Modbus TCP \geq V1.30)
% SW35 30	--	326 5	MBTCP_LAST_ERROR_TCP_PORT	Último erro detectado: porta TCP do servidor Modbus (\geq V1.30)
% SD353 2	--	326 6	MBTCP_LAST_ERROR_IP	Último erro detectado: endereço IP do servidor modbus TCP (\geq V1.30)
% SB353 6	--	326 8 (L)	MBTCP_LAST_ERROR_UNITID	Último erro detectado: Unit ID do servidor Modbus TCP (\geq V1.30)
% SB353 7	--	326 8 (H)	MBTCP_LAST_ERROR_TYPE	Último erro detectado: tipo de erro (\geq V1.30)
% SB353 8	--	326 9	MBTCP_LAST_ERROR_CODE	Último erro detectado: código do erro (\geq V1.30)

Marcadores de Sistema de Leitura / Escrita (Comando)

Ender eço	Bit	Mo dbus	Tag	Descrição
%CB3 000	0	0	ERASE_RET	Zera Retentivos - Zera marcadores retentivos na transição positiva

%CB3 000	1	2	CLEAR_ENC_ALARM	Limpa alarme de Falha de Encoder
%CB3 000	2	3	BUZZER_ACTIVE	Buzzer ativo
%CW 3030	--	301 5	WR_HOUR	Hora do RTC - Ajuste da Hora do RTC
%CW 3032	--	301 6	WR_MINUTE	Minuto do RTC - Ajuste do Minuto do RTC
%CW 3034	--	301 7	WR_SECOND	Segundo do RTC - Ajuste do Segundo do RTC
%CW 3036	--	301 8	WR_DAY	Dia do RTC - Ajuste do Dia do RTC
%CW 3038	--	301 9	WR_MONTH	Mês do RTC - Ajuste do Mês do RTC
%CW 3040	--	302 0	WR_YEAR	Ano do RTC - Ajuste do Ano do RTC
%CB3 052	--	302 6	CAN_ADDRESS	Endereço do PLC300 para interface CAN
%CB3 055	--	302 7	CAN_BAUDRATE	Taxa de comunicação para interface CAN
%CB3 061	--	303 0	RS232_MODE	Modo de operação da interface RS232: 0=Modbus RTU 2=ASCII
%CB3 062	--	303 1 (L)	RS232_BYT_E_FORMAT	Paridade e stop bits para interface RS232
%CB3 063	--	303 1 (H)	RS232_BAUDRATE	Taxa de comunicação para interface RS232
%CB3 068	--	303 4 (L)	RS485_ADDRESS	Endereço do PLC300 para interface RS485
%CB3 069	--	303 4 (H)	RS485_MODE	Modo de operação (mestre ou escravo) para interface RS485
%CB3 070	--	303 5 (L)	RS485_BYT_E_FORMAT	Paridade e stop bits para interface RS485
%CB3 071	--	303 5 (H)	RS485_BAUDRATE	Taxa de comunicação para interface RS485
%CW 3100	--	305 0	MBUS_DISABLE_COMM	Desabilita interface do mestre Modbus RTU (>= V1.30)
%CB3 114	0	912	RS232_RX_CLEAR	Inicia nova recepção ASCII via RS232

%CB3 120	--	306 0	CO_NMT_COMMAND	Envio de comando NMT pelo mestre CANopen: código do comando
%CB3 121	0	968	CO_NMT_TOGGLE	Envio de comando NMT pelo mestre CANopen: toggle bit
%CB3 121	7	975	CO_DISABLE	Desabilita comunicação CANopen
%CB3 122	--	306 1	CO_NMT_SLAVE_ADDR	Envio de comando NMT pelo mestre CANopen: endereço do escravo
%CB3 124	--	306 2	RS232_ASCII_STRING	Buffer de recepção do protocolo ASCII
%CB3 124	--	306 2	RS232_ASCII_BYTEBUFF ER	Buffer de recepção do protocolo ASCII
%CB3 400	0	320 0	INTERVAL_TASK1_DISA BLE	Desabilita interrupção 1 de tempo (\geq V1.20)
% CB34 00	1	320 1	INTERVAL_TASK2_DISA BLE	Desabilita interrupção 2 de tempo (\geq V1.20)
% CB34 00	2	320 2	INTERVAL_TASK3_DISA BLE	Desabilita interrupção 3 de tempo (\geq V1.20)
% CB34 00	3	320 3	INTERVAL_TASK4_DISA BLE	Desabilita interrupção 4 de tempo (\geq V1.20)
% CB34 00	4	320 4	INTERVAL_TASK5_DISA BLE	Desabilita interrupção 5 de tempo (\geq V1.20)
% CB34 00	5	320 5	INTERVAL_TASK6_DISA BLE	Desabilita interrupção 6 de tempo (\geq V1.20)
% CB34 00	6	320 6	INTERVAL_TASK7_DISA BLE	Desabilita interrupção 7 de tempo (\geq V1.20)
% CB34 00	7	320 7	INTERVAL_TASK8_DISA BLE	Desabilita interrupção 8 de tempo (\geq V1.20)
%CB3 401	0	320 8	INTERVAL_TASK9_DISA BLE	Desabilita interrupção 9 de tempo (\geq V1.30)
% CB34 01	1	320 9	INTERVAL_TASK10_DIS ABLE	Desabilita interrupção 10 de tempo (\geq V1.30)
% CB34 01	2	321 0	INTERVAL_TASK11_DIS ABLE	Desabilita interrupção 11 de tempo (\geq V1.30)
%	3	321	INTERVAL_TASK12_DIS	Desabilita interrupção 12 de tempo (\geq V1.30)

CB34 01		1	ABLE	
% CB34 01	4	321 2	INTERVAL_TASK13_DIS ABLE	Desabilita interrupção 13 de tempo (>= V1.30)
% CB34 01	5	321 3	INTERVAL_TASK14_DIS ABLE	Desabilita interrupção 14 de tempo (>= V1.30)
% CB34 01	6	321 4	INTERVAL_TASK15_DIS ABLE	Desabilita interrupção 15 de tempo (>= V1.30)
% CB34 01	7	321 5	INTERVAL_TASK16_DIS ABLE	Desabilita interrupção 16 de tempo (>= V1.30)
% CB34 02	0	321 6	SINGLE_TASK1_DISABLE	Desabilita interrupção 1 de evento (>= V1.20)
% CB34 02	1	321 7	SINGLE_TASK2_DISABLE	Desabilita interrupção 2 de evento (>= V1.20)
% CB34 02	2	321 8	SINGLE_TASK3_DISABLE	Desabilita interrupção 3 de evento (>= V1.20)
% CB34 02	3	321 9	SINGLE_TASK4_DISABLE	Desabilita interrupção 4 de evento (>= V1.20)
% CB34 02	4	322 0	SINGLE_TASK5_DISABLE	Desabilita interrupção 5 de evento (>= V1.20)
% CB34 02	5	322 1	SINGLE_TASK6_DISABLE	Desabilita interrupção 6 de evento (>= V1.20)
% CB34 02	6	322 2	SINGLE_TASK7_DISABLE	Desabilita interrupção 7 de evento (>= V1.20)
% CB34 02	7	322 3	SINGLE_TASK8_DISABLE	Desabilita interrupção 8 de evento (>= V1.20)
% CB34 03	0	322 4	SINGLE_TASK9_DISABLE	Desabilita interrupção 9 de evento (>= V1.30)
% CB34 03	1	322 5	SINGLE_TASK10_DISABLE	Desabilita interrupção 10 de evento (>= V1.30)
% CB34	2	322 6	SINGLE_TASK11_DISABLE	Desabilita interrupção 11 de evento (>= V1.30)

03				
% CB34 03	3	322 7	SINGLE_TASK12_DISABLE	Desabilita interrupção 12 de evento ($\geq V1.30$)
% CB34 03	4	322 8	SINGLE_TASK13_DISABLE	Desabilita interrupção 13 de evento ($\geq V1.30$)
% CB34 03	5	322 9	SINGLE_TASK14_DISABLE	Desabilita interrupção 14 de evento ($\geq V1.30$)
% CB34 03	6	323 0	SINGLE_TASK15_DISABLE	Desabilita interrupção 15 de evento ($\geq V1.30$)
% CB34 03	7	323 1	SINGLE_TASK16_DISABLE	Desabilita interrupção 16 de evento ($\geq V1.30$)
% CB34 04	0	323 2	EXT_EVENT_TASK1_DISABLE	Desabilita interrupção 1 de evento externo ($\geq V1.20$)
% CB34 04	1	323 3	EXT_EVENT_TASK2_DISABLE	Desabilita interrupção 2 de evento externo ($\geq V1.20$)
% CB34 04	2	323 4	EXT_EVENT_TASK3_DISABLE	Desabilita interrupção 3 de evento externo ($\geq V1.20$)
% CB34 06	0	324 8	COUNT_TASK1_DISABLE	Desabilita interrupção 1 de contagem ($\geq V1.20$)
% CB34 06	1	324 9	COUNT_TASK2_DISABLE	Desabilita interrupção 2 de contagem ($\geq V1.20$)
% CB34 06	2	325 0	COUNT_TASK3_DISABLE	Desabilita interrupção 3 de contagem ($\geq V1.20$)
% CB34 06	3	325 1	COUNT_TASK4_DISABLE	Desabilita interrupção 4 de contagem ($\geq V1.20$)
% CB34 06	4	325 2	COUNT_TASK5_DISABLE	Desabilita interrupção 5 de contagem ($\geq V1.20$)
% CB34 06	5	325 3	COUNT_TASK6_DISABLE	Desabilita interrupção 6 de contagem ($\geq V1.20$)
% CB34 06	6	325 4	COUNT_TASK7_DISABLE	Desabilita interrupção 7 de contagem ($\geq V1.20$)

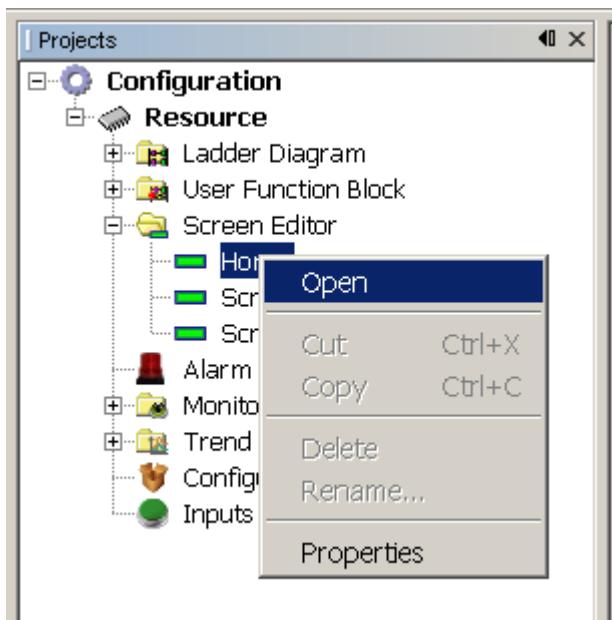
%CB34 06	7	325 5	COUNT_TASK8_DISABLE	Desabilita interrupção 8 de contagem (\geq V1.20)
%CB34 07	0	325 6	COUNT_TASK9_DISABLE	Desabilita interrupção 9 de contagem (\geq V1.30)
%CB34 07	1	325 7	COUNT_TASK10_DISABLE	Desabilita interrupção 10 de contagem (\geq V1.30)
%CB34 07	2	325 8	COUNT_TASK11_DISABLE	Desabilita interrupção 11 de contagem (\geq V1.30)
%CB34 07	3	325 9	COUNT_TASK12_DISABLE	Desabilita interrupção 12 de contagem (\geq V1.30)
%CB34 07	4	326 0	COUNT_TASK13_DISABLE	Desabilita interrupção 13 de contagem (\geq V1.30)
%CB34 07	5	326 1	COUNT_TASK14_DISABLE	Desabilita interrupção 14 de contagem (\geq V1.30)
%CB34 07	6	326 2	COUNT_TASK15_DISABLE	Desabilita interrupção 15 de contagem (\geq V1.30)
%CB34 07	7	326 3	COUNT_TASK16_DISABLE	Desabilita interrupção 16 de contagem (\geq V1.30)
%CD34 24	--	321 2	ETH_IP	Endereço IP (\geq V1.30)
%CD34 28	--	321 4	ETH_MASK	Máscara de sub-rede (\geq V1.30)
%CD34 32	--	321 6	ETH_GW	Gateway padrão (\geq V1.30)
%CB34 36	--	321 8 (L)	ETH_DHCP	Habilita DHCP (\geq V1.30)
%CB34 37	--	321 8 (H)	ETH_SPD_DUP	Modo de comunicação Ethernet (\geq V1.30)
%CD34 40	--	322 0	MBTCP_IP_AUTH	Autenticação de IP (\geq V1.30)

% CW34 44	--	322 2	MBTCP_PORT	Porta TCP do servidor Modbus TCP (\geq V1.30)
% CB34 46	--	322 3	MBTCP_UNITID	UnitID do servidor Modbus TCP (\geq V1.30)
% CW34 48	--	322 4	MBTCP_GW_TOUT	Timeout de recepção do escravo Modbus RTU (Gateway Modbus TCP) (\geq V1.30)
% CW34 52	--	322 6	MBTCP_DISABLE_COMM	Desabilita cliente Modbus TCP (\geq V1.30)
% CD34 56	--	322 8	SNTP_IP1	Endereço IP do servidor SNTP (\geq V1.30)
% CD34 60	--	323 0	SNTP_IP2	Endereço IP do servidor SNTP redundante (\geq V1.30)
% CW34 64	--	323 2	SNTP_UPD_FREQ	Frequência de atualização SNTP (\geq V1.30)
% CW34 66	--	323 3	SNTP_TIMEOUT	Timeout de recepção SNTP (\geq V1.30)

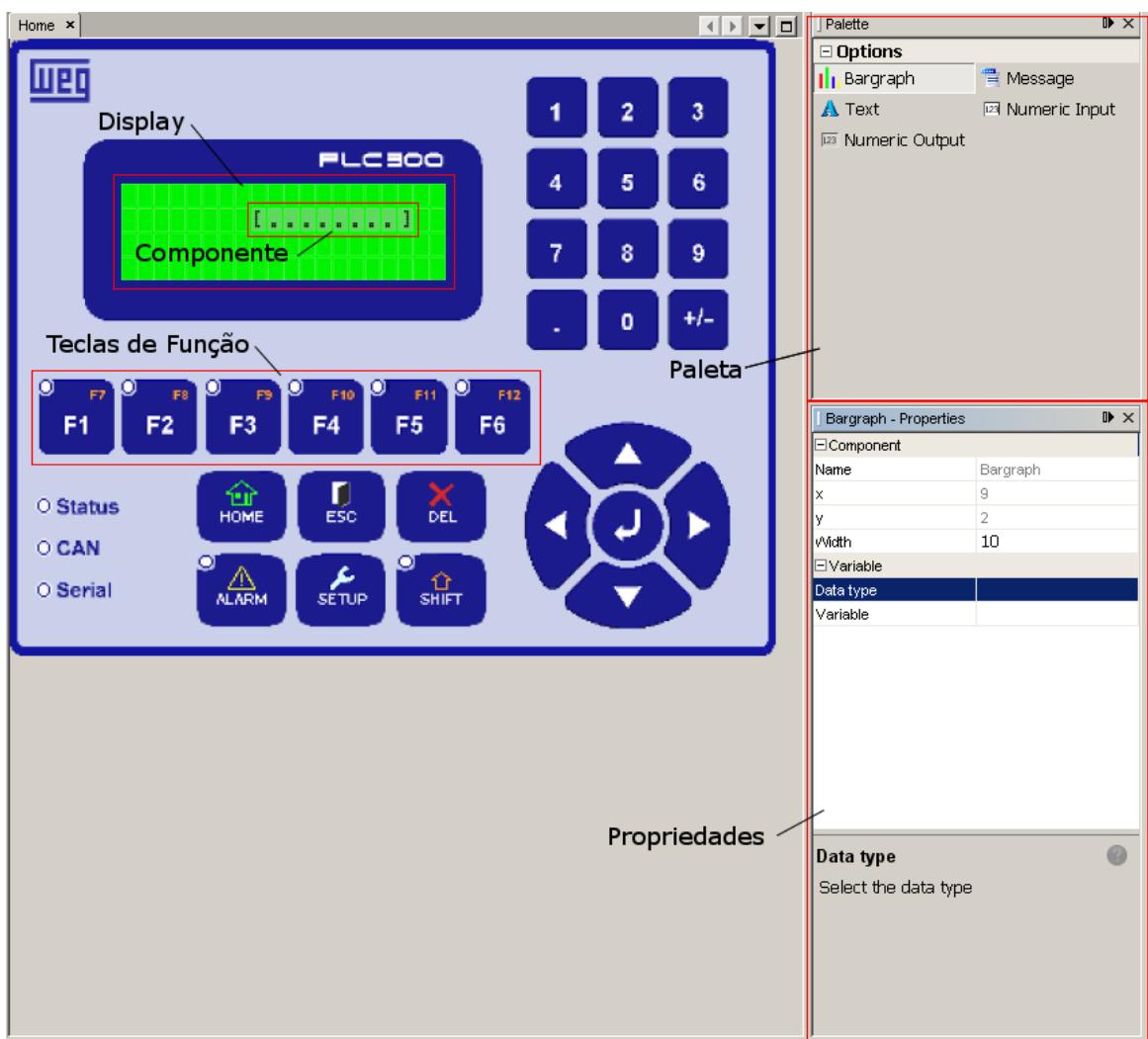
6.5 Editor de Telas

6.5.1 Configuração das Telas

O PLC 300 permite que até 240 telas possam ser programadas pelo usuário. Para iniciar o modo de edição das telas é preciso abrir uma tela na janela de projetos.



Após a abertura, a janela de edição das telas é visualizada. Esta janela é dividida nas seguintes áreas:



Componente

Componente configurável contendo determinada característica para composição da tela do equipamento.

Display

Local onde são inseridos e editados os componentes.

Teclas de Função

Teclas programáveis com funções de gerenciamento de telas e manipulação de bits.

Paleta

Janela contendo os componentes da tela do equipamento inseridos através de drag'n drop.

Janela de Propriedades

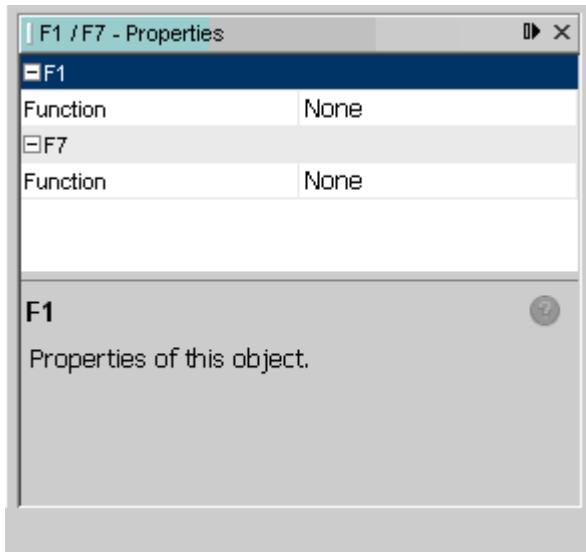
Janela sensível ao contexto onde são editadas e visualizadas as propriedades dos componentes e do display.

6.5.2 Teclas de Função

O gerenciamento de telas e a manipulação de bits são feitos através da configuração das teclas de função. Para tornar as propriedades de uma das teclas de função ativa é preciso selecioná-la clicando sobre ela na área das teclas de função.



As propriedades da tecla de função se tornarão ativas na tela de propriedades.



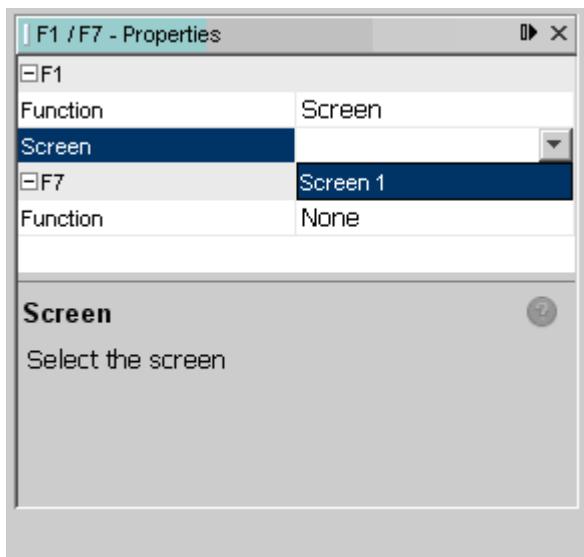
6.5.3 Gerenciamento de Telas

O editor de telas possui a função de gerenciamento de telas para o acesso navegação entre diferentes telas através das teclas de função.



Nota! Para esta configuração é necessário que uma tecla de função esteja selecionada e a janela de propriedades esteja aberta no ambiente de desenvolvimento.

A ligação entre as telas é feita através da edição da propriedade "Função" selecionando a opção "Screen". Após a seleção, uma lista com as telas disponíveis pode ser selecionada na opção "Tela".



Neste exemplo, a tela selecionada (Screen 1) será acessada através da tecla de função configurada (F1) pela tela atual.

6.5.4 Manipulação de Bits

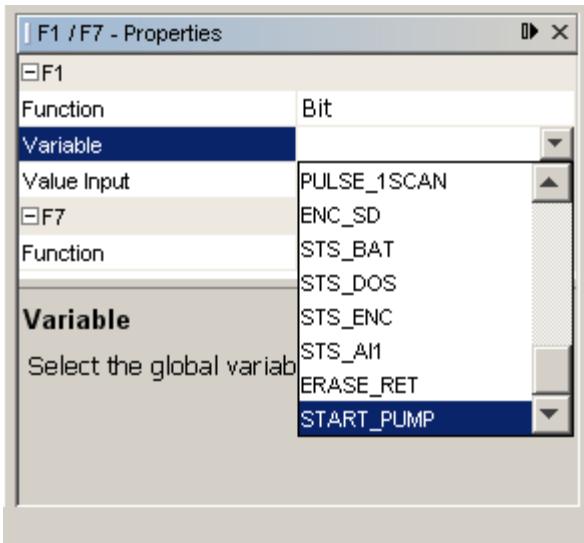
O editor de telas possui a função de manipulação de bits para alteração do estado de variáveis globais no ladder através das teclas de função.



Nota! Para esta configuração é necessário que uma tecla de função esteja selecionada e a janela de propriedades esteja aberta no ambiente de desenvolvimento.

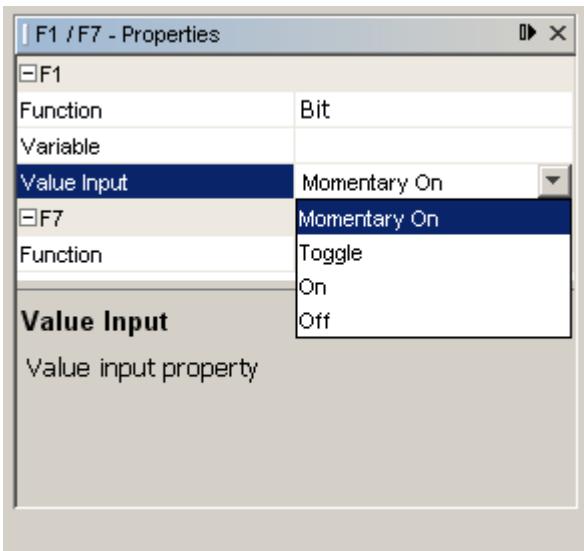
Na manipulação de bits é necessária a edição da propriedade "Função" selecionando a opção "Bit".

Após a seleção da função "Bit" uma variável global deve ser selecionada na propriedade "Variável".



Nota! A variável deve ser previamente configurada no editor ladder.

As ações disponíveis para a função BIT são:



Momentary On

Liga uma determinada variável do tipo booleana (marcador de bit) somente enquanto a tecla estiver pressionada. Ao soltar a tecla, o marcador volta a zero.

Toggle

Inverte o estado de uma determinada variável do tipo booleana (marcador de bit);

On

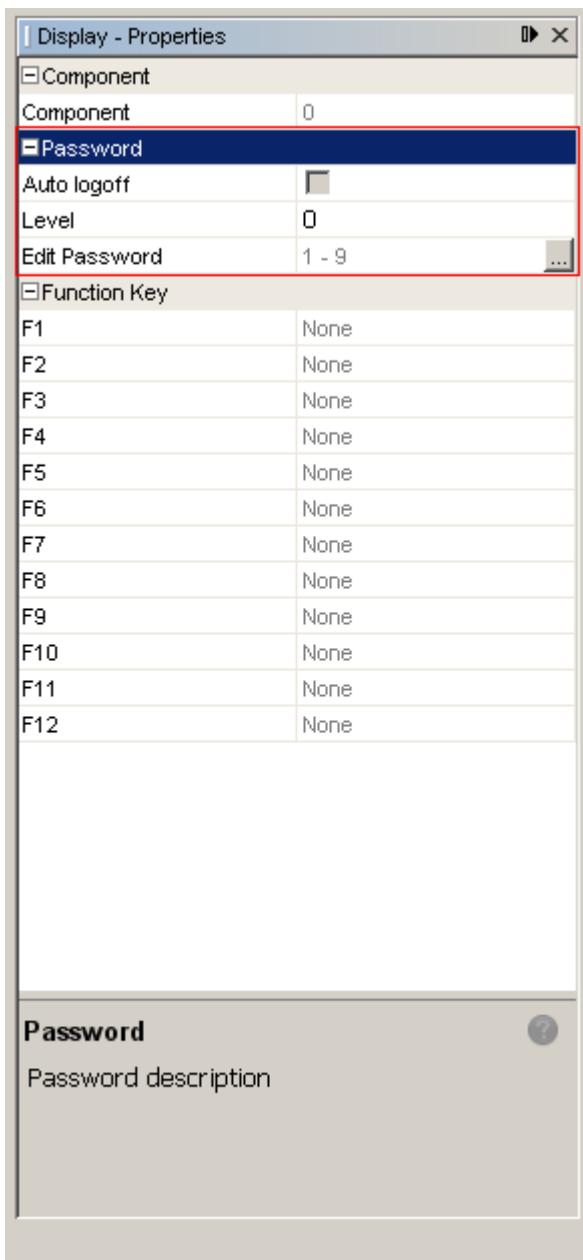
Liga uma determinada variável do tipo booleana (marcador de bit);

Off

Desliga uma determinada variável do tipo booleana (marcador de bit);

6.5.5 Senhas

O PLC300 possui dez níveis de senha (0 - 9). Destes dez níveis, o nível zero é escolhido quando se deseja o acesso à tela por qualquer usuário. Com exceção da tela zero (HOME), que possui nível de acesso zero sem possibilidade de alteração, as demais telas podem ser programadas com um nível de acesso. As senhas configuradas devem possuir seis dígitos decimais. As propriedades para configuração dos níveis de acesso são acessadas através da janela de propriedades do display.





Nota! Para esta configuração é necessário que a janela de propriedades esteja aberta no ambiente de desenvolvimento.

As propriedades de configuração das senhas são as seguintes:

Editar Senha

Configura as senhas de 1 - 9 de acesso às telas. Para edição das senhas deve ser selecionado o botão **...** na propriedade "Editar Senha".

Password	
Auto logoff	<input type="checkbox"/>
Level	0
Edit Password	1 - 9 <input type="button" value="..."/>

Então é aberta uma janela com uma tabela contendo campos para serem preenchidos com a senha.

Level	Password
1	123456
2	
3	
4	
5	
6	
7	
8	
9	

Após o final da edição basta fechar a janela que as senhas estarão gravadas.

Auto Logoff

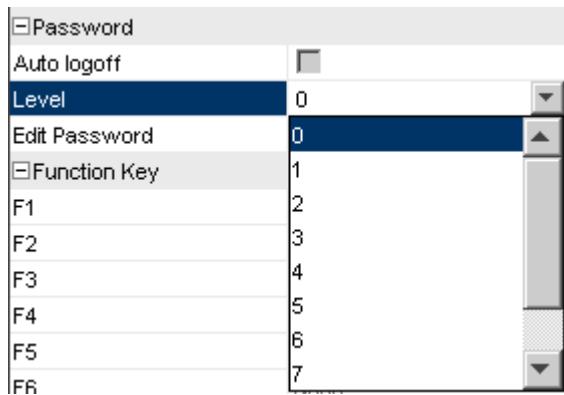
Desativa a senha de acesso digitada para o nível anterior ao da tela que acabou de sair.

Password	
Auto logoff	<input type="checkbox"/>
Level	0
Edit Password	1 - 9 <input type="button" value="..."/>

Nível

Seleciona o nível de acesso da tela. Quanto maior o nível de acesso, maior o privilégio. Até nove níveis de acesso podem ser programados, de modo a permitir diferentes tipos de usuários com acesso a determinadas telas. O nível de acesso zero

significa que a tela é de livre acesso.

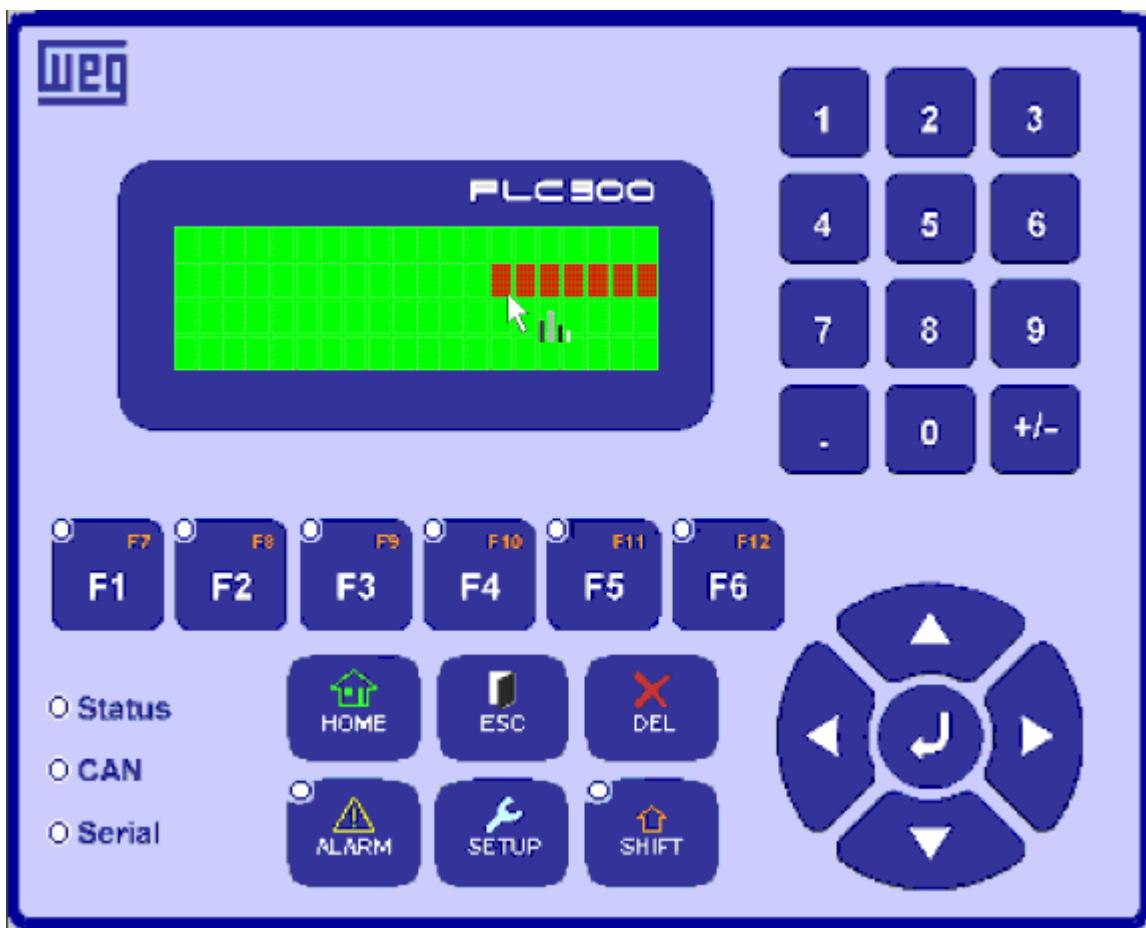


6.5.6 Programação

6.5.6.1 Visão Geral

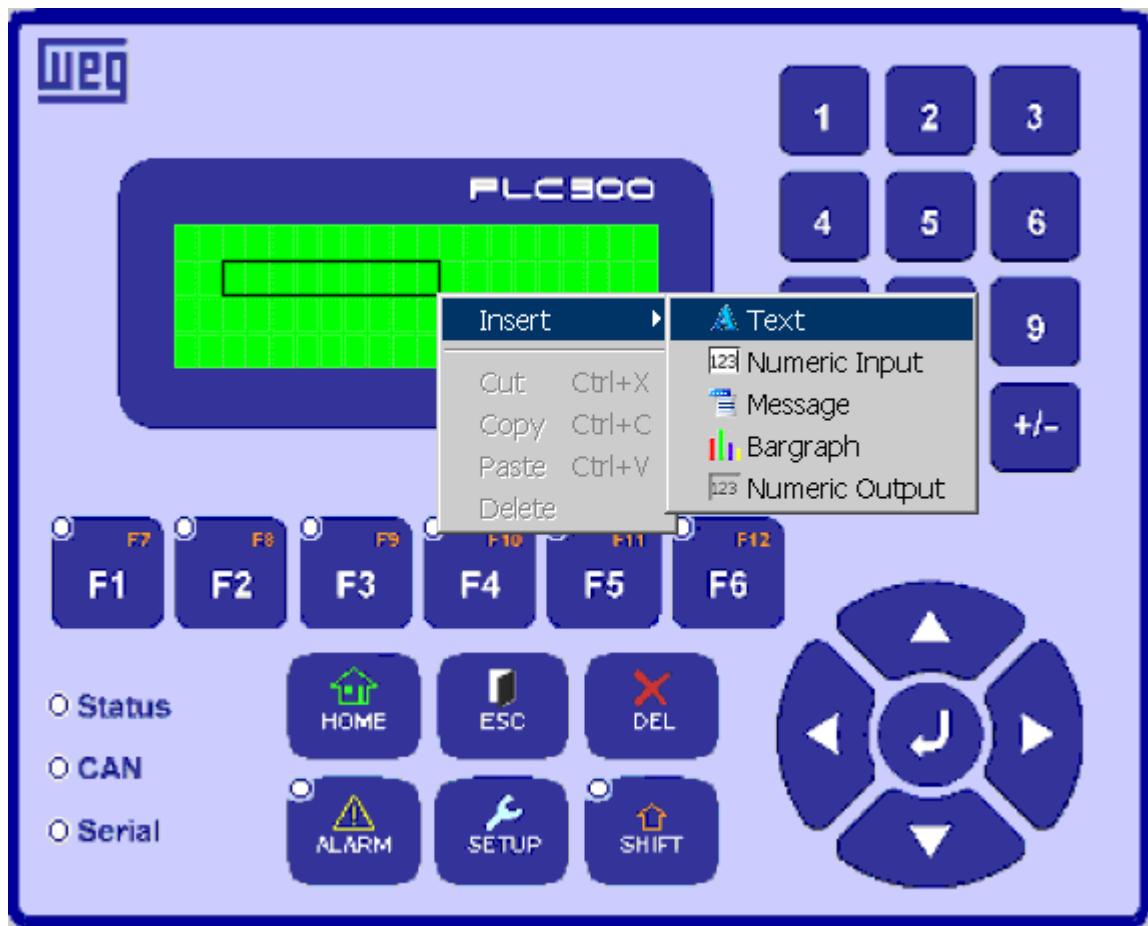
As telas são compostas por componentes do tipo Text A, Numeric Input i23, Numeric Output i23, Message e Bargraph i23. Para a programação das telas os componentes são inseridos no display de três maneiras:

Drag'n drop (Arrastar e Soltar)



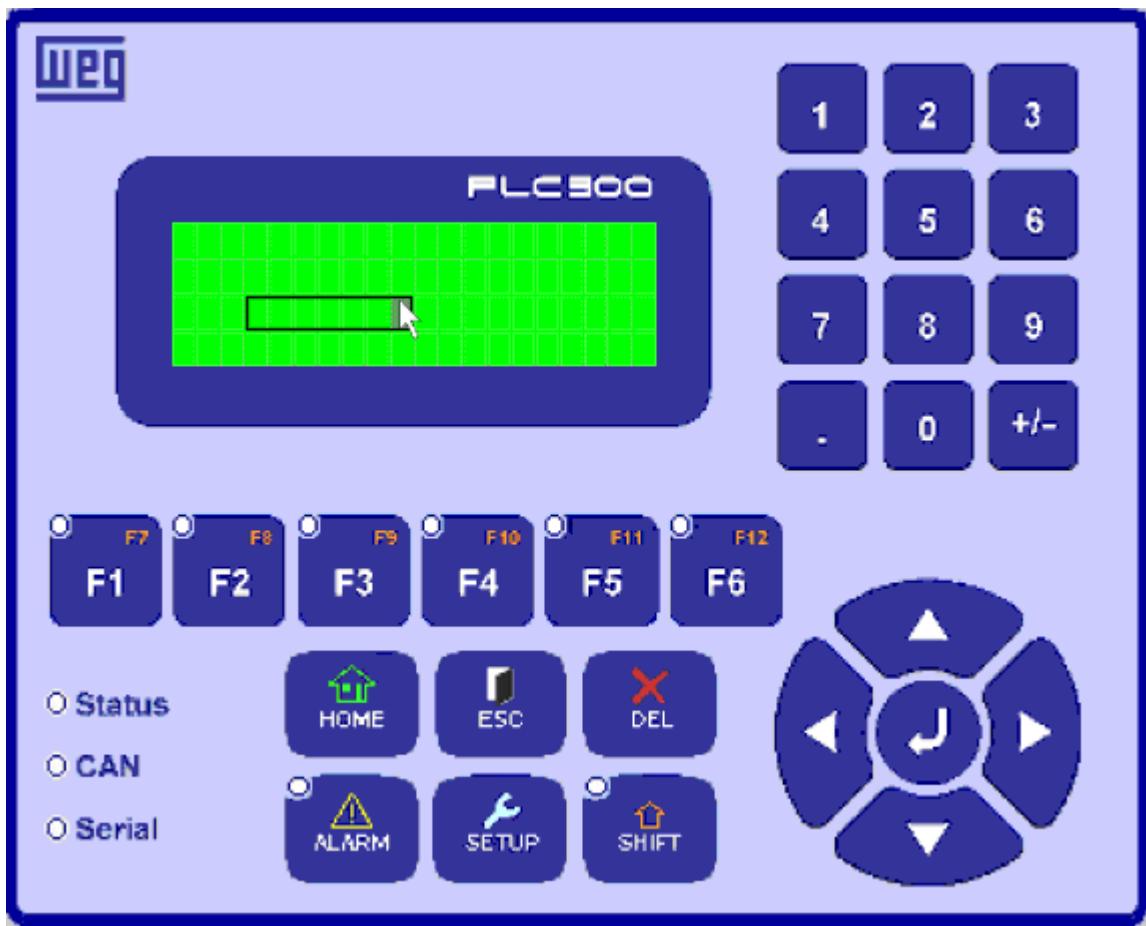
- Selecione o componente desejado na paleta pressionando o botão esquerdo do mouse,
- Mantenha o botão pressionado e arraste até a área do display na qual deseja inserir o componente,
- Certifique-se de que a sombra do componente está na cor cinza (caso contrário o componente não está posicionado corretamente), e
- Solte o botão do mouse.

Teclado



- Com o cursor selecionado dentro da área do display, pressione a tecla Ctrl,
- Seleciona-se o tamanho do componente com as teclas de direção ▲ ▶ ▼ ▹,
- Pressiona-se Ctrl + Space e aparecerá um pop-up contendo o item inserir, e
- No item inserir, selecione o componente desejado.

Mouse

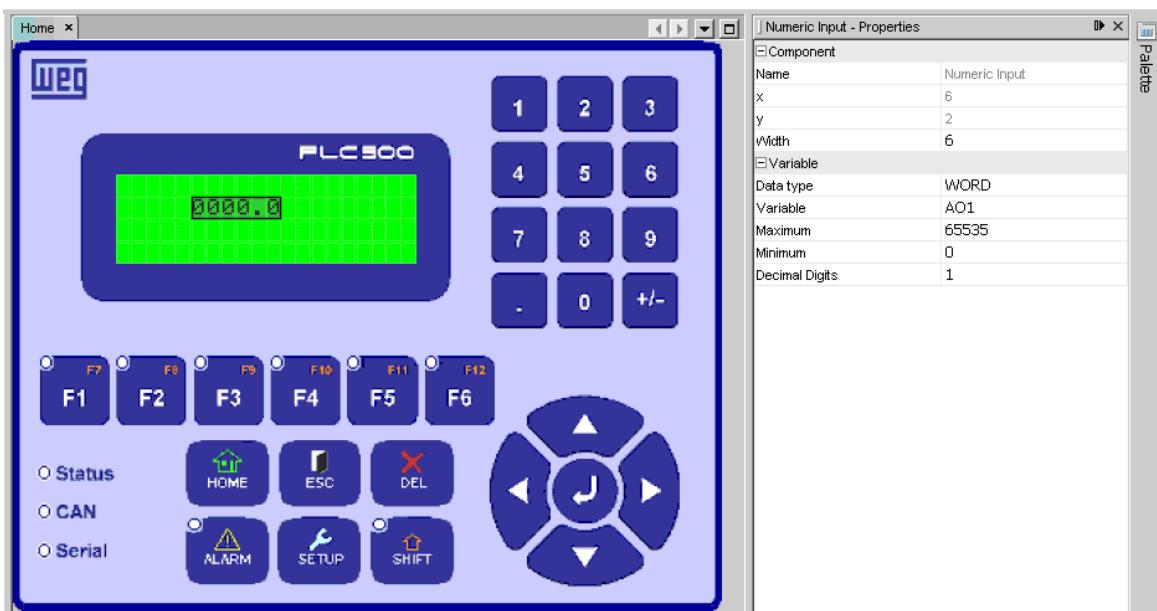


- Selecione uma área no display clicando e arrastando sobre o display,
- Pressione o botão direito do mouse sobre a área selecionada e aparecerá um pop-up contendo o item inserir, e
- No item inserir, selecione o componente desejado.

Configuração dos Componentes

A configuração dos componentes e das propriedades do display é feita através da janela de propriedades. Para tornar as propriedades de um componente ativas, o componente deve estar selecionado na área do display.

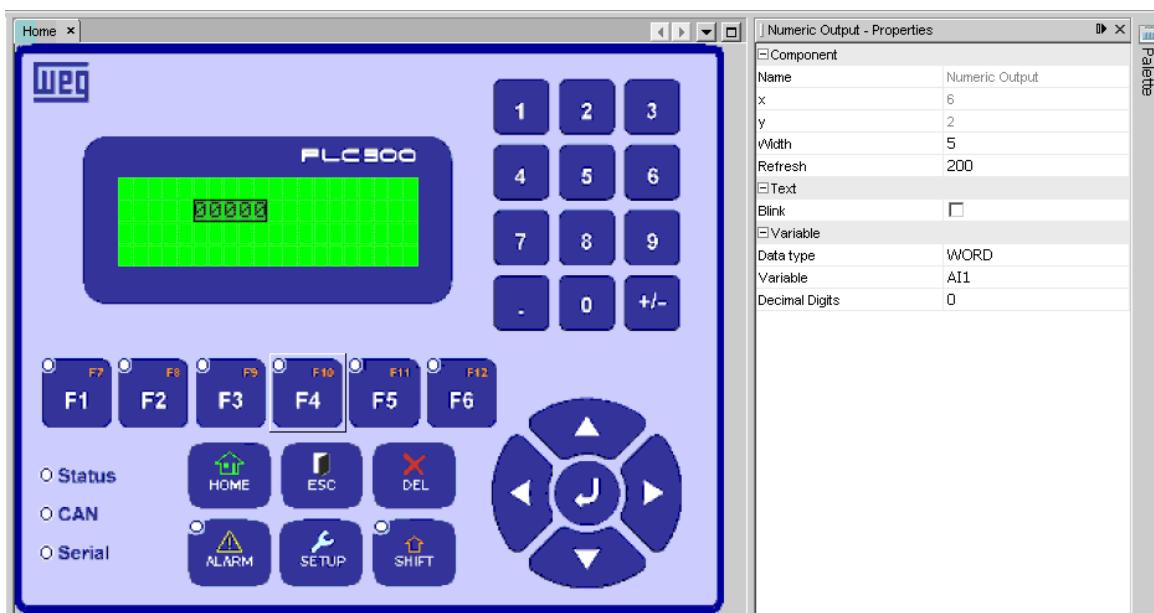
6.5.6.2 Componente Numeric Input



O componente Numeric Input possibilita a entrada de um valor numérico pelo usuário através do PLC 300. O conteúdo digitado é armazenado no marcador configurado na tabela de variáveis do editor ladder. As propriedades deste componente são:

- Componente
 - Nome: Nome do componente (somente leitura),
 - Linha: Número da linha do componente (somente leitura),
 - Coluna: Numero da coluna do componente (somente leitura),
 - Tamanho: Número de células do display ocupadas pelo componente,
- Variável
 - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
 - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima,
 - Máximo: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo, e
 - Mínimo: Valor mínimo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor mínimo.

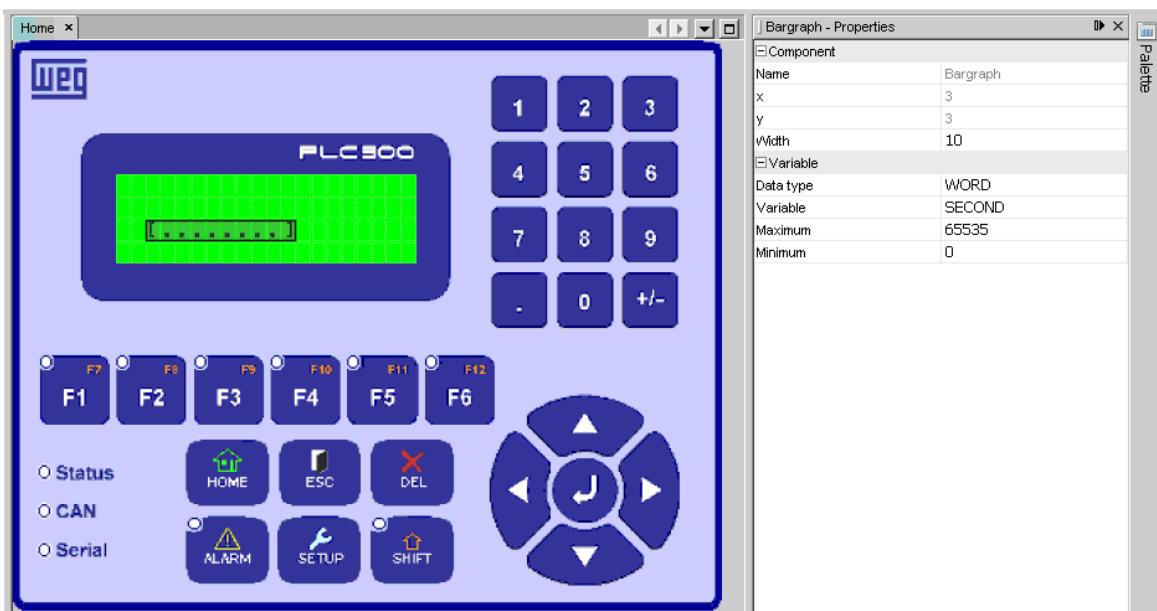
6.5.6.3 Componente Numeric Output



O componente Numeric Output apresenta o valor formatado da variável selecionada no display. As propriedades deste componente são:

- Componente
 - Nome: Nome do componente (somente leitura),
 - Linha: Número da linha do componente (somente leitura),
 - Coluna: Numero da coluna do componente (somente leitura),
 - Tamanho: Número de células do display ocupadas pelo componente,
- Variável
 - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
 - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima, e
 - Dígitos decimais: Formatação do número de casas decimais na apresentação da variável no display.

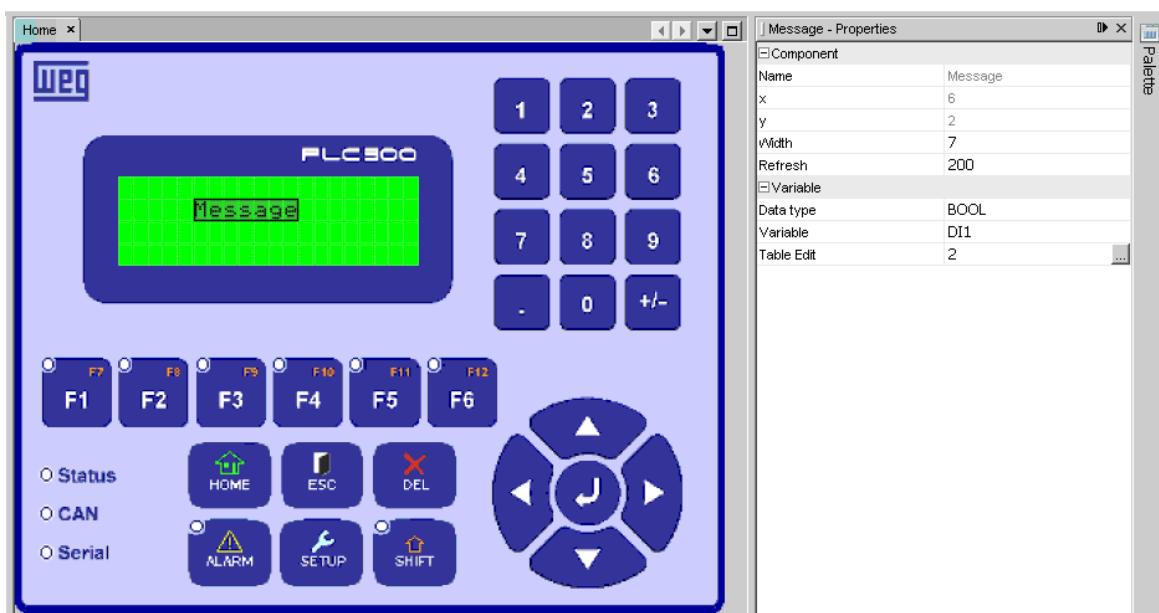
6.5.6.4 Componente Bargraph



O componente Bargraph mostra uma barra construída com caracteres tipo bloco, proporcional ao valor da variável. As propriedades deste componente são:

- Componente
 - Nome: Nome do componente (somente leitura),
 - Linha: Número da linha do componente (somente leitura),
 - Coluna: Número da coluna do componente (somente leitura),
 - Tamanho: Número de células do display ocupadas pelo componente,
- Variável
 - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD).
 - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima.
 - Máximo: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo.
 - Mínimo: Valor mínimo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor mínimo.

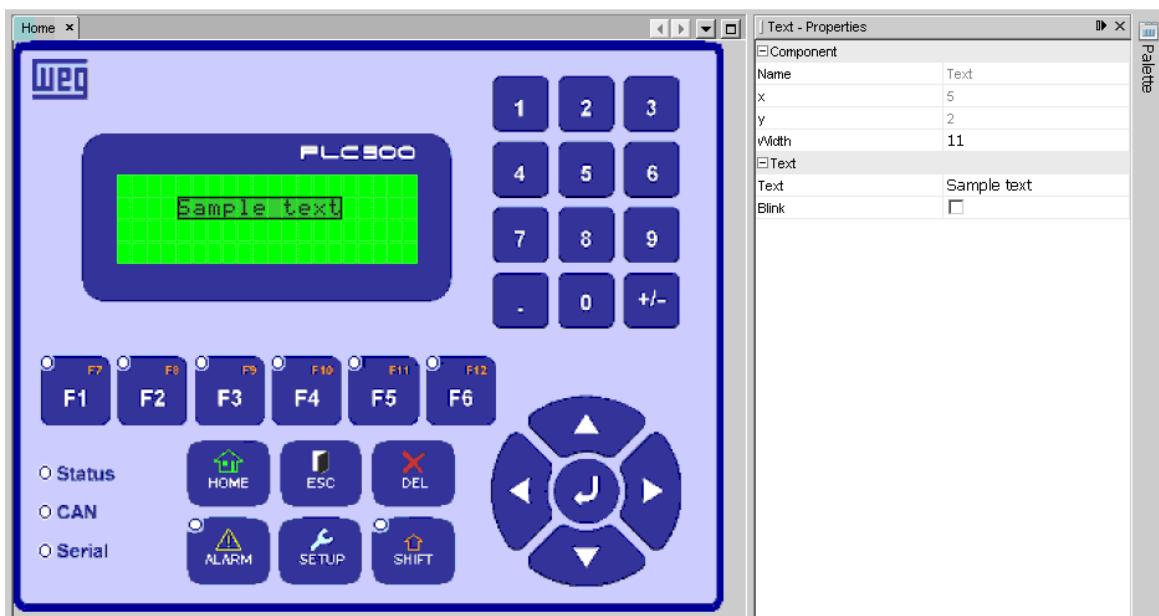
6.5.6.5 Componente Message



O componente Message apresenta textos configuráveis para valores de uma variável. As propriedades deste componente são:

- Componente
 - Nome: Nome do componente (somente leitura),
 - Linha: Número da linha do componente (somente leitura),
 - Coluna: Numero da coluna do componente (somente leitura),
 - Tamanho: Número de células do display ocupadas pelo componente,
- Variável
 - Tipo de dado: Lista de tipos de dados permitidos. Os tipos de dados permitidos para este componente são: BOOL, SINT, USINT(BYTE), INT, UINT(WORD), DINT, UDINT(DWORD) e REAL,
 - Variável: Lista de variáveis configuradas no diagrama ladder para o tipo de dado configurado acima,
 - Tabela: Valor máximo permitido como entrada para este campo. Caso o valor digitado seja superior a este valor, será atribuído o valor máximo.

6.5.6.6 Componente Text



O componente Text é utilizado para inserir textos fixos que na tela. As propriedades deste componente são:

- Componente
 - Nome: Nome do componente (somente leitura),
 - Linha: Número da linha do componente (somente leitura),
 - Coluna: Numero da coluna do componente (somente leitura),
 - Tamanho: Número de células do display ocupadas pelo componente,
- Texto
 - Texto: Texto apresentado na tela,
 - Piscante: Habilita a opção de texto piscante.

6.6 Alarmes

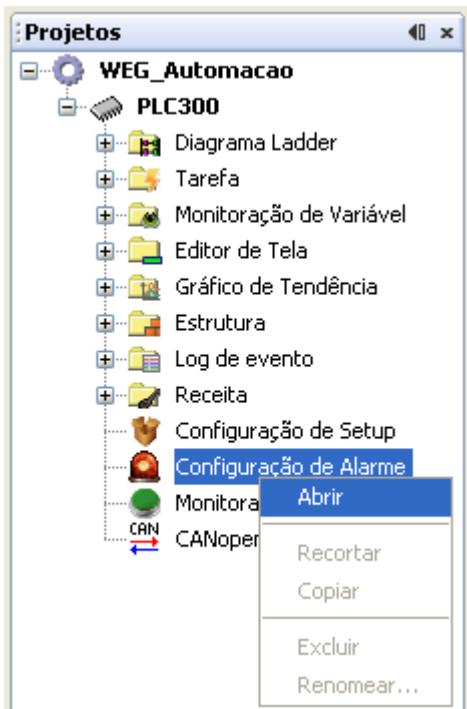
6.6.1 Visão Geral

Os alarmes são uma ferramenta importante na automação de processos, permitindo ao usuário monitorar sua planta, checando pontos críticos e sinalizando ao operador.

No PLC300 os alarmes configuráveis são programados pelo usuário, sendo acionados por um marcador de bit que pode ser ativado pelo programa.

Os alarmes internos são alarmes relativos a alguns componentes de hardware, que ocupam a memória interna.

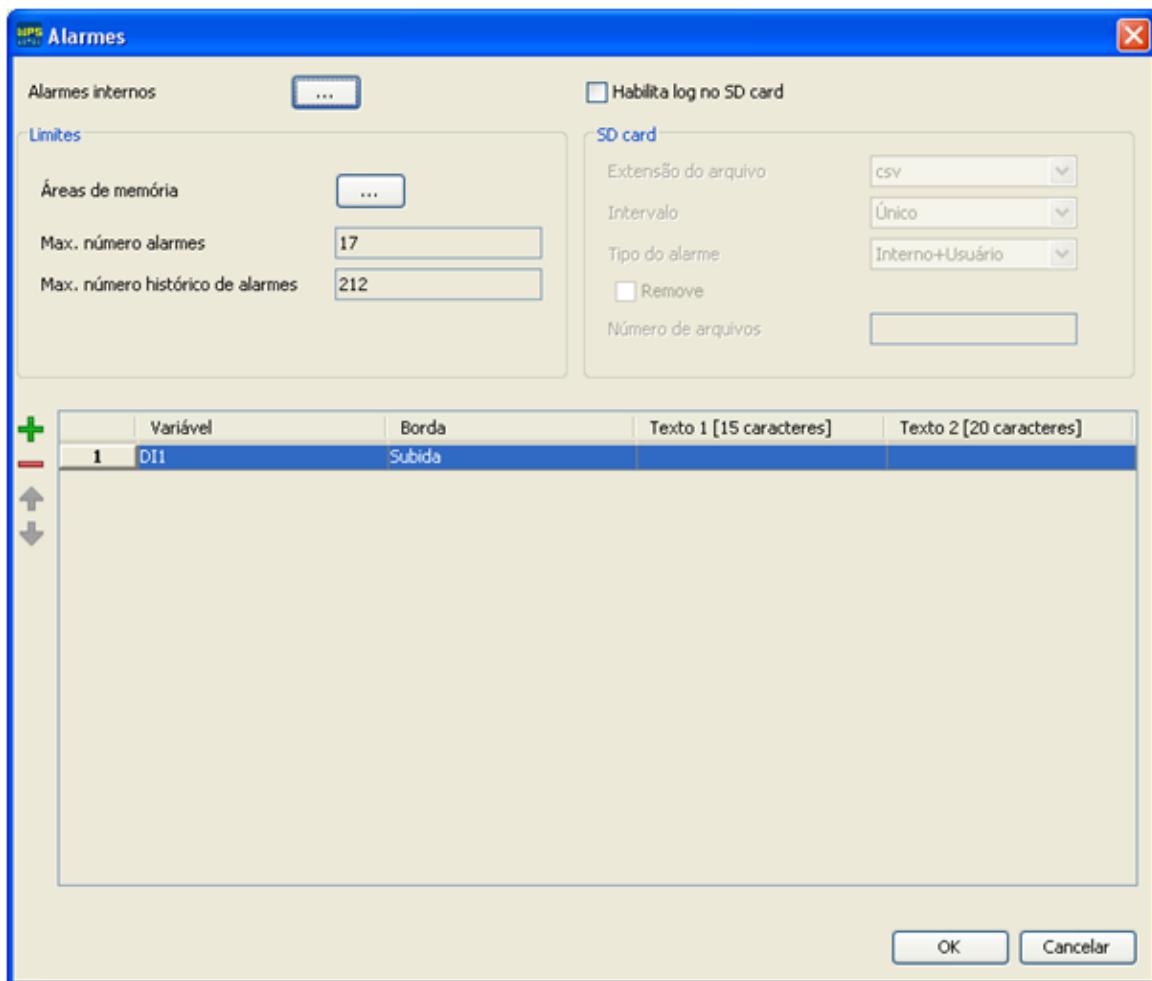
Os alarmes são configurados através da “Configuração de Alarme” acessada através das pastas de projeto:



6.6.2 Configuração dos Alarmes

Na janela de configuração dos alarmes são configurados:

- Alarms internos: Alarms generated by the device that can be enabled or disabled by the user;
- Limites: Maximum area occupied by user alarms and historical alarms;
- SD card: Configuration of alarm storage in the SD card;
- User alarm table: Table for configuring alarms triggered through device variables.



Alarmes Internos

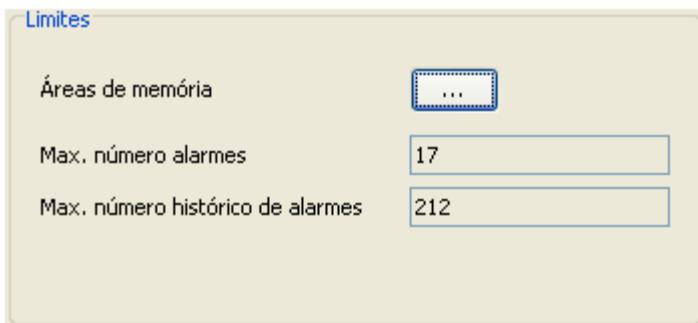
Os alarmes internos são alarmes gerados pelo dispositivo que podem ser habilitados ou desabilitados pelo usuário. Para ter acesso à janela de configuração de alarmes internos é necessário pressionar o botão na janela de alarmes.

O PLC 300 possui seis alarmes internos, cinco dos quais podem ser habilitados pelo usuário:



- Falha Saídas Digitais: Indica que algum problema está ocorrendo em alguma das saídas DO1 a DO8.
- Fio Partido: Corrente abaixo de 2 mA, quando a entrada analógica AI1 estiver em modo corrente 4 a 20mA.
- Falha Encoder: Falta de um dos sinais do encoder.
- Alimentação da CAN: Ausência da tensão de alimentação da interface CAN.
- SD card: Este alarme ocorre quando há problema na escrita ou leitura do SD card. Os problemas mais comuns são: SD card ausente, cartão protegido contra escrita e formatação do sistema de arquivos diferente do FAT32 .

Limites



No campo limites é configurada a área máxima ocupada pelos alarmes do usuário e pelo histórico de alarmes. Através do botão da área de limites dos alarmes, é aberta a janela de configuração das áreas de memória. Através desta janela é possível configurar o tamanho da área de memória que os alarmes de usuário e o histórico de usuário deverão ocupar.

O tamanho que os alarmes de usuário ocupam na memória é calculado pela fórmula:

$$\text{Tamanho alarmes (bytes)} = 24 + 56 * \text{núm. de alarmes}$$

O tamanho que o histórico de alarmes ocupa na memória é calculado pela fórmula:

Tamanho histórico (bytes) = 24 + 24 * núm. de históricos

SD Card



A opção SD card, quando habilitada, configura as características de armazenamento dos alarmes no SD card. O armazenamento dos arquivos de alarmes possui as seguintes opções:

- Extensão do arquivo: Formato no qual os alarmes serão armazenados no SD card. As opções são:

- txt: texto com formatação de fácil compreensão para os usuários.

Data	Hora	Alarme	Ação	Descrição
31/05/11	07:58:21	Alarme 0	A	Falha DOS
31/05/11	07:58:22	Alarme 0	A	Falha Encoder
31/05/11	07:58:22	Alarme 0	A	CAN Power Off
31/05/11	07:58:43	Alarme 0	A	Falha DOS
31/05/11	07:58:43	Alarme 0	A	Falha Encoder
31/05/11	07:58:43	Alarme 0	A	CAN Power OFF
31/05/11	07:59:41	Alarme 1	A	Bomba ligada
31/05/11	07:59:41	Alarme 1	N	Bomba ligada
31/05/11	07:59:44	Alarme 1	A	Bomba ligada
31/05/11	07:59:44	Alarme 1	N	Bomba ligada

- csv (comma separated value): valores separados por vírgula, geralmente utilizado em planilhas eletrônicas.

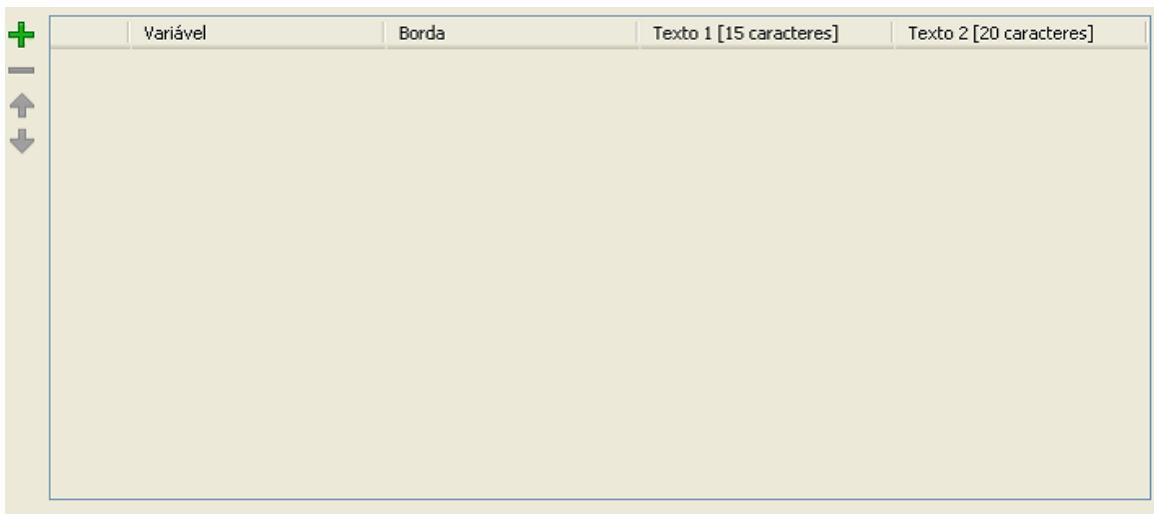
The screenshot shows a Windows Notepad window titled "log_alarms.csv - Notepad". The window contains a list of alarm logs in CSV format. Each log entry consists of a timestamp, date, hour, minute, second, followed by a status code (0 or 1), an action (A or N), and a description. The logs are as follows:

Timestamp	Date	Hour	Minute	Second	Status	Action	Description
31/05/11, 08:04:39	31/05/11	08	04	39	0	A	Falha DOS, Verificar Saidas.
31/05/11, 08:04:39	31/05/11	08	04	39	0	A	Falha Encoder, Falha Sinais Encoder
31/05/11, 08:04:39	31/05/11	08	04	39	0	A	CAN Power Off, CAN Bus Power Detect
31/05/11, 08:05:01	31/05/11	08	05	01	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:05:01	31/05/11	08	05	01	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:05:02	31/05/11	08	05	02	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:05:03	31/05/11	08	05	03	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:06:46	31/05/11	08	06	46	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:06:46	31/05/11	08	06	46	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:06	31/05/11	08	08	06	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:06	31/05/11	08	08	06	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:11	31/05/11	08	08	11	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:11	31/05/11	08	08	11	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:18	31/05/11	08	08	18	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:18	31/05/11	08	08	18	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:30	31/05/11	08	08	30	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:30	31/05/11	08	08	30	1	N	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:53	31/05/11	08	08	53	1	A	Bomba ligada, Bomba reserv. 1 lig.
31/05/11, 08:08:53	31/05/11	08	08	53	1	N	Bomba ligada, Bomba reserv. 1 lig.

Os valores armazenados são a data, hora, alarme (0 – interno, 1 – alarme do usuário), ação (A – atuado e N – normalizado) e descrição.

- Intervalo: No campo intervalo é configurado o tempo de duração da gravação dos dados em um único arquivo. As opções deste campo são as seguintes:
 - Único: Os dados serão gravados em um único arquivo.
 - Diário: Os dados são gravados em um arquivo por dia. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do dia no relógio do dispositivo. O arquivo é gravado com um sufixo contendo o dia, mês e ano em que iniciou sua gravação.
 - Mensal: Os dados são gravados em um arquivo por mês. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do mês no relógio do dispositivo. O arquivo é gravado com um sufixo contendo o mês e o ano em que iniciou sua gravação.
 - Anual: Os dados são gravados em um arquivo por ano. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do ano no relógio do dispositivo. O arquivo é gravado com um sufixo contendo o ano em que iniciou sua gravação.
- Tipo do alarme: Valores de alarme que serão armazenados no SD card. Podem ser dos tipos “Usuário” para armazenar somente os alarmes configurados pelo usuário ou “Usuário+Internos” para armazenar os alarmes configurados pelo usuário e os alarmes internos.
- Remover: Quando selecionada esta opção a caixa de diálogo “Número de arquivos” se torna habilitada, permitindo a entrada de um número inteiro. Este número representa o número de arquivos que serão mantidos no SD card. Sempre que um arquivo é criado diariamente, mensalmente ou anualmente é verificado o número de arquivos criados para este alarme e então removidos os arquivos mais antigos.

Tabela de Alarmes de Usuário



Na tabela de alarmes de usuário é realizada a configuração dos alarmes acionados através de variáveis do dispositivo com textos editados pelo usuário. A tabela dos alarmes possui quatro campos a serem preenchidos:

- Marcador: Marcador de bit que ativa o alarme. Variável global do tipo booleana.
- Borda: Borda de transição em que o alarme será ativado. Os valores possíveis são positiva (de 0 para 1) ou negativa (de 1 para 0).
- Texto 1: Texto para com a mensagem de alarme. Este campo poderá conter no máximo 15 caracteres. Este texto será visualizado na lista de alarmes ativos e histórico de alarmes.
- Texto 2: Texto com o descriptivo da mensagem de alarme. Este campo poderá conter no máximo 20 caracteres. O texto será visualizado na descrição detalhada dos alarmes ativos e histórico de alarmes.

6.7 Tarefas

6.7.1 Configuração de Tarefas

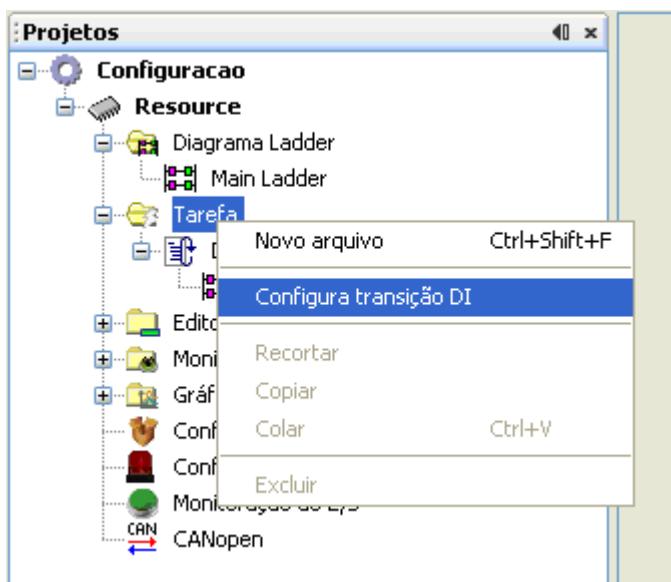
Visão Geral

A tarefa é um mecanismo de escalonamento muito útil em sistemas de tempo real para controlar a execução de programas, de forma periódica ou disparada por eventos.

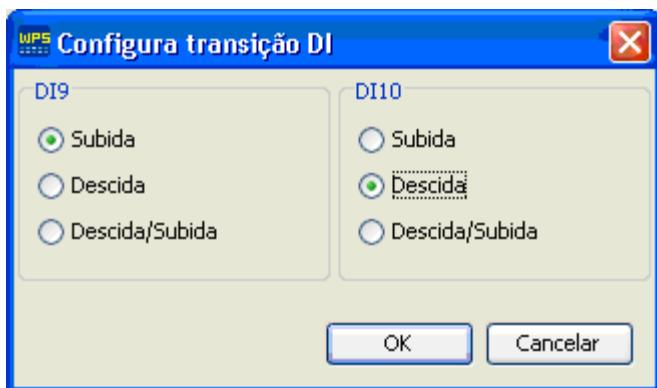
A configuração das tarefas é realizada adicionando arquivos de tarefas à pasta "Tarefa", dentro do recurso. A estes arquivos de tarefa, são então associados programas que executam sua lógica, quando a tarefa se torna ativa. Para cada tarefa pode ser configurado um watchdog (ver seção watchdog). Cada tarefa possui um marcador de sistema associado que pode ser habilitado ou desabilitado através do programa.

Dentro de um recurso, existe uma única tarefa chamada "Main", que não pode ser apagada, na qual o programa principal (Main Ladder) é executado.

No menu de opções da pasta de tarefas, é possível realizar as seguintes ações:

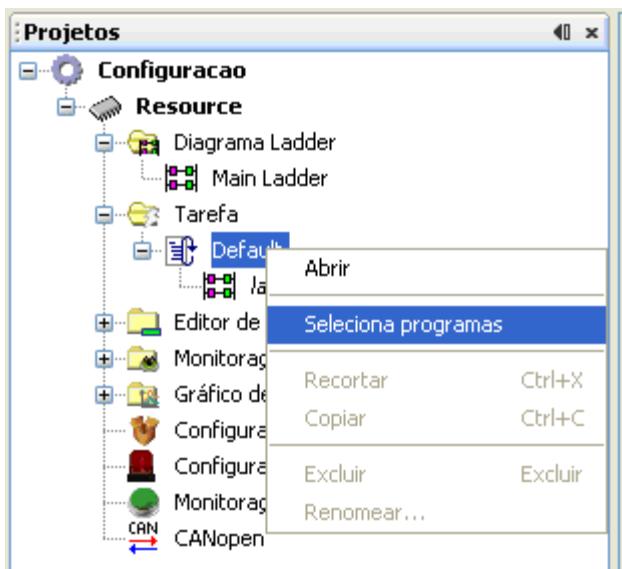


Adicionar uma nova tarefa: para adicionar uma nova tarefa, é preciso selecionar a opção "Novo arquivo" da pasta "Tarefa".



Estas DIs são utilizadas nas tarefas do tipo evento externo e contagem.

Através do menu de opções dos arquivos das tarefas, é possível realizar as seguintes ações:



Configurar tarefas: para configurar a tarefa, é preciso selecionar a opção "Abrir", no arquivo de tarefas que se deseja configurar. Para mais informações de como configurar as tarefas, ver seção "Configuração das tarefas".

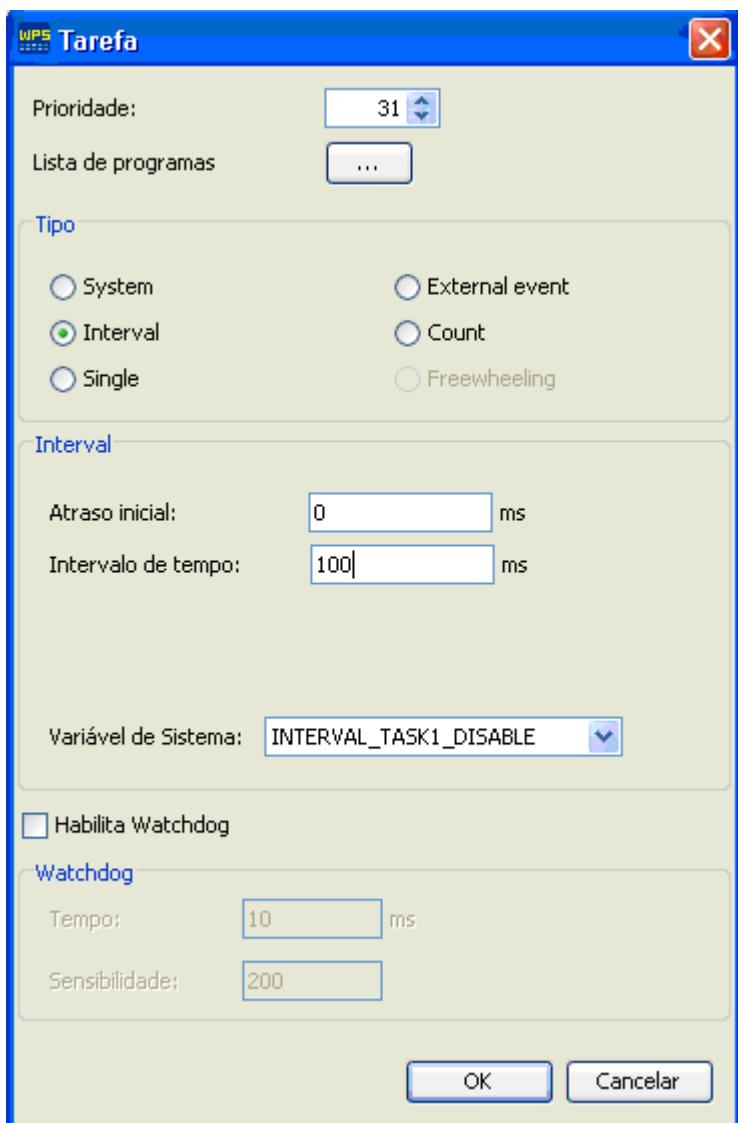
Adicionar, remover ou ordenar programas da tarefa: para adicionar, remover ou ordenar programas de uma determinada tarefa, é preciso selecionar a opção "Seleciona programas" no arquivo de tarefas que deseja configurar. Para mais informações de como selecionar os programas, ver na seção "Configuração das tarefas" o item "Lista de programas".

Configuração da borda de transição das entradas digitais (DIs): as entradas digitais DI9 e DI10 possuem a opção de disparar eventos através da transição da borda de subida, descida ou subida e descida. A configuração destas bordas de transição é feita através da janela de configuração das DIs, que é acessada selecionando a opção "Configura transição DI" da pasta de "Tarefa".

Configuração de Tarefas

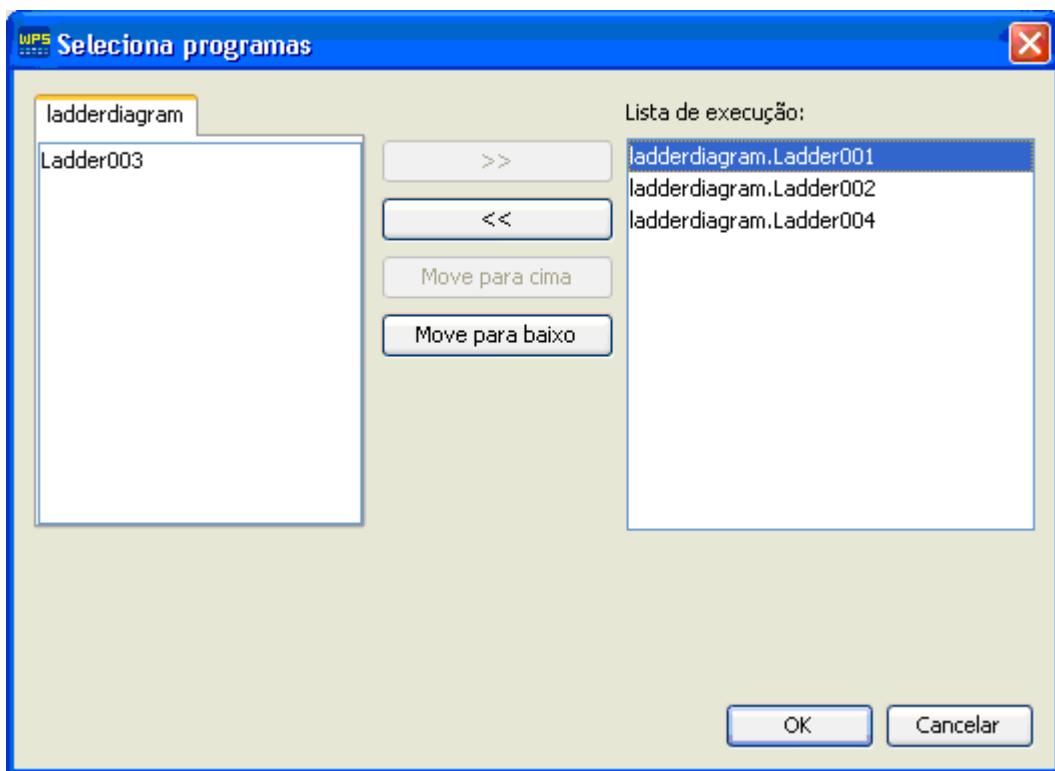
Na janela de configuração das tarefas, são configurados:

- Prioridade,
- Sequência dos programas,
- Tipo de tarefa, e
- Opções de watchdog da tarefa.



Prioridade: configurada com um número entre 0 e 30 (0 - prioridade mais alta, 30 - mais baixa) que define a prioridade em que a tarefa será executada. Caso uma tarefa de prioridade mais alta do que a tarefa em execução seja acionada, esta imediatamente entrará em execução. De outra forma, caso uma tarefa de prioridade mais baixa do que a atual seja acionada, esta aguarda o final da execução da tarefa atual para entrar em execução.

Lista de programas: a janela de seleção da sequência dos programas serve para adicionar, remover ou alterar a sequência dos programas associados a determinada tarefa. Na lista do lado esquerdo, estão todos os programas disponíveis no recurso e no lado direito, os programas selecionados para esta tarefa. A ordem que os programas entrarão em execução será a mesma ordem definida nesta lista.



Tipos de Tarefa

As tarefas estão divididas em: system, interval, single, external event, count e freewheeling.



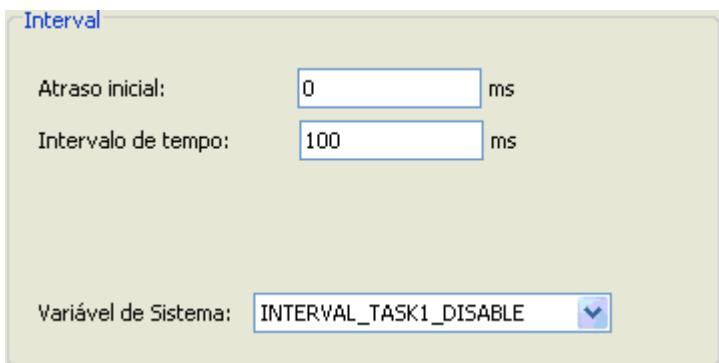
System

Esta tarefa pode ser do tipo Inicialização ou Parada. Quando selecionada a opção "Inicialização", a tarefa entrará em execução logo que o programa do usuário for executado pela primeira vez. Caso a opção "Parada" seja selecionada, os programas associados a esta tarefa, entram em execução logo após a parada do programa do usuário.



Interval

Os programas associados a esta tarefa, são executados repetidas vezes, no intervalo de tempo definido no campo "Intervalo de tempo", com o tempo de atraso para a primeira execução definido no campo "Atraso inicial". A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



Single

Os programas associados a esta tarefa, são executados uma única vez, sempre que a variável selecionada sofrer uma transição positiva, ou seja, de zero para algum outro valor. A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



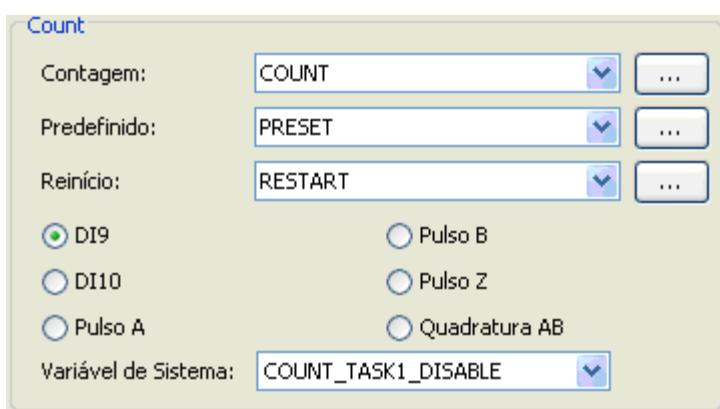
External Event

Os programas associados a esta tarefa, são executados uma única vez, sempre que o evento DI9, DI10 ou Pulso Z selecionado for ativado. Os eventos DI9 e DI10 possuem a opção de selecionar a transição que ativa a tarefa (ver item "Configuração da borda de transição das entradas digitais" na seção "Visão Geral"). A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.



Count

Os programas associados a esta tarefa, são executados toda vez que a contagem de pulsos na entrada selecionada (DI9, DI10, Pulso A, Pulso B, Pulso Z e Quadratura AB), ultrapassar o valor da variável definida no campo "Predefinido". O valor de contagem dos pulsos é armazenado na variável "Contagem". O valor da contagem dos pulsos é reiniciado quando este valor ultrapassar o valor definido no campo "Reinício". As variáveis configuradas nestes campos podem ser do tipo DWORD e UDINT para as entradas DI9, DI10, Pulso A, Pulso B e Pulso Z e DINT para entrada quadratura AB. A esta tarefa é associada uma variável de sistema que permite que a tarefa seja desabilitada através do programa.

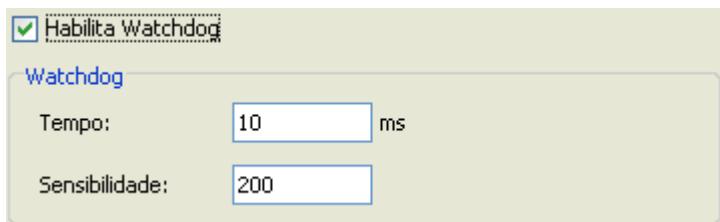


Freewheeling

Os programas associados a esta tarefa, são executados de forma cíclica. No final da execução da lista de programas, a lista de programas é reiniciada, até que o programa do usuário seja parado. O programa principal (Main Ladder) está associado a esta tarefa e não é possível removê-lo ou associá-lo a outra tarefa.

Watchdog

Quando a opção watchdog está ativa, o programa do usuário é parado com alarme de erro, caso o tempo de execução da tarefa demore mais do que o tempo definido no campo "Tempo", por um número de vezes maior que o definido no campo "Sensibilidade".



Tempo: Tempo máximo para execução da tarefa, sem que ocorra o erro de watchdog.

Sensibilidade: Número de erros de watchdog permitidos, antes da geração do alarme de erro e parada do programa do usuário.

Disponível

- PLC300 : a partir da versão 1.20

6.8 Log de Eventos

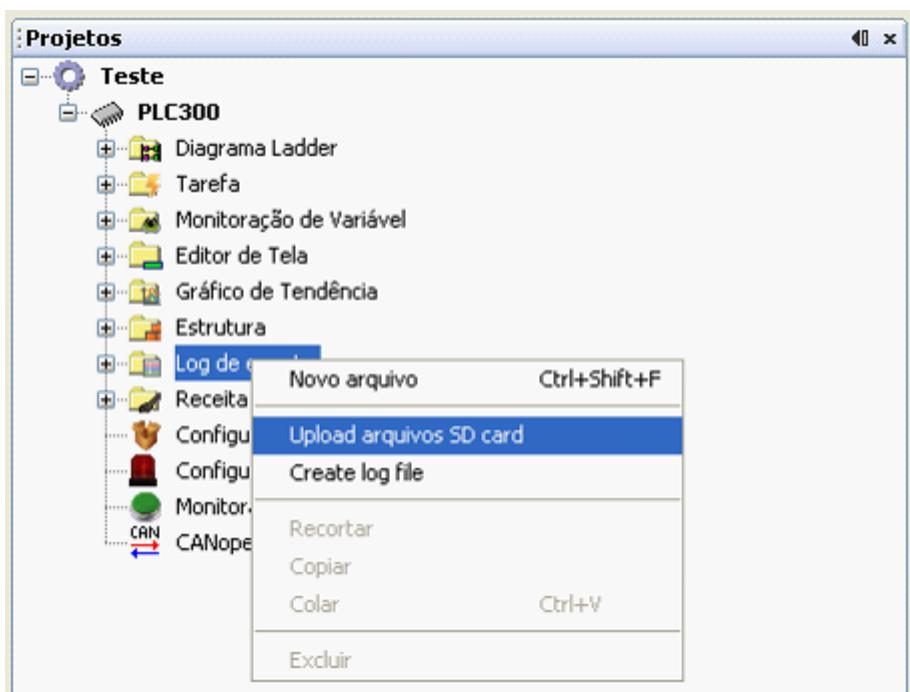
6.8.1 Configuração de Log de Eventos

Visão Geral

O log de eventos é um conjunto valores de variáveis que são armazenados, com data e hora, no SD card do dispositivo, no formato csv (comma separated value). Estes valores são gravados após a ocorrência de eventos que podem ser dos tipos: time, change of state ou trigger. Para cada arquivo de log de eventos é configurado o intervalo de gravação do arquivo e é associada uma variável booleana responsável por habilitar ou desabilitar o log através do programa.

Para cada arquivo de configuração de log pode ser associado um texto que será apresentado juntamente com as variáveis assim que o upload do log de eventos for realizado (Ver seção configuração do campo texto).

No menu de opções do log de eventos é possível realizar as seguintes ações:



- Adicionar um novo log de eventos: para adicionar um novo log de eventos é preciso selecionar a opção "Novo arquivo" da pasta "Log de evento". Para mais informações de como configurar o log de eventos ver seção "Configuração do log de eventos".
- Upload arquivos SD card: após o download dos arquivos de log de eventos configurados para o equipamento, é possível fazer o upload destes arquivos para visualizar os eventos ocorridos. Mais detalhes ver seção "Upload arquivos SD card".
- Criar arquivo de log: esta funcionalidade permite que o usuário crie um arquivo de log dentro de um período definido com os arquivos de log carregados no projeto. Mais detalhes ver seção "Criação arquivos de log".

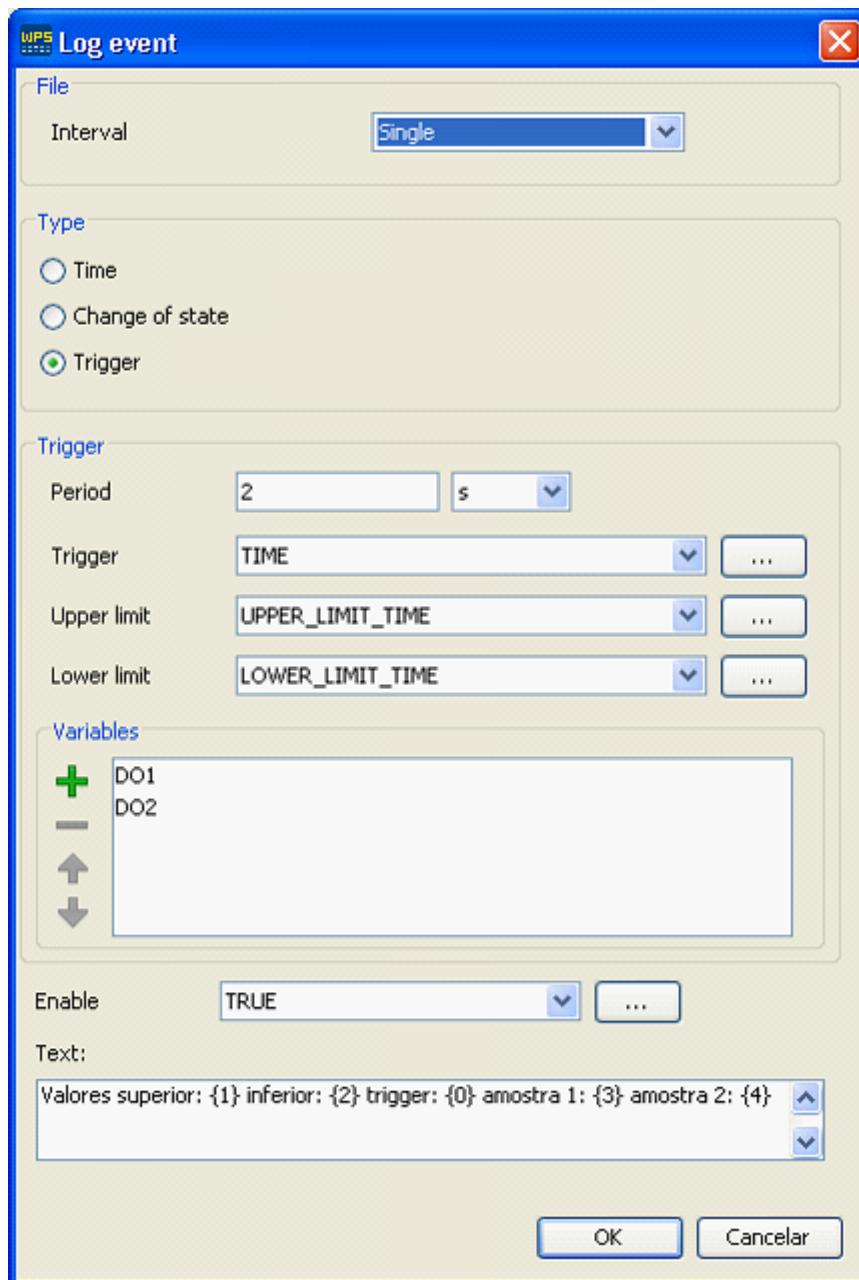
Configuração de Log de Eventos

Na janela de configuração de log de eventos são configurados: o intervalo de gravação do arquivo de log de eventos, o tipo do log de eventos, a variável booleana que habilita o log eventos e o texto a ser exibido no log de eventos.

- Intervalo do arquivo: No campo intervalo do arquivo é configurado o tempo de duração da gravação dos dados em um único arquivo. As opções deste campo são as seguintes:
 - Único: Os dados serão gravados em um único arquivo.
 - Diário: Os dados são gravados em um arquivo por dia. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do dia no relógio do dispositivo. O arquivo é gravado com um sufixo contendo o dia, mês e ano em que iniciou sua gravação.
 - Mensal: Os dados são gravados em um arquivo por mês. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do mês no relógio do dispositivo.

O arquivo é gravado com um sufixo contendo o mês e o ano em que iniciou sua gravação.

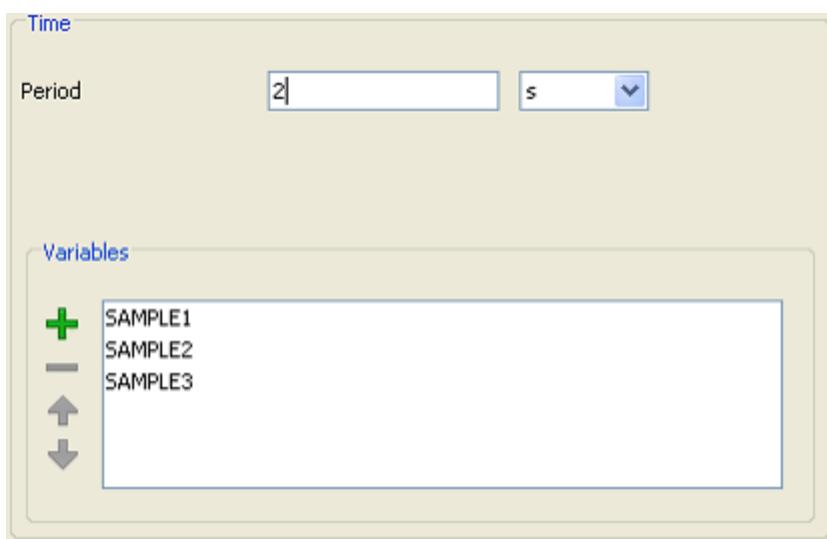
- Anual: Os dados são gravados em um arquivo por ano. A gravação de um novo arquivo é iniciada sempre que ocorrer a mudança do ano no relógio do dispositivo. O arquivo é gravado com um sufixo contendo o ano em que iniciou sua gravação.



Tipos de Log de Eventos

Os log de eventos podem ser dos tipos time, change of state e trigger.

- Time



O log de eventos é gravado em intervalos de tempo definidos no campo "Período". O valor configurado no campo "Período" tem que ser um valor inteiro maior que 0 e menor que 4294967295 s. A unidade de tempo pode ser selecionada entre segundos, minutos ou horas.

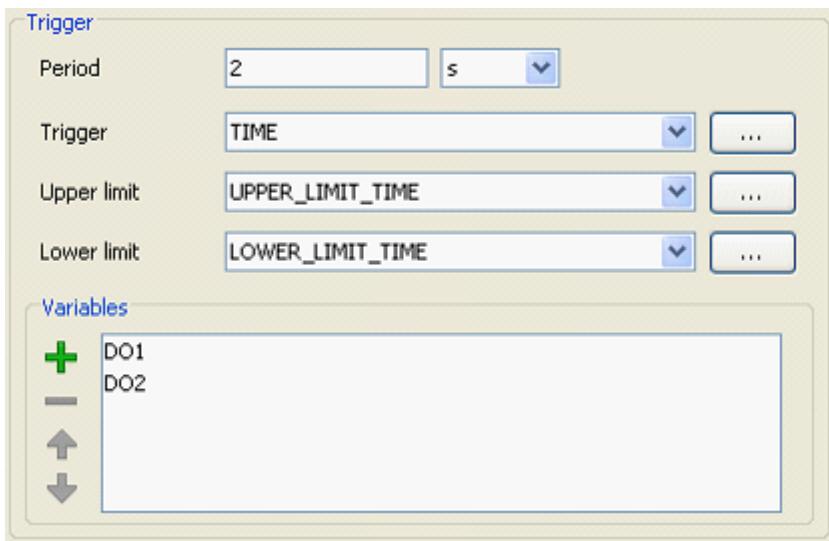
No campo "Variáveis" são selecionadas as variáveis que serão amostradas no período configurado.

- Change of State



O log de eventos é gravado após a mudança do valor da variável selecionada. Apenas o valor da variável selecionada é gravado no log de eventos.

- Trigger



Assim como no evento "Time", o log de eventos é gravado em intervalos de tempo definidos no campo "Período", mas somente quando o valor da variável configurada no campo "Trigger" estiver dentro dos limites selecionados nas variáveis dos campos "Limite superior" e "Limite inferior". O valor configurado no campo "Período" tem que ser um valor inteiro maior que 0 e menor que 4294967295 s. A unidade de tempo pode ser selecionada entre segundos, minutos ou horas.

No campo "Variáveis" são selecionadas as variáveis que serão amostradas no período configurado.

Configuração do Campo Texto

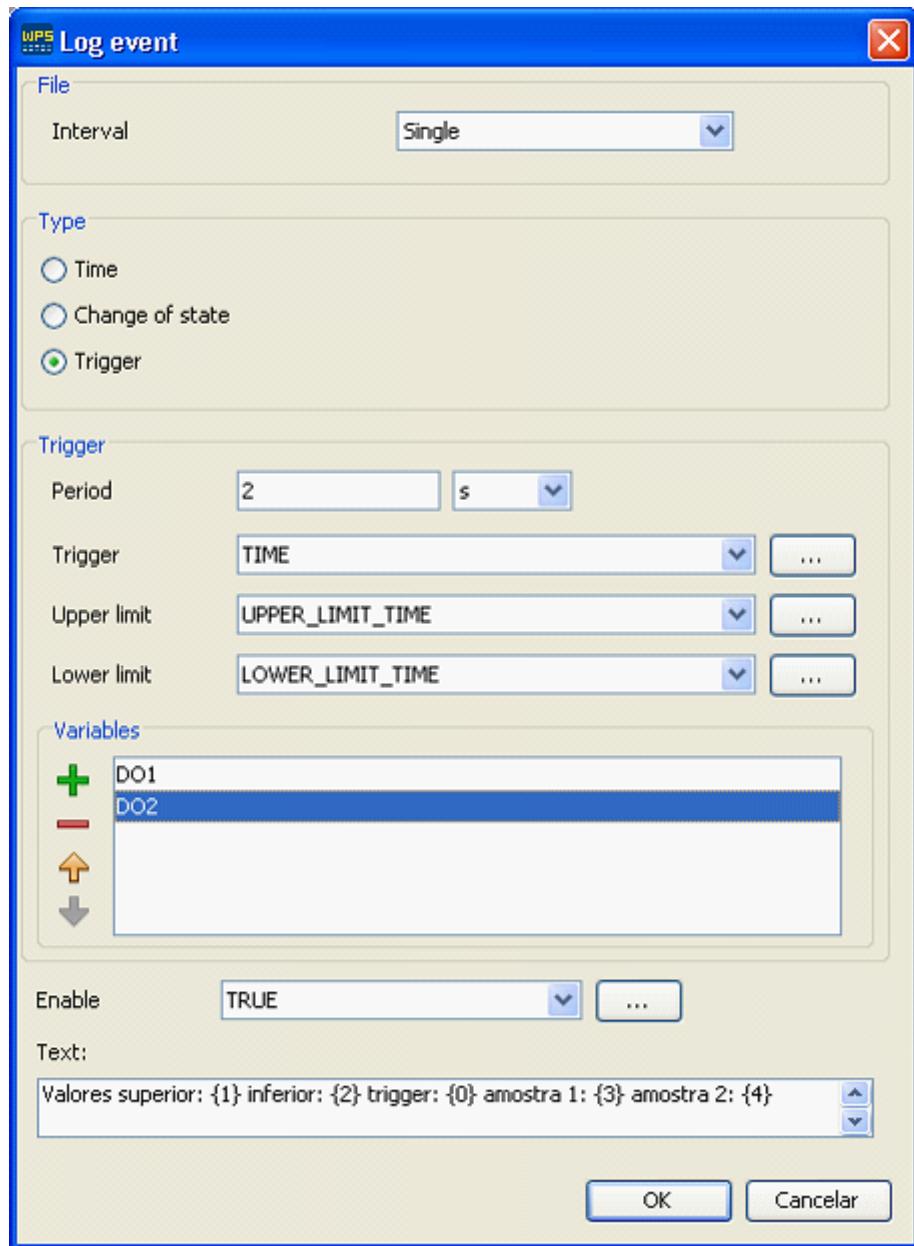
No campo "Texto" é possível adicionar um texto, para ser apresentado juntamente com as variáveis. Este texto será adicionado assim que o arquivo de log de eventos for carregado (através do upload de arquivos de log) para o WPS.

Para que os valores das variáveis possam ser inseridos em qualquer posição do texto, são adicionados marcadores no texto que serão substituídos pelas variáveis.

Conforme o tipo de log de eventos selecionado, os marcadores devem ser adicionados da seguinte forma:

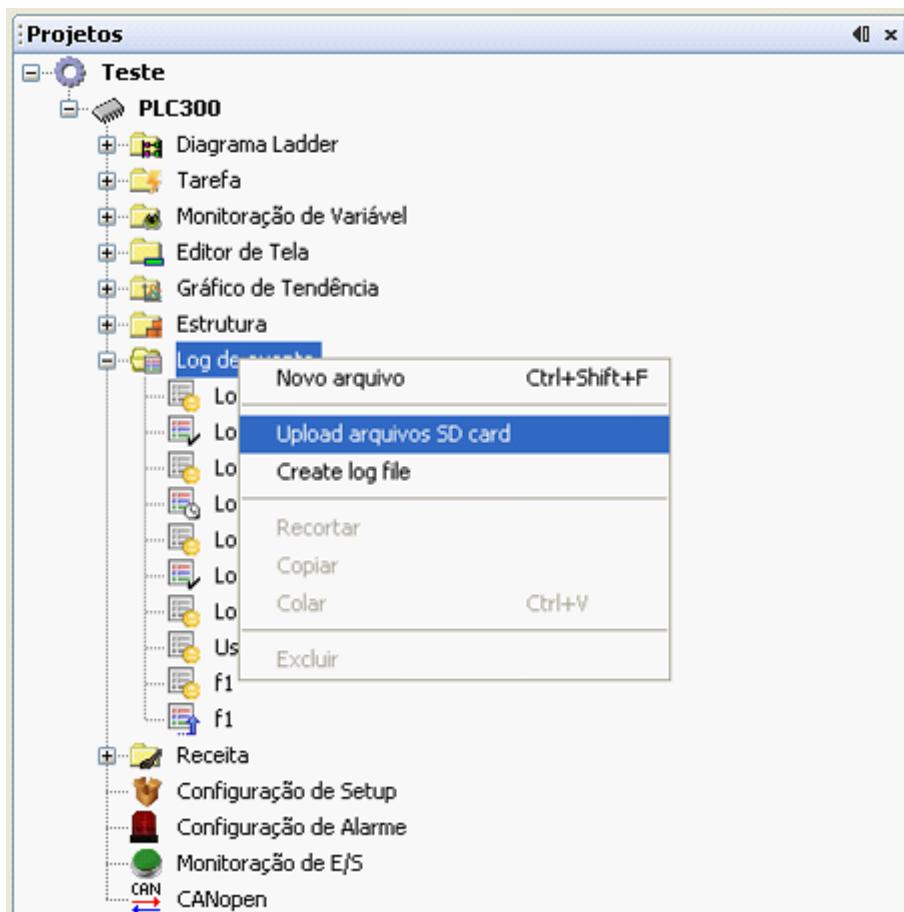
- Time: a primeira variável da lista é representada pelo marcador {0}, a segunda pelo marcador {1} e assim sucessivamente.
- Change of state: a única variável selecionada será representada pelo marcador {0}.
- Trigger: a variável "Trigger" é representada pelo marcador {0}, o "Limite superior" pelo marcador {1}, o limite inferior pelo marcador {2}, a primeira variável da lista pelo marcador {3}, a segunda pelo marcador {4} e assim sucessivamente. Como exemplo segue abaixo a configuração de um log de eventos do tipo "Trigger" e seu

arquivo de log.

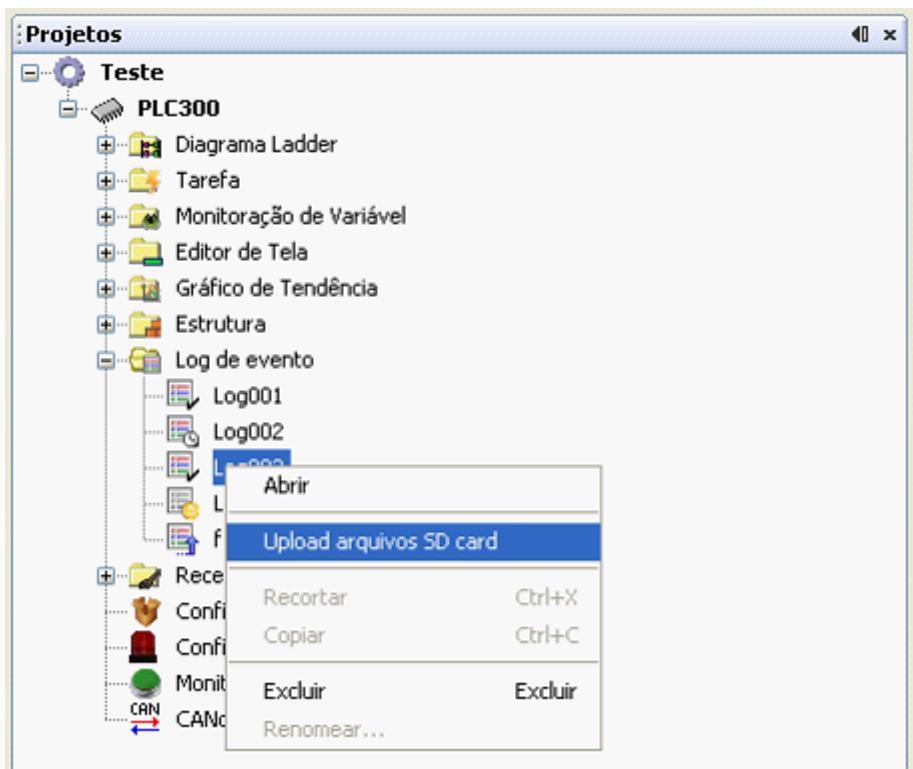


Upload de Arquivos de Log

Existem duas formas de realizar o upload dos arquivos de log: upload de todos os arquivos do projeto ou upload dos arquivos individualmente.



Para realizar o upload de todos os arquivos de log configurados no projeto, é preciso selecionar a pasta "Log de evento" com o botão direito e selecionar a opção "Upload arquivos SD card".

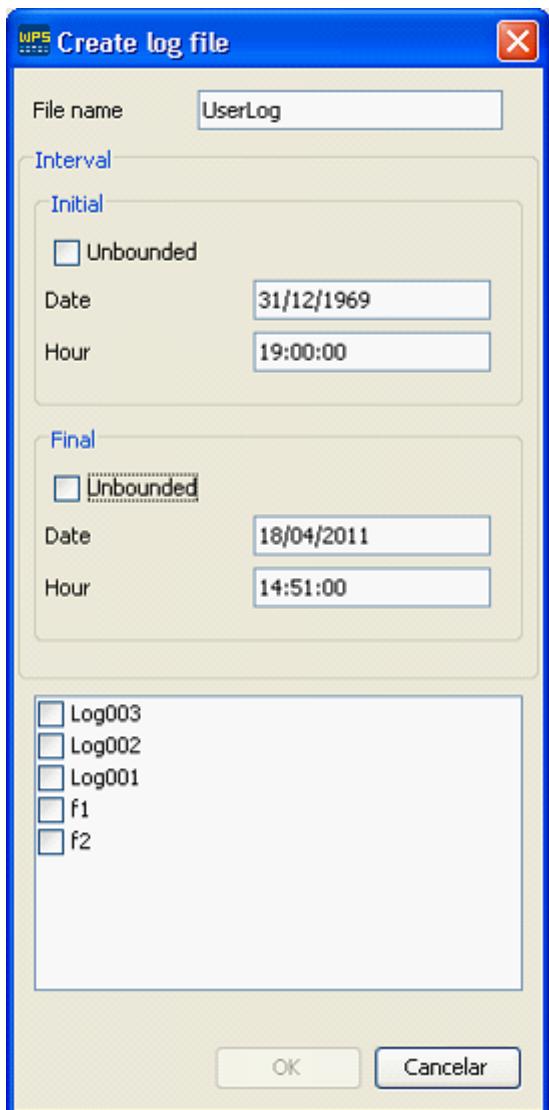


Para realizar o upload dos arquivos associados a apenas um arquivo de configuração de log de eventos é preciso clicar com o botão direito no arquivo de configuração do log de eventos e selecionar a opção "Upload arquivos SD card".

Atenção: Ao fazer o upload de arquivos os arquivos anteriores serão sobreescritos.

Criação de Arquivos de Log

Através da ferramenta de criação de arquivos de log é possível criar novos arquivos de log de eventos utilizando os arquivos de log previamente carregados no projeto. Com isso é possível definir o intervalo inicial e final do log e quais os arquivos de log que devem ser utilizados. Para iniciar a criação de um novo arquivo de log, é preciso clicar com o botão direito do mouse na pasta "Log de evento" e selecionar a opção "Criar arquivo de log". As opções para configuração do novo arquivo são as seguintes:



- Nome do arquivo: Nome que será utilizado no arquivo de log criado.

Atenção: Caso o nome já exista os dados serão sobreescritos!

- Intervalo inicial: Determina o dia e hora inicial dos logs do arquivo. Caso a opção "Ilimitado" seja selecionada, não haverá um valor mínimo para a data.
- Intervalo final: Determina o dia e hora final dos logs do arquivo. Caso a opção "Ilimitado" seja selecionada, não haverá um valor máximo para a data.
- Arquivos: Seleciona os arquivos de log que serão utilizados para gerar o novo arquivo de log.

6.9 Estruturas

6.9.1 Configuração de Estruturas

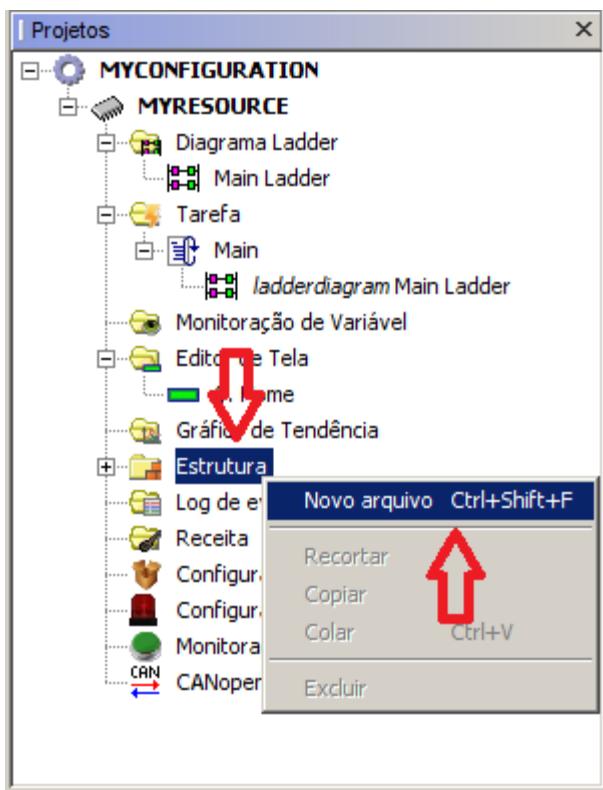
Estrutura

Estrutura consiste num agrupamento de dados utilizada para definir uma receita ou um objeto.

No programa Ladder é possível criar variáveis do tipo estrutura e utilizá-las nos blocos. Para acesso aos membros internos da estrutura, utiliza-se o '!' seguido do seu respectivo membro.

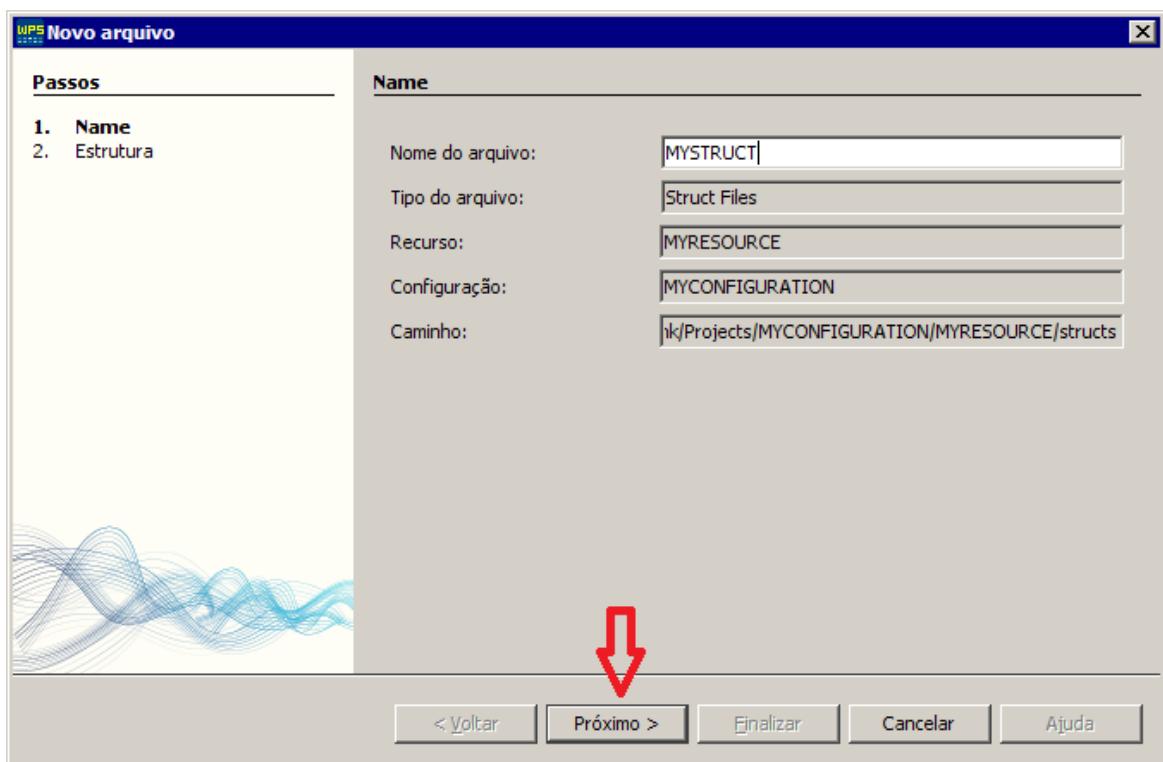
Criando uma Estrutura

Com o botão direito do mouse sobre a pasta "Estrutura", clique em "Novo arquivo".



(Figura 1: Criando uma Estrutura)

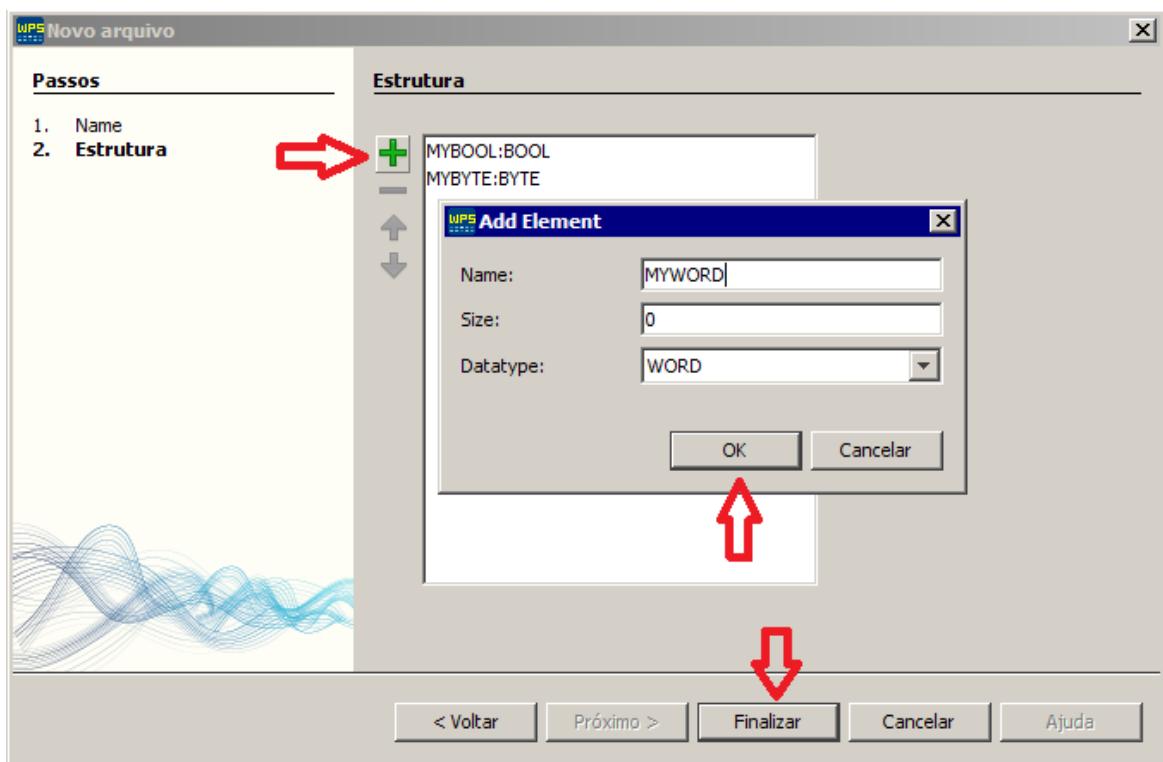
Defina o nome do arquivo e aperte o botão "Próximo".



(Figura 2: Definindo o nome da Estrutura)

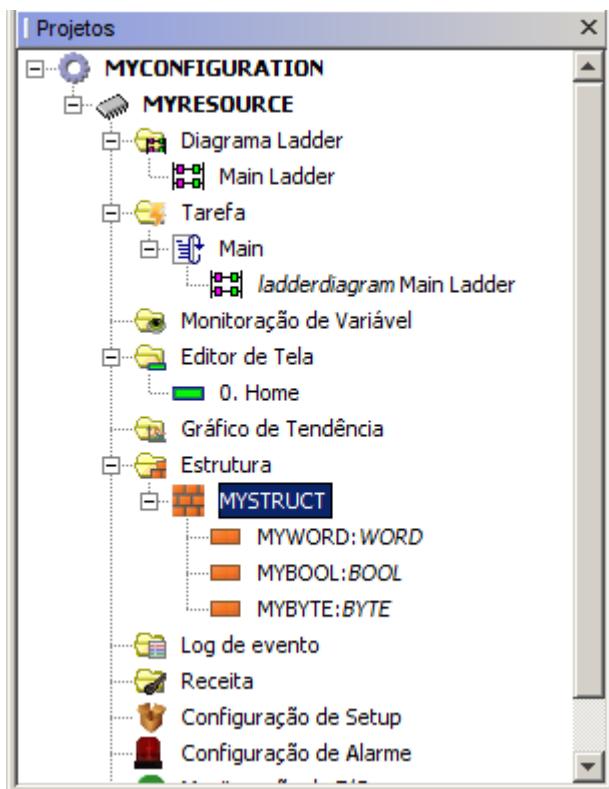
Criando uma Estrutura

Configure a estrutura utilizando os botões apresentados na figura abaixo.



(Figura 3: Editando a Estrutura)

Após concluir a edição da estrutura, clique no botão "Finalizar".



(Figura 4: Estrutura criada no projeto)

Editando uma Estrutura

Basta dar um duplo-clique na estrutura desejada, conforme a figura 4, e uma janela vai abrir conforme a figura 3, permitindo inserir novos dados, apagar ou mover de posição os dados.

6.10 Receitas

6.10.1 Configuração de Receitas

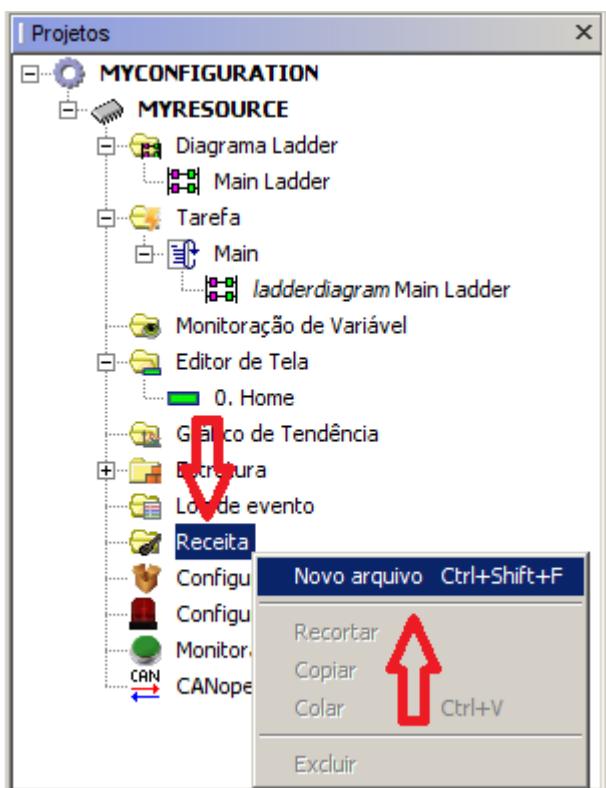
Receitas

Uma Receita constitui um conjunto de dados organizados na memória, que definem determinadas configurações para um processo, como por exemplo: tempo de cada etapa, valores mínimos e máximos, setpoint, número de repetições, etc.

Para se criar uma tabela de receitas, primeiramente é necessário definir os dados que a compõem, através de uma estrutura de dados.

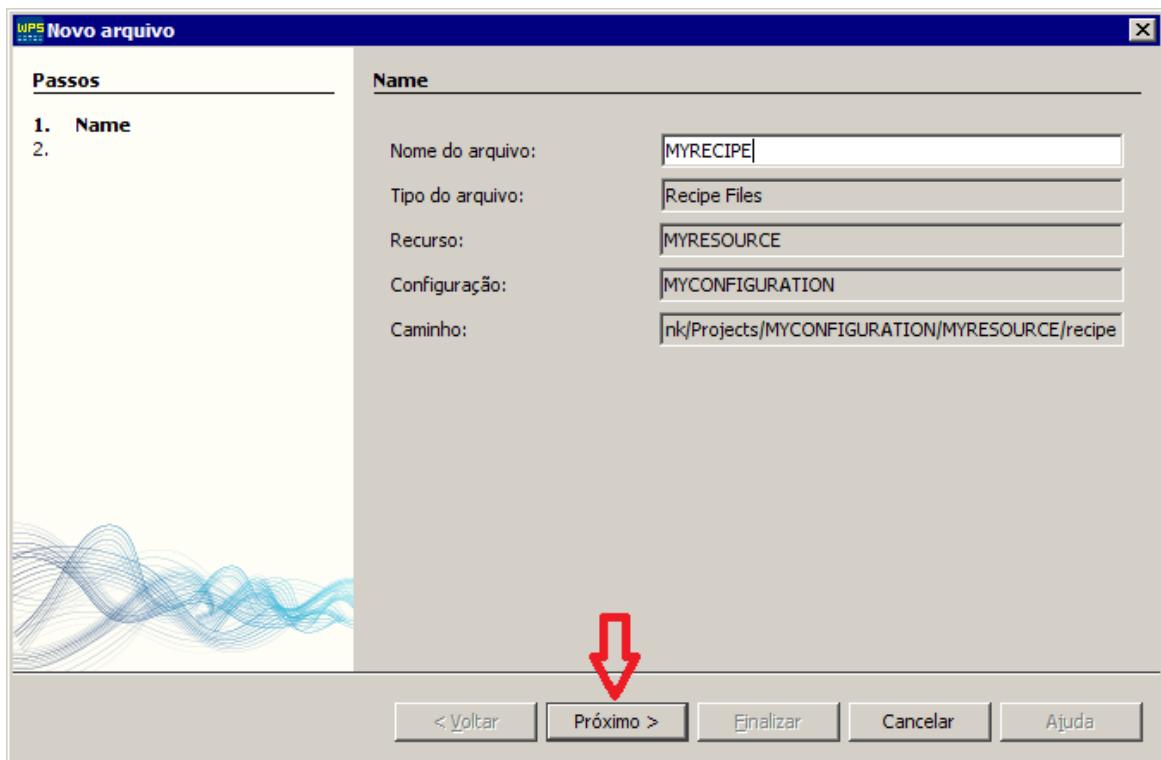
Criando uma Receita

Com o botão direito do mouse sobre a pasta "Receita", clique em "Novo arquivo".



(Figura 1: Criando uma Receita)

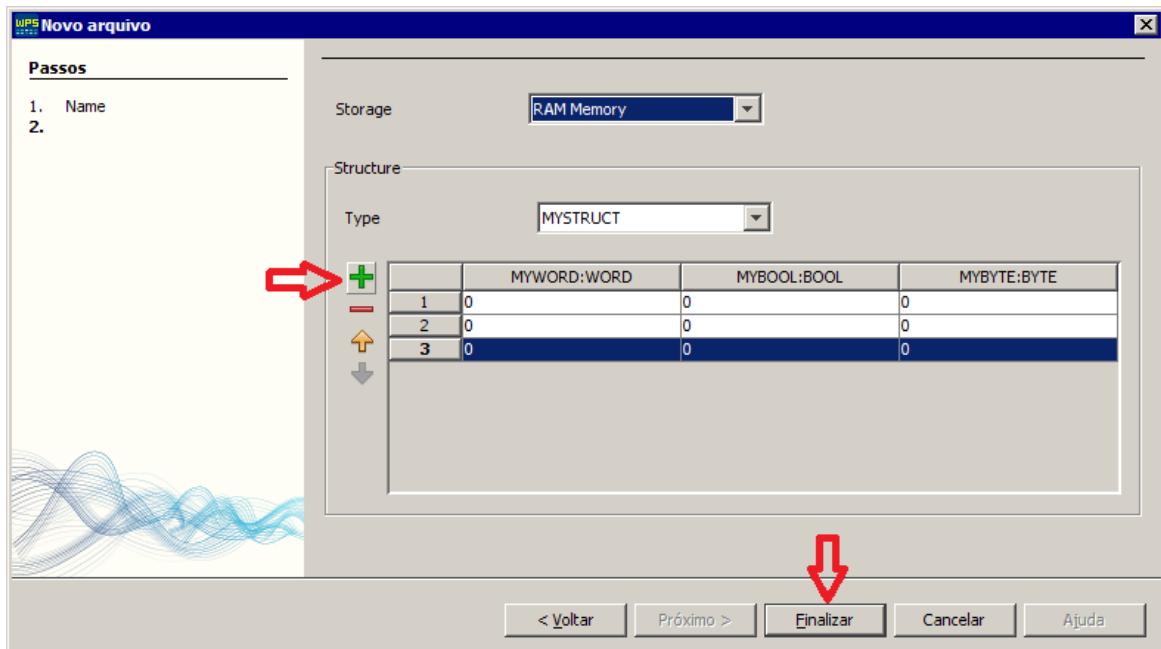
Defina o nome do arquivo e aperte o botão "Próximo".



(Figura 2: Definindo o nome da Receita)

Configurando a Receita

Configure a receita, configurando os campos conforme a figura abaixo.



(Figura 3: Editando a receita)

Campo "Storage"

- RAM Memory: os dados da receita são armazenados na memória RAM, tornando a sua manipulação mais rápida, porém com capacidade de armazenamento limitada a memória do PLC300.
- SD Card: os dados da receita são armazenados em arquivos no cartão de memória, tornando a sua manipulação bem mais lenta, mas com capacidade de armazenamento de acordo com o cartão SD.

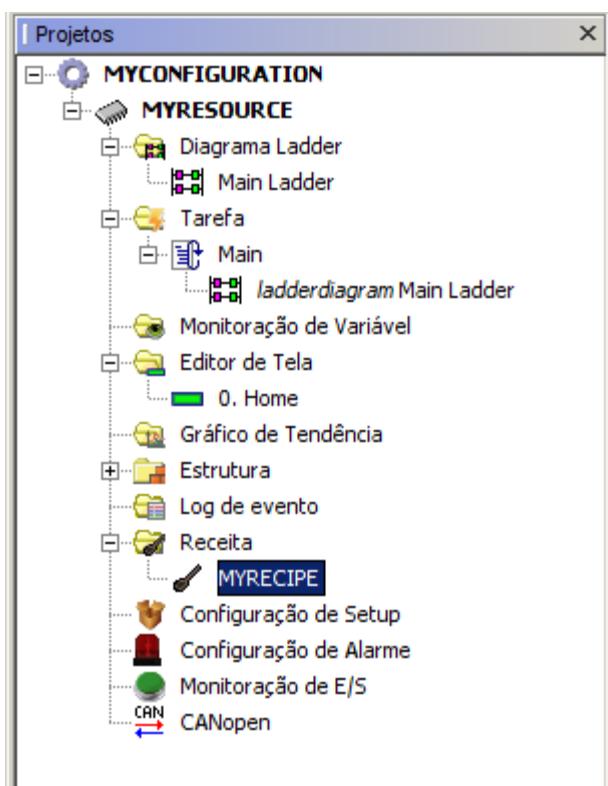
Campo "Type"

- Permite selecionar todas as estruturas que foram definidas pelo usuário no projeto.

Campo "Tabela de Dados":

- As colunas representam os elementos da estrutura. Observe que além do nome membro, também é apresentado após os ':' o seu respectivo tipo de dado;
- As linhas representam cada receita.

Após concluir a edição da estrutura, clique no botão "Finalizar".



(Figura 4: Receita criada no Projeto)

Editando uma Receita

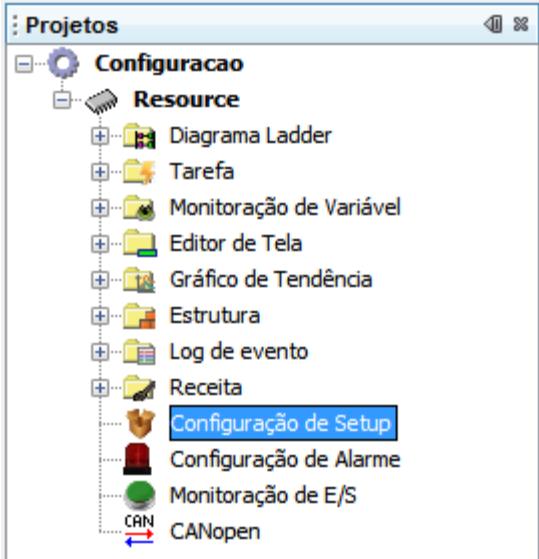
Basta dar um duplo-clique na estrutura desejada, conforme a figura 4, e uma janela vai abrir conforme a figura 3, permitindo inserir novos dados, apagar ou mover de posição os dados.

6.11 Setup

6.11.1 Configuração de Setup

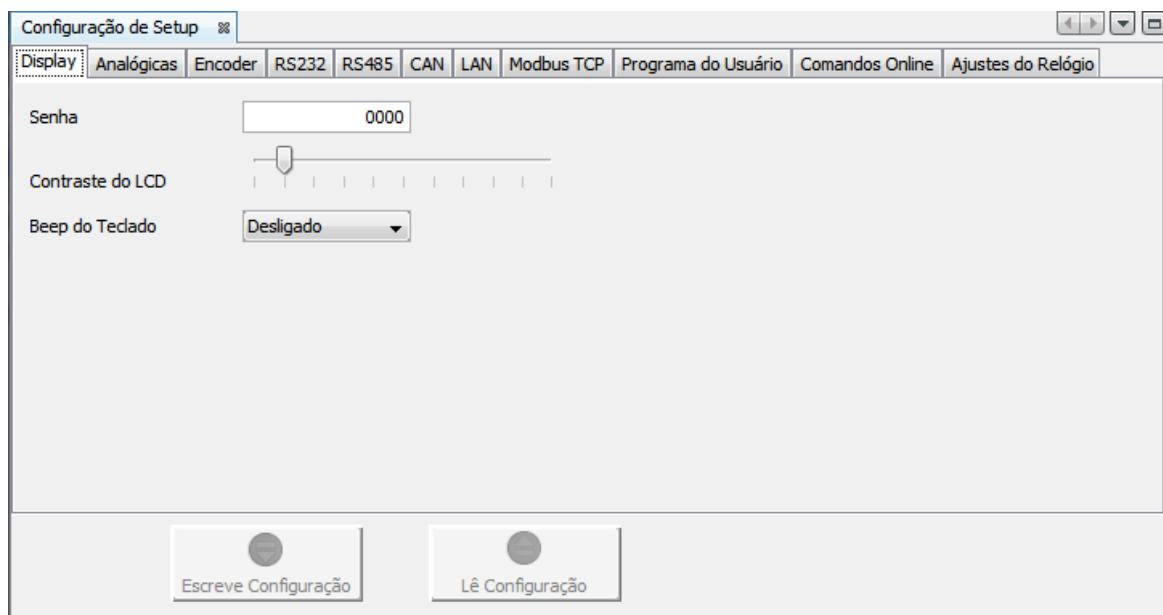
Visão Geral

A configuração do setup do PLC300 é acessada através do clique duplo no atalho disponível no recurso conforme figura a seguir.



 Atenção! Ao criar um novo recurso no WPS os valores da configuração de setup serão os valores padrões do PLC300, recomendamos a revisão desses valores conforme a necessidade da aplicação e o envio desses ajustes conforme explicação a seguir.

A janela de configuração de setup é dividida em abas e possui dois botões para acesso ao PLC300 conforme figura a seguir.



O botões de "Escreve Configuração" e "Lê Configuração" só estarão ativos quando o WPS estiver conectado ao PLC300 através do comando Conectar Dispositivo F12".

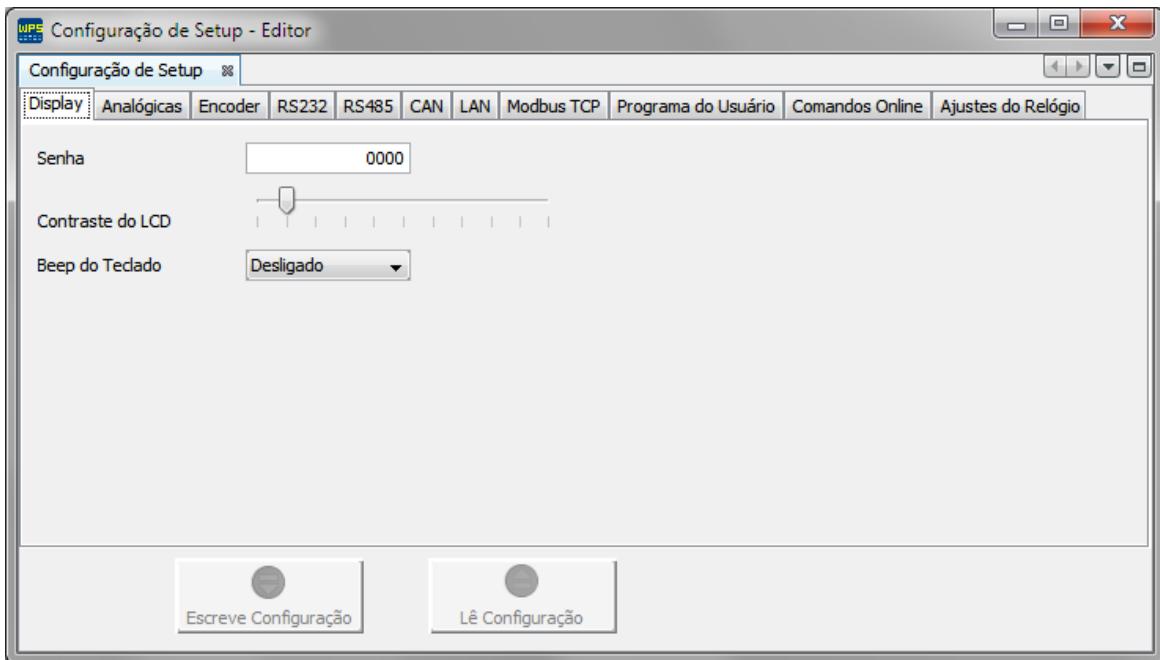
Função do Botões

Escreve Configuração: Envia todas as configurações ajustadas nas telas da configuração de setup para o equipamento.

Lê Configuração: Recebe todas as configurações ajustadas no equipamento e configura as telas conforme os valores recebidos.

6.11.2 Janelas Configuração de Setup

6.11.2.1 Display

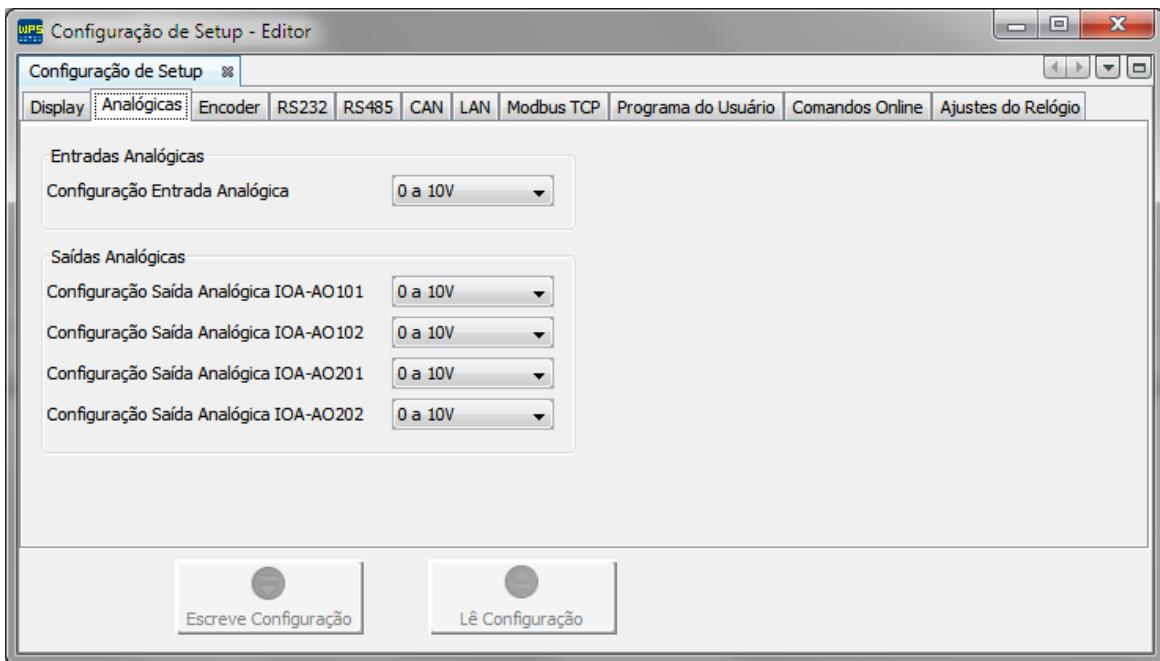


Senha: Permite alterar a senha da função setup do PLC300. A senha padrão é '0000'. A nova senha deve ser um número com quatro algarismos.

Contraste LCD: ajuste do contraste do LCD do PLC300.

Beep do Teclado: Habilita o beep das teclas do PLC300.

6.11.2.2 Analógicas

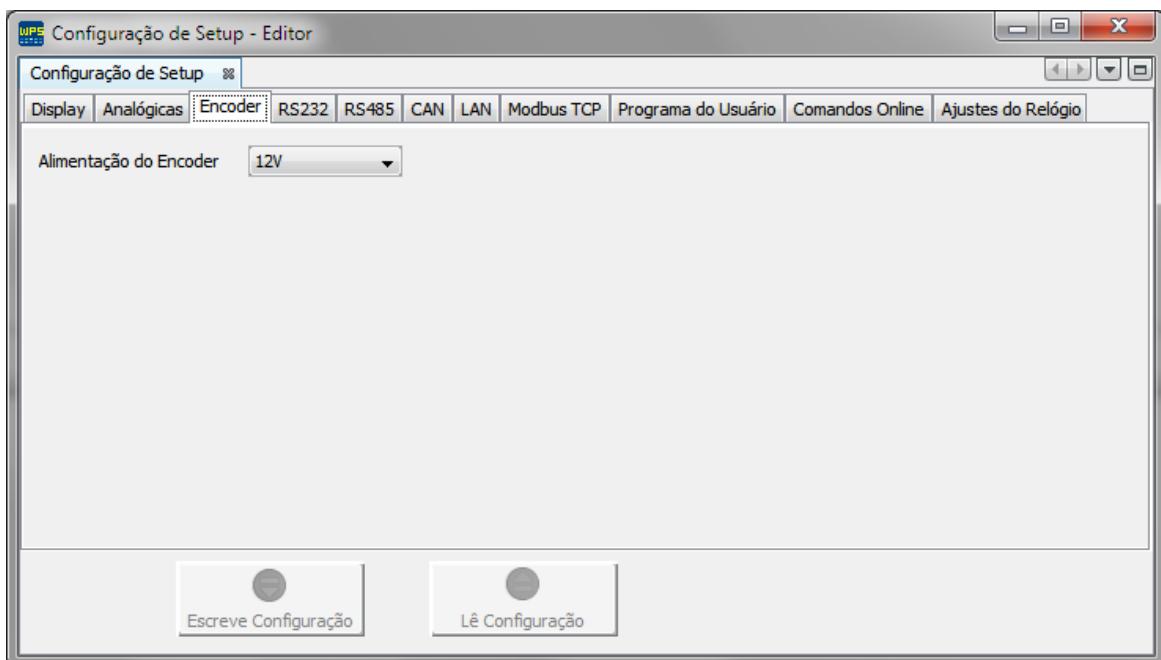


Entradas Analógicas: Seleciona um dos três modos de operação da entrada analógica AI1 do PLC300: 'Tensão 0 a 10 V', 'Corrente 0 a 20 mA' ou 'Corrente 4 a 20 mA'.

* Na opção 4 a 20 mA, o valor que o ladder enxerga é um valor proporcional, normalizado, ou seja 4 a 20 mA → 0 a 32767.

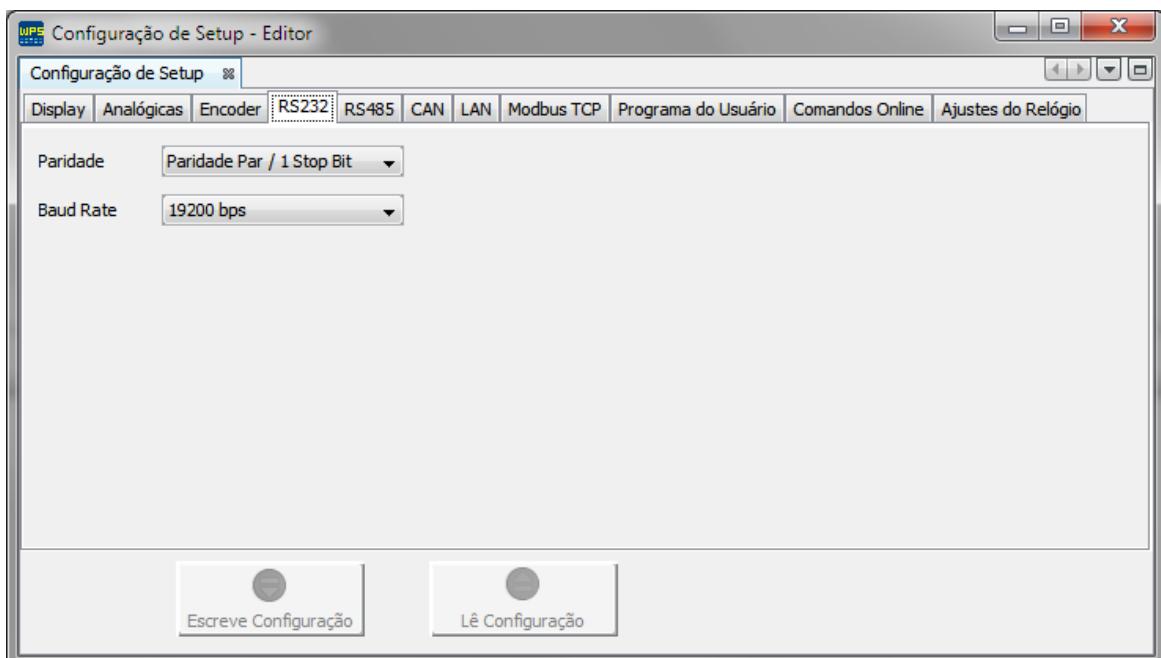
Saídas Analógicas: Seleciona um dos quatro modos de operação das saídas analógicas (AOs) dos acessórios IOA instalados no PLC300, sendo que as AOs 101 e 102, são as AOs 1 e 2 do cartão IOA instalado no slot 1 do PLC300 e as AOs 201 e 202, são as AOs 1 e 2 do cartão IOA instalado no slot 2.

6.11.2.3 Encoder



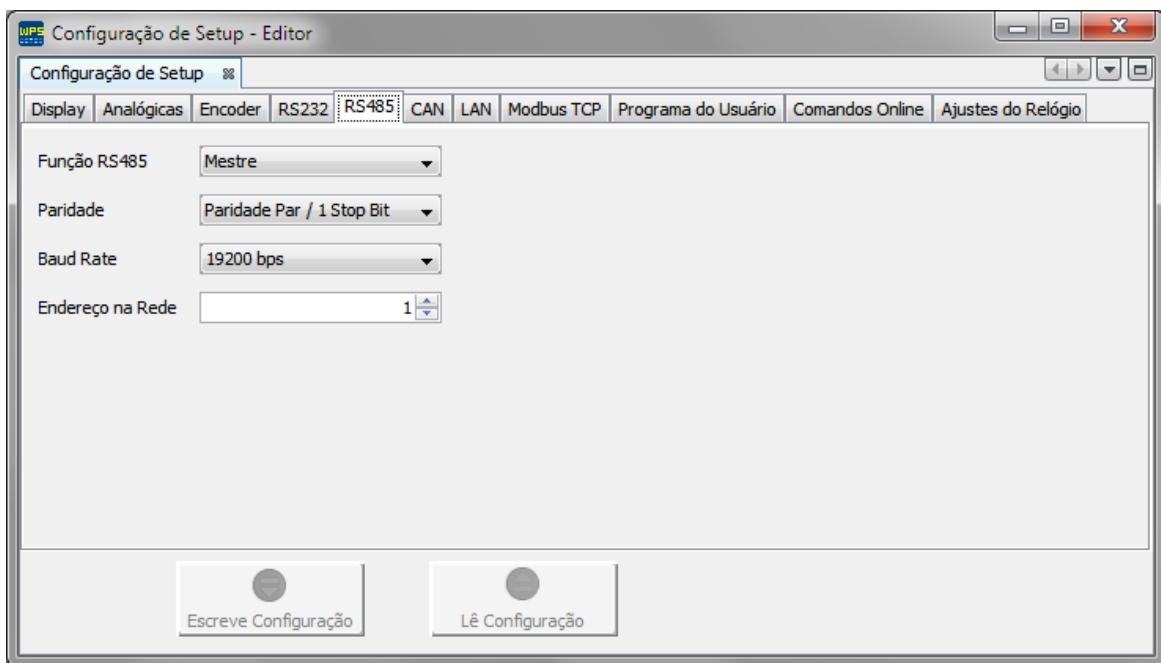
Seleciona a tensão de alimentação do encoder do PLC300 entre 5 ou 12V.

6.11.2.4 RS232



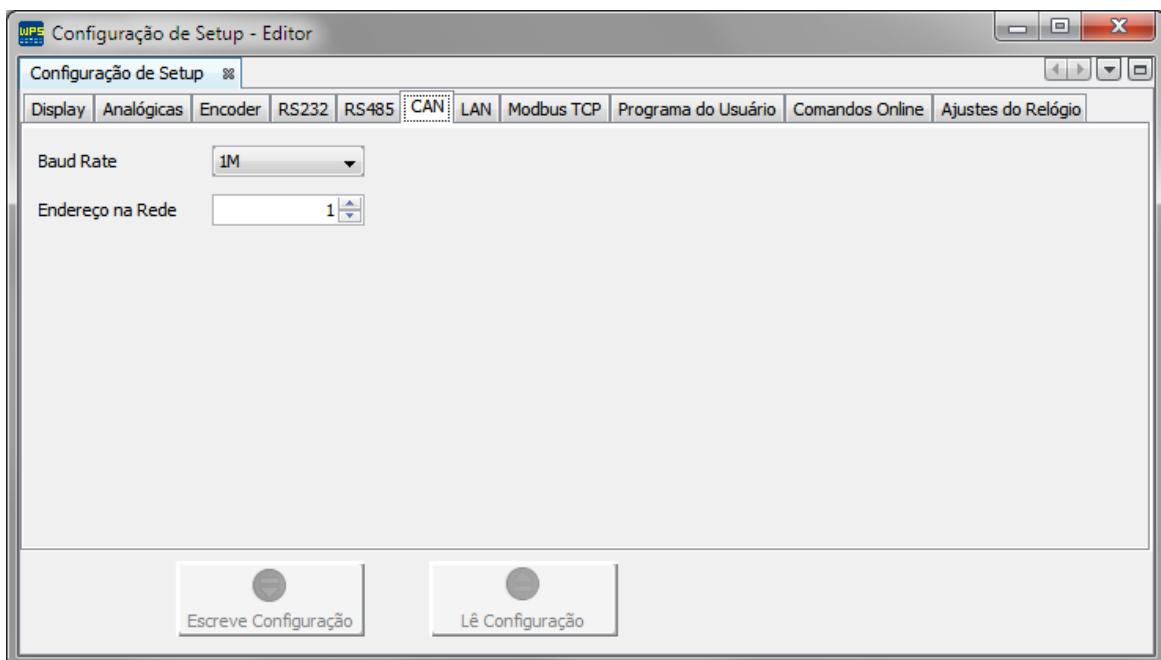
É possível configurar o baud rate, a paridade e o número de stop bits da interface serial RS232 do PLC300.

6.11.2.5 RS485



É possível configurar o baud rate, a paridade o número de stop bits, o modo (mestre/escravo) e o endereço do PLC300 numa rede ModbusRTU, através interface serial RS485 do PLC300.

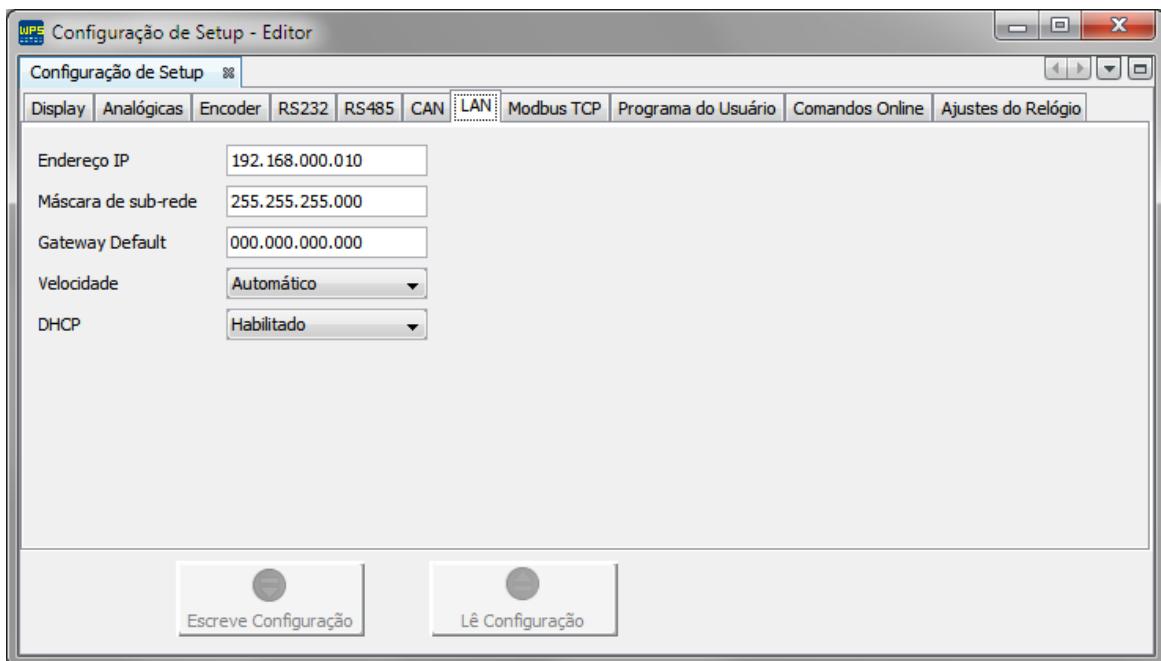
6.11.2.6 CAN



É possível configurar o baud rate e o endereço do PLC300 numa rede CANopen,

através da interface CAN do PLC300.

6.11.2.7 LAN



É possível configurar endereço IP, máscara de sub-rede, gateway padrão, DHCP, velocidade e modo duplex do PLC300 numa rede Ethernet.

Endereço IP: 4 bytes de endereço que identificam o PLC300 na rede IP;

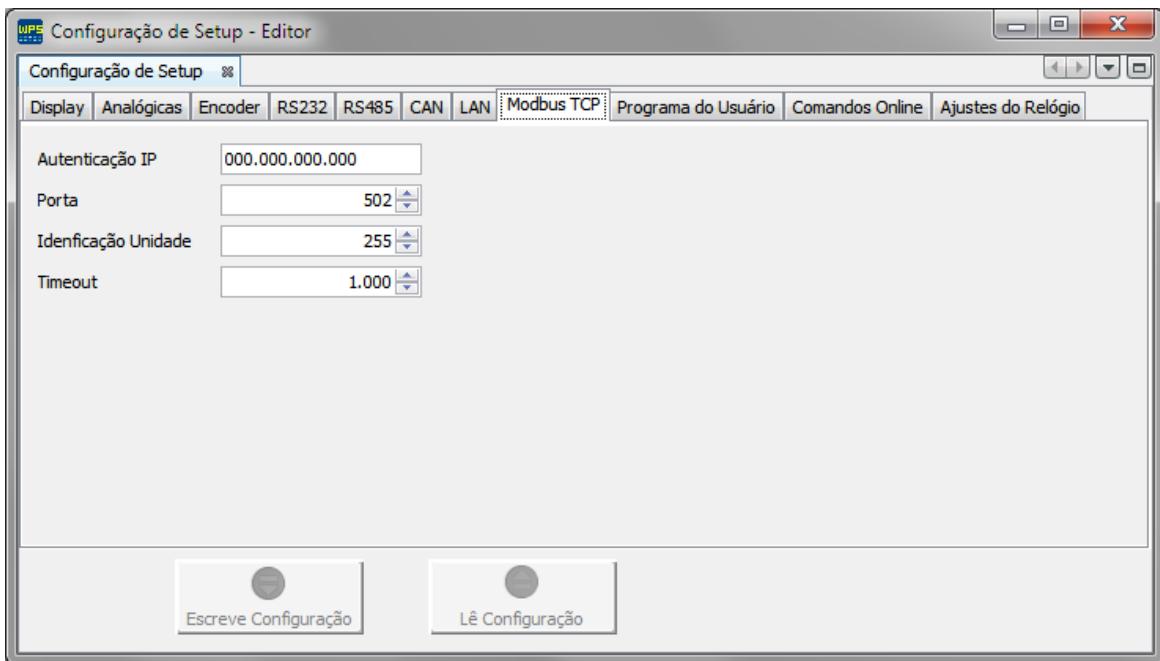
Máscara de Sub-rede: 4 bytes que identificam a sub-rede ao qual pertence o PLC300 na rede IP;

Gateway: 4 bytes de endereço que identificam o gateway padrão para acesso a outras sub-redes na rede IP;

DHCP: Desabilitado, Habilitado;

Speed/Duplex: Auto, 10MBps Full Duplex, 10MBps Half Duplex, 100MBps Full Duplex, 100MBps Half Duplex;

6.11.2.8 Modbus TCP



É possível configurar porta TCP, Unit ID, autenticação de IP e timeout do Gateway Modbus TCP/RTU do PLC300 numa rede Ethernet utilizando o protocolo ModbusTCP.

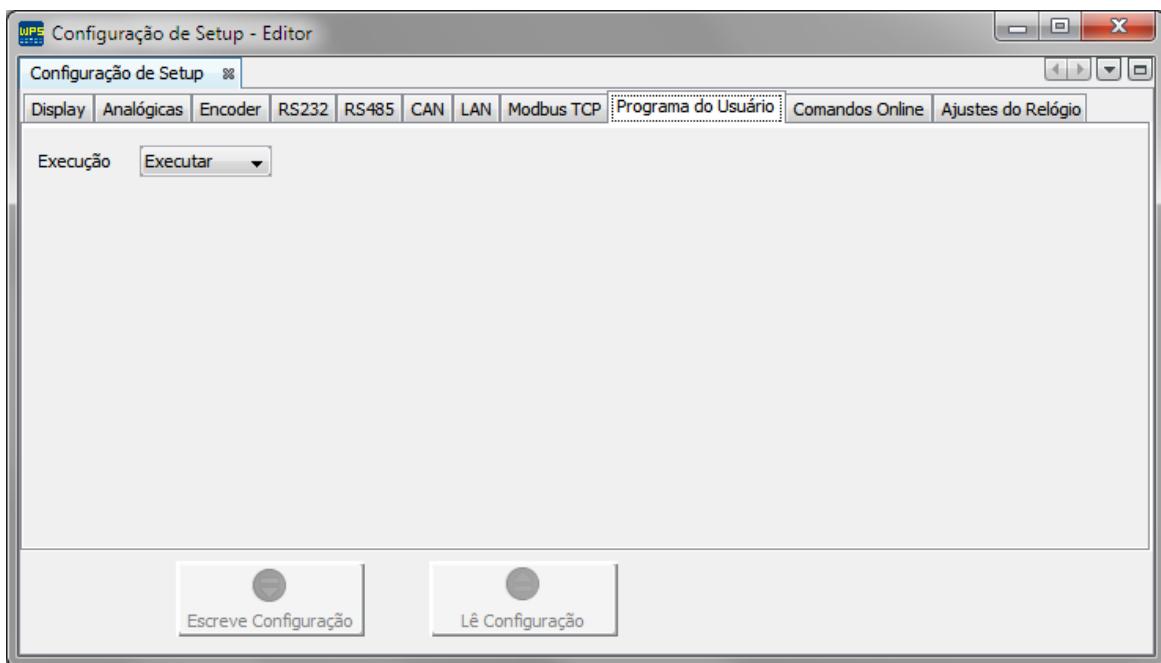
Autenticação IP: 4 bytes de endereço que identificam o único endereço IP remoto que pode se conectar ao PLC300. Todos os campos em zero desabilitam a autenticação de IP e qualquer endereço remoto pode se conectar ao PLC300;

Porta TCP: 0 a 65535,

Unit ID: 1 a 255,

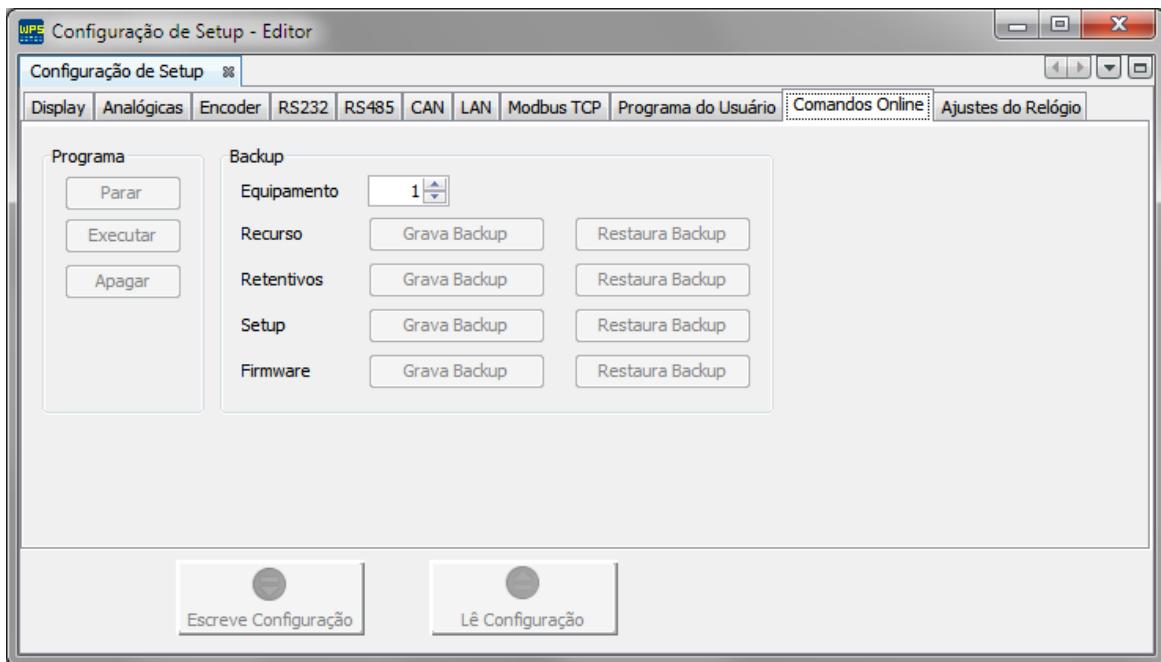
Gateway Timeout: 20 a 5000 ms,

6.11.2.9 Programa do Usuário



Permite parar/executar ou apagar o programa do usuário. Essa configuração não é online e é transmitida através do botão "Escreve Configuração".

6.11.2.10 Comandos Online



Os comandos online só estarão ativos quando o WPS estiver conectado ao PLC300 através do comando " Conectar Dispositivo F12".

Programa: Permite parar/executar ou apagar o programa do usuário.

Backup: Permite gravar (PLC300 -> SD card) ou restaurar (SD card -> PLC300) um recurso, configurações do setup e firmware no SD card conectado ao PLC300.

O cartão de memória deve ser do tipo SD com formatação FAT32. Quanto mais rápido o cartão (classe do cartão), menor o tempo de gravação. Isso é importante no caso de gravações periódicas de log de variáveis por exemplo.

O 'Equipamento' , ou, número do equipamento, indica a pasta onde serão gravados, ou de onde serão carregados os arquivos da função selecionada. Pode variar entre 0001 e 9999.

Ao selecionar as funções de gravação de recurso, setup, firmware ou geral, os arquivos são gravados nas seguintes pastas, respectivamente, de acordo com o número do equipamento.

```
\PLC300\0001\Resource  
\PLC300\0001\Setup  
\PLC300\0001\Firmware
```



Atenção! Os arquivos existentes nas pastas de destinos serão sobreescritos ao executar as funções de gravação.

Em caso de carga de firmware, após a confirmação e carga, o PLC é reinicializado.

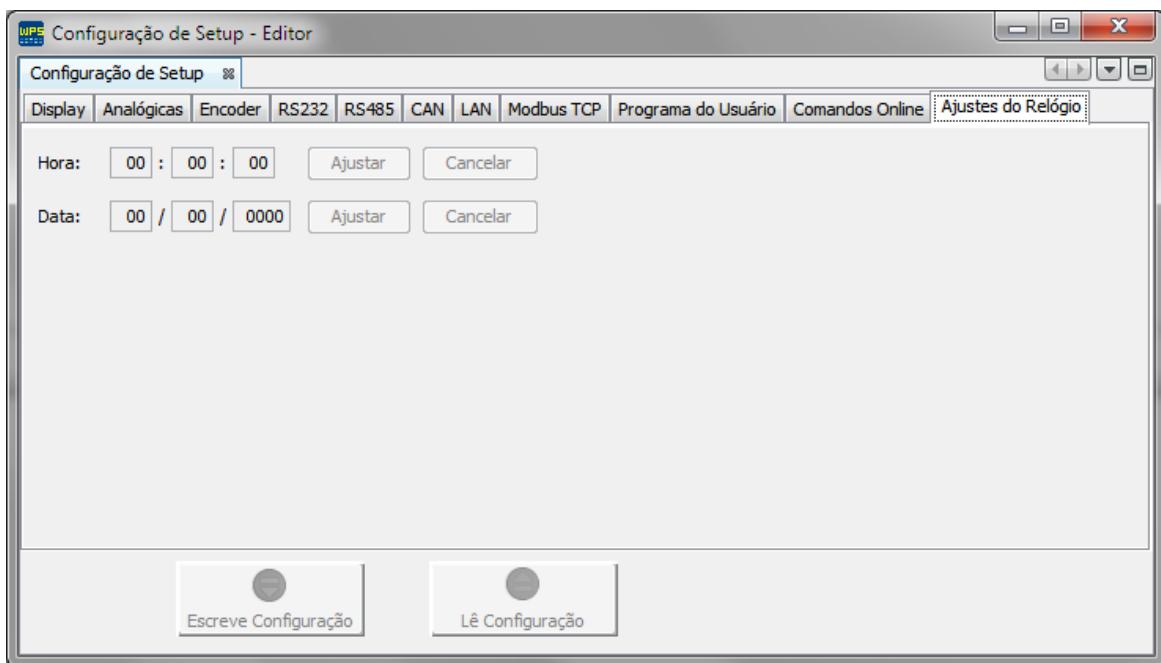
O backup geral nada mais é do que todas as operações em sequência, ou seja, firmware, setup e recurso.

No caso de carga backup geral, primeiro o firmware é carregado, então o PLC é reinicializado. Antes de iniciar o programa do usuário, o setup e o recurso são carregados, mostrando uma tela com o resultado da carga, sucesso ou erro.

Possíveis causas de erro:

- Arquivos inválidos ou inexistentes;
- Revisão dos arquivos incompatível com a versão de firmware do PLC300;
- Pasta com o número do equipamento diferente do configurado no setup;
- SD card sem espaço para escrita ou com proteção de escrita acionada;
- Versão do bootloader não permite carregar firmware através do SD card, verificar se marcador %SW3016 (versão do bootloader) é igual ou superior a 200.

6.11.2.11 Ajustes do Relógio



Os ajustes do relógio só estarão ativos quando o WPS estiver conectado ao PLC300 através do comando "🔧 Conectar Dispositivo F12".

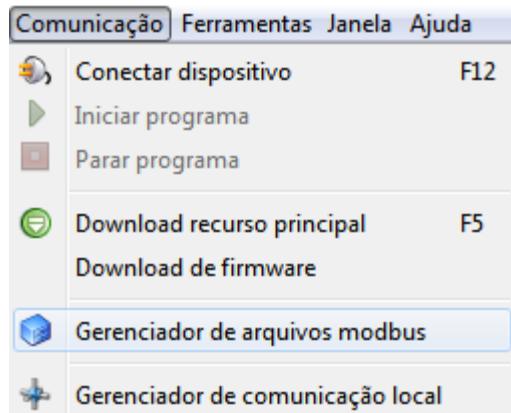
Permite o ajuste do relógio RTC do PLC300.

6.12 SD Card

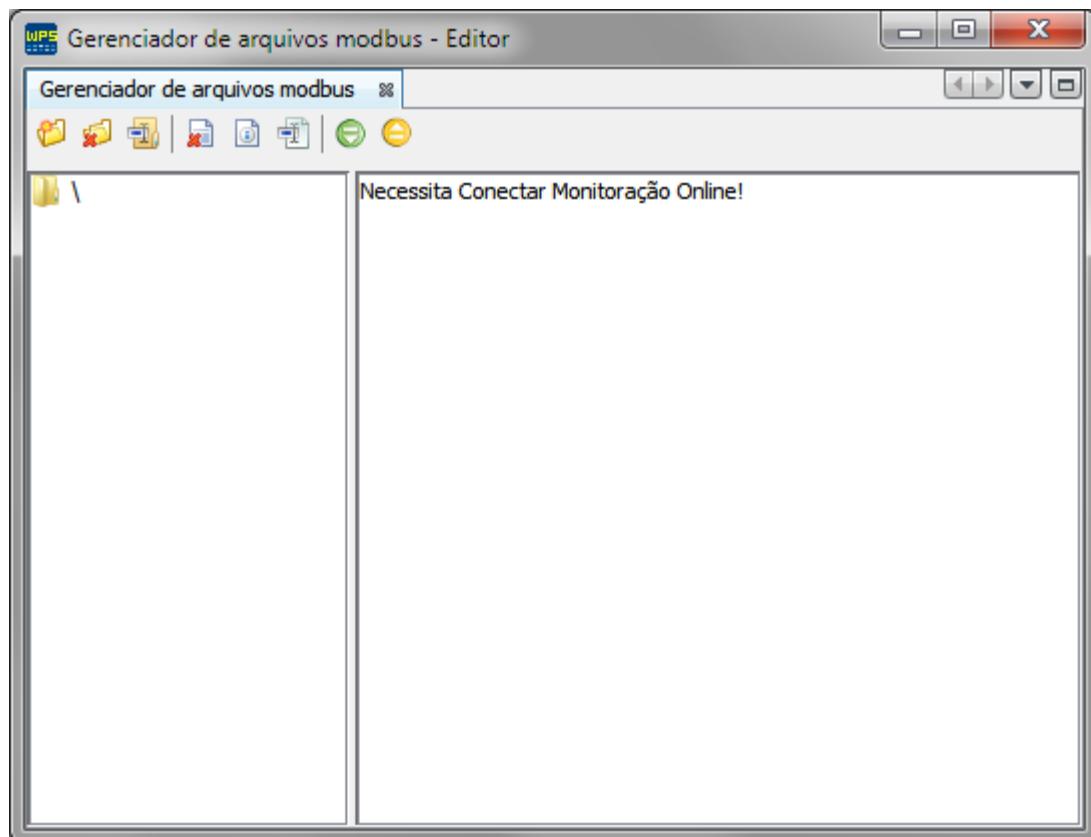
6.12.1 Gerenciador de arquivos modbus

Visão Geral

O gerenciador de arquivos modbus do PLC300 é acessada através do menu comunicação conforme figura a seguir.



A janela do gerenciador de arquivos modbus é composta das seguintes partes conforme figura a seguir.



Parte superior: Toolbar com todas as ferramentas do gerenciador de arquivos

Lado esquerdo: Lista de pastas do SD Card.

Lado direito: Lista de arquivos da pasta selecionada.

Só será possível visualizar os arquivos e pastas do SD Card quando o WPS estiver conectado ao PLC300 através do comando "  Conectar Dispositivo F12".

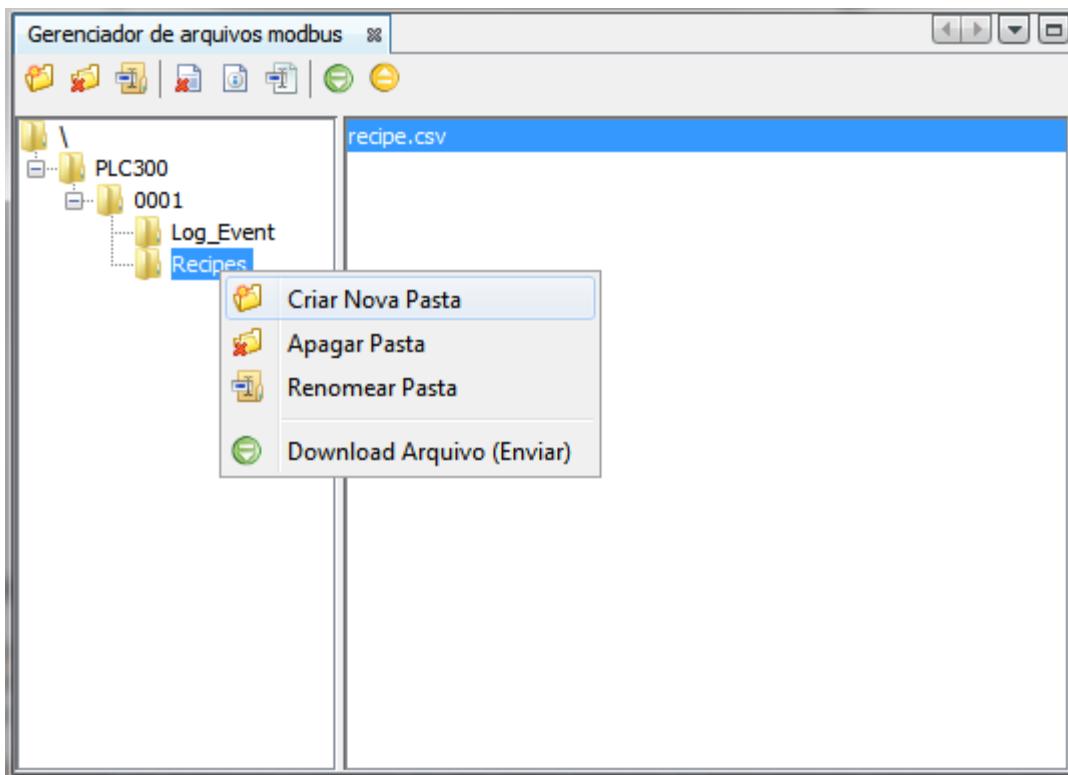
Toolbar

-  Cria nova pasta
-  Apaga pasta selecionada
-  Renomeia pasta selecionada
-  Apaga arquivo selecionado
-  Exibe propriedades do arquivo selecionado (caminho, tamanho e data)
-  Renomeia arquivo selecionado
-  Envia arquivo para o PLC300
-  Recebe arquivo selecionado do PLC300

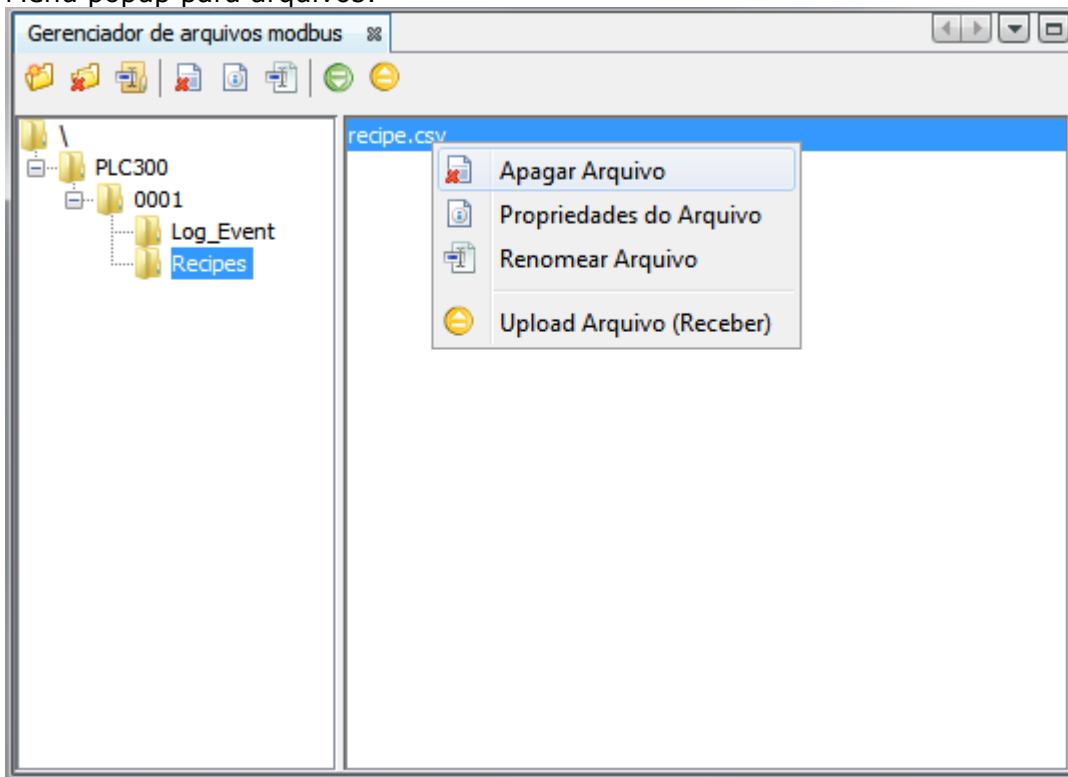
Menus popups

Para acessar o menu popup basta clicar com o botão direito sobre uma pasta ou sobre um arquivo e o mesmo será mostrado conforme figura a seguir.

Menu popup para pastas:



Menu popup para arquivos:



6.13 Componentes e Blocos

6.13.1 Contact

6.13.1.1 NOCONTACT

Figura



Configuração do Bloco

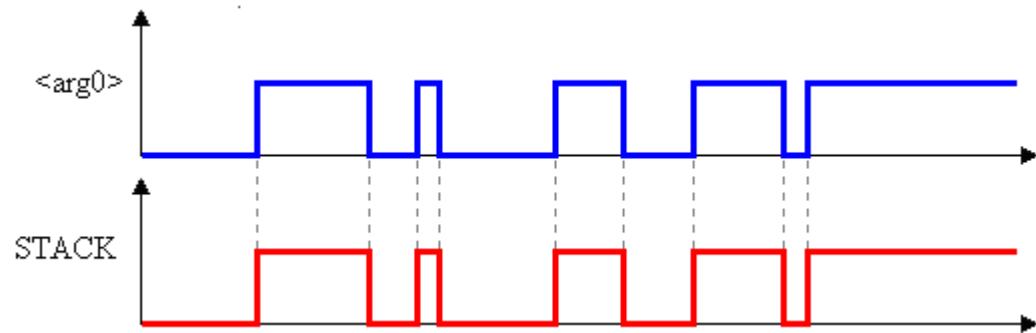
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega para o STACK o valor de <arg0>.

Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.1.2 NCCONTACT

Figura



Configuração do Bloco

<arg0> - VAR_IN: inserir uma variável (tag).

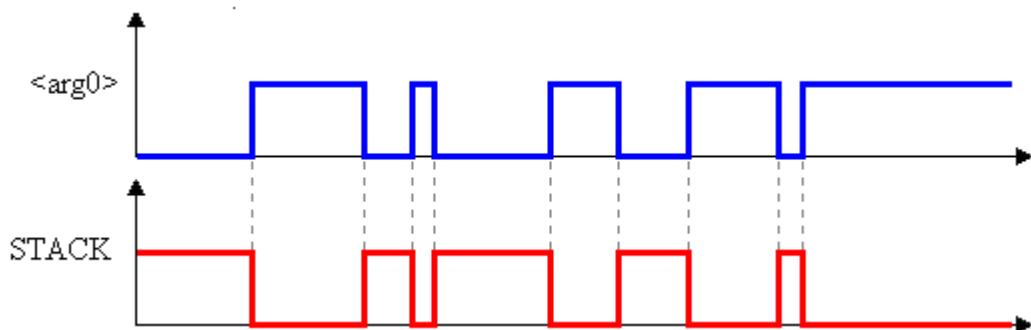
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega para o STACK o valor negado de <arg0>.

Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento

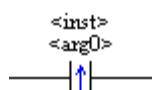


Disponível

- PLC300 : a partir da versão inicial

6.13.1.3 PTSCONTACT

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

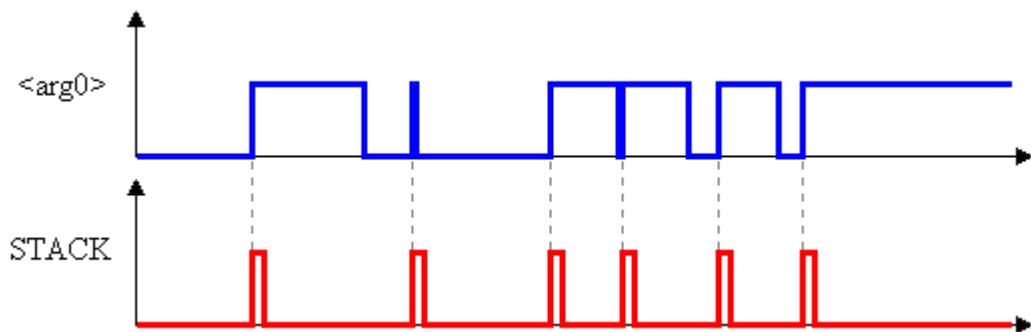
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega 1 para o STACK por 1 SCAN somente na transição de 0 para 1 de <arg0>, ou seja, na borda de subida do STACK.

Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.1.4 NTSCONTACT

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

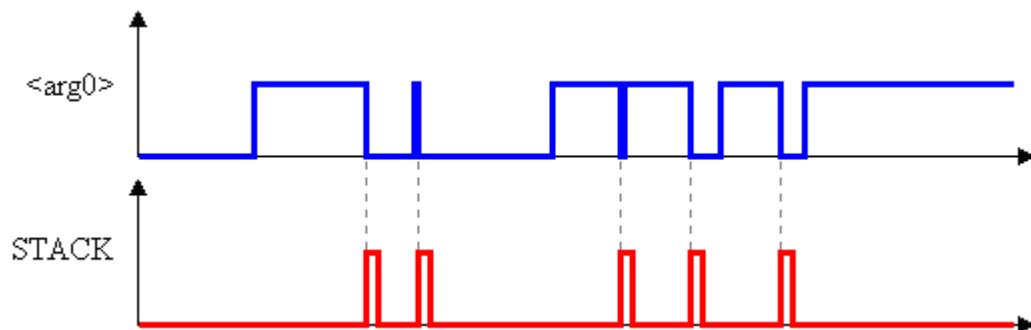
<arg0> - VAR_IN: inserir uma variável (tag).
Tipo de dado: BOOL.

Descrição de Funcionamento

Carrega 1 para o STACK por 1 SCAN somente na transição de 1 para 0 de <arg0>, ou seja, na borda de descida do STACK.

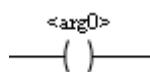
Veja também como funcionam as associações série e paralelo em LD com os elementos Contatos.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.2 Coil**6.13.2.1 DIRECTCOIL****Figura****Configuração do Bloco**

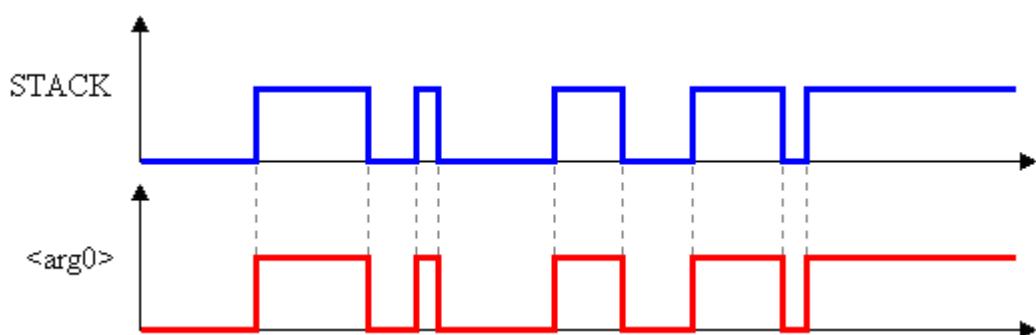
<arg0> - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

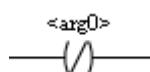
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Salva o valor do STACK em <arg0>.

Diagrama de Funcionamento**Disponível**

- PLC300 : a partir da versão inicial

6.13.2.2 INVERTEDCOIL**Figura****Configuração do Bloco**

<arg0> - VAR_OUT: inserir uma variável (tag).

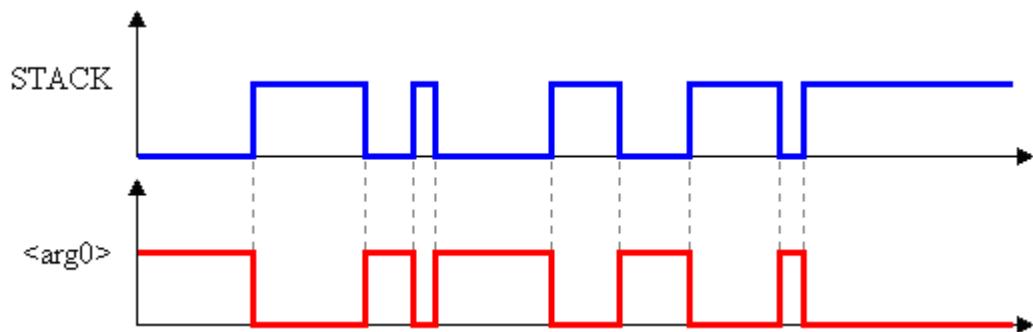
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Salva o valor negado do STACK em <arg0>.

Diagrama de Funcionamento

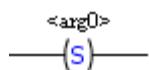


Disponível

- PLC300 : a partir da versão inicial

6.13.2.3 SETCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

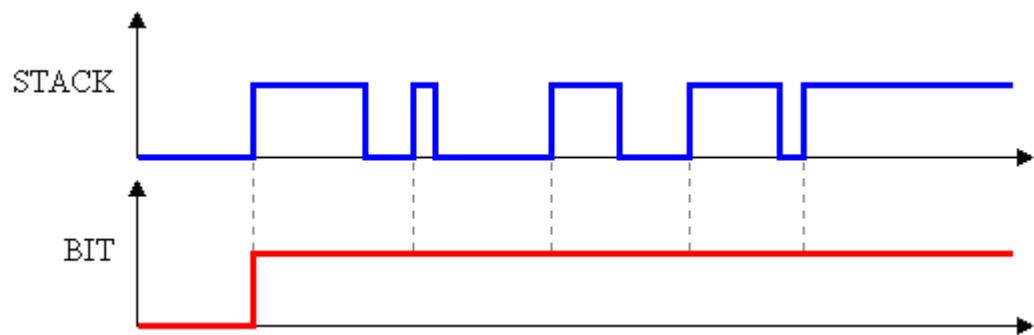
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Quando o valor do STACK for 1, <arg0> vai para 1.

Diagrama de Funcionamento

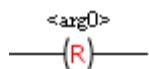


Disponível

- PLC300 : a partir da versão inicial

6.13.2.4 RESETCOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

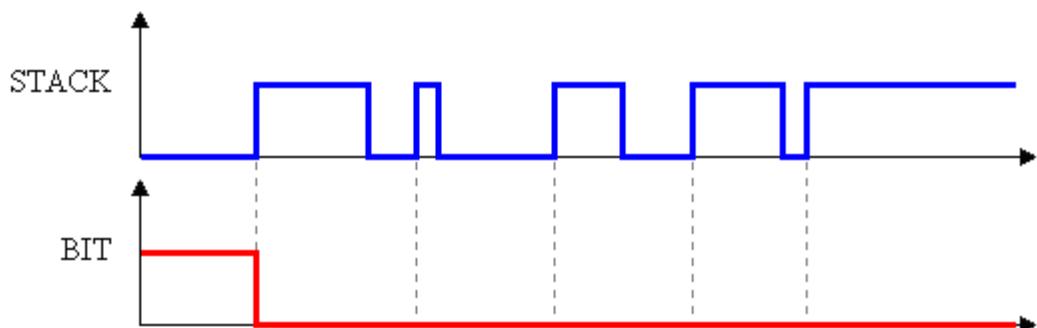
Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Quando o valor do STACK for 1, <arg0> vai para 0.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.2.5 TOGGLECOIL

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

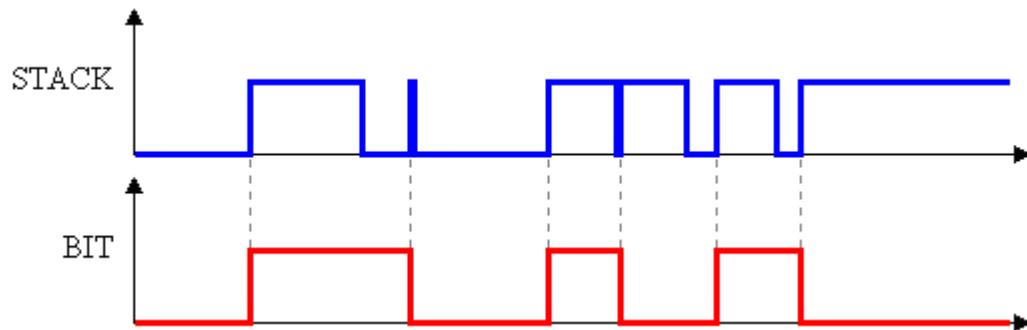
<arg0> - VAR_IN: inserir uma variável (tag).

Tipo de dado: BOOL.

Descrição de Funcionamento

A cada transição de 0 para 1 no STACK, a <arg0> inverte seu valor.

Diagrama de Funcionamento

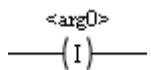


Disponível

- PLC300 : a partir da versão inicial

6.13.2.6 IMMEDIATECOIL

Figura



Configuração do Bloco

<arg0> - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

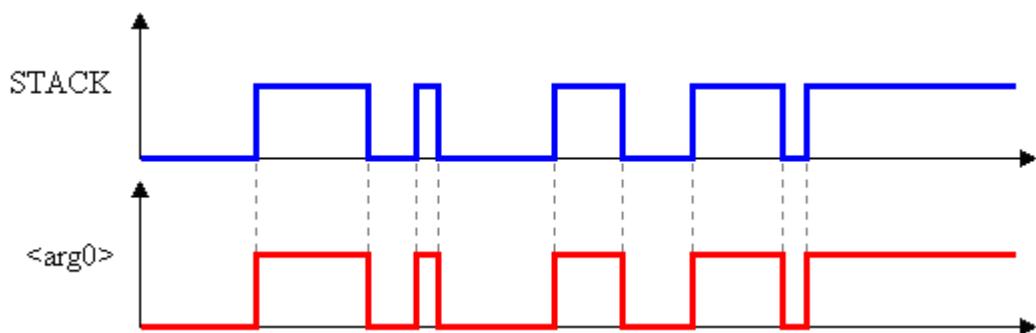
Nota: A variável deve ser uma saída digital do PLC300, excluindo-se as saídas digitais das expansões, isto é, apenas são válidas as DO's de 1 até 9.

Descrição de Funcionamento

Salva o valor do STACK em <arg0>. Diferentemente do DIRECTCOIL, este bloco seta as saídas digitais diretamente, sem aguardar o fim do SCAN do programa. Seu principal uso é nos POU's chamados pelas TASK's single, evento externo, interval e contagem.

A imagem da saída digital também é atualizada imediatamente.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão 1.20

6.13.3 Logic Bit

6.13.3.1 RESETBIT

Figura



Configuração do Bloco

<arg0>: "Data" - VAR_IN_OUT: inserir uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
 Nota: A variável tem que ter permissão de escrita.

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
 Tipos de dados: BYTE ou USINT.
 Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT ou SINT.
 Faixa de Valores: 0 a 15 se "Data" for WORD, UINT ou INT.

Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT ou DINT.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "Done" normalmente retorna o STACK o valor de "EN". Isto somente não ocorre quando "Position" for configurado de maneira incorreta, mantendo neste caso, "Data" inalterado.

Em execução ("EN"=1), o bit configurado em "Position" da variável em "Data" vai para 0.

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo1 ("EN"=1)

"Position" = 2
"Data" = 15 = 2#1111 (antes da execução)
"Data" = 11 = 2#1011 (depois da execução)

Exemplo2 ("EN"=1)

"Position" = 0
"Data" = 2 = 2#0010 (antes da execução)
"Data" = 2 = 2#0010 (depois da execução)

Disponível

- PLC300 : a partir da versão inicial

6.13.3.2 SETBIT

Figura



Configuração do Bloco

<arg0>: "Data" - VAR_IN_OUT: inserir uma variável (tag)
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT
Nota: A variável tem que ter permissão de escrita.

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
Tipos de dados: BYTE ou USINT.
Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT e SINT.
Faixa de Valores: 0 a 15 se "Data" for WORD, UINT e INT.

Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT, DINT.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "Done" normalmente retorna o STACK o valor de "EN". Isto somente não ocorre quando "Position" for configurado de maneira incorreta, mantendo neste caso, "Data" inalterado.

Em execução ("EN"=1), o bit configurado em "Position" da variável em "Data" vai para 1.

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo1 ("EN"=1)

"Position" = 2
"Data" = 15 = 2#1111 (antes da execução)
"Data" = 15 = 2#1111 (depois da execução)

Exemplo2 ("EN"=1)

"Position" = 0
"Data" = 2 = 2#0010 (antes da execução)
"Data" = 3 = 2#0011 (depois da execução)

Disponível

- PLC300 : a partir da versão inicial

6.13.3.3 TESTBIT

Figura



Configuração do Bloco

<arg0>: "Data" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT

<arg1>: "Position" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

Faixa de Valores: 0 a 7 se "Data" for BYTE, USINT e SINT.

Faixa de Valores: 0 a 15 se "Data" for WORD, UINT e INT.

Faixa de Valores: 0 a 31 se "Data" for DWORD, UDINT, DINT.

Nota: "Position" em 0 indica o bit menos significativo da variável.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega para a saída "Q" o valor do bit configurado em "Position" da variável em "Data".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Position" = 2

"Data" = 15 = 2#1111 e "Q" = 1

Exemplo2: ("EN"=1)

"Position" = 0

"Data" = 2 = 2#0010 e "Q" = 0

Disponível

- PLC300 : a partir da versão inicial

6.13.4 Logic Boolean

6.13.4.1 NOT

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = NOT "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 12 = 2#1100
 "Result" = 3 = 2#0011

Exemplo2 ("EN"=1)

"Value" = 18743 = 2# 0100_1001_0011_0111
 "Result" = -56 = 2#1100_1000 (se SINT)
 "Result" = 200 = 2#1100_1000 (se USINT)
 "Result" = -18744 = 2#1011_0110_1100_1000 (se INT)
 "Result" = 46762 = 2#1011_0110_1100_1000 (se UINT)

Disponível

- PLC300 : a partir da versão inicial

6.13.4.2 AND**Figura****Configuração do Bloco**

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" AND "Value2".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value1" = 12 = 2#1100

"Value2" = 5 = 2#0101
 "Result" = 4 = 2#0100

Exemplo2 ("EN"=1)

"Value1" = 18743 = 2# 0100_1001_0011_0111
 "Value2" = 43798 = 2#1010_1011_0001_0110
 "Result" = 22 = 2#0001_0110 (se SINT)
 "Result" = 2326 = 2#0000_1001_0001_0110 (se INT)

Disponível

- PLC300 : a partir da versão inicial

6.13.4.3 OR

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" OR "Value2".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value1" = 12 = 2#1100.
 "Value2" = 5 = 2#0101.
 "Result" = 14 = 2#1101.

Exemplo2 ("EN"=1)

"Value1" = 18743 = 2# 0100_1001_0011_0111.
 "Value2" = 43798 = 2#1010_1011_0001_0110.
 "Result" = 55 = 2#0011_0111 (se SINT).
 "Result" = -5321 = 2#1110_1011_0011_0111 (se INT).
 "Result" = 60215 = 2#1110_1011_0011_0111 (se UINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.4.4 XOR**Figura****Configuração do Bloco**

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" XOR "Value2", ou seja, "Result" = ("Value1" AND (NOT "Value2")) OR ((NOT "Value1") AND "Value2").

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value1" = 12 = 2#1100.
 "Value2" = 5 = 2#0101.
 "Result" = 9 = 2#1001.

Exemplo2 ("EN"=1)

"Value1" = 18743 = 2# 0100_1001_0011_0111.
 "Value2" = 43798 = 2#1010_1011_0001_0110.
 "Result" = 33 = 2#0010_0001 (se SINT).
 "Result" = -7647 = 2#1110_0010_0010_0001 (se INT).
 "Result" = 57889 = 2#1110_0010_0010_0001 (se UINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.4.5 XNOR

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
 Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é calculado: "Result" = "Value1" XNOR "Value2", ou seja, "Result" = ("Value1" AND "Value2") OR ("Value1" AND "Value2").

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value1" = 12 = 2#1100.
 "Value2" = 5 = 2#0101.
 "Result" = 6 = 2#0110.

Exemplo2 ("EN"=1)

"Value1" = 18743 = 2# 0100_1001_0011_0111.

"Value2" = 43798 = 2#1010_1011_0001_0110.
"Result" = -34 = 2#1101_1110 (se SINT).
"Result" = 222 = 2#1101_1110 (se USINT).
"Result" = 7646 = 2#0001_1101_1101_1110 (se INT).

Disponível

- PLC300 : a partir da versão inicial

6.13.5 Logic Shift

6.13.5.1 ASHL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: SINT, INT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: SINT, INT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento aritmético binário para a esquerda, conforme "Shift": "Result" = "Value" << "Shift". O sinal de "Value" é levado em consideração.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 3 = 2#0011

"Shift" = 1

"Result" = 6 = 2#0110

Exemplo2 ("EN"=1)

"Value" = -93 = 2#1010_0011

"Shift" = 4

"Result" = -80 = 2#1011_0000 (se SINT)

"Result" = -1488 = 2#1111_1010_0011_0000 (se INT)

Disponível

- PLC300 : a partir da versão inicial

6.13.5.2 ASHR**Figura****Configuração do Bloco**

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: SINT, INT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: SINT, INT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento aritmético binário para a direita, conforme "Shift": "Result" = "Value" >> "Shift". O sinal de "Value" é levado em consideração.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 76 = 2#0100_1100

"Shift" = 3

"Result" = 9 = 2#0000_1001 (se SINT)

Exemplo2 ("EN"=1)

"Value" = -71 = 2#1011_1001
 "Shift" = 5
 "Result" = -3 = 2#1111_1101 (se SINT)

Exemplo3 ("EN"=1)

"Value" = -3011 = 2#1111_0100_0011_1101
 "Shift" = 2
 "Result" = -113 = 2#1000_1111 (se SINT)
 "Result" = -753 = 2#1111_1101_0000_1111 (se INT)

Disponível

- PLC300 : a partir da versão inicial

6.13.5.3 SHL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento lógico binário para a esquerda, conforme "Shift": "Result" = "Value" << "Shift".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 163 = 2#1010_0011
 "Shift" = 4

"Result" = 48 = 2#0011_0000 (se SINT)
 "Result" = 2608 = 2#0000_1010_0011_0000 (se INT)

Disponível

- PLC300 : a partir da versão inicial

6.13.5.4 SHR

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executado um deslocamento lógico binário para a direita, conforme "Shift": "Result" = "Value" >> "Shift".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 163 = 2#1010_0011

"Shift" = 4

"Result" = 10 = 2#0000_1010

Exemplo2 ("EN"=1)

"Value" = 2944 = 2#0000_1011_1000_0000

"Shift" = 3

"Result" = 112 = 2#0111_0000 (se SINT)

"Result" = 368 = 2#0000_0001_0111_0000 (se INT)

Disponível

- PLC300 : a partir da versão inicial

6.13.6 Logic Rotate

6.13.6.1 ROL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota1: A variável tem que ter permissão de escrita.

Nota2: O tamanho do tipo de dado deve ser igual ao de "Value".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma rotação para a esquerda em "Value" conforme "Shift" e armazenado em "Result". Os bits mais significativos deslocados para fora, são transferidos para a parte menos significativa de "Result".

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 21 = 2#0001_0101.

"Shift" = 5

"Result" = 162 = 2#1010_0010

Disponível

- PLC300 : a partir da versão inicial

6.13.6.2 ROR

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Shift" - VAR_IN: inserir um número inteiro constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

Nota1: A variável tem que ter permissão de escrita.

Nota2: O tamanho do tipo de dado deve ser igual ao de "Value".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma rotação para a direita em "Value" conforme "Shift" e armazenado em "Result". Os bits menos significativos deslocados para fora, são transferidos para a parte mais significativa de "Result".

Se "EN" for 0, o valor em "Data" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 21 = 2#0001_0101.

"Shift" = 3.

"Result" = 162 = 2#1010_0010.

Disponível

- PLC300 : a partir da versão inicial

6.13.7 Math Basic

6.13.7.1 ABS

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = |"Value"|. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = -5.

"Result" = 5 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = -200.

"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = 200 (se USINT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.2 NEG

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: SINT, INT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = - "Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = -5
"Result" = 5 e "Done"=1.

Exemplo2 ("EN"=1)

"Value" = 10
"Result" = -10 e "Done"=1.

Exemplo3 ("EN"=1)

"Value" = -200
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 200 (se USINT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.3 ADD

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
 Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1, ou seja, o topo do STACK for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" + "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário ("EN"=0), o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value1" = 100.
 "Value2" = 150.
 "Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
 "Result" = 250 (se USINT) e "Done"=1.

Exemplo2 ("EN"=1)

"Value1" = -100.
 "Value2" = -150.
 "Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
 "Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
 "Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).
 "Result" = -250 (se INT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.4 SUB

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" - "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value1" = 400.
"Value2" = 210.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 190 (se USINT) e "Done"=1.

Exemplo2 ("EN"=1)

"Value1" = 500.
"Value2" = 730.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = -230 (se INT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.5 MUL

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" * "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"Value1" = 143.

"Value2" = 308.

"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = inalterado (se INT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = 44044 (se UINT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.6 DIV

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" / "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"Value1" = 751.

"Value2" = 3.

"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = 250 (se USINT) e "Done"=1.

"Result" = 250.333 (se REAL) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.7.7 MOD

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value1" % "Value2", ou seja, é calculado o resto entre a divisão de "Value1" com "Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value1" = 7.
"Value2" = 3.
"Result" = 1 e "Done"=1.

Exemplo2 ("EN"=1)

"Value1" = 1000.
"Value2" = 400.
"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).
"Result" = 200 (se USINT) e "Done"=1.

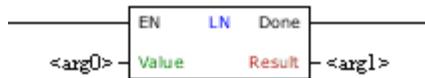
Disponível

- PLC300 : a partir da versão inicial

6.13.8 Math Extended

6.13.8.1 LN

Figura



Configuração do Bloco

`<arg0>`: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: REAL.

Faixa de valores: somente valores reais positivos.

`<arg1>`: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\ln(\text{Value})$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Nota: \ln é o log na base = 2.7182818284590452353602874.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 100.

"Result" = 4.605 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = 0.025.

"Result" = -3.689 e "Done"=1.

Exemplo3 ("EN"=1)

"Value" = 0.0 (valor fora da faixa).

"Result" = inalterado e "Done"=0.

Disponível

- PLC300 : a partir da versão inicial

6.13.8.2 EXP

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = e"Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Nota: $e = 2.7182818284590452353602874$.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 5.
"Result" = 148.413 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = -2.
"Result" = 0.135 e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.8.3 LOG10

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: REAL.

Faixa de valores: somente valores reais positivos.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\log_{10}(\text{Value})$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 110.53

"Result" = 2.043 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = 0.147.

"Result" = -0.883 e "Done"=1.

Exemplo3 ("EN"=1)

"Value" = -1.3 (valor fora da faixa).

"Result" = inalterado e "Done"=0.

Disponível

- PLC300 : a partir da versão inicial

6.13.8.4 ALOG10

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = 10"Value". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 2.5.

"Result" = 316.228 e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = -1.333.

"Result" = 0.046 e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.8.5 POW

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Power" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = "Value" "Power". Se não ocorrer

erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 2.

"Power" = 3.

"Result" = 8 e "Done"=1.

Exemplo2 ("EN"=1)

"Value" = -49.

"Power" = 0.5.

"Result" = inalterado e "Done"=0 (resulta num valor imaginário).

Disponível

- PLC300 : a partir da versão inicial

6.13.8.6 SQRT

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

Faixa de valores: somente valores reais positivos.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = RaizQuadrada("Value"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 81.

"Result" = 9. e "Done" = 1.

Exemplo2 ("EN"=1)

"Value" = -4.

"Result" = inalterado e "Done"=0 (resulta num valor imaginário).

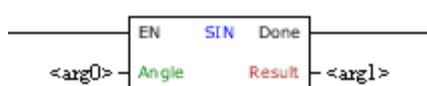
Disponível

- PLC300 : a partir da versão inicial

6.13.9 Math Trigonometry

6.13.9.1 SIN

Figura



Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: REAL.

Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\sin("Angle")$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"Angle" = 1.047 (equivalente a 60º).

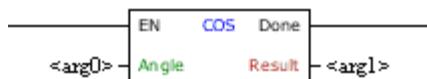
"Result" = 0.866 e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.2 COS

Figura



Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\cos(\text{Angle})$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

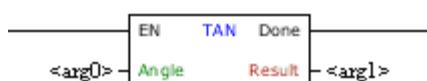
"Angle" = 1.047 (equivalente a 60º).
"Result" = 0.5 e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.3 TAN

Figura



Configuração do Bloco

<arg0>: "Angle" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\tan(\text{Angle})$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"Angle" = 1.047 (equivalente a 60º).
"Result" = 1.732 e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.4 ASIN

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Faixa de Valores: -1.0 a 1.0.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = $\arcsin(\text{Angle})$. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 0.866.

"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Exemplo2 ("EN"=1)

"Value" = 1.5 (valor fora da faixa permitida).

"Angle" = inalterado e "Done"=0.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.5 ACOS**Figura****Configuração do Bloco**

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

Faixa de Valores: -1.0 a 1.0.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Unidade: radianos

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-cos("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo1 ("EN"=1)

"Value" = 0.5.

"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Exemplo2 ("EN"=1)

"Value" = 1.5 (valor fora da faixa permitida).

"Angle" = inalterado e "Done"=0.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.6 ATAN

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Angle" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-tan("Angle"). Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"Value" = 1.732.
"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.9.7 ATAN2

Figura



Configuração do Bloco

<arg0>: "X" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: REAL.

<arg0>: "Y" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg2>: "Angle" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Unidade: radianos

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), é calculado: "Result" = arc-tan("Y" / "X"). Diferentemente do bloco ATAN, este bloco leva em consideração o quadrante para calcular o valor do ângulo. Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Se "EN"=0, "Result" permanece inalterado e a saída "Done" fica em 0.

Exemplo ("EN"=1)

"X" = 1.

"Y" = 1.732.

"Angle" = 1.047 (equivalente a 60º) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.10 Math Util

6.13.10.1 MAX

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou
REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou
REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução (EN=1), é calculado: Se "Value1" > "Value2", "Result" = "Value1", senão "Result"="Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário (EN=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando EN=1)

"Value1" = 100.

"Value2" = 150.

"Result" = 150.

Exemplo2 (considerando EN=1)

"Value1" = -1000.

"Value2" = -1500.

"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).

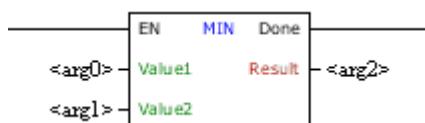
"Result" = -1000 (se INT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.10.2 MIN

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Value2" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução (EN=1), é calculado: Se "Value1" < "Value2", "Result" = "Value1", senão "Result"="Value2". Se não ocorrer erros, a saída "Done" vai para 1. Caso contrário, "Done" indica 0 e o "Result" permanece inalterado.

Em caso contrário (EN=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando EN=1)

"Value1" = 100.

"Value2" = 150.

"Result" = 100.

Exemplo2 (considerando EN=1)

"Value1" = -1000.

"Value2" = -1500.

"Result" = inalterado (se SINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = inalterado (se USINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = inalterado (se UINT) e "Done"=0 (valor fora do limite deste tipo de dado).

"Result" = -1500 (se INT) e "Done"=1.

Disponível

- PLC300 : a partir da versão inicial

6.13.10.3 SAT

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg1>: "Minimum" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg2>: "Maximum" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

<arg3>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada EN for 1.

Em execução ("EN"=1), a saída "Q" vai para 1 se houver saturação, ou seja: Se "Value" > "Maximum", "Result" = "Maximum" e "Q"=1; Se Value < "Minimum", Result = "Minimum" e "Q"=1; Senão "Result"="Value" e "Q"=0. Se ocorrer erros, "Q" indica 0 e o "Result" permanece inalterado.

Em caso contrário ("EN"=0), o valor em "Result" permanece inalterado.

Exemplo1 (considerando "EN"=1)

"Value" = 10.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 10 e "Q"=0.

Exemplo2 (considerando "EN"=1)

"Value" = 30.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 15 e "Q"=1.

Exemplo3 (considerando "EN"=1)

"Value" = 2.
"Minimum" = 5.
"Maximum" = 15.
"Result" = 5 e "Q"=1.

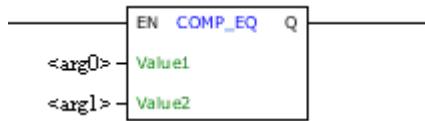
Disponível

- PLC300 : a partir da versão inicial

6.13.11 Comparação

6.13.11.1 COMPEQ

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)

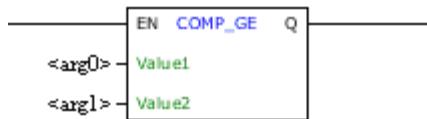
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

Disponível

- PLC300 : a partir da versão inicial

6.13.11.2 COMPGE

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)

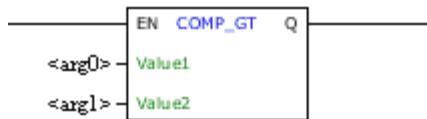
"Value1" = 4158
"Value2" = 36.5
"Q" = 1

Disponível

- PLC300 : a partir da versão inicial

6.13.11.3 COMPGT

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1">>"Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 0

Exemplo3: ("EN"=1)

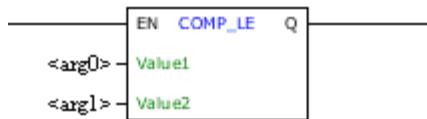
"Value1" = 4158
"Value2" = 36.5
"Q" = 1

Disponível

- PLC300 : a partir da versão inicial

6.13.11.4 COMPLE

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"="Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 1

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)

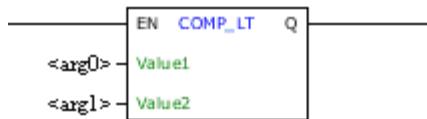
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

Disponível

- PLC300 : a partir da versão inicial

6.13.11.5 COMPLT

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1" < "Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)

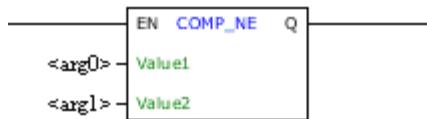
"Value1" = 4158
"Value2" = 36.5
"Q" = 0

Disponível

- PLC300 : a partir da versão inicial

6.13.11.6 COMPNE

Figura



Configuração do Bloco

<arg0>: "Value1" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg1>: "Value2" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1), o bloco carrega 1 para a saída "Q" se "Value1"?"Value2". Caso contrário, carrega 0 para "Q".

Se "EN" for 0, a saída "Q" vai para 0.

Exemplo1: ("EN"=1)

"Value1" = 2.3
"Value2" = 2.3
"Q" = 0

Exemplo2: ("EN"=1)

"Value1" = 4
"Value2" = 26
"Q" = 1

Exemplo3: ("EN"=1)

"Value1" = 4158
"Value2" = 36.5
"Q" = 1

Disponível

- PLC300 : a partir da versão inicial

6.13.12 Conversão

6.13.12.1 BCD

6.13.12.1.1 BCD_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para BCD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

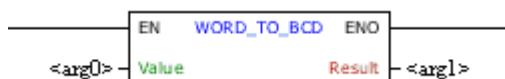
"Value" = 123 = 16#7B.
"Result" = 291 = 16#123.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.1.2 WORD_TO_BCD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BCD para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 291 = 16#123.
"Result" = 123 = 16#7B.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.2 BOOL

6.13.12.2.1 BYTE_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BYTE para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.
"Result" = 0.

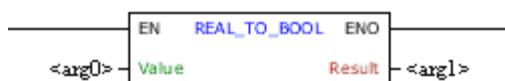
Exemplo2 ("EN"=1)

"Value" = 6.
"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.2.2 REAL_TO_BOOL

Figura**Configuração do Bloco**

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1.
Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.0.
"Result" = 0.

Exemplo2 ("EN"=1)

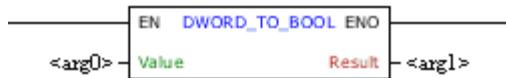
"Value" = -0.26.
"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.2.3 DWORD_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)

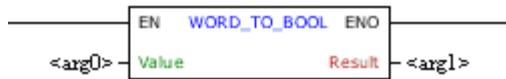
"Value" = 6.
"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.2.4 WORD_TO_BOOL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para BOOL, ou seja, para qualquer "Value" diferente de 0, "Result"=1. Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.

"Result" = 0.

Exemplo2 ("EN"=1)

"Value" = 6.

"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.3 BYTE

6.13.12.3.1 BOOL_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.

"Result" = 0.

Exemplo2 ("EN"=1)

"Value" = 1.

"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.3.2 REAL_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT ou SINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = -45.6.

"Result" = -45 = 16#D3 (se SINT).

"Result" = 211 = 16#D3 (se USINT).

Exemplo2 ("EN"=1)

"Value" = 1452.25.
"Result" = -84 = 16#AC (se SINT).
"Result" = 172 = 16#AC (se USINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.12.3.3 DWORD_TO_BYTE

Figura**Configuração do Bloco**

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 145.
"Result" = 145 e "Done"=1.

Exemplo2 ("EN"=1)

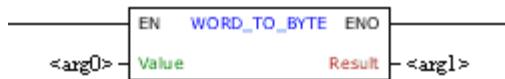
"Value" = 1452 = 16#5AC.
"Result" = -84 = 16#AC (se INT).
"Result" = 172 = 16#AC (se UINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.12.3.4 WORD_TO_BYTE

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para BYTE.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 145.
"Result" = 145 e "Done"=1.

Exemplo2 ("EN"=1)

"Value" = 1452 = 16#5AC.
"Result" = -84 = 16#AC (se INT).
"Result" = 172 = 16#AC (se UINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.12.4 WORD

6.13.12.4.1 BOOL_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.
"Result" = 0.

Exemplo2 ("EN"=1)

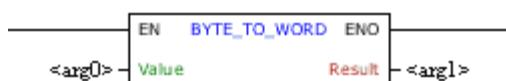
"Value" = 1.
"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.4.2 BYTE_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BYTE para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 145.

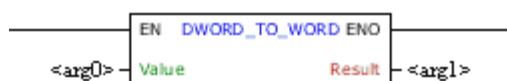
"Result" = 145.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.4.3 DWORD_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 145

"Result" = 145

Exemplo2 ("EN"=1)

"Value" = 104758

"Result" = -26314 = 16# 9936 (se INT)

"Result" = 39222 = 16# 9936 (se UINT)

Disponível

- PLC300 : a partir da versão inicial

6.13.12.4.4 REAL_TO_WORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para WORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = -45.74.
"Result" = -45 = 16#FFD3 (se INT).
"Result" = 65491= 16#FFD3 (se UINT).

Exemplo2 ("EN"=1)

"Value" = 104386.234.
"Result" = -26686= 16#97C2 (se INT).
"Result" = 38850= 16#97C2 (se UINT).

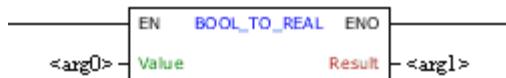
Disponível

- PLC300 : a partir da versão inicial

6.13.12.5 REAL

6.13.12.5.1 BOOL_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para REAL, ou seja, para qualquer "Value" diferente de 0, "Result"=1.
Senão, "Result"=0.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.
"Result" = 0.0.

Exemplo2 ("EN"=1)

"Value" = 1.
"Result" = 1.0

Disponível

- PLC300 : a partir da versão inicial

6.13.12.5.2 DWORD_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de DWORD para REAL.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = -45

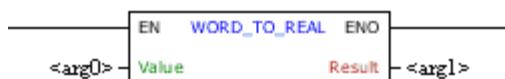
"Result" = -45.0.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.5.3 WORD_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para REAL.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = -45.

"Result" = -45.0.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.5.4 BYTE_TO_REAL

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BYTE para REAL.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = -45.

"Result" = -45.0.

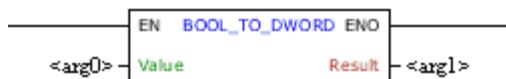
Disponível

- PLC300 : a partir da versão inicial

6.13.12.6 DWORD

6.13.12.6.1 BOOL_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BOOL para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 0.

"Result" = 0.

Exemplo2 ("EN"=1)

"Value" = 1.

"Result" = 1.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.6.2 REAL_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de REAL para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = -45.75

"Result" = -45 = 16#FFFF_FFD3 (se DINT).

"Result" = 4294967251 = 16#FFFF_FFD3 (se UDINT).

Exemplo2 ("EN"=1)

"Value" = 11275503078.234.

"Result" = -1609398810 = 16#A012_85E6 (se DINT).

"Result" = 2685568486 = 16#A012_85E6 (se UDINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.12.6.3 WORD_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de WORD para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 145.

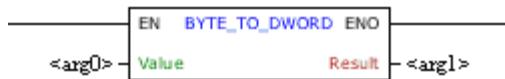
"Result" = 145.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.6.4 BYTE_TO_DWORD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de BYTE para DWORD.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 145.

"Result" = 145.

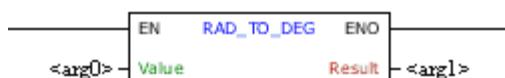
Disponível

- PLC300 : a partir da versão inicial

6.13.12.7 Rad-Deg

6.13.12.7.1 RAD_TO_DEG

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: REAL.
Unidade: radianos

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Unidade: graus

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de radianos para graus.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 1.047.

"Result" = 60.

Disponível

- PLC300 : a partir da versão inicial

6.13.12.7.2 DEG_TO_RAD

Figura



Configuração do Bloco

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).

Tipo de dado: REAL.

Unidade: graus

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: REAL.

Unidade: radianos

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é executada a conversão de graus para radianos.

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Value" = 60.
"Result" = 1.047.

Disponível

- PLC300 : a partir da versão inicial

6.13.13 Data Transfer

6.13.13.1 STORE

Figura**Configuração do Bloco**

<arg0>: "SRC" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL.

Descrição: dado fonte.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BYTE, USINT ou SINT se "SRC" for BYTE, USINT ou SINT.

Tipo de dado: WORD, UINT ou INT se "SRC" for WORD, UINT ou INT.

Tipo de dado: DWORD, UDINT, DINT ou REAL se "SRC" for DWORD, UDINT, DINT ou REAL.

Nota: A variável tem que ter permissão de escrita.

Descrição: dado destino.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo de "SRC" é armazenado em "DST". O tamanho do tipo de dado de "DST" deve ser o mesmo que o de "SRC".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"SRC" = 145 (SINT).
"DST" = 145 (USINT).

Exemplo2 ("EN"=1)

"SRC" = 4.85 (REAL).
"DST" = 859020096 = 16#3333_9B40 (DINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.2 ILOADBOOL

Figura



Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_STATUS (GLOBAL_SYSTEM_STATUS),
SYS_COMMAND (GLOBAL_SYSTEM_COMMAND), HW_INPUT (GLOBAL_IO_INPUT),
HW_OUTPUT (GLOBAL_IO_OUTPUT), NET_INPUT (GLOBAL_NETWORK_INPUT) ou
NET_OUTPUT (GLOBAL_NETWORK_OUTPUT).

Descrição: grupo em que variável global está armazenada.

<arg1>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg2>: "Bit" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Descrição: posição do bit a ser verificado da variável global.

<arg3>: "Value" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.
Descrição: variável com o conteúdo lido.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo da variável global localizada no "Bit" de "Address" do "Group#" é transferido para "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)
"Group#"=HW_OUTPUT
"Address"=0

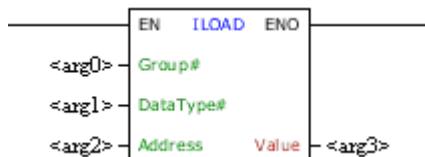
"Bit" = 4.
 "Value"=indicará o conteúdo da saída digital 5 (DO5).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.3 ILOAD

Figura



Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
 Opções: GLOBAL, GLOBAL_RETAIN, SYS_STATUS (GLOBAL_SYSTEM_STATUS),
 SYS_COMMAND (GLOBAL_SYSTEM_COMMAND), HW_INPUT (GLOBAL_IO_INPUT),
 HW_OUTPUT (GLOBAL_IO_OUTPUT), NET_INPUT (GLOBAL_NETWORK_INPUT) ou
 NET_OUTPUT (GLOBAL_NETWORK_OUTPUT).
 Descrição: grupo em que variável global está armazenada.

<arg1>: "DataType#" - VAR_IN: selecionar uma das seguintes opções.
 Opções: BYTE, USINT, SINT, WORD UINT, INT, DWORD UDINT, DINT ou REAL.
 Descrição: tipo de dado da variável global.

<arg2>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
 Tipo de dado: DWORD, UDINT ou DINT.
 Descrição: endereço da variável global, conforme o seu grupo.

<arg3>: "Value" - VAR_OUT: inserir uma variável (tag).
 Tipo de dado: conforme selecionado em "DataType#".
 Nota: A variável tem que ter permissão de escrita.
 Descrição: variável com o conteúdo lido.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo da variável global localizada em "Address" do "Group#", que é do tipo "DataType#", é transferido para "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Group#"=HW_INPUT
"DataType#"=WORD
"Address"=6
"Value"=indicará o conteúdo da entrada analógica 1 (AI1).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.4 ISTOREBOOL

Figura



Configuração do Bloco

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
Opções: GLOBAL, GLOBAL_RETAIN, SYS_COMMAND (GLOBAL_SYSTEM_COMMAND),
HW_OUTPUT (GLOBAL_IO_OUTPUT) ou NET_OUTPUT
(GLOBAL_NETWORK_OUTPUT).
Descrição: grupo em que variável global está armazenada.

<arg1>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: DWORD, UDINT ou DINT.
Descrição: endereço da variável global, conforme o seu grupo.

<arg2>: "Bit" - VAR_IN: selecionar uma constante ou uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT.
Descrição: posição do bit a ser verificado da variável global.

<arg3>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipo de dado: BOOL.
Descrição: valor a ser escrito.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), "Value" é transferido para a variável global localizada no "Bit" de "Address" do "Group#".

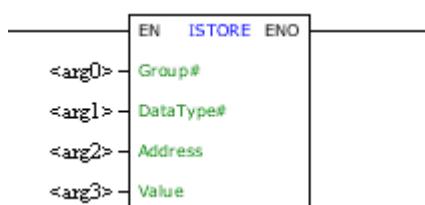
Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo ("EN"=1)

"Group#"=HW_OUTPUT
 "Address"=0
 "Bit" = 4.
 "Value"=1. Seta a saída digital 5 (D05).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.5 ISTORE**Figura****Configuração do Bloco**

<arg0>: "Group#" - VAR_IN: selecionar uma das seguintes opções.
 Opções: GLOBAL, GLOBAL_RETAIN, SYS_COMMAND (GLOBAL_SYSTEM_COMMAND), HW_OUTPUT (GLOBAL_IO_OUTPUT) ou NET_OUTPUT (GLOBAL_NETWORK_OUTPUT).
 Descrição: grupo em que variável global está armazenada.

<arg1>: "DataType#" - VAR_IN: selecionar uma das seguintes opções.
 Opções: BYTE, USINT, SINT, WORD UINT, INT, DWORD UDINT, DINT ou REAL.
 Descrição: tipo de dado da variável global.

<arg2>: "Address" - VAR_IN: selecionar uma constante ou uma variável (tag).
 Tipo de dado: DWORD, UDINT ou DINT.
 Descrição: endereço da variável global, conforme o seu grupo.

<arg3>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).
 Tipo de dado: conforme selecionado em "DataType#".
 Descrição: valor a ser escrito.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o conteúdo de "Value" , que é do tipo "DataType#", é transferido para a variável global localizada em "Address" do "Group#".

Se "EN" for 0, o valor em "Result" permanece inalterado.

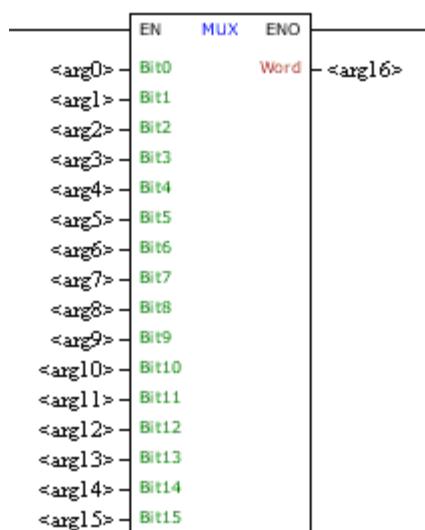
Exemplo ("EN"=1)
"Group#"=HW_OUTPUT
"DataType#"=WORD
"Address"=6
"Value"=100. Novo valor da entrada analógica 1 (AI1).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.6 MUX

Figura



Configuração do Bloco

<arg0>: "Bit0" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Bit1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg2>: "Bit2" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg3>: "Bit3" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg4>: "Bit4" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg5>: "Bit5" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg6>: "Bit6" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg7>: "Bit7" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg8>: "Bit8" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg9>: "Bit9" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg10>: "Bit10" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg11>: "Bit11" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg12>: "Bit12" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg13>: "Bit13" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg14>: "Bit14" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg15>: "Bit15" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg16>: "Word" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

Nota: A variável tem que ter permissão de escrita.

Descrição: bits concatenados.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), os valores "Bit0", "Bit1", "Bit2", ... e "Bit15" são transferidos para "Word"...

Se "EN" for 0, o valor em "Word" permanece inalterado.

Exemplo ("EN"=1)

"Bit0" = 1; "Bit1" = 0; "Bit2" = 0; "Bit3" = 0; "Bit4" = 1; "Bit5" = 1;
"Bit6" = 0; "Bit7" = 0; "Bit8" = 0; "Bit9" = 1; "Bit10" = 0; "Bit11" = 0;
"Bit12" = 0; "Bit13" = 0; "Bit14" = 1; "Bit15" = 1.
"Word" = -15823 = 16#1100_0010_0011_0001 (se INT).

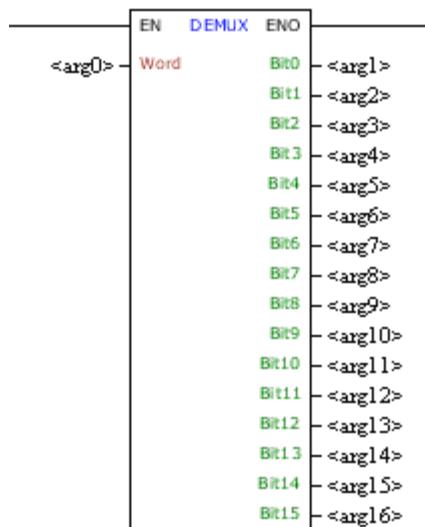
"Word" = 49713 = 16#1100_0010_0011_0001 (se UINT).

Disponível

- PLC300 : a partir da versão inicial

6.13.13.7 DEMUX

Figura



Configuração do Bloco

<arg0>: "Word" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipo de dado: WORD, UINT ou INT.

<arg1>: "Bit0" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg2>: "Bit1" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg3>: "Bit2" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg4>: "Bit3" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg5>: "Bit4" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BOOL.
Nota: A variável tem que ter permissão de escrita.

<arg6>: "Bit5" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "Bit6" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Bit7" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "Bit8" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg10>: "Bit9" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg11>: "Bit10" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg12>: "Bit11" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg13>: "Bit12" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg14>: "Bit13" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg15>: "Bit14" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

<arg16>: "Bit15" - VAR_OUT: inserir uma variável (tag).

Tipo de dado: BOOL.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o valor de "Word" são distribuído respectivamente ao "Bit0",

"Bit1", "Bit2", ... e "Bit15".

Se "EN" for 0, os valores em "Bit0", "Bit1", "Bit2", ... e "Bit15" permanecem inalterados.

Exemplo ("EN"=1)

```
"Word" = 49713 = 16#1100_0010_0011_0001.
"Bit0" = 1; "Bit1" = 0; "Bit2" = 0; "Bit3" = 0; "Bit4" = 1; "Bit5" = 1;
"Bit6" = 0; "Bit7" = 0; "Bit8" = 0; "Bit9" = 1; "Bit10" = 0; "Bit11" = 0;
"Bit12" = 0; "Bit13" = 0; "Bit14" = 1; "Bit15" = 1.
```

Disponível

- PLC300 : a partir da versão inicial

6.13.13.8 SEL

Figura



Configuração do Bloco

<arg0>: "Selector" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BOOL.

<arg1>: "Value0" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg2>: "Value1" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL

<arg3>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT, SINT, WORD, UINT, INT, DWORD, UDINT, DINT ou REAL
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), se "Selector"=0, "Result"="Value0" e se "Selector"=1, "Result"="Value1".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Selector"=0.
"Value0" = 5.
"Value1" = 10.
"Result" = 5.

Exemplo2 ("EN"=1)

"Selector"=1.
"Value0" = 5.
"Value1" = 10.
"Result" = 10.

Disponível

- PLC300 : a partir da versão inicial

6.13.13.9 SWAP**Figura****Configuração do Bloco**

<arg0>: "Value" - VAR_IN: inserir um número constante ou uma variável (tag).
Tipo de dado: WORD, UINT, INT, DWORD, UDINT ou DINT.

<arg1>: "Result" - VAR_OUT: inserir uma variável (tag).
Tipo de dado: BYTE, USINT ou SINT se "Value" for BYTE, USINT ou SINT.
Tipo de dado: WORD, UINT ou INT se "Value" for WORD, UINT ou INT.
Tipo de dado: DWORD, UDINT ou DINT se "Value" for DWORD, UDINT ou DINT.
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito uma troca em "Value" e transferido ao "Result". O tamanho do tipo de dado de "Result" deve ser o mesmo que o de "Value".

Se "EN" for 0, o valor em "Result" permanece inalterado.

Exemplo1 ("EN"=1)

"Value" = 4660 = 16#1234
"Result" = 13330 = 16#3412

Exemplo2 ("EN"=1)

"Value" = 19088743 = 16#0123_4567
"Result" = 587294533 = 16#2301_6745

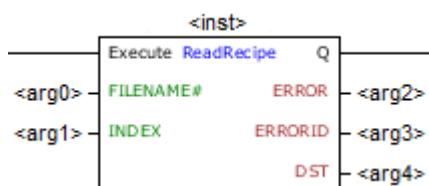
Disponível

- PLC300 : a partir da versão inicial

6.13.13.1|ReadRecipe

Leitura de Receitas

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "FILENAME#" - VAR_IN: inserir uma String com o nome e a extensão do arquivo.

Tipos de dados: STRING

Descrição: Nome do arquivo com as receitas.

Formato: 'RECIPEFILENAME.CSV', onde CSV define o padrão do arquivo quando armazenado no SD Card.

<arg1>: "INDEX" - VAR_IN: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Índice da receita a ser escrita no arquivo. Pode variar de 1 ao número total de receitas do arquivo.

<arg2>: "ERROR" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da escrita.

Nota: A variável deve possuir permissão de escrita.

<arg3>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na escrita, indica o tipo de erro ocorrido. Resultados possíveis: 1="Receita incompleta"; 2="Receita inválida (estrutura a receita diferente da estrutura de SRC)"; 3="Receita inexistente (índice da receita inexistente)"; 4="Arquivo inválido"; 5="Arquivo inexistente ou SD card inválido"; 6="SD card bloqueado para escrita".

Nota: A variável deve possuir permissão de escrita.

<arg4>: "DST" - VAR_IN: inserir uma variável (tag).

Tipos de dados: STRUCT (definido pelo usuário)

Descrição: Variável do tipo STRUCT de onde serão salvos os dados lidos da receita.

Nota: O tipo STRUCT da variável deve ser o mesmo da receita lida.

Descrição de Funcionamento

Bloco para leitura dos arquivos que podem estar armazenados no cartão de memória (SD_CARD) ou na memória interna do PLC300 (RAM).

O arquivo de receitas consiste num conjunto de dados formando uma matriz, onde uma linha representa uma estrutura, e cada coluna desta linha representa cada elemento desta estrutura de dados.

O bloco possui uma entrada de habilitação "Execute" e uma saída "Q", que é ativada quando houver sucesso na leitura desejada. O bloco é executado a cada transição positiva de "Execute", transferindo o conteúdo da posição indexada por "INDEX" no arquivo configurado em "FILENAME" para o "DST".

Se houver sucesso na leitura a saída "Q" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da leitura, a saída "ERROR" é ativada, e o código do erro é colocado em "ERRORID".

Disponível

- PLC300 : a partir da versão 1.30

6.13.13.1 WriteRecipe

Escrita de Receitas

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "FILENAME#" - VAR_IN: inserir uma String com o nome e a extensão do arquivo.

Tipos de dados: STRING

Descrição: Nome do arquivo com as receitas.

Formato: 'RECIPEFILENAME.CSV', onde CSV define o padrão do arquivo quando armazenado no SD Card.

<arg1>: "INDEX" - VAR_IN: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Índice da receita a ser escrito no arquivo. Pode variar de 1 ao número total de receitas do arquivo.

<arg2>: "SRC" - VAR_IN: inserir uma variável (tag).

Tipos de dados: STRUCT (definido pelo usuário)

Descrição: Variável do tipo STRUCT de onde serão carregados os dados para a receita desejada.

Nota: O tipo STRUCT da variável deve ser o mesmo da receita a ser escrita.

<arg3>: "ERROR" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da escrita.

Nota: A variável deve possuir permissão de escrita.

<arg4>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na escrita, indica o tipo de erro ocorrido. Resultados possíveis: 1="Receita incompleta"; 2="Receita inválida (estrutura a receita diferente da estrutura de SRC)"; 3="Receita inexistente (índice da receita inexistente)"; 4="Arquivo inválido"; 5="Arquivo inexistente ou SD card inválido"; 6="SD card bloqueado para escrita".

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de receitas em arquivos que podem estar armazenados no cartão de memória (SD_CARD) ou na memória interna do PLC300 (RAM).

O arquivo de receitas consiste num conjunto de dados formando uma matriz, onde uma linha representa uma estrutura, e cada coluna desta linha representa cada elemento desta estrutura de dados.

O bloco possui uma entrada de habilitação "Execute" e uma saída "Q", que é ativada quando houver sucesso na escrita desejada. O bloco é executado a cada transição positiva de "Execute", transferindo o conteúdo de "SRC" para a posição indexada por "INDEX" no arquivo configurado em "FILENAME".

Se houver sucesso na leitura a saída "Q" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da escrita, a saída "ERROR" é ativada, e o código do erro é colocado em "ERRORTID".

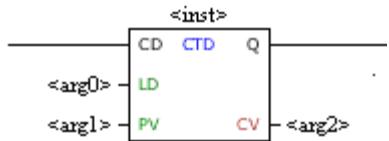
Disponível

- PLC300 : a partir da versão 1.30

6.13.14 Counter

6.13.14.1 CTD

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "LD" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BOOL

Descrição: Load

<arg1>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Value

<arg2>: "CV" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

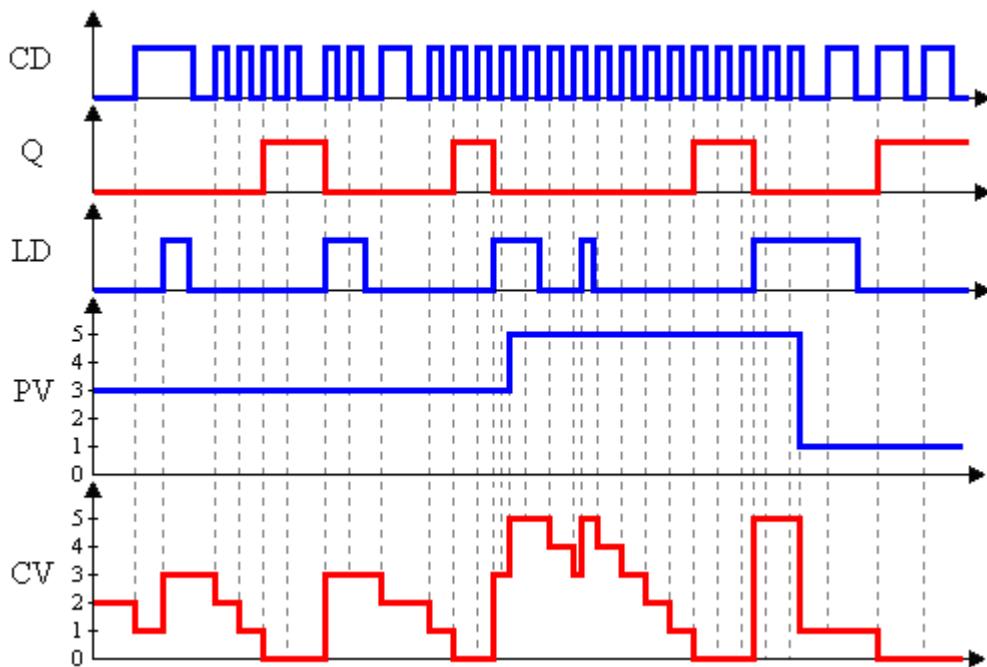
Descrição: Counter Value

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

A cada pulso detectado em "CD", "CV" é decrementado. Quando "CV" for 0, o bloco carrega 1 para a saída "Q". Caso contrário, "Q"=0. Quando "LD"=1, "CV"= "PV".

Diagrama de Funcionamento

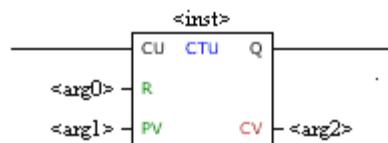


Disponível

- PLC300 : a partir da versão inicial

6.13.14.2 CTU

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "R" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BOOL

Descrição: Reset

<arg1>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Value

<arg2>: "CV" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

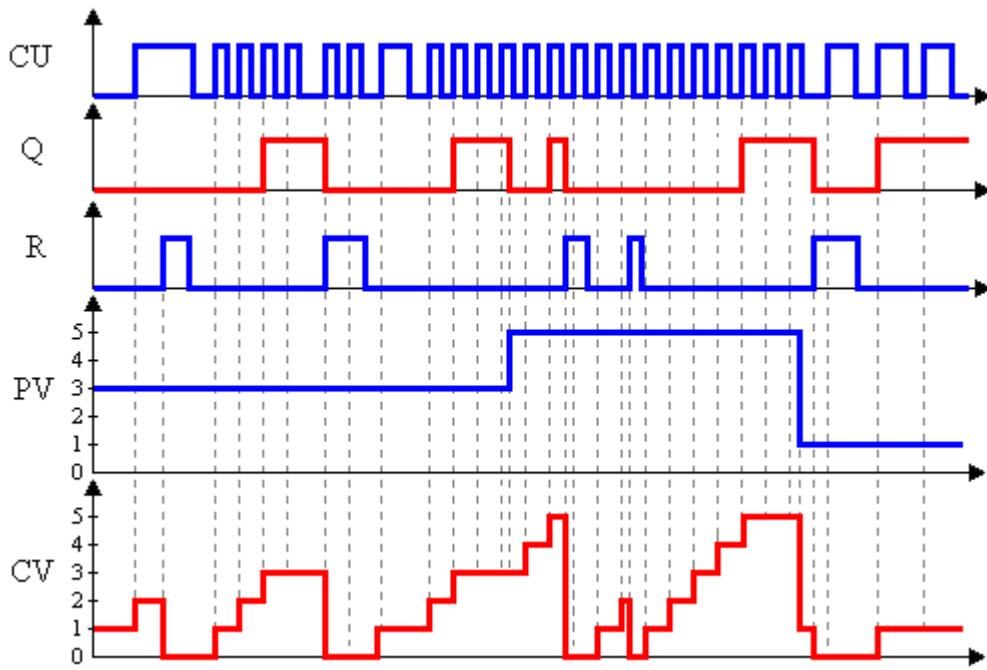
Descrição: Counter Value

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

A cada pulso detectado em "CU", "CV" é incrementado. Quando "CV" atingir "PV", o bloco carrega 1 para a saída "Q". Caso contrário, "Q"=0. Quando "R"=1, "CV"=0.

Diagrama de Funcionamento

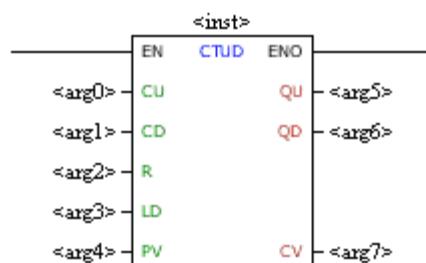


Disponível

- PLC300 : a partir da versão inicial

6.13.14.3 CTUD

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "CU" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Count Up

<arg1>: "CD" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Count Down

<arg2>: "R" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Reset

<arg3>: "R" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Load

<arg4>: "PV" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Value

<arg5>: "QU" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Output Up

Nota: A variável tem que ter permissão de escrita.

<arg6>: "QD" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Output Down

Nota: A variável tem que ter permissão de escrita.

<arg7>: "CV" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Counter Value

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

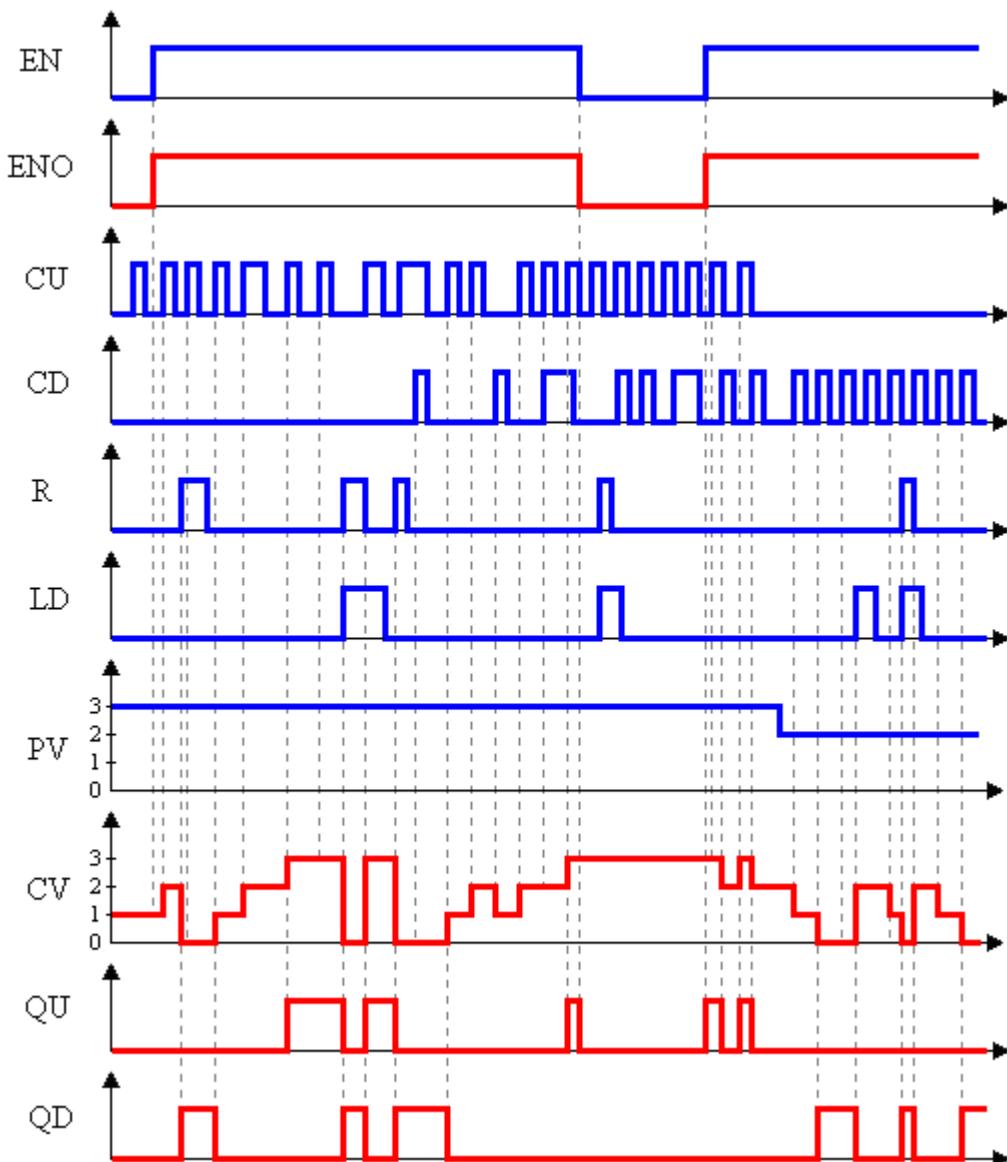
O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

A cada pulso detectado em "CU", "CV" é incrementado. Quando "CV" atingir "PV", o bloco carrega 1 para a saída "QU". Caso contrário, "QU"=0. Quando "R"=1, "CV"=0 e "QD"=1 .

Por outro lado, a cada pulso detectado em "CD", "CV" é decrementado. Quando "CV" for 0, o bloco carrega 1 para a saída "QD". Caso contrário, "QD"=0. Quando "LD"=1, "CV"= "PV" e "QU"=1.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.15 Timer

6.13.15.1 TON

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Time [ms - milisegundos]

<arg1>: "ET" - VAR_IN: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Elapsed Timer [ms - milisegundos]

Nota: A variável tem que ter permissão de escrita.

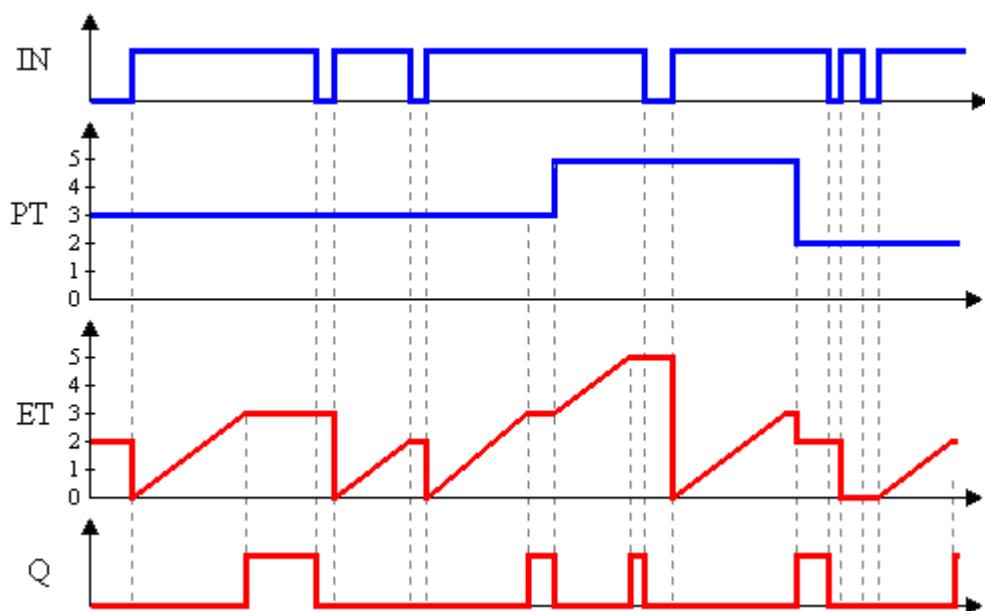
Descrição de Funcionamento

O bloco somente é executado quando a entrada "IN" for 1.

A saída "Q" vai para 0 enquanto "IN"=1 e "ET" < "PT". Após este tempo decorrido, a saída "Q" vai para 0, desde que a entrada "IN" permaneça em 1..

Se "IN"=0, "ET" permanece inalterado e a saída "Q"=0. Na borda de subida de "IN", "ET"=0.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.15.2 TOF

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Time [ms - milisegundos]

<arg1>: "ET" - VAR_IN: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Elapsed Timer [ms - milisegundos]

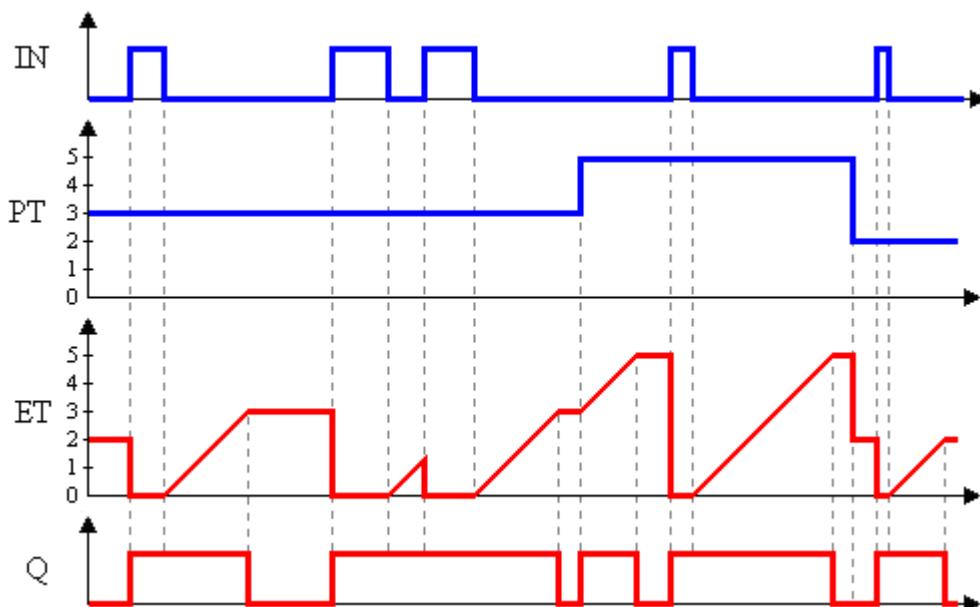
Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco é iniciado na transição positiva de "IN" e a saída Q vai para 1.

No instante em que "IN"=0, a temporização é iniciada e quando "ET" atingir "PT", a saída "Q"=0.

Diagrama de Funcionamento

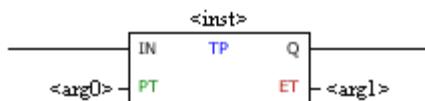


Disponível

- PLC300 : a partir da versão inicial

6.13.15.3 TP

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "PT" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Preset Time [ms - milisegundos]

<arg1>: "ET" - VAR_IN: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Elapsed Timer [ms - milisegundos]

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

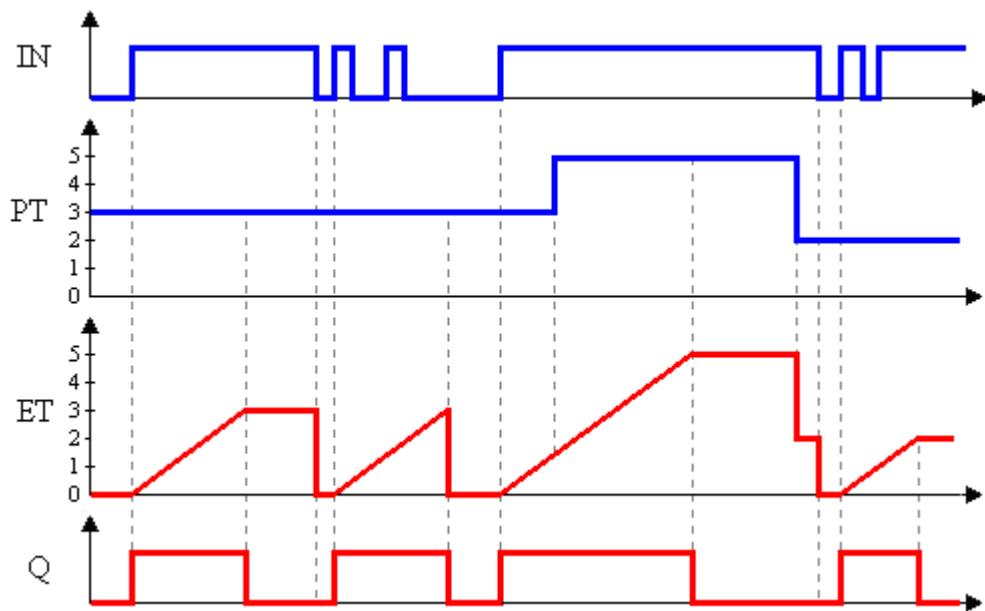
O bloco somente é executado quando ocorre uma borda de subida em "IN".

A saída "Q" vai para 1 enquanto "ET" < "PT". Após este tempo decorrido, a saída "Q" vai para 0.

Quando estiver em execução:

- Se "ET" atingir "PT" e se "IN" já estiver em 0, então "ET" = 0
- Se "ET" atingir "PT" e se "IN" for 1, então, "ET" = "PT".

Diagrama de Funcionamento



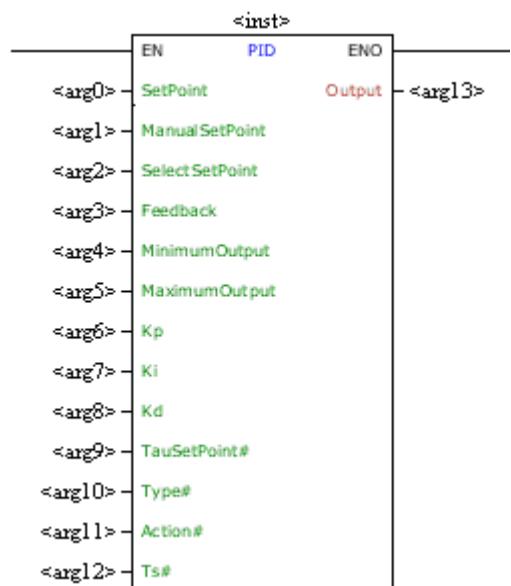
Disponível

- PLC300 : a partir da versão inicial

6.13.16 Control

6.13.16.1 PID

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SetPoint" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: REAL

Descrição: Referência automática, ou seja, é o valor desejado para a variável de processo.

<arg1>: "ManualSetPoint" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Referência manual, ou seja, é o valor transferido à saída do controlador.

<arg2>: "SelectorSetPoint" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: BOOL

Descrição: Seleciona a referência ativa: 0= "SetPoint"; 1= "ManualSetPoint".

<arg3>: "Feedback" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Realimentação da saída do processo, ou seja, é a variável do processo.

<arg4>: "MinimumOutput" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Valor mínimo permitido para a saída do controlador.

<arg5>: "MaximumOutput" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Valor máximo permitido para a saída do controlador.

<arg6>: "Kp" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Ganho proporcional do controlador, ou seja, é a correção proporcional ao erro.

Nota: A correção a ser aplicado ao processo deve crescer na proporção que cresce o erro entre o valor real (saída do processo) e desejado (referência).

<arg7>: "Ki" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Ganho integral do controlador, ou seja, é a correção proporcional ao produto erro- tempo.

Nota: Erros pequenos mas que existem há muito tempo requerem correção mais intensa.

<arg8>: "Kd" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Ganho derivativo do controlador, ou seja, é a correção proporcional à taxa de variação do erro em função do tempo.

Nota: Se o erro está variando muito rápido, esta taxa de variação deve ser reduzida para evitar oscilações.

<arg9>: "TauSetPoint#" - VAR_IN: inserir uma constante.

Tipos de dados: REAL

Descrição: Constante de tempo do filtro da referência automática.

<arg10>: "Type#" - VAR_IN: escolher uma das opções: "Academic"; "Parallel".

Tipos de dados: BYTE

Descrição: Tipo de controlador

<arg11>: "Action#" - VAR_IN: escolher uma das opções: "Direct"; "Reverse".

Tipos de dados: BYTE

Descrição: Ação de controle. Na ação direta, se a variável do processo aumenta, a saída do controlador também aumenta. Na ação reversa, se a variável do processo aumenta, a saída do controlador diminui.

<arg12>: "Ts#" - VAR_IN: inserir uma constante.

Tipos de dados: UINT

Descrição: Período de amostragem do controlador [ms].

<arg13>: "Output" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: REAL

Descrição: Saída do controlador que atua na entrada do processo.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Se "EN"=0, "Output" permanece inalterado e a saída "ENO"=0. Na borda de subida de "EN", "Output"=0.

Diagrama de Blocos

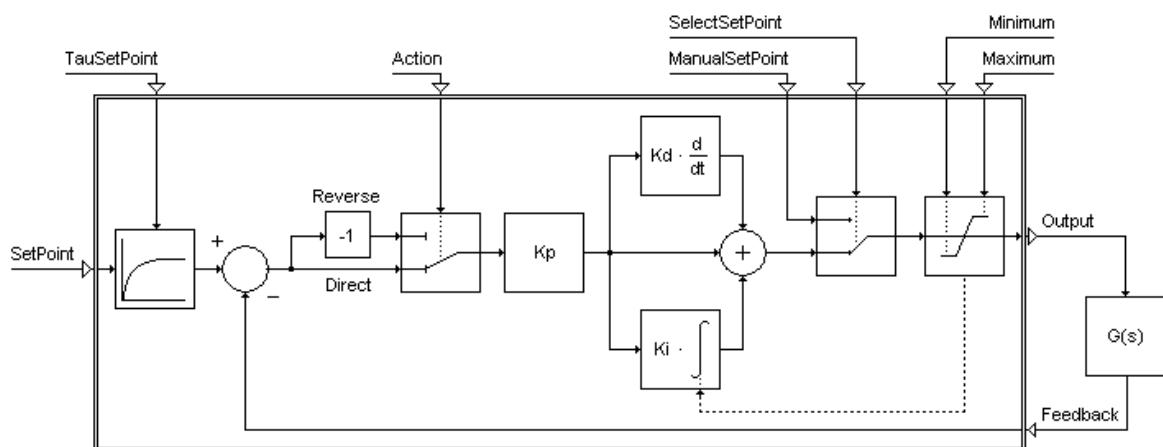
- Acadêmico:

SPf: "SetPoint" após o filtro

Se "Action"="Direct", então: $E(k) = SPf(k) - Feedback(k)$

Se "Action"="Reverse", então: $E(k) = Feedback(k) - SPf(k)$

$$Output(k) = Output(k-1) + K_p \cdot ((1 + K_i \cdot Ts + (K_d / Ts)) \cdot E(k) - (K_d / Ts) \cdot E(k-1))$$



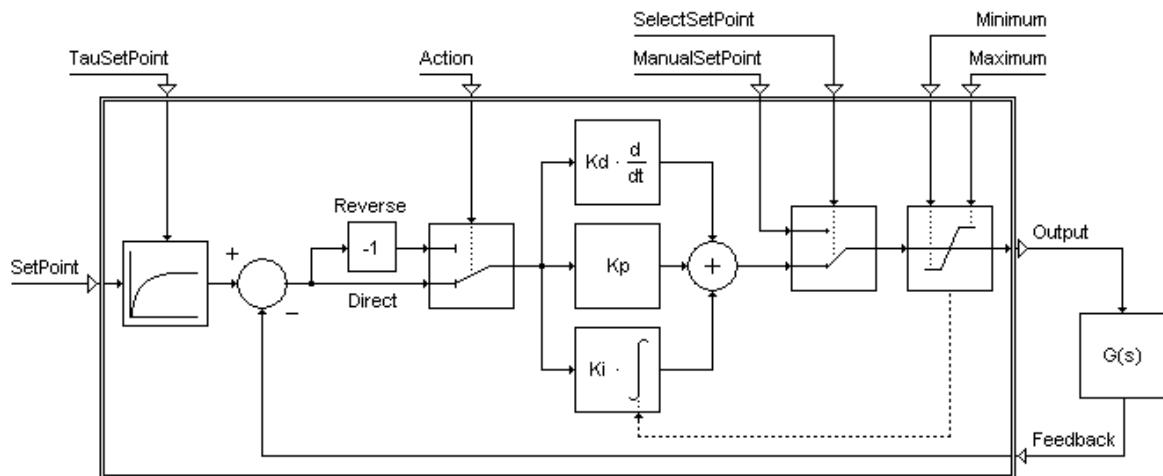
- Paralelo:

SPf: "SetPoint" após o filtro

Se "Action"="Direct", então: $E(k) = SPf(k) - Feedback(k)$

Se "Action"="Reverse", então: $E(k) = Feedback(k) - SPf(k)$

$$Output(k) = Output(k-1) + (K_p + K_i \cdot T_s + (K_d / T_s)) \cdot e(k) - (K_d / T_s) \cdot e(k-1)$$



Efeito da alteração dos ganhos sobre o processo

- Se diminuir K_p , o processo torna-se mais lento; geralmente mais estável ou menos oscilante; tem menos sobre-sinal (overshoot).
- Se aumentar K_p , o processo responde mais rápido; pode ficar instável ou mais oscilante; tem mais sobre-sinal (overshoot).
- Se diminuir K_i , o processo torna-se mais lento, demorando para atingir o "SetPoint"; fica mais estável ou menos oscilante; tem menos sobre-sinal (overshoot).
- Se aumentar K_i , o processo torna-se mais rápido, atingindo rapidamente o "SetPoint"; fica mais instável ou mais oscilante; tem mais sobre-sinal (overshoot).
- Se diminuir K_d , o processo torna-se mais lento; tem menos sobre-sinal (overshoot).
- Se aumentar K_d , tem mais sobre-sinal (overshoot).

Como melhorar o desempenho do processo pelo ajuste dos ganhos (válido para o PID Acadêmico)

- Se o desempenho do processo está quase bom, mas o sobre-sinal (overshoot) está um pouco alto, tente: (1) diminuir o K_p em 20%, (2) diminuir K_i em 20% e/ou (3) diminuir K_d em 50%.
- Se o desempenho do processo está quase bom, mas não tem o sobre-sinal (overshoot) e demora para atingir o "SetPoint", tente: (1) aumentar K_p em 20%, (2) aumentar K_i em 20%, (3) aumentar K_d em 50%.
- Se o desempenho do processo está bom, mas a saída do processo está variando

demais, tente: (1) aumentar KD em 50%, (2) diminuir Kp em 20%.

- Se o desempenho do processo está ruim, ou seja, após a partida, o transitório dura vários períodos de oscilação, que reduz muito lentamente ou não reduz, tente: (1) diminuir Kp em 50%.

- Se o desempenho do processo está ruim, ou seja, após a partida avança lentamente em direção ao "SetPoint", sem sobre-sinal (overshoot), mas ainda está muito longe e a saída do processo é menor que o valor nominal, tente: (1) aumentar Kp em 50%, (2) aumentar Ki em 50%, (3) aumentar Ki em 70%.

Disponível

- PLC300 : a partir da versão inicial

6.13.17 Filter

6.13.17.1 LOWPASS

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Input" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: REAL

Descrição: Entrada do filtro.

<arg1>: "Tau" - VAR_IN: inserir uma constante ou variável (tag).

Tipos de dados: REAL

Descrição: Constante de tempo do filtro.

<arg2>: "Ts#" - VAR_IN: inserir uma constante.

Tipos de dados: UINT

Descrição: Período de amostragem do filtro [ms].

<arg3>: "Output" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: REAL

Descrição: Saída do filtro.

Nota: A variável tem que ter permissão de escrita.

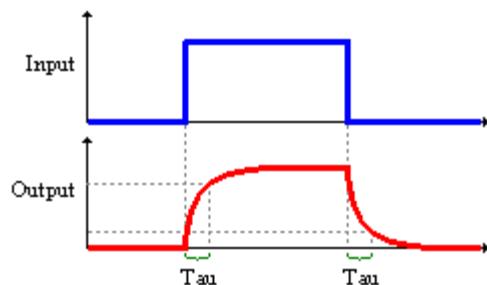
Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Se "EN"=0, "Output" permanece inalterado e a saída "ENO"=0. Na borda de subida de "EN", "Output"=0.

Diagrama de Funcionamento



Disponível

- PLC300 : a partir da versão inicial

6.13.18 Hardware

6.13.18.1 READENC

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".

Tipos de dados: BYTE

Descrição: Fonte de contagem

<arg1>: "Value" - VAR_OUTN: inserir uma variável (tag).

Tipos de dados: DWORD ou UDINT

Descrição: Valor de contagem

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura do número de pulsos, conforme "Source" e transferidos para "Value". A cada transição de subida em "EN", o conteúdo de "Value" é inicializado em 0.

Se "EN" for 0, o valor em "Value" permanece inalterado.

Disponível

- PLC300 : a partir da versão inicial

6.13.18.2 READENC2

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".

Tipos de dados: BYTE

Descrição: Fonte de contagem

<arg1>: "Pulses#" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: DWORD ou UDINT

Descrição: Número de pulsos para reiniciar contagem.

<arg2>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: DWORD ou UDINT

Descrição: Valor de contagem

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura do número de pulsos, conforme "Source" e transferidos para "Value". A cada transição de subida em "EN", o conteúdo de "Value" é inicializado em 0. Além disso, toda vez que a contagem for crescendo e "Value" atingir "Pulses", "Value" é reiniciado para 0. Porém, se a contagem decrescer, ou seja, "Value" tender a ficar menor que 0, então "Value" vai para "Pulses".

Se "EN" for 0, o valor em "Value" permanece inalterado.

Disponível

- PLC300 : a partir da versão inicial

6.13.18.3 READENC3

Figura



Configuração do Bloco

<arg0>: "Source#" - VAR_IN: selecionar uma das seguintes opções: "Quadrature_AB"; "Pulse_A"; "Pulse_B"; "Pulse_Z".

Tipos de dados: BYTE

Descrição: Valor da contagem.

<arg1>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: DWORD ou UDINT

Descrição: Número de pulsos contados pelo PLC300.

Nota: A variável tem que ter permissão de escrita.

<arg2>: "Delta" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: DINT

Descrição: Diferença entre o número de pulsos contados nessa execução do bloco com a execução anterior.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feita a leitura do número de pulsos contados desde quando o PLC300 foi energizado, conforme "Source#" e transferidos para "Value". O valor de "Delta" será a diferença entre os pulsos contados nessa execução do bloco com a execução anterior, ou seja, se o bloco foi executado por uma TASK de evento externo DI9 e depois executado por uma TASK de evento externo DI10, o valor de "Delta" será o número de pulsos contados entre o acionamento da DI9 e DI10.

Se "EN" for 0, o valor em "Value" e "Delta" permanecem inalterados.

Disponível

- PLC300 : a partir da versão 1.20

6.13.18.4 PWM

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "Frequency" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: DWORD ou UDINT

Descrição: Freqüência do PWM - 0 a 300000 Hz

<arg1>: "Value" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Largura do pulso - 0 a 1000 (onde 1000 equivale a 100%)

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Pelo fato deste bloco utilizar a saída digital 9 (DO9), quando em execução ("EN"=1), é escrito nesta saída um PWM conforme a "Frequency" e "Width".

IMPORTANTE: Este bloco ativo ("EN"=1) tem prioridade de comando em relação a qualquer bobina que tente comandar a saída digital 9.

Se "EN" for 0, o valor em "Value" permanece inalterado.

Disponível

- PLC300 : a partir da versão inicial

6.13.18.5 IMMEDIATE_INPUT

Figura



Configuração do Bloco

<arg0>: "Source#" - VAR_IN: inserir uma das seguintes opções: "DI01... DI10";

"AI01".

Tipos de dados: BYTE

Descrição: Origem dos dados (DI1 à DI10 ou AI1).

<arg1>: "Destination" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD, INT ou UINT

Descrição: Valor instantâneo da(s) entrada(s).

Nota: A variável tem que ter permissão de escrita.

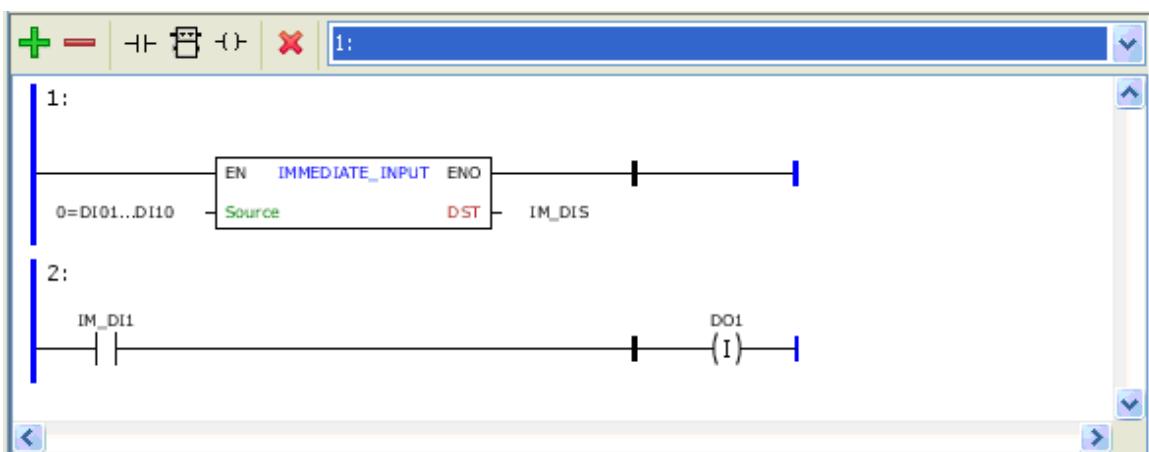
Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), é feito a leitura instantânea das entradas DI1 à DI10 ou AI1, conforme "Source#" e transferidos para "Destination", sem alterar as respectivas imagens (entradas digitais ou entrada analógica). As imagens somente são atualizadas no início do próximo SCAN do programa.

Para ler uma entrada digital individualmente, basta criar uma variável no mesmo endereço da variável "arg1" e programar o bit desejado. No exemplo abaixo, "IM_DIS" é a variável "arg1"(WORD) e os bits IM_DI1 a IM_DI10, são booleanos mapeados sobre a IM_DIS.



Se "EN" for 0, o valor em "Destination" permanece inalterado.

Disponível

- PLC300 : a partir da versão 1.20

6.13.18.6 IMMEDIATE_OUTPUT

Figura



Configuração do Bloco

<arg0>: "Source" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD, INT ou UINT
Descrição: Valor de escrita.

<arg1>: "Destination#" - VAR_OUT: inserir uma das seguintes opções: "DO01 à DO9" ou "AO1".
Tipos de dados: BYTE
Descrição: Destino de escrita.

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

A saída "ENO" retorna para o STACK o valor de "EN".

Em execução ("EN"=1), o valor de "Source" é transferido imediatamente para as saídas digitais DO01 à DO9 ou AO1, conforme "Destination#", sem aguardar o fim do SCAN do programa. A imagem da saída selecionada também é alterada.

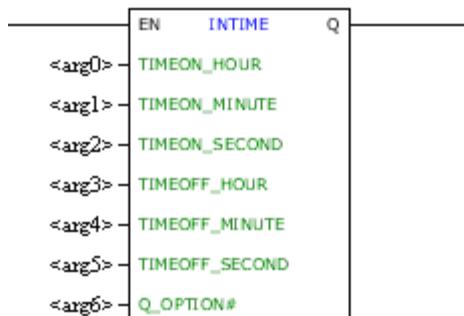
Disponível

- PLC300 : a partir da versão 1.20

6.13.19 RTC

6.13.19.1 INTIME

Figura



Configuração do Bloco

<arg0>: "TIMEON_HOUR" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Hora para ligar

<arg1>: "TIMEON_MINUTE" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Minuto para ligar.

<arg2>: "TIMEON_SECOND" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Segundo para ligar

<arg3>: "TIMEOFF_HOUR" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Hora para desligar

<arg4>: "TIMEOFF_MINUTE" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Minuto para desligar.

<arg5>: "TIMEOFF_SECOND" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Segundo para desligar

<arg6>: "Q_OPTION#" - VAR_IN: inserir uma das seguintes opções: Normal;
Invertido
Tipos de dados: BYTE
Descrição: Opção de funcionamento da saída "Q".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1) e se "Q_OPTION"=Normal, a saída "Q" somente vai a 1 quando a hora atual do relógio for maior ou igual ao horário configurado em "TIMEON_HOUR", "TIMEON_MINUTE" e "TIMEON_SECOND" e menor que o horário configurado em "TIMEOFF_HOUR", "TIMEOFF_MINUTE" e "TIMEOFF_SECOND".
Senão, a saída "Q" fica em 0.

Em execução ("EN"=1) e se "Q_OPTION"=Inverted, então o funcionamento da saída "Q" é o oposto da descrição do parágrafo anterior.

Se "EN" for 0 então a saída "Q" também será 0.

Exemplo ("EN"=1 e "Q_OPTION#"=Normal)

"TIMEON_HOUR"=14
"TIMEON_MINUTE"=00
"TIMEON_SECOND"=00

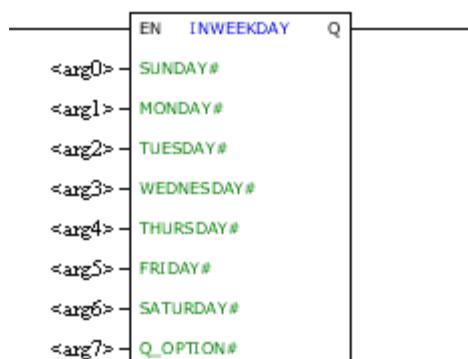
```
"TIMEOFF_HOUR"=16  
"TIMEOFF_MINUTE"=00  
"TIMEOFF_SECOND"=00  
Se a hora corrente=13:59:59, então "Q"=0  
Se a hora corrente=14:00:00, então "Q"=1  
Se a hora corrente=16:00:00, então "Q"=0
```

Exemplo ("EN"=1 e "Q_OPTION#"=Inverted)

```
"TIMEON_HOUR"=14  
"TIMEON_MINUTE"=00  
"TIMEON_SECOND"=00  
"TIMEOFF_HOUR"=16  
"TIMEOFF_MINUTE"=00  
"TIMEOFF_SECOND"=00  
Se a hora corrente=13:59:59, então "Q"=1  
Se a hora corrente=14:00:00, então "Q"=0  
Se a hora corrente=16:00:00, então "Q"=1
```

Disponível

- PLC300 : a partir da versão inicial

6.13.19.2 INWEEKDAY**Figura****Configuração do Bloco**

<arg0>: "SUNDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg1>: "MONDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.
Tipos de dados: BOOL

<arg2>: "TUESDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.

Tipos de dados: BOOL

<arg3>: "WEDNESDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.

Tipos de dados: BOOL

<arg4>: "THURSDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.

Tipos de dados: BOOL

<arg5>: "FRIDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.

Tipos de dados: BOOL

<arg6>: "SATURDAY#" - VAR_IN: inserir uma das seguintes opções: Habilitado; Desabilitado.

Tipos de dados: BOOL

<arg7>: "Q_OPTION#" - VAR_IN: inserir uma das seguintes opções: Normal; Invertido

Tipos de dados: BYTE

Descrição: Opção de funcionamento da saída "Q".

Descrição de Funcionamento

O bloco somente é executado quando a entrada "EN" for 1.

Em execução ("EN"=1) e se "Q_OPTION"=Normal, a saída "Q" somente vai a 1 quando o Dia da Semana: "SUNDAY#", "MONDAY#", "TUESDAY#", "WEDNESDAY#", "THURSDAY#", "FRIDAY#" ou "SATURDAY#" estiver "Enabled" e o dia da semana corrente do relógio for o mesmo. Senão, a saída "Q" fica em 0.

Em execução ("EN"=1) e se "Q_OPTION"=Inverted, então o funcionamento da saída "Q" é o oposto da descrição do parágrafo anterior.

Se "EN" for 0 então a saída "Q" também será 0.

Exemplo1 ("EN"=1 e "Q_OPTION"=Normal)

"SUNDAY#"=0
"MONDAY#"=1
"TUESDAY#"=1
"WEDNESDAY#"=1
"THURSDAY#"=1
"FRIDAY#"=1
"SATURDAY#"=0

Se o dia corrente =Domingo, então "Q"=0

Se o dia corrente =Terça, então "Q"=1

Se o dia corrente =Sábado, então "Q"=0

Exemplo2 ("EN"=1 e "Q_OPTION"=Inverted)

"SUNDAY#"=0
"MONDAY#"=1

```
"TUESDAY#"=1  
"WEDNESDAY#"=1  
"THURSDAY#"=1  
"FRIDAY#"=1  
"SATURDAY#"=1  
Se o dia corrente =Domingo, então "Q"=1  
Se o dia corrente =Terça, então "Q"=0  
Se o dia corrente =Sábado, então "Q"=1
```

Disponível

- PLC300 : a partir da versão inicial

6.13.20 Screen

6.13.20.1 SETSCREEN

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NUMBER" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Número da Tela a ser mostrada na HMI.

Descrição de Funcionamento

O bloco somente é executado na transição de subida da entrada "EN".

Na transição de subida de EN, a saída "Q" vai para 1 somente por 1 ciclo de SCAN, desde que a número da tela seja válido. Em qualquer outra condição, a saída "Q" fica em 0.

Nota: Ao utilizar este bloco, verificar a necessidade ou não de ter senha na TELA configurada. Caso a senha tenha sido programada, ao mudar para a TELA utilizando este bloco, o sistema irá solicitar a devida senha.

Disponível

- PLC300 : a partir da versão inicial

6.13.21 Rede de Comunicação

6.13.21.1 CANopen

6.13.21.1.1 Visão Geral do CANopen

Operação na Rede CANopen - Modo Mestre

Além da operação como escravo, o controlador programável PLC300 também permite a operação como mestre da rede CANopen. A seguir serão descritas características e funções do PLC300 como mestre da rede CANopen.

Habilitação da Função CANopen Mestre

Como padrão, o controlador programável PLC300 está programado para operar como escravo da rede CANopen. A programação do equipamento como mestre da rede deve ser feita utilizando o software WSCAN, que permite também a configuração de toda a rede CANopen. A descrição detalhada das janelas e funções do software WSCAN deve ser obtida no menu "Ajuda" do próprio software.

Depois de elaborada a configuração do mestre, é necessário fazer o download das configurações, utilizando uma das interfaces de programação do produto - consulte o manual do usuário para maiores informações. Uma vez programado como mestre da rede, caso seja necessário apagar estas configurações, a função para apagar o programa do usuário - disponível no menu Setup - também apaga as configurações do mestre CANopen.

Nota: A rede CANopen é uma rede flexível e que permite várias formas de configuração e operação. No entanto, esta flexibilidade exige que o usuário tenha bom conhecimento das funções e objetos de comunicação utilizados para configuração da rede, bem como o conhecimento do software de programação WSCAN.

Características do Mestre CANopen

O controlador programável PLC300 permite controlar um conjunto de até 63 escravos, utilizando os seguintes serviços e recursos de comunicação:

- Serviço de gerenciador da rede (NMT)
- 63 PDOs de transmissão
- 63 PDOs de recepção
- 63 Consumidores Heartbeat
- Produtor Heartbeat
- Cliente SDO
- Produtor/consumidor SYNC
- 512 bytes de marcadores de rede de entrada (input)
- 512 bytes de marcadores de rede de saída (output)

As características físicas - instalação, conector, cabo, etc. - são as mesmas, tanto para o PLC300 operando como mestre quanto como escravo. As configurações de endereço e taxa de comunicação também são necessárias para a operação como mestre, mas estas configurações são programadas pelo software WSCAN de acordo

com as propriedades definidas para o mestre no próprio software.

Nota: Os marcadores de rede de entrada são utilizados para mapear dados nos RPDOs, enquanto que os marcadores de rede de saída são utilizados para mapear dados nos TPDOs. Eles podem ser acessados em Byte (%IB ou %QB), Word (%IW ou %QW) ou Double Word (%ID ou %QD). Sua função, no entanto, não é pré-definida, e depende do aplicativo em ladder desenvolvido para o controlador PLC300.

Operação do Mestre

Uma vez programado para operar como mestre, o controlador programável PLC300 executará as seguintes etapas para realizar a inicialização, em seqüência, para cada um dos escravos:

- 1 - Enviado o comando de reset da comunicação para toda a rede, para que os escravos iniciem com valores conhecidos para os objetos de comunicação.
- 2 - Identificação do equipamento na rede, através da leitura via SDO do objeto 1000h/00h - Object Identification.
- 3 - Escrita via SDO de todos os objetos programados para o escravo, que usualmente inclui a configuração e mapeamento dos TPDOs e RPDOs, node guarding, heartbeat, além dos objetos específicos do fabricante, caso sejam programados.
- 4 - Iniciado serviço de controle de erros - node guarding ou heartbeat - caso sejam programados.
- 5 - Envio do escravo para modo operacional.

Se uma destas etapas falhar, será indicado erro de comunicação com o escravo. Dependendo das configurações, a inicialização dos escravos será abortada, e o mestre fará a inicialização do escravo seguinte, retornando para o escravo com erro após tentar inicializar todos os demais escravos da rede.

De forma semelhante, se, durante a operação de um escravo, for identificado erro no serviço de controle de erros, dependendo das configurações feitas para o mestre, o escravo será automaticamente resetado e o procedimento de inicialização será executado novamente.

Nota: O estado da comunicação e o estado de cada escravo podem ser observados em marcadores de sistema de entrada.

Blocos para o Mestre CANopen

Além dos objetos de comunicação e das configurações feitas no software WSCAN, também estão disponíveis blocos para monitoração e envio de comandos, que podem ser utilizados durante a elaboração do aplicativo em ladder para o controlador programável PLC300. Não é necessário utilizar estes blocos durante a operação do equipamento, mas seu uso confere maior flexibilidade e facilita o diagnóstico de problemas de comunicação durante a operação do controlador programável PLC300.

6.13.21.1.2 CO_SDORead

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NodeID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 127.

<arg1>: "Index#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Índice do objeto acessado, dentre os objetos disponíveis no dicionário de objetos do escravo - 0 a 65535.

<arg2>: "SubIndex#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Sub-índice do objeto acessado - 0 a 255.

<arg3>: "Size#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Tamanho do dado acessado, em bytes - 1 a 4.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 5 a 5000 ms.

<arg5>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg6>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (cliente SDO enviando outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Cartão não pode executar a função" (exemplo: mestre não habilitado); 2= "Timeout na resposta do escravo"; 3= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

<arg9>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 4] ou USINT[1 ... 4]

Descrição: Variável ou array onde serão salvos os dados lidos do escravo.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de dados via SDO de um escravo remoto. Permite realizar a leitura de objetos na rede com tamanho de até 4 bytes.

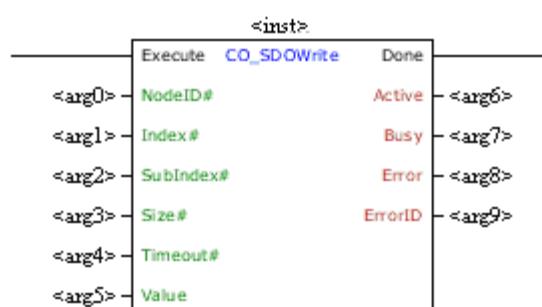
Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução com sucesso da função. Na transição positiva de "Execute", quando o cliente SDO do mestre estiver livre, uma nova requisição é enviada para o servidor SDO do escravo. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Disponível

- PLC300 : a partir da versão inicial

6.13.21.1.3 CO_SDOWrite

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "NodeID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 127.

<arg1>: "Index#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Índice do objeto acessado, dentre os objetos disponíveis no dicionário de objetos do escravo - 0 a 65535.

<arg2>: "SubIndex#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Sub-índice do objeto acessado - 0 a 255.

<arg3>: "Size#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Tamanho do dado acessado, em bytes - 1 a 4.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 5 a 5000 ms.

<arg5>: "Value" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 4] ou USINT[1 ... 4]

Descrição: Variável ou array com dados para enviar para o escravo.

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (cliente SDO enviando outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados

possíveis: 0= "Executado com sucesso"; 1= "Cartão não pode executar a função" (exemplo: mestre não habilitado); 2= "Timeout na resposta do escravo"; 3= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de dados via SDO para um escravo remoto. Permite realizar a escrita de objetos na rede com tamanho de até 4 bytes.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução com sucesso da função. Na transição positiva de "Execute", quando o cliente SDO do mestre estiver livre, uma nova requisição é enviada para o servidor SDO do escravo. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Nota

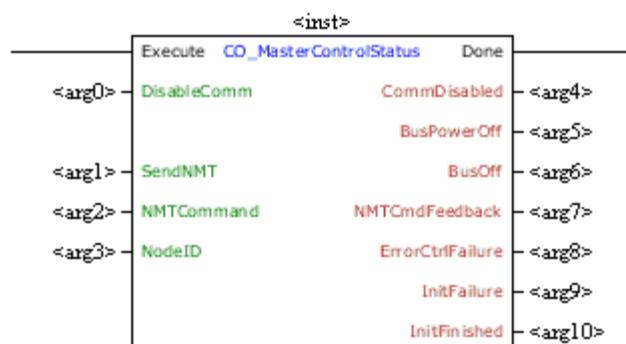
- É importante que a quantidade de dados lidos ou escritos programado nos blocos seja compatível com o tamanho da variável ou do array com o valor,
- Em caso de erro retornado pelo escravo, é possível obter o código do último erro recebido através dos marcadores de sistema de leitura. Consulte o item "Marcadores de Sistema do CAN" para a lista de marcadores disponíveis.

Disponível

- PLC300 : a partir da versão inicial

6.13.21.1.4 CO_MasterControlStatus

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "DisableComm" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BOOL

Descrição: Desabilita a comunicação CANopen. Ao desabilitar o mestre, os contadores e marcadores de status do mestre CANopen também são zerados - 0 ou 1.

<arg1>: "SendNMT" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BOOL

Descrição: Na transição deste sinal o mestre CANopen dispara o envio de um comando de gerenciamento - NMT - de acordo com o comando e o endereço programado neste bloco - 0 ou 1.

<arg2>: "NMTCommand" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BYTE

Descrição: Indica qual comando deve ser enviado para o escravo: 1= "Start node"; 2= "Stop node"; 128= "Enter pre-operational"; 129= "Reset node"; 130= "Reset communication".

<arg3>: "NodeID" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Endereço do escravo para envio do comando NMT - 0= Broadcast (mensagem para todos os escravos); 1 a 127= Endereço específico do escravo.

<arg4>: "CommDisabled" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre e a comunicação na interface CAN foi desabilitada. É indicado sempre que o comando do usuário para desabilitar a interface for recebido, mas também é indicado nas situações de falta de alimentação na interface CAN ou bus off: 0= "Comunicação habilitada"; 1= "Comunicação desabilitada".

Nota: A variável tem que ter permissão de escrita.

<arg5>: "BusPowerOff" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que foi detectado falha na alimentação da interface CAN: 0= "Interface CAN alimentada"; 1= "Interface CAN sem alimentação".

Nota: A variável tem que ter permissão de escrita.

<arg6>: "BusOff" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que foi detectado erro de bus off na interface CAN: 0= "Sem erro de bus off"; 1= "Com erro de bus off".

Nota: A variável tem que ter permissão de escrita.

<arg7>: "NMTCmdFeedback" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o comando de gerenciamento foi enviado pelo mestre: 0= "Sem comando ou comando não enviado"; 1= "Comando NMT enviado".

Nota: A variável tem que ter permissão de escrita.

<arg8>: "ErrorCtrlFailure" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre detectou erro no serviço de controle de erros (node guarding ou heartbeat) em pelo menos um escravo da rede: 0= "Sem erro

detectado"; 1= "Mestre detectou erro de node guarding ou heartbeat em pelo menos um escravo da rede".

Nota: A variável tem que ter permissão de escrita.

<arg9>: "InitFailure" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre detectou erro durante a inicialização de pelo menos um escravo da rede: 0= "Sem erro detectado"; 1= "Mestre detectou erro na inicialização em pelo menos um escravo da rede".

Nota: A variável tem que ter permissão de escrita.

<arg10>: "InitFinished" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre tentou fazer a inicialização de todos os escravos da rede. Não necessariamente a inicialização foi executada com sucesso, podem ter ocorrido erros durante a inicialização: 0= "Mestre ainda não executou o procedimento de inicialização de todos os escravos"; 1= "Mestre fez a inicialização (com ou sem sucesso) de todos os escravos".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para controle e monitoração do mestre da rede CANopen. Mostra o estado do mestre da rede para diagnóstico e identificação de problemas de comunicação, bem como permite o envio de comandos para os serviços de gerenciamento da rede - NMT.

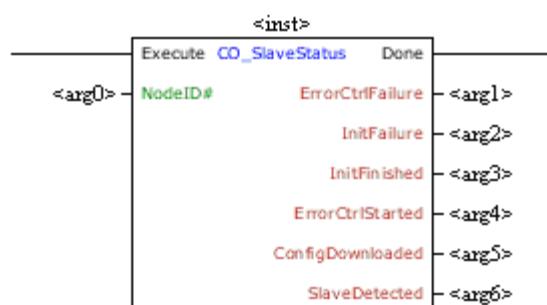
Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Se a entrada "Execute" está ativa, ele atualiza os valores das entradas e saídas e ativa a saída "Done". Se a entrada "Execute" não estiver ativa, os demais valores das entradas são ignorados e todas as saídas são zeradas.

Disponível

- PLC300 : a partir da versão inicial

6.13.21.1.5 CO_SlaveStatus

Figura



Configuração do Bloco

<arg0>: "NodeID" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Endereço do escravo para identificação do estado da comunicação com o mestre - 1 a 127.

<arg1>: "ErrorCtrlFailure" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre detectou erro no serviço de controle de erros (node guarding ou heartbeat) no escravo indicado: 0= "Sem erro detectado"; 1= "Mestre detectou erro de node guarding ou heartbeat no escravo".

Nota: A variável tem que ter permissão de escrita.

<arg2>: "InitFailure" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre detectou erro durante a inicialização do escravo indicado: 0= "Sem erro detectado"; 1= "Mestre detectou erro na inicialização no escravo".

Nota: A variável tem que ter permissão de escrita.

<arg3>: "InitFinished" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre fez a inicialização completa e com sucesso do escravo indicado: 0= "Mestre não concluiu o procedimento de inicialização do escravo"; 1= "Mestre fez a inicialização do escravo com sucesso".

Nota: A variável tem que ter permissão de escrita.

<arg4>: "ErrprCtrlStarted" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre iniciou o serviço de controle de erros (node guarding ou heartbeat) com o escravo indicado. Se este serviço não for habilitado para o escravo, este bit será ativado após realizar a configuração: 0= "Controle de erros com o escravo não iniciado"; 1= "Controle de erros com o escravo iniciado".

Nota: A variável tem que ter permissão de escrita.

<arg5>: "ConfigDownloaded" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre finalizou com sucesso o download das configurações via SDO para o escravo indicado: 0= "Mestre não finalizou o download das configurações para o escravo"; 1= "Download das configurações para o escravo finalizado com sucesso".

Nota: A variável tem que ter permissão de escrita.

<arg6>: "SlaveDetected" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica que o mestre conseguiu fazer a leitura da identificação via SDO do escravo indicado: 0= "Escravo não foi contactado"; 1= "Escravo contactado com sucesso".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para monitoração do escravo da rede CANopen. Mostra o estado do escravo da rede para diagnóstico e identificação de problemas de comunicação.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Se a entrada "Execute" está ativa, ele atualiza os valores das entradas e saídas e ativa a saída "Done". Se a entrada "Execute" não estiver ativa, os demais valores das entradas são ignorados e todas as saídas são zeradas.

Nota

- Os dados acessados utilizando este bloco também estão disponíveis através de marcadores de sistema de leitura e escrita, conforme descrito no item "Marcadores de Sistema do CAN".

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2 Modbus RTU

6.13.21.2.1 Visão Geral do Modbus RTU

Operação na Rede Modbus RTU - Modo mestre

Além da operação como escravo, o controlador programável PLC300 também permite a operação como mestre da rede Modbus RTU. Para esta operação, é necessário observar os seguintes pontos:

- Somente a interface RS485 permite operação como mestre da rede.
- É necessário programar, nas configurações do produto, o modo de operação como "Mestre", além da taxa de comunicação, paridade e stop bits, que devem ser as mesmas para todos os equipamentos da rede.
- O mestre da rede Modbus RTU não possui endereço, logo o endereço configurado no PLC300 não é utilizado.
- O envio e recepção de telegramas via interface RS485 utilizando o protocolo Modbus RTU é programado utilizando blocos em linguagem de programação ladder. É necessário conhecer os blocos disponíveis e o software de programação em ladder para poder programar o mestre da rede.
- As seguintes funções estão disponíveis para envio de requisições pelo mestre Modbus:
 - Função 01: Read Coils
 - Função 02: Read Discrete Inputs
 - Função 03: Read Holding Registers
 - Função 04: Read Input Registers
 - Função 05: Write Single Coil
 - Função 06: Write Single Register
 - Função 15: Write Multiple Coils
 - Função 16: Write Multiple Registers

Blocos para a programação do mestre

Para o controle e monitoração da comunicação Modbus RTU utilizando o controlador programável PLC300, foram desenvolvidos os seguintes blocos, que devem ser utilizados durante a programação em ladder.

6.13.21.2.2 MB_ReadBinary

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de leitura: 1= "Read Coils"; 2= "Read Discrete Inputs".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do bit inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de bits lidos em sequência a partir do endereço inicial - 1 a 128.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com

Offset de 1".

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

<arg10>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL[1 ... 128]

Descrição: Variável ou array onde serão salvos os dados lidos do escravo.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de bits. Permite fazer a leitura de até 128 bits em sequência do escravo destino, utilizando as funções 1 (Read Coils) e 2 (Read Discrete Inputs) do Modbus.

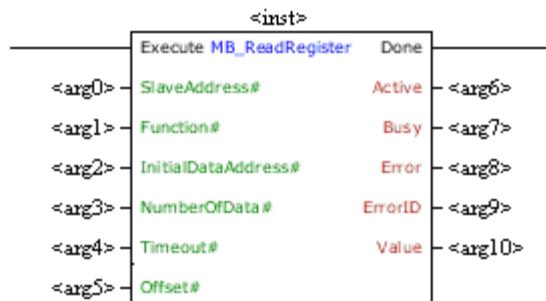
Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2.3 MB_ReadRegister

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de leitura: 3= "Read Holding Registers"; 4= "Read Input Registers".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do registrador inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de registradores lidos a partir do endereço inicial - 1 a 8.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de leitura enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

<arg10>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 16], SINT[1 ... 16], USINT[1 ... 16], WORD[1 ... 8], UINT

[1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1 ... 4]

Descrição: Variável ou array onde serão salvos os dados lidos do escravo.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de registradores de 16 bits. Permite fazer a leitura de até 8 registradores em sequência do escravo destino, utilizando as funções 3 (Read Holding Registers) e 4 (Read Input Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Nota

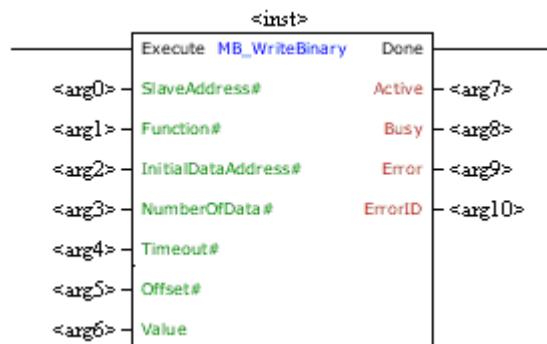
- O protocolo Modbus RTU, utilizando as funções 3 e 4, permite a leitura de registradores de 16 bits apenas; para leitura de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a leitura de múltiplos registradores, e salvar o valor em uma variável com tamanho maior que 16 bits, e
- É importante que a quantidade de registradores lidos seja compatível com o tamanho da variável ou do array onde os dados serão salvos.

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2.4 MB_WriteBinary

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de escrita: 5= "Write Single Coil"; 15= "Write Multiple Coils".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do bit inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de bits escritos em sequência a partir do endereço inicial - 1 a 128.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#"

possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Value" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL[1 ... 128]

Descrição: Variável ou array com os dados que serão escritos no escravo.

<arg7>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg10>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de bits. Permite fazer a escrita de até 128 bits utilizando as funções 5 (Write Single Coil) e 15 (Write Multiple Coils) do Modbus.

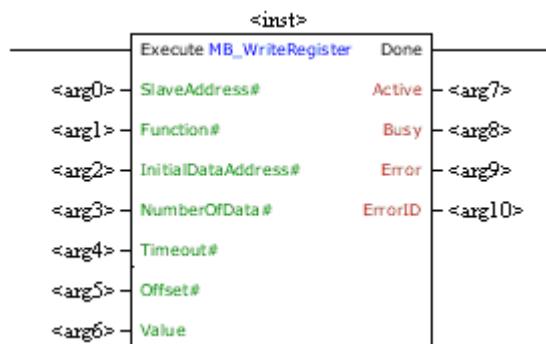
Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2.5 MB_WriteRegister

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "SlaveAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Endereço do escravo destino - 1 a 247.

<arg1>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de escrita: 6= "Write Single Register"; 16= "Write Multiple Registers".

<arg2>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do registrador inicial - 0 a 65535.

<arg3>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de registradores escritos a partir do endereço inicial - 1 a 8.

<arg4>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do escravo, a partir do início do envio pelo mestre - 20 a 5000 ms.

<arg5>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg6>: "Value" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 16], USINT[1 ... 16], SINT[1 ... 16], WORD[1 ... 8], UINT[1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1 ... 4]

Descrição: Variável ou array com os dados que serão escritos no escravo.

<arg7>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de escrita enviada para o escravo e aguardando resposta.

Nota: A variável tem que ter permissão de escrita.

<arg8>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (interface RS485 ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável tem que ter permissão de escrita.

<arg9>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável tem que ter permissão de escrita.

<arg10>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Mestre não habilitado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de registradores. Permite fazer a escrita de um ou mais registradores utilizando a função 6 (Write Holding Register) ou 16 (Write Multiple Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo mestre Modbus RTU quando a interface serial RS485 estiver livre. Ao término com sucesso da operação - resposta recebida do escravo - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

Nota

- O protocolo Modbus RTU, utilizando a função 16, permite a escrita de registradores de 16 bits apenas. Para escrita de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a escrita de múltiplos registradores, e utilizar como fonte dos dados uma variável com tamanho maior que 16 bits,
- É importante que a quantidade de registradores escritos seja compatível com o tamanho da variável ou do array de onde os dados serão utilizados.

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2.6 MB_MasterControlStatus

Figura



Configuração do Bloco

<arg0>: "DisableComm" - VAR_IN: inserir uma constante ou uma variável (tag).

Tipos de dados: BOOL

Descrição: Permite desabilitar o mestre Modbus. Ao desabilitar o mestre, os contadores e marcadores de status do mestre Modbus RTU também são zerados: 0= "Mestre em execução"; 1= "Desabilita mestre".

<arg1>: "CommDisabled" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Indica se o mestre está ou não desabilitado. Pode ocorrer por solicitação do usuário ou caso a interface esteja programada para operar como escravo da rede: 0= "Mestre habilitado"; 1= "Mestre desabilitado".

Nota: A variável tem que ter permissão de escrita.

<arg2>: "TxCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de requisições enviadas pelo mestre da rede para os escravos. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.

Nota: A variável tem que ter permissão de escrita.

<arg3>: "RxCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de telegramas recebidos pelo mestre da rede. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.

Nota: A variável tem que ter permissão de escrita.

<arg4>: "NoAnswerCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de requisições do mestre que não foram respondidas pelos

escravos. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.

Nota: A variável tem que ter permissão de escrita.

<arg5>: "ErrorResponseCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de requisições do mestre e que os escravos responderam com alguma resposta de erro. O código do erro pode ser obtido no marcador que indica o código do último erro detectado. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 65535.

Nota: A variável tem que ter permissão de escrita.

<arg6>: "LastErrorSlaveAddress" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Indica o endereço do escravo no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 247.

Nota: A variável tem que ter permissão de escrita.

<arg7>: "LastErrorMessage" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Indica o resultado da operação - timeout ou resposta de erro, conforme ERROR ID do bloco - para o escravo no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado: 0= "Sem erro detectado"; 4= "Timeout na resposta do escravo"; 5= "Escravo retornou erro".

Nota: A variável tem que ter permissão de escrita.

<arg8>: "LastErrorCode" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Indica o código do erro, no caso do mestre receber resposta de erro de algum escravo. É zerado sempre que o equipamento for desligado ou o mestre for desabilitado - 0 a 255.

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para controle e monitoração do mestre da rede Modbus RTU. Sempre que uma rede Modbus RTU for montada com o PLC300 como mestre da rede, recomenda-se utilizar este bloco para obter informações sobre o estado da comunicação.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Enquanto a entrada de habilitação "Execute" estiver ativa, os dados de entrada são utilizados e os dados de saída são atualizados. Caso a entrada seja zerada, os valores de entrada são desconsiderados e os argumentos de saída são zerados. A saída "Done" reflete o valor da entrada.

Nota

- Os dados acessados utilizando este bloco também estão disponíveis através de marcadores de sistema de leitura e escrita, conforme descrito no item "Marcadores de Sistema do Modbus".

Disponível

- PLC300 : a partir da versão inicial

6.13.21.2.7 MB_SlaveStatus

Figura



Configuração do Bloco

<arg0>: "ErrorsToSetOffline#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Permite programar, para este bloco, a quantidade de erros de comunicação que o mestre deve identificar até que a comunicação com um escravo da rede seja considerada offline. É considerado erro de comunicação toda requisição (leitura ou escrita) que o mestre enviou para um escravo e não recebeu resposta, a resposta recebida possuía erro de CRC ou o escravo retornou resposta de erro - 1 a 99.

<arg1>: "AddressSlave1#" - VAR_IN: inserir uma constante.

<arg2>: "AddressSlave2#" - VAR_IN: inserir uma constante.

<arg3>: "AddressSlave3#" - VAR_IN: inserir uma constante.

<arg4>: "AddressSlave4#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Permite programar o endereço de até 4 escravos, cuja quantidade de erros de comunicação serão monitorados para saber se estão online ou offline. Caso a quantidade de erros de comunicação em sequência, detectados nos blocos de leitura e escrita via Modbus, atinja o valor programado em "ErrorsToSetOffline", a saída respectiva é acionada. Caso deseje-se monitorar um número menor de escravos, pode-se deixar qualquer das entradas em zero: 0= "Ignora entrada"; 1 a 247.

<arg5>: "GeneralOffline#" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Se qualquer uma das saídas dos escravos indicados for acionada, esta saída também será acionada. Funciona como uma lógica OU entre as 4 saídas de indicação dos escravos - 0 a 65535.

Nota: A variável tem que ter permissão de escrita.

<arg6>: "Slave1Offline#" - VAR_OUT: inserir uma variável (tag).

<arg7>: "Slave2Offline#" - VAR_OUT: inserir uma variável (tag).

<arg8>: "Slave3Offline#" - VAR_OUT: inserir uma variável (tag).
<arg9>: "Slave4Offline#" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Saída acionada caso a quantidade de erros de comunicação em sequência para os escravos indicados nas respectivas entradas atinja o valor programado em "ErrorsToSetOffline".

Nota: A variável tem que ter permissão de escrita.

Descrição de Funcionamento

Bloco para monitoração dos escravos da rede Modbus RTU. Deve ser utilizado caso seja desejado identificar problemas na comunicação do mestre com algum escravo da rede Modbus RTU.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Enquanto a entrada de habilitação "Execute" estiver ativa os dados de entrada são utilizados e os dados de saída são atualizados a cada execução do bloco. A saída "Done" reflete o valor da entrada.

Disponível

- PLC300 : a partir da versão inicial

6.13.21.3 Modbus TCP

6.13.21.3.1 Visão Geral do Modbus TCP

Operação na Rede Modbus TCP - Modo Cliente

Além da operação como servidor, o controlador programável PLC300 também permite a operação como cliente da rede Modbus TCP. Para esta operação, é necessário observar os seguintes pontos:

- O envio e recepção de telegramas via interface Ethernet utilizando o protocolo Modbus TCP é programado utilizando blocos em linguagem de programação ladder. É necessário conhecer os blocos disponíveis e o software de programação em ladder para poder programar o cliente da rede.
- As seguintes funções estão disponíveis para envio de requisições pelo cliente Modbus TCP:
 - Função 01: Read Coils
 - Função 02: Read Discrete Inputs
 - Função 03: Read Holding Registers
 - Função 04: Read Input Registers
 - Função 05: Write Single Coil
 - Função 06: Write Single Register
 - Função 15: Write Multiple Coils
 - Função 16: Write Multiple Registers

Blocos para a Programação do Cliente

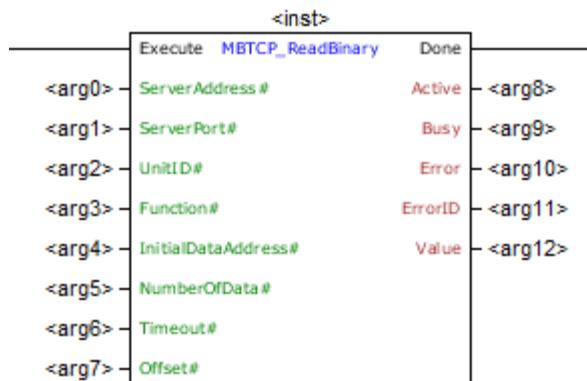
Para o controle e monitoração da comunicação Modbus TCP utilizando o controlador programável PLC300, foram desenvolvidos os seguintes blocos, que devem ser

utilizados durante a programação em ladder.

6.13.21.3.2 MBTCP_ReadBinary

Leitura de Bits

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "ServerAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: DWORD

Descrição: Endereço IP do servidor.

Formato: XXX.XXX.XXX.XXX, onde XXX é um número entre 0 e 255.

<arg1>: "ServerPort#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Porta Modbus TCP do servidor. O padrão é 502.

<arg2>: "UnitID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: UnitID do servidor. O padrão é 255.

<arg3>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de leitura: 1= "Read Coils"; 2= "Read Discrete Inputs".

<arg4>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do bit inicial - 0 a 65535.

<arg5>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de bits lidos em sequência a partir do endereço inicial - 1 a 128.

<arg6>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do servidor, a partir do início do envio pelo cliente - 20 a 5000 ms.

<arg7>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg8>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de leitura enviada para o servidor e aguardando resposta.

Nota: A variável deve possuir permissão de escrita.

<arg9>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (conexão ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável deve possuir permissão de escrita.

<arg10>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável deve possuir permissão de escrita.

<arg11>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Cliente não habilitado"; 4= "Timeout na resposta do servidor"; 5= "Servidor retornou erro".

Nota: A variável deve possuir permissão de escrita.

<arg12>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL[1 ... 128]

Descrição: Variável ou array onde serão salvos os dados lidos do servidor.

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de bits. Permite fazer a leitura de até 128 bits em sequência do servidor, utilizando as funções 1 (Read Coils) e 2 (Read Discrete Inputs) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo cliente Modbus TCP quando a conexão cliente estiver livre. Ao término com sucesso da operação - resposta

recebida do servidor - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

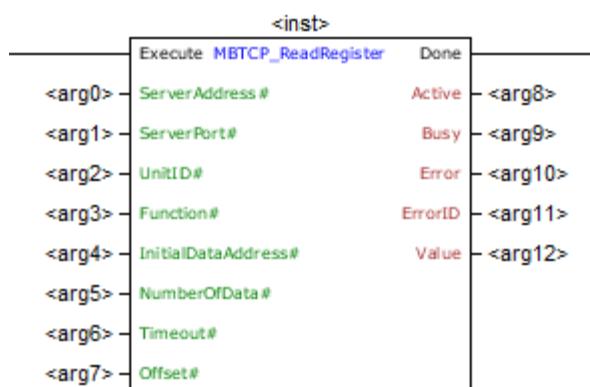
Disponível

- PLC300 : a partir da versão 1.30

6.13.21.3.3 MBTCP_ReadRegister

Leitura de Registradores

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "ServerAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: DWORD

Descrição: Endereço IP do servidor.

Formato: XXX.XXX.XXX.XXX, onde XXX é um número entre 0 e 255.

<arg1>: "ServerPort#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Porta Modbus TCP do servidor. O padrão é 502.

<arg2>: "UnitID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: UnitID do servidor. O padrão é 255.

<arg3>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de leitura: 3= "Read Holding Registers"; 4= "Read Input Registers".

<arg4>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do registrador inicial - 0 a 65535.

<arg5>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de registradores lidos a partir do endereço inicial - 1 a 8.

<arg6>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do servidor, a partir do início do envio pelo cliente - 20 a 5000 ms.

<arg7>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg8>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de leitura enviada para o servidor e aguardando resposta.

Nota: A variável deve possuir permissão de escrita.

<arg9>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (conexão ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável deve possuir permissão de escrita.

<arg10>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável deve possuir permissão de escrita.

<arg11>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Cliente não habilitado"; 4= "Timeout na resposta do servidor"; 5= "Servidor retornou erro".

Nota: A variável deve possuir permissão de escrita.

<arg12>: "Value" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 16], SINT[1 ... 16], USINT[1 ... 16], WORD[1 ... 8], UINT [1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1 ... 4]

Descrição: Variável ou array onde serão salvos os dados lidos do servidor.

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para leitura de registradores de 16 bits. Permite fazer a leitura de até 8 registradores em sequência do servidor, utilizando as funções 3 (Read Holding Registers) e 4 (Read Input Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo cliente Modbus TCP quando a conexão estiver livre. Ao término com sucesso da operação - resposta recebida do servidor - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa, e os dados recebidos são copiados para "Value". Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

NOTA!

O protocolo Modbus, utilizando as funções 3 e 4, permite a leitura de registradores de 16 bits apenas. Para leitura de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a leitura de múltiplos registradores, e salvar o valor em uma variável com tamanho maior que 16 bits.

É importante que a quantidade de registradores lidos seja compatível com o tamanho da variável ou do array onde os dados serão salvos.

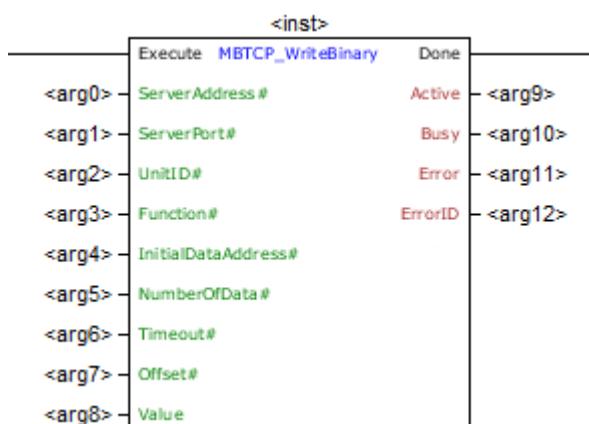
Disponível

- PLC300 : a partir da versão 1.30

6.13.21.3.4 MBTCP_WriteBinary

Escrita de Bits

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "ServerAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: DWORD

Descrição: Endereço IP do servidor.

Formato: XXX.XXX.XXX.XXX, onde XXX é um número entre 0 e 255.

<arg1>: "ServerPort#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Porta Modbus TCP do servidor. O padrão é 502.

<arg2>: "UnitID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: UnitID do servidor. O padrão é 255.

<arg3>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de escrita: 5= "Write Single Coil"; 15= "Write Multiple Coils".

<arg4>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do bit inicial - 0 a 65535.

<arg5>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de bits escritos em sequência a partir do endereço inicial - 1 a 128.

<arg6>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do servidor, a partir do início do envio pelo cliente - 20 a 5000 ms.

<arg7>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#" possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg8>: "Value" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BOOL[1 ... 128]

Descrição: Variável ou array com os dados que serão escritos no servidor.

<arg9>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de escrita enviada para o servidor e aguardando resposta.

Nota: A variável deve possuir permissão de escrita.

<arg10>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (conexão ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo

bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável deve possuir permissão de escrita.

<arg11>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável deve possuir permissão de escrita.

<arg12>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Cliente não habilitado"; 4= "Timeout na resposta do servidor"; 5= "Servidor retornou erro".

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de bits. Permite fazer a escrita de até 128 bits utilizando as funções 5 (Write Single Coil) e 15 (Write Multiple Coils) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo cliente Modbus TCP quando a conexão estiver livre. Ao término com sucesso da operação - resposta recebida do servidor - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

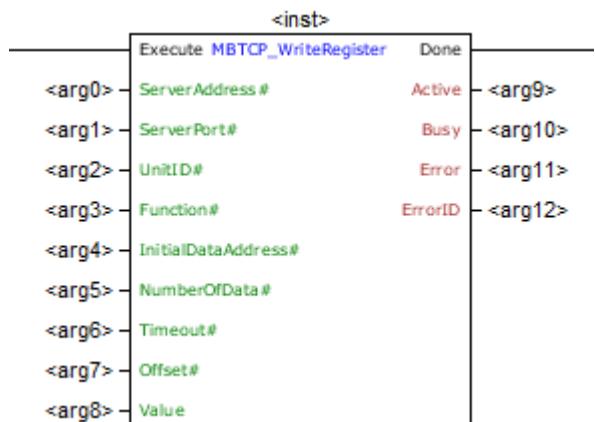
Disponível

- PLC300 : a partir da versão 1.30

6.13.21.3.5 MBTCP_WriteRegister

Escrita de Registradores

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "ServerAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: DWORD

Descrição: Endereço IP do servidor.

Formato: XXX.XXX.XXX.XXX, onde XXX é um número entre 0 e 255.

<arg1>: "ServerPort#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Porta Modbus TCP do servidor. O padrão é 502.

<arg2>: "UnitID#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: UnitID do servidor. O padrão é 255.

<arg3>: "Function#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Código da função de escrita: 6= "Write Single Register"; 16= "Write Multiple Registers".

<arg4>: "InitialDataAddress#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Endereço do registrador inicial - 0 a 65535.

<arg5>: "NumberOfData#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Número de registradores escritos a partir do endereço inicial - 1 a 8.

<arg6>: "Timeout#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Tempo de espera para chegada da resposta do servidor, a partir do início do envio pelo cliente - 20 a 5000 ms.

<arg7>: "Offset#" - VAR_IN: inserir uma constante.

Tipos de dados: BOOL

Descrição: Indica se o endereço do dado programado em "InitialDataAddress#"

possui offset, ou seja, se o endereço do dado programado no bloco deve ser subtraído de 1 para enviar pela rede Modbus: FALSE= "Sem Offset"; TRUE= "Com Offset de 1".

<arg8>: "Value" - VAR_IN: inserir uma variável (tag).

Tipos de dados: BYTE[1 ... 16], USINT[1 ... 16], SINT[1 ... 16], WORD[1 ... 8],
UINT[1 ... 8], INT[1 ... 8], DWORD[1 ... 4], UDINT[1 ... 4], DINT[1 ... 4] ou REAL[1
... 4]

Descrição: Variável ou array com os dados que serão escritos no escravo.

<arg9>: "Active" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco ativo, requisição de escrita enviada para o servidor e aguardando resposta.

Nota: A variável deve possuir permissão de escrita.

<arg10>: "Busy" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Bloco habilitado, mas recurso não está disponível (conexão ocupada com outra requisição), aguardando liberação para que a solicitação seja enviada pelo bloco. Se a entrada de habilitação for retirada enquanto o bloco faz esta indicação, a requisição é descartada.

Nota: A variável deve possuir permissão de escrita.

<arg11>: "Error" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Erro na execução da requisição.

Nota: A variável deve possuir permissão de escrita.

<arg12>: "ErrorID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Em caso de erro na requisição, indica o tipo de erro ocorrido. Resultados possíveis: 0= "Executado com sucesso"; 1= "Algum dado de entrada inválido"; 2= "Cliente não habilitado"; 4= "Timeout na resposta do servidor"; 5= "Servidor retornou erro".

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para escrita de registradores. Permite fazer a escrita de um ou mais registradores utilizando a função 6 (Write Holding Register) ou 16 (Write Multiple Registers) do Modbus.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done", que é ativada após o término da execução com sucesso da função. Após a transição positiva de "Execute" um novo telegrama é enviado pelo cliente Modbus TCP quando a conexão estiver livre. Ao término com sucesso da operação - resposta recebida do servidor - a saída "Done" é ativada, permanecendo ativa enquanto a entrada estiver ativa. Em caso de erro na execução da requisição, a saída "Error" é ativada, e o código do erro é colocado em "ErrorID".

NOTA!

O protocolo Modbus, utilizando a função 16, permite a escrita de registradores de 16 bits apenas. Para escrita de dados com mais de 16 bits (um REAL, por exemplo), é possível fazer a escrita de múltiplos registradores, e utilizar como fonte dos dados uma variável com tamanho maior que 16 bits.

É importante que a quantidade de registradores escritos seja compatível com o tamanho da variável ou do array de onde os dados serão utilizados.

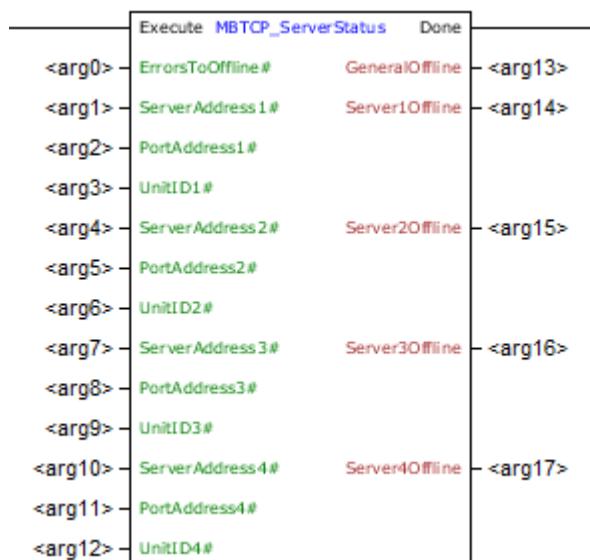
Disponível

- PLC300 : a partir da versão 1.30

6.13.21.3.6 MBTCP_ServerStatus

Estado dos Servidores da Rede Modbus TCP

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "ErrorsToSetOffline#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Permite programar, para este bloco, a quantidade de erros de comunicação que o cliente deve identificar até que a comunicação com um servidor seja considerada offline. É considerado erro de comunicação toda requisição (leitura ou escrita) que o cliente enviou para um servidor e não recebeu resposta.

<arg1>: "ServerAddress1#" - VAR_IN: inserir uma constante.

<arg4>: "ServerAddress2#" - VAR_IN: inserir uma constante.

<arg7>: "ServerAddress3#" - VAR_IN: inserir uma constante.

<arg10>: "ServerAddress4#" - VAR_IN: inserir uma constante.

Tipos de dados: DWORD

Descrição: Permite programar o endereço IP de até 4 servidores, cuja quantidade de erros de comunicação serão monitorados para saber se estão online ou offline. Caso a quantidade de erros de comunicação em sequência, detectados nos blocos de leitura e escrita via Modbus TCP, atinja o valor programado em "ErrorsToSetOffline", a saída respectiva é acionada. Caso deseje-se monitorar um número menor de servidores, pode-se deixar qualquer das entradas em zero: 0= "Ignora entrada".

<arg2>: "ServerPort1#" - VAR_IN: inserir uma constante.
<arg5>: "ServerPort2#" - VAR_IN: inserir uma constante.
<arg8>: "ServerPort3#" - VAR_IN: inserir uma constante.
<arg11>: "ServerPort4#" - VAR_IN: inserir uma constante.

Tipos de dados: WORD

Descrição: Permite programar a porta Modbus TCP de até 4 servidores, cuja quantidade de erros de comunicação serão monitorados para saber se estão online ou offline. Caso a quantidade de erros de comunicação em sequência, detectados nos blocos de leitura e escrita via Modbus TCP, atinja o valor programado em "ErrorsToSetOffline", a saída respectiva é acionada. O padrão é 502.

<arg3>: "UnitID1#" - VAR_IN: inserir uma constante.
<arg6>: "UnitID2#" - VAR_IN: inserir uma constante.
<arg7>: "UnitID3#" - VAR_IN: inserir uma constante.
<arg12>: "UnitID4#" - VAR_IN: inserir uma constante.

Tipos de dados: BYTE

Descrição: Permite programar o UnitID de até 4 servidores, cuja quantidade de erros de comunicação serão monitorados para saber se estão online ou offline. Caso a quantidade de erros de comunicação em sequência, detectados nos blocos de leitura e escrita via Modbus TCP, atinja o valor programado em "ErrorsToSetOffline", a saída respectiva é acionada. O padrão é 255.

<arg13>: "GeneralOffline#" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Se qualquer uma das saídas dos servidores indicados for acionada, esta saída também será acionada. Funciona como uma lógica OU entre as 4 saídas de indicação dos servidores.

Nota: A variável deve possuir permissão de escrita.

<arg6>: "Server1Offline#" - VAR_OUT: inserir uma variável (tag).
<arg7>: "Server2Offline#" - VAR_OUT: inserir uma variável (tag).
<arg8>: "Server3Offline#" - VAR_OUT: inserir uma variável (tag).
<arg9>: "Server4Offline#" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BOOL

Descrição: Saída acionada caso a quantidade de erros de comunicação em sequência para os servidores indicados nas respectivas entradas atinja o valor programado em "ErrorsToSetOffline".

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para monitoração dos servidores da rede Modbus TCP. Deve ser utilizado caso seja desejado identificar problemas na comunicação do cliente com algum servidor da rede Modbus TCP.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função. Enquanto a entrada de habilitação "Execute" estiver ativa os dados de entrada são utilizados e os dados de saída são atualizados a cada execução do bloco. A saída "Done" reflete o valor da entrada.

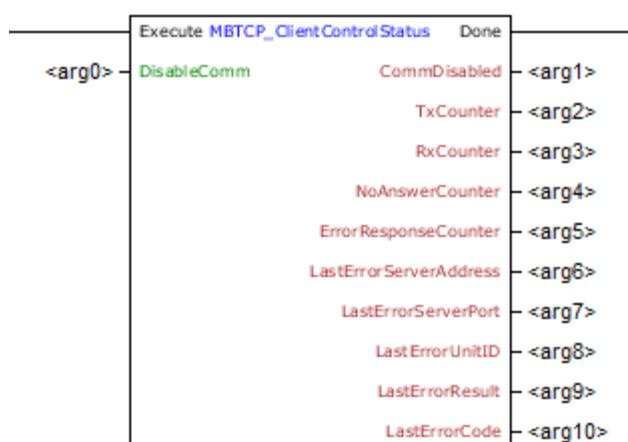
Disponível

- PLC300 : a partir da versão 1.30

6.13.21.3.7 MBTCP_ClientControlStatus

Controle e Estado do Cliente Modbus TCP

Figura



Configuração do Bloco

<inst> - inserir uma instância (tag).

<arg0>: "DisableComm" - VAR_IN: inserir uma constante ou uma variável (tag).
Tipos de dados: BOOL

Descrição: Permite desabilitar o cliente Modbus TCP. Ao desabilitar o cliente, os contadores e marcadores de status do cliente Modbus TCP também são zerados: 0= "Cliente em execução"; 1= "Desabilita cliente".

<arg1>: "CommDisabled" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: BOOL
Descrição: Indica se o cliente está ou não desabilitado: 0= "Cliente habilitado"; 1= "Cliente desabilitado".
Nota: A variável deve possuir permissão de escrita.

<arg2>: "TxCounter" - VAR_OUT: inserir uma variável (tag).
Tipos de dados: WORD ou UINT
Descrição: Contador de requisições enviadas pelo cliente para os servidores. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 65535.

Nota: A variável deve possuir permissão de escrita.

<arg3>: "RxCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de telegramas recebidos pelo cliente. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 65535.

Nota: A variável deve possuir permissão de escrita.

<arg4>: "NoAnswerCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de requisições do cliente que não foram respondidas pelos servidores. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 65535.

Nota: A variável deve possuir permissão de escrita.

<arg5>: "ErrorResponseCounter" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Contador de requisições do cliente em que os servidores responderam com alguma resposta de erro. O código do erro pode ser obtido no marcador que indica o código do último erro detectado. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 65535.

Nota: A variável deve possuir permissão de escrita.

<arg6>: "LastErrorServerAddress" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: DWORD

Descrição: Indica o endereço IP do servidor no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 247.

Nota: A variável deve possuir permissão de escrita.

<arg7>: "LastErrorUnitID" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: WORD ou UINT

Descrição: Indica a porta TCP do servidor no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 65535.

Nota: A variável deve possuir permissão de escrita.

<arg8>: "LastErrorResult" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Indica o resultado da operação - timeout ou resposta de erro, conforme ERROR ID do bloco - para o servidor no qual foi detectado o último erro de comunicação. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado: 0= "Sem erro detectado"; 4= "Timeout na resposta do servidor"; 5= "Servidor retornou erro".

Nota: A variável deve possuir permissão de escrita.

<arg8>: "LastErrorCode" - VAR_OUT: inserir uma variável (tag).

Tipos de dados: BYTE ou USINT

Descrição: Indica o código do erro, no caso do cliente receber resposta de erro de algum servidor. É zerado sempre que o equipamento for desligado ou o cliente for desabilitado - 0 a 255.

Nota: A variável deve possuir permissão de escrita.

Descrição de Funcionamento

Bloco para controle e monitoração do cliente Modbus TCP. Sempre que uma rede Modbus TCP for montada com o PLC300 como cliente, recomenda-se utilizar este bloco para obter informações sobre o estado da comunicação.

Possui uma entrada de habilitação do bloco "Execute" e uma saída "Done" que é ativada após o término da execução da função.

Enquanto a entrada de habilitação "Execute" estiver ativa, os dados de entrada são utilizados e os dados de saída são atualizados. Caso a entrada seja zerada, os valores de entrada são desconsiderados e os argumentos de saída são zerados. A saída "Done" reflete o valor da entrada.

Disponível

- PLC300 : a partir da versão 1.30

Índice

- V -

Variável 35

- W -

Watchdog; Estouro 36

- A -

arquivos 144

At; Representação Direta 34

- B -

Bloco Funcional; Function Block 34

- C -

configuração 133, 135, 136, 137, 138, 139, 140, 141, 143

Configuração; Config; Configuration 34

- F -

Função; Bloco de Função 34

- M -

modbus 144

- P -

Programa; Programa; Aplicativo 35

- R -

Recurso; Resource; Equipamento 35

- S -

SD Card 144

setup 133, 135, 136, 137, 138, 139, 140, 141, 143

- T -

Tarefa; Task 35

Tipo de Dado; Data Type 35